Intonation in Spoken Language Generation

Intonation
in
Spoken Language Generation

Een wetenschappelijke proeve op het gebied van de Letteren

Proefschrift

ter verkrijging van de graad van doctor
aan de Katholieke Universiteit Nijmegen,
volgens besluit van het College van Decanen
in het openbaar te verdedigen
op woensdag 4 juli 2001
des namiddags om 3.30 uur precies

door

Erwin Christian Marsi

geboren op 27 november 1968
te Deventer

Promotor:	Prof. Dr. C.H.M. Gussenhoven
Co-promotores:	Dr. P.A.J.M. Coppen Dr. A.C.M. Rietveld
Manuscriptcommissie:	Prof. Dr. L.W.J. Boves Prof. Dr. J. Nerbonne (RUG) Dr. J.M.B. Terken (TUE)

Cover illustration: view from Monte Perdido (3355 m) in the north of Spain, by M. van Beekvelt.

Published by
LOT
Trans 10
3512 JK Utrecht
The Netherlands

phone: +31 30 253 6006
fax: +31 30 253 6000
e-mail: lot@let.uu.nl
http://www.let.uu.nl/LOT/

ISBN 90-76864-07-1
NUGI 941

Contents

List of Abbreviations

A,AP	the syntactic categories Adjective and Adjective Phrase (p. 44)
σ	the prosodic category Syllable (p. 18)
Σ	the prosodic category Foot (p. 18)
ω	the prosodic category Prosodic Word (p. 18)
φ	the prosodic category Phonological Phrase (p. 18)
π	the prosodic category Intonational Phrase (p. 18)
AD	Association Domain (p. 23)
Adv,AdvP	the syntactic categories Adverb and Adverb Phrase (p. 44)
AGFL	Affix Grammars over Finite Lattices; a type of attribute grammar (p. 174)
APS	Abstract Phrase Specification (p. 142)
Aux	the syntactic category Auxiliary verb (p. 44)
Base	the base component of SynR (p. 172)
C	the syntactic category Complementizer (p. 44)
$\mathcal{CAS}$	the intonational phrasing algorithm by Caspers (p. 96)
CONPAS	CONtent to Prosodically Adequate Speech; the reusable content-to-speech module of our SLG system (p. 137)
Det	the syntactic category Determiner (p. 44)
F_0	Fundamental frequency (p. 2)
Flex	the inflection component of SynR (p. 172)
GRAMTSY	*GRAMmaticaal Transformationeel Systeem* 'Grammatical Transformational System' (p. 182)

$\mathcal{GUS}$	the intonational phrasing algorithm by Gussenhoven (p. 100)
H	high tone (p. 21)
%H	initial high boundary tone (p. 23)
H%	final high boundary tone (p. 23)
H*L	High-Low pitch accent (p. 22)
H*LH	High-Low-High pitch accent (p. 22)
KUNTTS	the TTS system for Dutch developed at the University of Nijmegen (p. 157)
L	low tone (p. 21)
%L	initial low boundary tone (p. 23)
L%	final low boundary tone (p. 23)
L*H	Low-High pitch accent (p. 22)
LRR	Lexical Rhythm Rule (p. 223)
MBROLA	MultiBand Resynthesis OverLap-Add; a diphone speech synthesis system (p. 167)
N,NP	the syntactic categories Noun and Noun Phrase (p. 44)
$\mathcal{NEW}$	the intonational phrasing algorithm for Dutch proposed in the current work (p. 112)
NLG	Natural Language Generation (p. 9)
NR	the text material consisting of Numbers (p. 245)
Num,NumP	the syntactic categories Numeral and Numeral Phrase (p. 44)
$\mathcal{N}\&\mathcal{V}$	the intonational phrasing algorithm by Nespor & Vogel (p. 87)
P,PP	the syntactic category Preposition and Prepositional Phrase (p. 44)
PlanR	the text planner of our SLG system (p. 137)
ProsR	the prosodic realization module of our SLG system (p. 157)
PTD	Public Transport Domain; the application domain involving travel descriptions (p. 139)
PTDGen	the SLG system for the public transport domain (p. 139)

PTDPlanR	the text planner of our SLG system for the public transport domain (p. 144)
Q	the syntactic category Quantifier (p. 44)
RR	Rhythm Rule (p. 41)
S	the syntactic category Sentence (p. 43)
SAAR	Sentence Accent Assignment Rule (p. 41)
SLG	Spoken Language Generation (p. 9)
SLH	Strict Layer Hypothesis (p. 19)
SpeechS	the speech synthesis module of our SLG system (p. 157)
SS	Speech Synthesis (p. 133)
SynR	the syntactic realization module of our SLG system (p. 153)
TD	the text material consisting of Travel Descriptions (p. 245)
Trans	the transformational component of SynR (p. 172)
U	the prosodic category Utterance (p. 18)
URR	Unit Rhythm Rule (p. 228)
V, VP	the syntactic categories Verb and Verb Phrase (p. 43)
WFD	Wild Flower Domain; the application domain involving botanical descriptions of wild flowers (p. 139)
WFDGen	the SLG system for the wild flower domain (p. 139)
WFDPlanR	the text planner of our SLG system for the wild flower domain (p. 149)

Acknowledgements

I would like to take the opportunity to thank all the people who have helped me to complete this thesis. To begin with, there are three people who initially started this project, supervised my work, and provided invaluable support all the way. My promotor, Carlos Gussenhoven, has been my primary source of knowledge with regard to the phonology and phonetics of intonation. He also managed to gently correct my English into something readers may actually understand. Peter-Arno Coppen, my first co-promotor, has been largely responsible for my education in generative grammar, and Dutch syntax in particular. His system for interpreting transformational grammars (Gramtsy) has proved to be essential during this project. Toni Rietveld, my second co-promotor, has contributed his extensive knowledge of phonetics and speech synthesis, and offered great assistance in all statistical matters. In general, all three of them have always been very involved and supportive.

Thanks to all my colleagues at the Department of Language and Speech, who offered me a room, a computer, and coffee, even when it took some time to finish my thesis. Thanks also to Esther Grabe, Judith Haan, Brechtje Post and the others members of the prosody group for the interesting collaboration. Joop Kerkhoff provided support for the KUNTTS speech synthesizer, and compiled special versions for use in my SLG system. Lisanne Theunissen read parts of this thesis and provided useful comments. My former roommate, Simon van Dreumel, managed to read almost all of the text, and provided so many suggestions and critical questions that I could easily fill ten more PhD projects.

I am grateful to the students of the course on Natural Language Generation for showing me all the things I did not yet know about this topic. Frouke Hermens made an initial implementation of the text planner for generating flower descriptions.

I also indebted to Johanneke Casper, Judith Haan, Cecilia Odé, and Marc Swerts for their intonational annotations, which formed an essential contribution to the final evaluation experiment. Jaques Terken, as a member of the manuscript commission, made many useful corrections and suggestions.

Finally, I would like to thank my friends for keeping me in touch with everyday life, my parents for supporting and encouraging me all these years, and Mireille for all her help and love (and who is probably even more happy than I am that this project is finished).

Chapter 1

Introduction

When two theories purport to explain the same set of data, we will choose the one that is simpler, or if they seem much the same on that score, the one that additionally explains new data. In intonation research, where variant theories commonly exist, evaluation is in practice hampered by the circumstance that the new data required for testing predictions of these theories do not exist. The data that *are* available – examples used in the literature – are limited in number, and possibly selective in various ways. One way of overcoming this problem is to implement theories in a computer program, get them to *generate* spoken language, and see whether their predictions with respect to the intonation make sense. The question then is whether the generative capacity of the theory is similar – by some criterion – to that of humans. This is the question that the investigation reported here tries to answer.

To pursue our aim, we would need to build a *spoken language generation system*, whose output can be observed as an acoustic signal. Given such a system, however, we would still be faced with the task of evaluating the output. As will be explained below, comparing outputs by just listening to them will not lead to decisive answers as to which theory is to be preferred. In order to make convincing comparisons, we will need to isolate just those aspects which the theories that are tested actually describe, and generate, and see how these match up with what humans do. In the case of intonation, there is the additional problem of variability, in the sense that different speakers may choose different intonation contours for the same text, as well as inappropriateness, in the sense that many speakers are not capable of reading texts without disfluencies, or even obviously incoherent placements of pitch accents or boundaries. As we will explain below, obtaining a responsible data set that can serve as a reference in this respect is a problem that we have solved by eliciting the best estimate of the most acceptable realization of a text.

This chapter is organised as follows. In Section 1.1, we introduce three aspects of intonation: pitch accent, intonational phrase, and tune. We discuss linguistic intuitions about intonation in the form of acceptability judgements, how to cope with variance in acceptability judgements, and how to properly

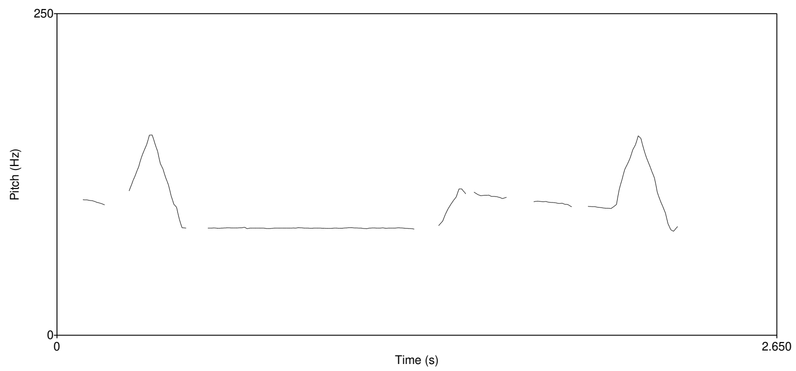


Figure 1.1: Pitch contour of the (synthesised) utterance [De sn$\overset{*}{\text{e}}$ltrein naar Nijmegen] [is al gew*ee*st] *'The fast-train to Nijmegen has already left'.*

assess these judgements. In Section 1.2, we introduce spoken language generation, and the way it is used in the current work. In Section 1.3, we describe our main research objective, identifying both a number of restrictions on its scope, and several more specific research questions. An outline of the remainder of the thesis concludes this chapter.

1.1 Intonation

1.1.1 Pitch accent, intonational phrase, and tune

Languages contain sounds that are *voiced* (all vowels and some of the consonants). The source of voiced sounds is a vibration of the speaker's vocal cords. The inverse time of one cycle of opening and closure of the vocal cords is called *fundamental frequency*, (F_0) or *pitch*.[1] Pitch can be measured during the production of a voiced sound, and is traditionally expressed in hertz (Hz), i.e. the number of cycles per second. If we measure the pitch of an utterance at fixed intervals of time, we obtain a sequence of pitch values through which can we draw a *pitch contour* (Figure 1.1).

Intonation can now be defined as:

> "[..] all aspects of the perceived pitch pattern that the speaker intends for the hearer to use in understanding the utterance, or that the hearer does use whether intentionally controlled by the speaker or not." (Beckman 1995)

[1] Strictly *pitch* refers to the auditory sensation of tone height, but the term will be used here to mean fundamental frequency.

Although not all theories of intonation are equally specific on this point, most of them distinguish at least three relevant aspects of the pitch pattern: the distribution of pitch accents, the structuring of speech by intonational phrases, and the specification of tune. A *pitch accent* can be defined as a perceptually salient pitch event, which usually results from a relatively sudden change in pitch level marking a particular syllable. An *intonational phrase* is a stretch of speech covered by a perceptually coherent intonation contour. Intonational phrase boundaries are also marked by a change in pitch level. *Tune* (or *melody*) refers to the particular shape of the intonation contour, which may vary independently of the locations of accents and boundaries. For instance, a statement may be turned into a question by changing only the tune, leaving the place of pitch accents and intonational boundaries unchanged. More precise definitions of these terms will be given in Chapter 2.

As an example, consider the question-answer pair in (1), and imagine the most natural intonation for the answer. The pitch accents in the answer are indicated by a star (*): the first is aligned with the stressed syllable of the word *sneltrein*; the second with the stressed syllable of *geweest*. These correspond to the pitch peaks in Figure 1.1, also referred to as *pointed hats* ('t Hart, Collier, and Cohen 1990). Furthermore, there are two intonational phrases, as indicated by by the square brackets. The boundary between them corresponds to the pitch discontinuity in Figure 1.1, i.e. a rise to a high pitch level at the end of *Nijmegen*, and a subsequent reset to the start pitch level beginning on *is*. Again, we leave a precise account of the relation between this symbolic representation of the intonation and the actual pitch track to Chapter 2.

(1) Question: Waarom nemen we de stoptrein naar Nijmegen?
why take we the slow-train to Nijmegen?
'why are we taking the slow train to Nijmegen?'

Answer: [$_{\text{AD}}$ De sn*eltrein naar Nijmegen] [$_{\text{AD}}$ is al
The fast-train to Nijmegen has already
gew*eest]
left

In the current work, we are primarily concerned with pitch accents and intonational boundaries, largely ignoring the aspect of tune.

1.1.2 Acceptability of intonation

Native speakers have linguistic intuitions: they can usually tell if some expression is acceptable in their language. A native speaker of Dutch, for example, would agree that (2-a) is an acceptable expression, whereas (2-b) is unacceptable.[2]

[2]Following common practice, unacceptable expressions are preceded by a star (*).

(2) a. de trein vertrekt naar Amsterdam
the train leaves to Amsterdam

b. *de trein vertrekken naar Amsterdam
the train leaves to Amsterdam

From a syntactic point of view, the challenge is to account for these acceptability judgements. This requires analysing an expression's syntactic structure in abstract terms. For instance, we can say that (2-b) is ill-formed because the *subject* (*de trein*) is not in agreement with the *verb* (*vertrekken*) with respect to the feature *number*. In the tradition of generative linguistics, the syntactic analysis is formalised as a grammar that should generate all and only well-formed expressions.

Similarly, native speakers have linguistic intuitions about the intonation of an expression. Listening to the utterances[3] in (3), Dutch speakers would agree that the intonation of the first is perfectly acceptable, while that of the second is not.

(3) a. [$_{\text{AD}}$ de trein vertrekt naar Amsterdam]
the train leaves to Amsterdam

b. *[$_{\text{AD}}$ de trein] [$_{\text{AD}}$ vertrekt naar Amsterdam]
the train leaves to Amsterdam

Again, accounting for these judgements requires analysing an expression's intonational structure. We can attribute the unacceptability of (3-b) to the fact that: (1) the last intonational phrase does not contain any accents (and would presumably have low pitch), and (2) the accent falls on the wrong syllable of *vertrekt*. In the next chapter, we will present a more elaborate analysis, and a grammar of Dutch intonation that is intended to generate all and only well-formed intonational structures.

However, in addition to being structurally well-formed, an expression must also fit into the context. Compare the two pieces of text in (4) and (5), which differ only with respect to their second sentence. Most native speakers of Dutch would probably agree that (4) is better than (5). Although (5-b) is perfectly acceptable in isolation, it is less appropriate in the given context when compared with the alternative in (4-b).[4]

(4) a. U neemt de trein naar Amsterdam.
You take the train to Amsterdam.

b. Die vertrekt om acht uur tien.
Which leaves at eight hour ten.

[3]In general, we will use the term *utterance* to refer to the audible, phonological form, and the term *sentence* to refer to the readable, written form.

[4]While comparing expressions, the less acceptable expression is preceded by a question mark.

(5) a. U neemt de trein naar Amsterdam.
You take the train to Amsterdam.

b. ?Om acht uur tien vertrekt de trein.
At eight hour ten leaves a train.

Accounting for this type of observation requires not only an analysis of the syntactic structure of an expression, but also an analysis of its context. Although it is not the topic of this thesis, we will elaborate on the relation between the syntactic structure of an expression and its context in Chapter 2 and Chapter 6. The point here is that acceptability judgements may be *context-dependent*. Another way to look upon this is to say that the set of possible syntactic forms to express a piece of information is reduced when it is placed in a more specific context.

The same goes for acceptability judgements about intonation. Accentuation is highly context-dependent. For example, most speakers would agree that the accentuation in (6-b) is inappropriate as a reply to question (6-a).

(6) a. Waarom de stoptrein naar Nijmegen?
why the slow-train to Nijmegen?

b. ?De sneltrein naar Nijmegen is al geweest
The fast-train to Nijmegen has already past

By contrast, the same accentuation is perfectly acceptable as a reply to question (7-b)

(7) a. Waarom de sneltrein naar Arnhem?
why the slow-train to Arnhem?

b. De sneltrein naar Nijmegen is al geweest
The fast-train to Nijmegen has already past

It is generally assumed that the reason for this is that pitch accents signal the information structure of an utterance.[5] In general, words that convey important or new information will be accented, whereas those conveying less important, old, or predictable information will remain unaccented. Hence, *sneltrein* can remain unaccented in answer (7-b), as it was already mentioned in the preceding question. whereas *Nijmegen* and *geweest* are both accented, because they contribute important new information. The fact that (6-b) is unacceptable can be explained in the same way. Although this relation between the pitch accents of an utterance and its information structure is backed up by substantial experimental evidence (Birch and Clifton 1995; Brown 1983; van Donzel 1999; Fowler and Housum 1987; Horne 1991; Kruyt 1985; Nooteboom

[5] We will elaborate on information structure, including the notion of *focus*, in Section 2.5.1.

and Terken 1982; Nooteboom and Kruyt 1987), the correspondence is definitely not as simple as these examples might suggest. Not only because an adequate definition of 'new' or 'important' has proved to be difficult, but also because of interactions with other aspects of the semantic, morpho-syntactic, and phonological structure of an utterance, preventing a one-to-one relation between 'new' and 'accented'.

In sum, native speakers can judge the acceptability of an expression's intonation. In order to be acceptable, the intonation has to be structurally well-formed and contextually appropriate. From a linguistic point of view, the challenge is to account for these judgements by means of a grammar.

1.1.3 Variance in acceptability

Despite the clear-cut examples above, it is not uncommon to find variation in acceptability judgements. There appear to at least three sources of variation:

1. acceptability judgements may be gradual rather than categorical, ranging from entirely unacceptable to highly acceptable
2. alternative intonational forms may be judged *equally* acceptable
3. acceptability judgements may vary from person to person

Any linguistic theory or computational system that tries to account for the acceptability of intonational forms has to deal with this variation. In the current work, the following assumptions are made.

With respect to the gradual nature of acceptability, the aim is to predict the *most acceptable* intonation of an utterance, alternatively referred to as the *preferred intonation*. For example, it is claimed in the current work that the accentuation in (8-a) is more acceptable than that in (8-b), and that the intonational phrasing of (9-a) is more acceptable than that of (9-b). This is not to say that the b-examples are unacceptable, but rather that a-examples are preferred in unemotional utterances in informationally neutral contexts.[6]

(8) a. honderd v$\overset{*}{\text{i}}$eren d$\overset{*}{\text{e}}$rtig
hundred four-and thirty
'134'

b. ?h$\overset{*}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren d$\overset{*}{\text{e}}$rtig

(9) a. [Die vertr$\overset{*}{\text{e}}$kt van perron $\overset{*}{\text{e}}$lf] [om $\overset{*}{\text{a}}$cht uur ti$\overset{*}{\text{e}}$n]
Which leaves from platform eleven at eight hour ten

b. ?[Die vertr$\overset{*}{\text{e}}$kt van perron $\overset{*}{\text{e}}$lf om $\overset{*}{\text{a}}$cht uur ti$\overset{*}{\text{e}}$n]

[6]Kruyt (1985, p. 115) proposed a similar distinction between *acceptable accentuation* and *optimal accentuation* in order to explain the variation between subjects in a number of perception and production experiments on accentuation in Dutch.

If alternative intonational forms are equally acceptable, we are satisfied with predicting at least one of these. In general, we will ignore equally acceptable alternatives, because they are of no value in distinguishing one theory from another. After all, if an utterance *with* a certain accent (boundary) is equally acceptable *without* it, the theory that predicts this accent (boundary) is just as good as the one that does not.

Finally, the possibility of intersubjective differences in acceptability judgements is addressed by relying on the judgements of *experts*. That is, the most acceptable intonation is determined by combining the judgements of a number of subjects who are experienced in analysing and transcribing Dutch intonation. The point is that experts are better judges of what is normal than laymen, because they have been exposed more to linguistic data and have developed a greater awareness of them. This situation is parallel to one in which a person who has learnt to write alphabetically is better able to identify the segments in a CVC syllable than an illiterate (Morais, Cary, Alegria, and Bertelson 1979).

In sum, the challenge addressed in the current study is to account for the intonation that is *most acceptable* (preferred) according to the judgements of experts, ignoring differences between equally acceptable forms.

1.1.4 Acoustic versus symbolic assessment of acceptability

The scientific paradigm of the current study is that of generative linguistics. Accordingly, the facts on which the theories are built are the intuitions of native speakers about the acceptability of intonational forms (competence), rather than intonational forms that are observed in actual speech (performance). Whereas intuitions about syntactic form are relatively easy to obtain, intuitions about intonational form are less easily accessed, and bring along a number of methodological considerations.

In order to tap the intuitions of *naive* subjects, the data, i.e. carrier utterances with the relevant patterns of accents and intonational phrases, will have to be presented acoustically. That is, rather than reading them, subjects must listen to the utterances, because naive subjects cannot be expected to judge a symbolic representation of the intonation. To make the data audible, we need either a human speaker, some form of speech manipulation, or speech synthesis. Whichever way this is done, the danger will be that acceptability judgements are influenced by external factors, such as segmental quality or speaker characteristics. Although listeners may be explicitly instructed to ignore these aspects, and to pay attention to the intonation only, it is doubtful whether naive listeners are capable of this. The only reliable method to prevent segmental quality and speaker characteristics from affecting a subject's judgement appears to be to present pairs of utterances that differ only with respect to their intonation.

However, this method is unsuitable for comparing long utterances or paragraph-sized text, due to listeners' memory limitations. Moreover, it is difficult, if not impossible, for a human speaker to produce two versions of an utterance that differ only with respect to their intonation. Alternatively, a speech processing technique like LPC resynthesis or PSOLA may be used to manipulate the pitch and duration of the stimuli. The problem with this approach is that the quality of the phonetic realization of pitch accents and boundaries in terms of F_0 may be inadequate. For instance, the precise alignment of intonational peaks to the segmental content is a difficult problem that is not yet fully understood; see Wichmann, House, and Rietveld (2000) for an overview. This introduces the possibility that an accent or boundary is in principle correctly placed, but is judged unacceptable only because of an inadequate phonetic realization. In addition, it is known that a natural realization of pitch accents involves increasing duration and intensity, and altering spectral tilt (Sluijter 1995). Creating these effects by hand is a nontrivial task. The same problems occur when using a speech synthesiser, with the addition that errors in segmental synthesis may disturb a subject's judgement as well. In sum, acoustically-based approaches for assessing the intuitions of native speakers are faced with methodological problems, all of which are related to the fact that other aspects of the wave form may interfere with judgement of the intonation.

As an alternative, intuitions of native speakers may be assessed at the symbolic level. This means that, rather than acoustically, the data are presented orthographically as strings of words annotated with intonational symbols indicating pitch accents, intonational boundaries, and perhaps also tune. Subjects are then asked to judge the acceptability of the intonation on the basis of this symbolic representation. This type of *symbolic assessment* has been used for evaluating accent placement in TTS systems (Baart 1987; van Bezooijen 1989; Monaghan 1993), and is most extensively presented as a sound evaluation method in Monaghan and Ladd (1990).

Evidently, symbolic assessment precludes the use of naive subjects, as only experts can interpret the symbolic representation. However, the use of expert judgements was already proposed in the previous section in order to limit intersubjective variation. Moreover, it avoids the methodological problems associated with an acoustic presentation of the data. At the same time, results obtained with both types of presentation appear to be comparable: van Bezooijen (1989, p. 29) reports a significant, positive correlation between judgements about accent placement in a symbolic and acoustic evaluation of the output of a Dutch speech synthesiser.

In the current work, therefore, symbolic assessment was used to assess the intuitions of experts with respect to the preferred accent placement and intonational phrasing. The latter has, as far as we are aware, not been studied before using symbolic assessment. Furthermore, we follow van Bezooijen (1989) in asking subjects to *prescribe* the locations of pitch accents and intonational boundaries rather than to judge whether a certain distribution of accents and boundaries is acceptable. The result will be referred to as a *prescription* or *reference*. The advantage of this approach is that the resulting prescription remains a valuable piece of data, because it can be used time and again as a reference for verifying predictions.

1.2 Spoken language generation

Spoken language generation (SLG) is a form of *natural language generation* (NLG). According to two prominent researchers in the field, NLG may be defined in the following way:

> "The area of study called natural language generation is about twenty-five years old, having started as an offshoot of the discipline natural language processing (also called computational linguistics). It shares interests with a healthy mixture of linguistics, computer science (artificial intelligence), psycholinguistics, and occasionally, philosophy, sociology, and neurolinguistics. The central question studied is how one can program computers to produce high-quality text from some computer-internal representation of information. Motivation for this study ranges from entirely theoretic (linguistics, psycholinguistic) to the entirely practical (to produce output systems for computer programs)." (Bateman and Hovy 1991, p. 53)

NLG systems come in different sizes and flavours. Some of them are modest applications that use canned text to produce simple messages, while others are complex systems with ambitious goals such as generating weather forecasts in multiple languages, the technical documentation of an aircraft, social security forms, or personalised letters to convince a person to quit smoking. See Adorni and Zock (1993) and Paiva (1998) for a survey of NLG systems.

The more advanced NLG systems are firmly rooted in linguistic theory. For instance, discourse grammars are employed to generate coherent multi-sentence texts. Likewise, syntactic grammars are used to map abstract phrase specifications into grammatically correct expressions in the target language. These aspects, and NLG in general, will be discussed more thoroughly in Chapter 5. The key point here is that NLG systems incorporate a fair amount of linguistic knowledge in the form of grammars and intermediary levels of representation.

The topic of this thesis, *spoken language generation* (SLG), is an extension of natural language generation to spoken rather than written output. Analogously, the central question is how one can program computers to produce high-quality *speech* from some computer-internal representation of information. It is therefore best viewed as a combination of NLG and speech synthesis.

Our motivation for working on SLG is primarily linguistic, that is, our perspective is that of a theoretical/computational linguist working in the paradigm of generative linguistics. Our intention is to use SLG as a tool to test and improve linguistic theories that try to account for the intuitions of native speakers with respect to the intonation.

The advantages of using SLG to this end may need some clarification. As far as accent placement and intonational phrasing are concerned, theories are typically based on a small set of crucial examples. Since alternative theories account for the well-known examples equally well, it is difficult to compare them. These theories should be confronted with a wider range of material in a systematic way to see to what extent their empirical predictions hold good. The problem is that test material is not readily available. Testing theories about accent placement and intonational phrasing typically requires discourse, semantic and morpho-syntactic information. Although prosodically-annotated speech corpora have become available, these appear to lack the amount of structural analysis that is desirable for this purpose.

There is, however, a way of bypassing the problem of lack of adequate data. The NLG system that is a necessary part of any SLG system is in principle capable of generating all of the required discourse, semantic and syntactic information. It can be used to generate large quantities of reliable test material. An SLG system thus offers an excellent test bed for linguistic theories about intonation. In addition to verifying a theory, its implementation in an SLG system may reveal hidden assumptions or inconsistencies. In short, we think that both the observational and descriptive adequacy of a linguistic theory may benefit from its implementation in an SLG system.

It should be noted that we have no psycholinguistic ambitions in the sense that our SLG systems might model human language production. Neither do we have ambitions with respect to language engineering or language and speech technology in the strict sense of intending to solve some immediate practical problem by means of SLG technology. Nevertheless, theoretical research has often been relevant for applied research and vice versa, and some practical implications of our work will be discussed in Chapter 9.

According to the above definition of SLG, the goal is to produce *high-quality speech*. Evidently, a definition of the quality of the output, and of the quality of an SLG system in general, directly depends on the motivation for engaging in SLG. From a psycholinguistic point of view, the quality of the output is determined by the extent to which it resembles human speech performance, for example, with respect to timing, hesitations, speech errors and repairs (De Smedt 1996). From an applied perspective, however, it is defined in terms of its practical value, for instance, how actual users judge the intelligibility, naturalness, or pleasantness of the output, or how well it enables a user to accomplish a certain task. The motivation that underlies the current work is primarily linguistic: to account for the intonation that is most acceptable according to the judgements of experts. Hence, high-quality output is defined as speech with the pitch accents and intonational boundaries preferred by experts.

We think this goal is well suited to the research paradigm of NLG and SLG. With respect to syntax, for instance, few NLG systems aim at generating merely acceptable, or even all acceptable, syntactic forms. Rather, systems try to generate the *best* syntactic form, given the content and context of the message. In the same vein, an SLG system should try to generate the *best* intonational form.

1.3 Research objective

To sum up so far, our intention is to use SLG as a tool to test and improve linguistic theories about the relation between the intonation of an utterance and other aspects of its structure and context. We focus on the placement of pitch accents and intonational boundaries, largely ignoring matters of tune choice. Given our linguistic motivation, the SLG system should output the intonation that is preferred by experts. In short, the *research objective* of this study is:

> To design, build, and test a linguistically-based SLG system that generates speech with pitch accents and intonational boundaries that are most acceptable to experts.

There are a number of other restrictions on the scope of the current work. First, and rather trivial, the target language for our SLG system is *Dutch*. This means that not all of our conclusions are necessarily valid for other languages, but they will, at a minimum, be suggestive of how other languages could be dealt with. Second, it is concerned only with speech read aloud; the generation and synthesis of spontaneous speech is likely to bring along some different problems. Third, although the output is real speech, we focus on generating intonation at the symbolic, phonological level, and largely abstract away from matters of phonetic realization. Fourth, due to the limitations inherent in NLG, system output is limited to certain domains. In our case, these are botanical descriptions of wild flowers (the *Wild Flower Domain* or WFD), and instructions on travelling by means of the public transport system (the *Public Transport Domain* or PTD); the reasons for choosing these particular domains will be described in Chapter 5. Fifth, it is not practically feasible to study all of the competing theories about accent placement and intonational phrasing by means of SLG. Therefore, we commit ourselves to a number of specific linguistic theories, which will be introduced in Chapter 2.

Given the above objective and restrictions, the research questions addressed in this work belong to two categories: (1) questions regarding the linguistic theory, and (2) questions regarding the architecture of an SLG system.

As explained above, we use SLG as a tool for evaluating particular linguistic theories, more specifically, by asking the following questions:

1. How adequate is the theory?

2. How can it be improved?

Adequacy is primarily addressed in terms of observational adequacy, that is, correctness of predictions and coverage of the facts.

Although NLG has a longstanding tradition, the body of research on SLG is relatively small. There are still a number of moot questions regarding the architecture of an SLG system, including, but not limited to, the following:

1. What is the best way to combine NLG and speech synthesis?
2. What is the best place in the system to control the intonation?
3. What modifications to the 'standard' architecture of NLG and speech synthesis systems are required?
4. What additional information should be generated by the NLG system, and how can this be accomplished?

Admittedly, some of these questions may overlap, but given our current state of knowledge, more focused formulations cannot be provided yet. These questions underly the work presented in this thesis, the content of which will be outlined below.

1.4 Outline of thesis

This thesis is divided in four parts. The first part (Chapter 2) presents the theoretical framework, which forms the point of departure for all subsequent work on accent placement and intonational phrasing. The second part involves preliminary explorations into phonological phrasing (Chapter 3) and intonational phrasing (Chapter 4). This involves corpus-based work, rather than SLG proper, but provides a substantial part of the linguistic theory ultimately implemented in the SLG system. The third part (Chapters 5, 6, and 7) describes the SLG system we developed. The final part concerns an evaluation of its output (Chapter 8), followed by a general discussion (Chapter 9). The role and content of the individual chapters will be described below.

Chapter 2: Theoretical Framework
This chapter presents the linguistic framework for our work on accent placement and intonational phrasing in SLG.

First, we define the concept of *prosody*, and introduce *prosodic phonology*, which is a phonological theory about the hierarchical organisation of speech. It assumes the existence of a *prosodic structure* consisting of prosodic constituents (the phonological counterpart of a syntactic tree) which serves as the interface between semantics, syntax, and morphology on the one hand, and prosodic phenomena on the other. One of these prosodic constituents is the *phonological phrase*, which plays an important role in our approach to accent placement. It is commonly assumed that the *intonational phrase* is a prosodic constituent as well, but it is argued here that this view gives rise to contradictions, and an alternative is presented.

Second, we introduce *intonational phonology*, and a phonological description of Dutch intonation. This allows us to describe the intonation of utterances at a phonological level, and to abstract away from the details of phonetic implementation.[7] Furthermore, the research objective is interpreted as generating pitch accents and intonational boundaries that are not only well-formed according to the intonational phonology, but also fit the grammatical structure, meaning, and context of an utterance.

Third, we present a preliminary rule-based grammar for pitch accent placement, which forms the point of departure for accentuation in the SLG system. The role of focus and argument structure in accent placement is discussed. It is shown that phonological phrasing is crucial for pitch accent placement, as the phonological phrase forms the application domain for the *Rhythm Rule*. This is a phonological rule that deletes accents under certain conditions, which are also explained.

Finally, this chapter outlines the syntactic analysis of Dutch assumed throughout our work.

Chapter 3: Phonological Phrasing
As already mentioned, correct phonological phrasing is considered crucial for accent placement. In this chapter, it is argued that the algorithms for phonological phrasing that have been proposed by other researchers are inadequate for Dutch. Therefore, a new algorithm for phonological phrasing in Dutch is proposed.

This algorithm is subsequently evaluated at the symbolic level: its consequences for accent placement are compared with a reference accentuation, resulting from a symbolic assessment of the preferences of several experts. The text material used in this evaluation consists of botanical descriptions of wild flowers (the wild flower domain or WFD). These involve rather long complex sentences, and are therefore highly challenging for a phonological phrasing algorithm. The results show that the newly proposed algorithm is fairly adequate.

Chapter 4: Intonational Phrasing
This chapter addresses intonational phrasing in Dutch in much the same way as phonological phrasing was addressed in Chapter 3. We start by describing several algorithms for intonational phrasing that have been proposed in the literature. Next, these algorithms are evaluated by comparing the intonational boundaries that they predict with those prescribed by a panel of experts. A detailed analysis of the results reveals that the algorithms have various flaws, and that none of them is entirely satisfactory. From these findings, a number of requirements for a better algorithm are derived.

The second part of this chapter presents a new algorithm for intonational phrasing. A similar evaluation of this algorithm shows that it performs better than the earlier algorithms.

[7]Although phonetic implementation will be addressed to some extent at the end of Chapter 5.

Chapter 5: Spoken Language Generation
This chapter primarily addresses the research questions regarding the architecture of an SLG system. It discusses alternative ways of combining NLG and speech synthesis. It summarises the 'standard' architecture for NLG systems and proposes an extension for SLG systems. In particular, it addresses the question where intonation should be generated in an SLG system.

Next, we introduce a reusable front-end for SLG systems called CONPAS ('CONtent to Prosodically Adequate Speech'). This component is shared in the two SLG systems developed in the current work. The first system is for generating flower descriptions (WFDGen), and is best considered as an attempt to reproduce the results of Chapter 3 and Chapter 4 in the context of an actual SLG system. The second system generates travel descriptions (PTDGen), and is intended as an independent domain to test the developed grammars for accent placement and intonational phrasing.

The remainder of this chapter provides an overview of the modules and intermediary representations in the SLG system. It describes the modules for text planning and speech synthesis, but leaves a description of the modules for syntactic and prosodic realization to the next two chapters.

Chapter 6: Syntactic Realization
Here, we describe a syntactic realization module for Dutch, which translates abstract phrase specifications into grammatically correct expressions and their syntactic structures. The main purpose of this chapter is to show how all of the syntactic information that is required for accentuation and intonational phrasing can indeed be generated in an SLG system.

Chapter 7: Prosodic Realization
This chapter brings together work of the preceding chapters by describing how the prosodic realization module handles the derivation of prosodic structure, pitch accents, and tune choice. The main contribution in this chapter is a further development of the preliminary algorithm for pitch accent placement presented in Chapter 2. One of the issues is the correct accentuation of numerals.

Chapter 8: Evaluation
This chapter reports on an evaluation of the SLG system. This is again carried out at the symbolic level by comparing the predicted pitch accents and intonational phrases with those prescribed by a panel of experts. However, the domain is different: travel descriptions generated by PTDGen rather than the flower descriptions that figured in the previous evaluations. Moreover, the prescription is provided by experts who did not participate in the previous evaluation. Roughly, this evaluation shows that the performance of our algorithms is comparable to, or even better

(depending on the evaluation metric) than that of other approaches, but still somewhat worse than that of humans.

Chapter 9: General Discussion
In this final chapter, we examine the current work in the light of the research objective, in particular with respect to the research question formulated above. In addition, we discuss some issues that were not explictly addressed, including the practical implications for speech technology.

Chapter 2

Theoretical Framework

2.1 Introduction

The previous chapter introduced the two main issues addressed in this thesis: pitch accent placement and intonational phrasing in SLG. This chapter presents the theoretical framework for our work. It provides a more precise interpretation of these issues in terms of phonological theories, and a first suggestion as to how we may account for them. This will then be worked out in the following chapters.

Section 2.2 defines prosody, and summarises prosodic phonology, the study of prosody from a phonological point of view. Section 2.3.1 introduces intonational phonology, and describes the intonational phonology of Dutch. It should be noted, however, that material in this chapter is not intended as review of prosodic phonology or intonational phonology, but focuses on those aspects that are relevant to our research aims.[1] This theoretical framework enables a more precise definition of the challenges involved in generating intonation in Section 2.4. As a first step towards a solution, Section 2.5 proposes a linguistic account of pitch accent placement in the form of a generative grammar. As most of the work reported in this thesis is in one way or another dependent on syntax, Section 2.6 briefly describes the syntactic analysis of Dutch used throughout our work. The chapter ends with a summary in Section 2.7.

2.2 Prosodic phonology

2.2.1 Prosody

Speech is more than a succession of vowels and consonants. These speech segments are organised into higher units: segments are structured into syllables,

[1] General introductions to prosodic phonology can be found in Booij (1983), Nespor and Vogel (1986), Inkelas and Zec (1995), and Shattuck-Hufnagel and Turk (1996); the best source for intonational phonology is Ladd (1992, 1996).

syllables into feet, etc. In addition, there appear to be patterns of relative prominence among these higher units. For example, the most prominent foot in a word is called the *main-stressed* foot, which contains the main stressed syllable. These intuitions are supported by phonetic measurements which show that these perceived higher units, and their relative prominence relations, are correlated with acoustic parameters such as pitch, duration, and segmental quality. Accordingly, syllables with main-stress tend to be longer and more carefully pronounced than syllables with secondary or no stress (see for example Sluijter (1995) and Rietveld, Kerkhoff, and Gussenhoven (1999) for Dutch).

The hierarchical organisation of segments into higher units, the patterns of relative prominence among these units, and their phonetic reflection in terms of phonetic parameters are collectively referred to as *prosody*. Although other definitions of prosody have been proposed, the above definition is the one that emerges from recent work that aims at integrating insights from phonological and phonetic research. Accordingly, a recent tutorial paper on prosody defines it as:

> "(1) acoustic patterns of F_0, duration, amplitude, spectral tilt, and segmental reduction, and their articulatory correlates, that can be best accounted for by reference to higher-level structures, and (2) the higher-level structures that best account for these patterns" (Shattuck-Hufnagel and Turk 1996, p. 196).

2.2.2 A phonological account of prosody

Initially, phonological accounts of prosody attempted to identify the hierarchical organisation of an utterance with its syntactic structure. However, it became clear that some important aspects of the syntactic structure are not reflected in the prosody, and conversely, that some prosodic aspects cannot be predicted from the syntactic structure.

> "It has become a majority view among researchers [...] that syntax does not provide domains for phonological rules in a direct fashion. Both the impoverished amount of syntactic information needed by the phonological module and the variety of mismatches between phonological rule domains and syntactic constituency argue for positing another level of representation" (Inkelas and Zec 1995, p. 537).

In reaction to this discrepancy between syntax and prosody, a number of phonologists, most prominently Selkirk (1984, 1996), Nespor and Vogel (1986), and Hayes (1986), developed a theory called *prosodic phonology*. It assumes the existence of a *prosodic structure*, an independent structure that consists of *prosodic constituents* labelled according to their *prosodic category*. An example of a prosodic structure is given in Figure 2.1.

Following to Nespor and Vogel (1986) and Hayes (1986), we distinguish seven prosodic categories, which are strictly ranked on a scale called the *proso-*

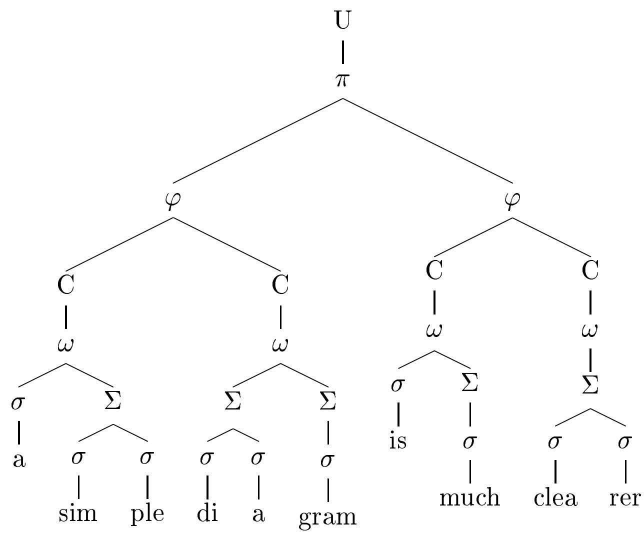


Figure 2.1: Example of a prosodic structure.

dic hierarchy; see Table 2.1. It is assumed that every utterance must be *fully* parsed in terms of *all* prosodic categories. Moreover, the resulting structure must adhere to the prosodic hierarchy. That is, a prosodic constituent of a certain rank cannot be dominated by a prosodic constituent of lower rank. A stronger version of this assumption claims that a prosodic constituent of a certain rank can only dominate a constituent of the directly subordinated rank, thereby excluding the possibility of recursion. This is known as the *Strict Layer Hypothesis*.

(1) **Strict Layer Hypothesis** (SLH)

The categories of the Prosodic Hierarchy may be ranked in a sequence $\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_n$ such that

a. all segmental material is directly dominated by the category C_n, and

b. for all categories C_i, $i \neq n$, C_i directly dominates all and only constituents of category C_{i+1}.

(Hayes 1986)

By implication, prosodic trees must have n-ary branching nodes. Notice, however, that the prosodic structure in Figure 2.1 partly violates the SLH: the determiner *a* is directly dominated by a ω rather than a Σ. The reason is that a schwa is unstressed, and cannot constitute a Σ or ω of its own. Some limited violation of the SLH is assumed to be necessary for prosodic constituents below

Table 2.1: Hierarchy of prosodic categories.

Rank:	Category:	Symbol:
1.	Utterance	U
2.	Intonational Phrase	π
3.	Phonological Phrase	φ
4.	Clitic Group	C
5.	Prosodic Word	ω
6.	Foot	Σ
7.	Syllable	σ

the φ (Booij 1995, p.144-145). Yet, in the current work, we will adhere to the SLH for φ's and higher categories.

The motivation for distinguishing these particular prosodic categories is based on two types of arguments. First, prosodic constituents may be the application domain for certain segmental rules. These *P-rules* are shown to apply within (or at) the boundaries of a prosodic constituent of a certain rank, but not across its boundaries. An example is *Degemination* in Dutch (Booij 1995, p. 151): in a sequence of two identical consonants, one of them may be deleted. In (2-a), which constitutes a single π, Degemination is responsible for deleting one of the /n/'s. In contrast, it cannot apply in (2-b), because of the intervening π-boundary. As we will see later, the reason for this boundary is that tag questions such as *niet* obligatory form an π of their own (Section 4.4.1).

(2) a. [$_\pi$ Is banaa/n/ /n/iet gezond dan?]
Is banana not sound then

b. [$_\pi$ Apen houden van banaa/n/] [$_\pi$ /n/iet?]
Monkeys like of banana not
'Monkeys like banana, don't they?'

However, undisputed evidence of this type is rare, especially for the prosodic categories between ω and π. The application of P-rules is thought to be dependent on speaking rate and speaking style as well. In all fairness, the generalisation appears to be that most P-rules apply more frequently in smaller domains than in larger domains (Booij 1995, p. 146).

The second type of argument in support of a prosodic constituent is based on the assumption that prosodic constituents account for the prosodic boundaries observed in an utterance. The higher the rank of a prosodic category, the more prominently its boundaries are marked by phonetic cues such as pause and lengthening of the preboundary syllable (Ferreira 1993; Menert 1994; Hofhuis, Gussenhoven, and Rietveld 1995; Rietveld, Kerkhoff, and Gussenhoven 1999; Cambier-Langeveld 2000). For instance, Gussenhoven and Rietveld (1992b) present evidence from a perception experiment that English listeners expect longer preboundary syllable durations as the rank of the prosodic boundary is higher.

The prosodic structure can be seen as the interface that connects morphology, syntax and semantics to phonology. It is obtained by filtering out all linguistic aspects that are not relevant to phonology. For instance, the complex, recursive syntactic structure of an utterance is reduced to a strictly-layered prosodic structure, which has only a few fixed categories and no recursion. This reduction is accomplished by means of so-called *mapping rules*, which map segmental, morpho-syntactic and semantic information onto prosodic constituents. Various mapping rules have been proposed, with slightly different predictions. Since only φ's and π's are directly relevant for accent placement and intonational phrasing, we will concentrate on the mapping rules for these constituents in Chapter 3 and Chapter 4, respectively. Some remarks regarding the lower prosodic constituents and their mapping rules can be found in Section 5.6.

2.3 Intonational Phonology of Dutch

2.3.1 Intonational phonology

Segmental phonology is concerned with describing the sound of a language at the phonemic level. As is well-known, the definition of phonemes is based on their capability to distinguish one word from another. For instance, Dutch has a phonemic distinction between the vowels /e/ and /u/, as the two words of the minimal pair /pen/ (*peen*, 'carrot') versus /pun/ (*poen*, slang for 'money') have different meanings. Other differences between speech sounds are non-distinctive, and gradual in nature. For example, although the vowel /e/ is 'coloured' by the following /r/ in /per/ (*peer* 'pear'), it is not distinct from the /e/ in /pen/ in the sense that it can differentiate Dutch words, and is therefore not a different phoneme. The distinction between phonemes is categorical, whereas the difference between allophones (non-distinctive variants of a phoneme) may be gradient in nature.

Analogously, *intonational phonology* assumes that intonation is phonologically structured, and that intonation contours can be analysed in terms of a limited inventory of categorical, contrastive intonational patterns. Additional gradient, non-distinctive variation in pitch is assumed to be a matter of phonetics. Accordingly, intonational phonology is concerned with the phonological aspects of pitch, representing pitch at an abstract, symbolic level. It describes intonation in the form of a grammar, that is, a combinatorial system consisting of a limited inventory of atomic elements, and rules to combine these elements into more complex structures. The complement of intonational phonology is *phonetic implementation*, which concerns the actual realization of these phonological abstractions as concrete pitch contours.

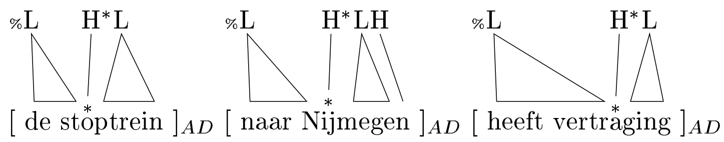


Figure 2.2: Example of Gussenhoven's description of Dutch intonation. The utterance means 'The slow-train to Nijmegen has delay'. The top line is the tone string consisting of boundary tones and pitch accents. The bottom line is the segmental string, where square brackets indicate association domains, and stars indicate accented syllables. In between are the tiers that align the two level of description.

The first model of intonation that introduced an explicit distinction between phonological description and phonetic implementation has been attributed to Pierrehumbert (1980). Her description of American-English intonation is called *autosegmental*, because it describes the intonation not in terms of features of phonemes, but as a separate string of intonational events, which are synchronised with the segmental string by means of *tiers*. The tonal events on the tone string are either *pitch accents* or *boundary tones*, both of which are composed of one or more *tone segments* (or *tones* for short). There are only two of these, *H* and *L*, which represent phonological abstractions of a high and a low pitch level, respectively.[2]

Following the work of Pierrehumbert (1980) for American-English, intonational phonologies of other languages have appeared. Gussenhoven (1984, 1988, 1991b, 1992b) developed an autosegmental description of English and Dutch intonation. It is this model that forms the theoretical basis of the work on intonation in SLG presented in this thesis. The relevant aspects of Gussenhoven's model will be presented in the remainder of this section.

2.3.2 Pitch accents

Earlier on, a *pitch accent* was defined as a perceptually salient pitch event, which results from a relatively sudden change in pitch level. A phonological description must specify more precisely which pitch patterns count as a pitch accent. According to Gussenhoven's description, Dutch has three types of pitch accents: H*L, L*H, and H*LH. The diacritic * indicates that the preceding tone segment, the *starred tone*, must be linked to a syllable in the segmental string. For instance, the first H* in Figure 2.2 is linked to the syllable *stop*, which is

[2]Notice that these are pitch *levels*, not pitch *movements*; in other words, points, not trajectories. This is one of the differences with primarily phonetic descriptions of intonation like the IPO-description of Dutch intonation ('t Hart, Collier, and Cohen 1990), which describe an intonation contour as a concatenation of pitch movements.

the main stressed syllable of the word *stoptrein*. Anchoring the starred tones of the tone string to stressed syllables in the segmental string is one of the ways in which the two descriptions are aligned.

It is important to keep apart the notions of (secondary and primary) *stress* and *pitch accent*. Although *stress* has been used as a term to cover prominence both at the lexical level (i.e. relative prominence patterns among syllables) and at the phrasal level (i.e. prominence patterns among words and phrases) current convention reserves *stress* for prominence at the lexical level and *pitch accent* for prominence at the phrasal level. This difference in prominence categories is supported by phonetic evidence (Beckman and Edwards 1994): the primary empirical cues for stress are segmental quality and duration, whereas the main cue for a pitch accent is pitch.

2.3.3 Boundary tones

An *intonational phrase* was defined earlier as a stretch of speech covered by a perceptually coherent intonation contour. This implies that the boundaries of an intonational phrase are marked by some discontinuity in the intonation contour. There appear to be at least two ways in which an intonational phrase boundary can be signalled: a change in pitch level, described by the occurrence of a boundary tone, or a change in register, described by the interruption of downstep. It is sometimes assumed they can be also be signalled by a pause of sufficient length. Although we agree that a tonally marked boundary is often supported by a pause, we reject the view that a pause is a sufficient cue for an intonational phrase boundary, because this gives rise to the inconsistencies (to be discussed in Section 2.3.7).

The change in pitch level near an intonational boundary is analysed in autosegmental approaches as a *boundary tone*, either H or L. Just as the starred tone segment of a pitch accent, it must be aligned with the segmental string. In Gussenhoven's model, it is linked to an edge of *association domain* (AD). For the time being, we can think of an AD as equivalent to an intonational phrase; we will discuss the differences in Section 2.3.7. As an example, the utterance in Figure 2.2 has three AD's. The edge the boundary tone is linked to is indicated by a prefixed (left edge) or suffixed (right edge) %-symbol. This amounts to four possibilities: a high initial boundary tone (%H), a low initial boundary tone (%L), a high final boundary tone (H%), and a low final boundary tone (L%). For example, the first %L in Figure 2.2 is an initial boundary tone that is linked to the beginning of the first AD. We postpone a discussion of boundary marking by a change in register to Section 2.3.6.

2.3.4 Intonation contours

So far, we have introduced the segmental string and the tonal string, which are synchronised by association lines which link every starred tone to a stressed syllable, and every boundary tone to an AD-boundary. Not every stressed syllable or AD-boundary needs to be linked. In Figure 2.2, for example, the syllables *de*, *naar*, and *heeft* are not linked to a starred tone; in other words, they are not accented. Similarly, the first AD in Figure 2.2 is not linked to a final boundary tone. The tone string in Figure 2.2 (%L H*L %L H*LH %L H*L) forms a complete description of the intonation, but before the representation is sent to the phonetic implementation rules for conversion to an acoustic signal, two things will happen.

First, tonal rules may modify the representation as a result of *AD-restructuring*, as described in Section 2.3.5. Second, default boundary tones are inserted. For one thing, the final H of a H*LH pitch accent in AD-final position functions as an AD-final boundary tone, H%. Second, an Utterance-final H*L is provided with a final L%, as shown in Figure 2.3, while an Utterance-final L*H is provided with a H%. Since we want to be able to discuss different intonational patterns without going into all the details, we will make use of an abstract, graphical representation of the pitch contour, one that dispenses with the default boundary tones and thus shows only AD-structure, the AD-initial boundary tone, and the pitch accents.

In order to derive the intonation contour, it is assumed that every tone yields a *target*. For instance, the first H* in Figure 2.3 yields a high target, indicated by a bullet. The dashed arrow indicates the tone of which each target is the phonetic implementation. Some tone segments, such as the initial boundary tones in Figure 2.3, produce two targets. The first target is located at the boundary, whereas the second target is located at some fixed distance from some other target. This is called *spreading*, and it is indicated in the figure by a triangle between the tone and the text showing the extent of the spreading. In addition to initial boundary tones, the *second* tone of the *last* pitch accent in an AD spreads up to the next tone in the AD, or up to the end of the AD if it is the last tone. For example, the L of the first H*L in Figure 2.3 spreads up to the end of the AD; the L of the H*LH spreads up to the following H, which, as a boundary tone, is in turn located at the end of the AD. When all targets are determined, connecting the consecutive targets yields the intonation contour. The computation of the actual pitch contour by the phonetic implementation module will be described more fully in Section 5.6.2.2.

2.3.5 Tonal rules

In addition to being autosegmental, Gussenhoven's description is also transformational. Analogously to segmental phonology, it assumes an underlying representation of the intonation that is rewritten into a surface representation by a number of optional *tonal rules*. The underlying representation requires that the utterance is exhaustively parsed into one or more AD's, and that each of these contains exactly one pitch accent.[3]

One of the tonal rules that may modify the underlying representation is called *AD-restructuring*. It merges two AD's into a single one by deleting their

[3]There is one exception: AD's without a pitch accent may occur for reporting clauses, where a rule called *Tone Copy* provides a pitch contour to the AD without an accent (Gussenhoven 1990; Gussenhoven 1992a).

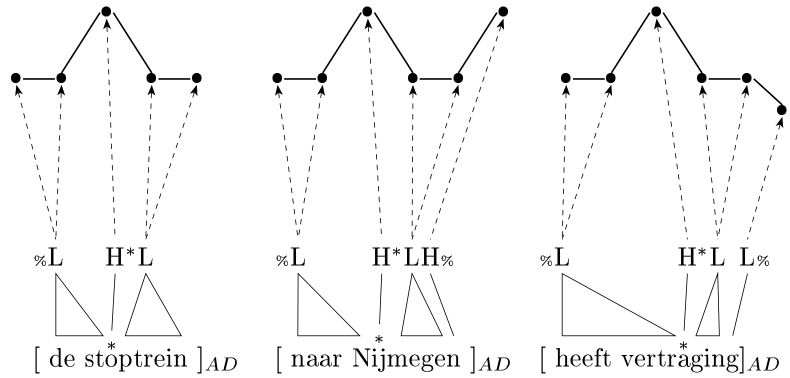


Figure 2.3: Abstract representation of the intonation contour corresponding to the description of Figure 2.2.

adjacent boundaries. As explained above, AD's serve as a reference for linking the non-starred tone segments of a pitch accent. Therefore, AD-restructuring will alter the alignment between the segmental string and the tone string, and consequently the shape of the intonation contour. An example of restructuring is given in Figure 2.4. Notice that the %L that was linked to the beginning of the second AD in Figure 2.3 is no longer present, since there is no legitimate position available to link it to. Furthermore, the first H*L accent is no longer the last accent in an AD. Recall that spreading only applies to the second tone segment of the *last* pitch accent in an AD. Therefore, the L tone segment of the first H*L in Figure 2.3 does not spread, but moves up to the following H*LH. The result is known as a *partly linked* contour.

Another tonal rule is called *Tone Linking*. It deletes the non-starred tone segments of the penultimate pitch accent in an AD. Figure 2.5 shows the effect of applying Tone Linking to the first AD in Figure 2.4. The result is an intonation contour that is known as a *fully linked* or *flat hat* contour.

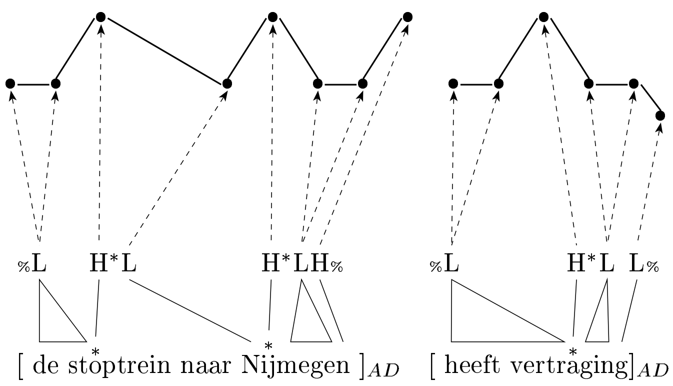


Figure 2.4: Example of AD-restructuring, resulting in a partly linked contour.

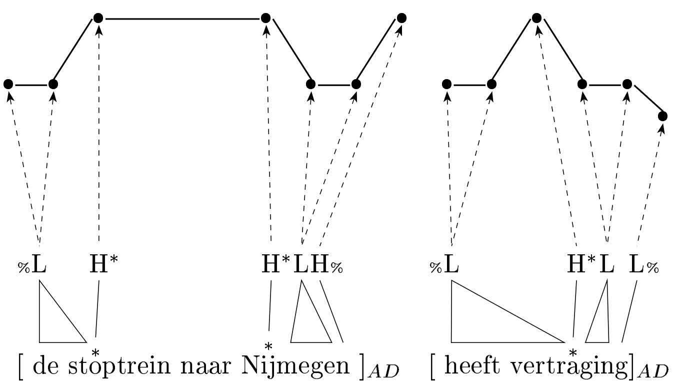


Figure 2.5: Example of Tone Linking, resulting in a fully linked contour.

2.3.6 Downstep

As mentioned earlier, an intonational phrase boundary can be signalled by a boundary tone or by a register change. The register corresponds roughly to the pitch span between H and L tones.[4] It can be compressed or expanded, and shifted upwards or downwards. One of its uses is to set off the boundaries of appositions, such as parenthetical remarks and non-restrictive relative clauses, where the register is compressed and shifted downwards at the start of the apposition, and restored to its former size and level at the end of the apposition.

Related to this is the phenomenon of gradual lowering of the pitch in the course of an intonational phrase, sometimes referred to as *declination* ('t Hart, Collier, and Cohen 1990). Declination has not been implemented in our intonation model, since it is believed that *accentual downstep* and *phrasal downstep* cause the intonation contour to have the required effect. The former involves stepwise lowering of the H* targets in an AD during phonetic implementation, the latter involves stepwise lowering of the AD's in an utterance (see Section 5.6.2.2). In Gussenhoven's model, these are described by an optional *tone morpheme* ('!') that is attached to an AD or an utterance, respectively.

The domain of accentual downstep is the AD. This means that it can be used as a cue to signal intonational phrasing: as long as there is downstep, there can be no intervening intonational boundary, whereas the register reset following downstep indicates the presence of a boundary.[5]

2.3.7 Association Domain

Section 2.2 defined prosody as the hierarchical organisation of segments into higher units, the patterns of relative prominence among these units, and their reflection in terms of phonetic parameters such as F_0 and duration. Clearly, the intonational phrase meets the criteria for a prosodic unit, and has therefore been incorporated into prosodic phonology as well. Most descriptions of intonation define an intonational phrase as a stretch of speech covered by a coherent, continuous pitch contour, and delimited by audible breaks. This definition of intonational phrase was also adopted by prosodic phonology: "The formulation of the basic I[ntontonational phrase] formation rule is based on the notions that the intonational phrase is the domain of an intonation contour and that the end of intonational phrases coincide with the positions in which pauses may be introduced in a sentence." (Nespor and Vogel 1986, p. 188). The intonational phrase was incorporated as a prosodic constituent (π), slotted in between the Utterance (U) and the Phonological Phrase (φ). In addition, mapping rules were formulated to derive π's from the semantic and syntactic structure of an utterance.

Incorporating the intonational phrase as a prosodic constituent (π) implies that it becomes subject to the general constraints on prosodic structure imposed

[4] See van den Berg, Gussenhoven, and Rietveld (1992) for a more careful definition.

[5] This not always true, since there is a variant of downstep in which the last accent of a downstepped AD is unaffected.

by the theory of prosodic phonology. For instance, the Strict Layer Hypothesis (Section 2.2) requires that every π is directly dominated by an U, and that every π directly dominates one or more φ's. In addition, the empirical predictions regarding prosodic constituents must hold for π as well. One of these is that the prominence of the edge of a prosodic constituent is proportional to its rank in the prosodic hierarchy. In other words, the higher the constituent, the more its boundaries will be emphasised by prosodic means such as preboundary lengthening and pausing. This explains the remark about pause in the citation above: since the π is the last but one highest in rank, its boundaries are likely to coincide with pauses. Consequently, whenever a clear silent[6] pause occurs, this must be considered strong evidence for the presence of a π-boundary. Another prediction is that the application of segmental rules is likely to be blocked by the intervention of an π-boundary.

However, this combination of theoretical constraints and empirical predictions, which result from incorporating the intonational phrase in the framework of prosodic phonology, leads to some serious problems. The core of these problems is a discrepancy between the two criteria for an intonational phrase, i.e. a coherent intonation contour and an audible pause.

Gussenhoven and Rietveld (1992a) gives the following pair of utterances.

(3) a. [$_\pi$ But we're n$\overset{*}{\text{o}}$t telling John]

b. [$_\pi$ But we're n$\overset{*}{\text{o}}$t going] [$_\pi$ John]

While (3-a) has a single intonation contour corresponding to a single π, (3-b) has the same contour, but two π's. This is because John is a indirect object in (3-a), but a vocative in (3-b). The latter constitutes an obligatory π according to the rule for π-formation (see Section 4.4.1), but is unaccented, like all final vocatives. On the one hand, the intonational boundary in (3-b) is correct, since *John* is preceded by a prosodic break, which is clearly missing in (3-a). On the other hand, the boundary is incorrect, since not every π in (3-b) corresponds to a coherent intonation contour, which violates the definition of an intonational phrase.

Another example of the same phenomenon is (4).

(4) a. [$_\pi$ To b$\overset{*}{\text{o}}$rrow the book may be che$\overset{*}{\text{a}}$per]

b. [$_\pi$ Tom$\overset{*}{\text{o}}$rrow] [$_\pi$ the book may be che$\overset{*}{\text{a}}$per]

The preposed adverb *tomorrow* can occur in exactly the same intonation contour as the verb *to borrow*. That is, from an intonational point of view, both utterances ought to be analysed as single π's. Yet, from a rhythmical point of view, there is an audible break after *tomorrow*, which is absent after *to borrow*. For additional evidence involving the application of segmental rules, see Gussenhoven and Rietveld (1992a)

[6] That is, not a filled pause, hesitation, etc.

Whereas these examples concern instances where a single intonation contour spans two π's, (5) involves a single π that contains at least two intonation contours. Gussenhoven points out that in the following pair of utterances the contours on *grey* and *bread* (5-b) can be identical to the contours on *Grey* and *bread* in (5-a), that is, a sequence of rise-falls, with an intonational break before *bread*.

(5) a. I brought [$_{\varphi}$ Gr$\overset{*}{\text{e}}$y] [$_{\varphi}$ br$\overset{*}{\text{e}}$ad]

b. I brought [$_{\varphi}$ gr$\overset{*}{\text{e}}$y br$\overset{*}{\text{e}}$ad]

While (5-a) may be accounted for by claiming that the φ-boundary after *Grey* coincides with a π-boundary, this solution is unavailable in (5-b), since the principles of prosodic phonology forbid a π to be dominated by a φ.

The crucial point is to show that *grey bread* in (5-b) really constitutes a single φ. One argument is that a pause may be inserted after *Grey* in (5-a), which is far less acceptable after *grey* in (5-b). A better argument is found in the facts of 'stress shift'. It is generally accepted that the φ forms the domain for 'stress shift' phenomena (see Section 2.5.3). A word like *unbaked* normally has its main stress on the second syllable (cf. (6-a)).[7] However, if it is immediately followed by another stressed word within the *same* φ, its main stress retracts to the first syllable in order to avoid a stress clash with the next word. As this is exactly what happens when we substitute *unbaked* for *grey* (cf. (6-a) versus (6-b)), we can infer that *grey bread* is indeed a single φ. In contrast, substituting *Marie-Lou* for *Grey* does not trigger stress shift (cf. (6-c) versus (6-d)), indicating that *Mary-Lou bread* is not a single φ, and by implication, cannot contain a π-boundary either without violating the principles of prosodic structure.[8]

(6) a. I brought bread that is ùnbáked.

b. I brought únbàked bréad

c. I brought Màrie-Lóu bréad

d. *I brought Márie-Lòu bréad

To sum up, the incorporation of the intonational phrase as a prosodic constituent (π) in the framework of prosodic phonology gives rise to contradictions. Gussenhoven and Rietveld (1992a, p. 286) conclude: "[...] in general terms, the problem is that the stretch of speech covered by an intonation contour cannot consistently be identified with a particular prosodic constituent." The core of the problem is a mismatch between the two criteria for an intonational phrase,

[7] Main stress is indicated by ´, secondary stress by `.

[8] Yet another problem is that a φ, by definition, must contain at least one lexical head, i.e. a noun, verb, or adjective that is a content word. However, a tag question as in *You're not going, are you?* does not contain a lexical head; just an auxiliary and a pronoun. Therefore, it cannot be a φ. However, tag questions are considered to constitute separate π's, for both intonational and durational reasons (cf. Section 4.4.1). As a result, we have a π that does not dominate a φ, which violates the principles of prosodic structure.

i.e. a coherent intonation contour and an audible pause, which is also discussed in Ladd (1986).

The introduction of AD's can be seen as a solution to this problem: "[...] there is an independent constituent over which the tone segments of a pitch accent are distributed: the association domain (AD). The AD is determined by the structure provided by the conventional prosodic constituents, but is not itself part of the prosodic hierarchy. On this view the status of the conventional I[π, in our notation] is similar to that of the other prosodic constituents, like Ph[φ, in our notation]. It will be pause-defined, and may play a role in the structural description of phonological rules, but it would not be definable on the basis of the intonation contour." (Gussenhoven and Rietveld 1992a, p. 287).

Unfortunately, the resulting terminology is a potential source of confusion, so let us state clearly what we mean by the terms *intonational phrase*, π, and AD. We will keep using *intonational phrase* as a general, theory-neutral term that refers to a stretch of speech with a perceptually coherent intonation contour. The π and AD are theory-specific interpretation of this: π is the intonational phrase as a prosodic constituent in prosodic phonology; AD is the intonational phrase in Gussenhoven's intonational phonology. We will use *π-phrasing* for the determination of π's, and *AD-derivation* for the derivation of AD's. However, determining intonational phrases is largely independent of the question about the correct theoretical status of the intonational phrase. That is, whether we prefer to analyse intonational phrases as π's, AD's, or some other theoretical construct, the problem of how to derive them for a particular utterance in a particular context remains essentially the same. Therefore, we will also continue to use the generally accepted term *intonational phrasing* whenever the difference between π's and AD's is irrelevant.

2.3.8 Summary

To sum up, we have assumed an intonational model with the following properties.

- Following the principles of intonational phonology, there are two levels of analysis: phonological description and phonetic implementation.
- The description of the intonation of an utterance is autosegmental, consisting of a tone string of tonal events that is aligned with a segmental string by means of tiers. The tonal events are either pitch accents (H*L, L*H, or H*LH) or boundary tones (%H, %L, H%, or L%), which are both composed of the primitive tones H and L. Every starred tone of a pitch accent must be linked to a stressed syllable, and every boundary tone must be linked to an AD-boundary; the remaining tones either spread, or move up to the next tone or AD-boundary.
- The point of departure is an underlying representation in which every utterance is exhaustively parsed into AD's, and every AD contains exactly one pitch accent. AD-restructuring allows for merging of AD's. Tonal

rules like Tone Linking may transform the underlying representation to a surface representation. This will then be mapped onto an actual pitch contour by phonetic implementation rules, taking into account tone morphemes like Downstep.

- The intonational phrase cannot be incorporated into the prosodic hierarchy (π), but must be analysed as a independent domain called the association domain (AD).

2.4 Generating intonation

The intonational phonology of Dutch described in the previous section aims to define all well-formed intonational patterns, or more precisely, all well-formed combinations of tone strings and segmental strings. It allows for variation along three dimensions.

1. An utterance may be divided into AD's in many different ways. Taking Figure 2.6 as a (arbitrary) reference, Figure 2.7 shows an alternative division into AD's, in which the first two AD's have been restructured to a single one. We will refer to the choice of AD's as *AD-derivation*, or generally as *intonational phrasing* if there is no chance of confusion with π-phrasing.

2. The number of pitch accents, or their locations, may be different. Example Figure 2.8 shows a variant in which the word *stoptrein* is unaccented.[9] The choice of the number of accents and their locations in an utterance will be referred to as *pitch accent placement*.

3. The kind of pitch accent may vary. In Figure 2.9, both accents of the type L*H; the H tone at the end of the final AD will cause this utterance to be interpreted as a question. In addition, tonal rules like Tone Linking and tone morphemes like Downstep may alter the shape of the contour. The choice of the type of pitch accents and boundary tones, and the application of tonal rules and tone morphemes in an utterance will be called *tune choice*.

The intonational grammar describes the variation along these three dimensions from a purely structural point of view, analogous to the way an autonomous syntactic grammar attempts to define all grammatical strings of a language without taking their meaning into account. However, many, if not most, of the intonational variants that are structurally well-formed will not be appropriate for a particular utterance in a particular context. It is generally accepted that this is because the intonational form is constrained by

[9] As a consequence, the utterance can have at most two AD's, since AD's without an accent are ill-formed. This shows that the three dimensions of variation are not independent of each other.

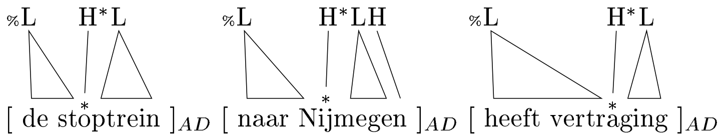


Figure 2.6: An example of an intonational form for the utterance 'The slow-train to Nijmegen has delay'.

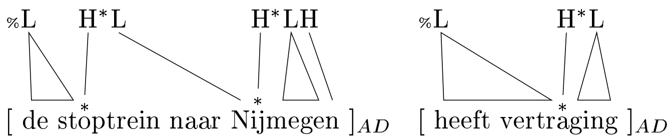


Figure 2.7: The same utterance with a different division into AD's.

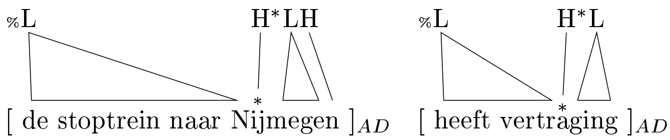


Figure 2.8: The same utterance with a different number of pitch accents as well.

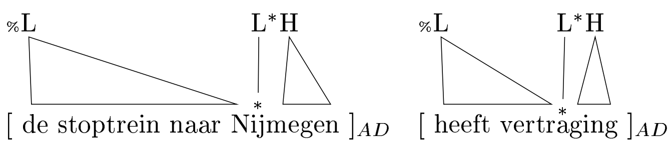


Figure 2.9: The same utterance with a different type of pitch accent as well.

the semantic, morpho-syntactic and phonological structure of the utterance, in combination with contextual and pragmatic factors. This has proved to be a complex relation, and has given rise to a large body of research in theoretical and computational linguistics, psycholinguistics, phonetics and speech technology. In the current work, we will address this issue from the point of view of generating intonational forms in an SLG system. The challenge of *generating intonation* is thus to perform intonational phrasing, pitch accents placement and tune choice in accordance with the linguistic structure and context of an utterance.

The complement of generating intonation may be called *interpreting intonation*, and concerns the way the intonation of an utterance constrains its structure and meaning, and that of the discourse it is part of. Whereas the former is typically found in TTS and SLG, the latter may occur in automatic speech recognition (ASR) and spoken dialogue systems. Although the present work is primarily concerned with generating intonation, a considerable part of the underlying theory actually concerns the general relation between intonation on the one hand and syntax, semantics, phonology and discourse on the other hand, and may therefore be relevant to the problem of intonation interpretation as well.

Pitch accent placement, as was already demonstrated in Section 1.1.2, is sensitive to the distinction between 'new' and 'given' information. In the next section, we will present a provisional algorithm for pitch accent placement that requires other input as well. It will be argued that other relevant factors include the predicate-argument structure, adjacency relations in the surface string, and the prosodic structure (phonological phrases). The main purpose of presenting this grammar is to explain why we will be concerned with phonological phrasing in Chapter 3, designing an appropriate SLG environment that can supply the required information in Chapter 5, and generating syntactic structures in Chapter 6. A more elaborate algorithm for pitch accent placement will then be presented in Chapter 7.

Intonational phrasing appears to be mainly dependent on the syntactic structure. In Chapter 4, several algorithms for intonational phrasing proposed in the literature will be systematically compared. This comparison will reveal a number of flaws in these algorithms. On the basis of this, we will propose and test a new algorithm.

Tune choice appears to be closer to pitch accent placement in the sense that it is affected by many factors, among others, the sentence mood (e.g. statement versus question) and the discourse structure. However, tune choice is beyond the scope of our research objective, and will only be superficially addressed in the current work.

The theoretical framework described so far contributes in at least two ways to our research goals. First, it allows us to focus on generating intonational forms at a phonological level, abstracting away from the details of phonetic implementation. Second, it defines well-formed intonational forms, ruling out those forms that are unacceptable on purely phonological grounds. In addition, it provides a precise definition of tune choice, from which we intend to abstract away, versus pitch accent placement and intonational phrasing.

2.5 Pitch accent placement

Pitch accent placement, or accentuation for short, refers to predicting the distribution of pitch accents in an utterance, i.e. the locations of pitch accents. At the level of intonational phonology, a pitch accent is linked to a stressed syllable, whereas at the level of the phonetic implementation, it is even more precisely located, when its targets are timed relative to the segments that make up the syllable (Section 5.6.2.2). For current purposes, however, it is convenient to specify the locations of pitch accents at a higher level of abstraction: we will assume that pitch accents are attached to words. Our goal is thus to explain and predict which words of an utterance are accented.

Distributing pitch accents over words abstracts away from the fact that some words can carry more than one pitch accent. For example, the word *roodbruin* 'redbrown', which is a compound adjective, may in some contexts carry pitch accents on both constituents. This implies that, in principle, we have to go down to the morphological level of non-compound words to provide a pitch accent with a unique location. This issue will be further discussed in Section 7.3.1.1. In most cases, however, it will suffice to locate each pitch accent by anchoring it to a word. We will continue to use a star (*) above the stressed syllable(s) of a word to indicate that it is pitch accented.

In accordance with our linguistic motivation, our goal is a *grammar for pitch accent placement*, that is, a generative grammar that takes into account all relevant factors in order to produce an acceptable accent distribution. A number of such grammars have been proposed, for example by Selkirk (1984, 1995) for English, or Baart (1987), Dirksen (1992, Dirksen and Quené (1993), and van Wijk (1987b) for Dutch. The grammar used in the present work is due to Gussenhoven (1984, 1991a, 1992b, 1999). Below, we will review the relevant factors and consider how they can be modelled the in grammar. The general strategy is one of deletion. That is, we start from the assumption that every word is is accented in the underlying representation, and our ambition is to predict which accents survive in the surface representation.

2.5.1 Focus

Language is typically used to exchange information by means of utterances that are not just sets of primitive propositions, but complex structured messages, which convey more aspects of meaning than just the propositional content. One of these aspects concerns the relative importance of the parts of information that constitute the message, which has been called the *information structure*.

> "Information structure is the encoding of the relative salience of the constituents of a clause, especially nominals, and is realized as

> choices among alternative syntactic arrangements. The information structure of a clause is determined by the larger sentence or discourse of which it is part (i.e. its context). The communicative effect of the information structure is to foreground certain aspects of the message of the clause, but to background others. The need to encode information structure is a language universal, but the formal means to do so vary widely across the languages of the world" (Foley 1994)

Almost every theory about grammar encompasses some notion of information structure, although the differences in terminology and definitions is bewildering, making it one of the most confusing issues in linguistics (Grosz, Joshi, and Weinstein 1995; van Kuppevelt 1991; Rooth 1995; Bosch and van der Sandt 1999). Nevertheless, some notion of information structure is essential for the purpose of pitch accent placement. In addition, it is important for the NLG component, because it affects syntactic variations in word choice and word order. We therefore will attempt to explicate our assumptions about information structure.

Foley's definition of information structure has three important aspects:

1. some information is more salient than others;
2. the distinction is formally (syntactically) encoded;
3. and depends on the context.

With respect to the first point, we will assume a binary distinction between *focus* and *background*. The partition of an utterance into focused and unfocused parts will be called its *focus distribution*. The focus distribution is not necessarily equivalent to the information structure; rather, the former constitutes a part of the latter, as the information structure is often considered to include additional distinctions. For instance, that between *topic* and *comment*, where the former represents what the sentence is about, and the latter what the sentence has to say about this: "The topic is presupposed information that the remainder of the sentence comments upon." (Foley 1994). In functional approaches to grammar (Halliday 1994), this is referred to as *theme* versus *rheme*. We will ignore other aspects of the information structure here, and concentrate on the focus distribution.

The focus-background distinction is closely related to the new-given distinction. Generally, new information, which is added to the discourse for the first time, will be focused. Conversely, old information, which has been recently mentioned in the discourse, will be left unfocused. The prototypical example of this involves a question-answer pair. Since a question is a request for a specific part of new information, it requires this piece of information to be focused in the reply. For example, *where* in (7-a) forces focus on the requested location, i.e. *a party*, in (7-b), where focus is indicated by underlining.

(7) a. Where did Sue go to?

b. She went to $\underline{\text{a p}\overset{*}{\text{a}}\text{rty}}$.

The question arises how a listener recognises the focus in (7). This brings us to the second point of the above definition of information structure: the distinction is formally encoded, by morphological, syntactic and phonological means. The reason for assuming this focus distribution is that the speaker of (7-b) used several linguistic means to signal it. To begin with, the speaker chose to refer to *Sue* by means of the pronoun *she*. A pronoun usually refers to an accessible antecedent, i.e. one that has been recently mentioned in the discourse or is contextually evoked, and is therefore unfocused. Second, *a party* is an indefinite noun phrase, which is normally used to add a new, and therefore focused, element to the discourse. Furthermore, the word order is such that the focused information is located at the end of the clause. This is in accordance with the canonical word order for a clause in English (and Dutch) in which unfocused information precedes focused information. Deviation of the normal word order often results in a distinct pragmatic effect (Rochemont and Cullicover 1990; Rats 1996); cf. *To a party she went.* Finally, and most importantly for the discussion at hand, the speaker chose to put a pitch accent on *party*. This illustrates the general tendency for focused information to be pitch accented, whereas unfocused information is not (Birch and Clifton 1995; Brown 1983; van Donzel 1999; Fowler and Housum 1987; Horne 1991; Kruyt 1985; Nooteboom and Terken 1982; Nooteboom and Kruyt 1987). This focus-accent correspondence forms the point of departure for our grammar for pitch accent placement.

If we alter the focus cues of an utterance, its focus distribution, and consequently its interpretation, will be affected to some extent. For instance, changing the pronoun *she* to the proper noun *Sue*, as in (8-a), renders the utterance somewhat formal. On the other hand, using a pronoun for the focused information, as in (8-b), makes it an incoherent answer, as its antecedent cannot be identified by the listener.[10] A similar effect arises from changing the pitch accent distribution in (7-b) to (8-c). Apparently, the pitch accent on *party* is essential for signalling the focus distribution required by preceding the wh-question. Putting a pitch accent on old information may result in pragmatic side-effects, as in (8-d), which appears to imply something about another person than Sue as well.

(8) a. Sue went to a party.

b. She went there.

c. $\text{S}\overset{*}{\text{u}}\text{e}$ went to a party.

d. $\text{S}\overset{*}{\text{u}}\text{e}$ went to a $\text{p}\overset{*}{\text{a}}\text{rty}$.

[10] Assuming there is no additional context from which the antecedent can be inferred. The antecedent of *there* would be clear if it were accompanied by a pointing gesture.

The listing of focus cues above is by no means exhaustive; different languages have shown to employ different strategies (Foley 1994). Moreover, there is no straightforward, fixed, or one-to-one correspondence between focus and its linguistic cues, because they may serve to signal other functions or may be bound by other grammatical constraints. For instance, an *indefinite* noun phrase may be *unfocused*, when its indefiniteness is meant to signal a generic reading instead of a specific one. For instance, *a printer* in (9) refers to a generic, or prototypical printer, and not to a specific individual we can point at.

(9) Printers are a good example of unreliable technology.
A printer <u>will always break down when you need it the most</u>.

Likewise, the preferred word order with unfocused preceding focused information may be disturbed due to purely syntactic constraints on word order (Section 6.3.7), or for stylistic purposes as in left-dislocation or similar marked word orders (Rochemont and Cullicover 1990).

(10) It is <u>Sue</u> who went to a party.

Often, there will be interaction among different cues. For instance, although a pronoun normally refers to unfocused information, it may be focused by giving the pronoun a pitch accent (Verhagen 1986). This often occurs in a context where previously introduced discourse entities are contrasted.

(11) She* went to the party (while I had to stay home).

The linguistic realization of focus poses a challenge to linguistic theories, and is an important aspect of NLG and SLG. Although some aspects of the relation between focus and word order will be addressed in Chapter 6, our work is primarily concerned with the accent-focus relation. Important for the discussion at hand is that there is no one-to-one correspondence between the focus distribution of an utterance and its pitch accent distribution either. Consider example (12). Given its similarity to (7), it is reasonable to assume that the whole phrase *a very big party* is focused. Nevertheless, the word *big* remains unaccented. In fact, the determiner *a* also remains unaccented in both examples.

(12) a. Where did Sue go?

b. She went to <u>a ve*ry big pa*rty</u>.

A grammar of pitch accent placement has to account for these type of observations; they will be addressed in Section 2.5.3.

We continue with the third point of the definition of information structure: the it depends on the context. This has been demonstrated by the wh-question in (7), which determines the focus distribution in the answer. This raises the question where the focus distribution comes from. In most cases, the relation between the focus distribution and the context is less obvious. It was already mentioned that the focus-background distinction is closely related to the given-new distinction. However, the focus distribution cannot be equated to, or derived from, the given-new distinction. In (13), for instance, *Mike* and *Sue* are new information at the start of the utterance, and consequently focused, as indicated by the pitch accents. Even though the second instance of the word *Sue* can be assumed to convey old information, as it was recently mentioned, it remains focused and accented. A possible explanation is that the second time, it is focused to contrast it with the elements of a set that was introduced earlier (Theune 1997b; Theune 1997a; van Deemter and Odijk 1997). Thus contrast may be one of the reasons that old information is nevertheless focused.

(13) Both M$\overset{*}{\text{i}}$ke and Su$\overset{*}{\text{e}}$ like h$\overset{*}{\text{i}}$king in the mo$\overset{*}{\text{u}}$ntains, although Su$\overset{*}{\text{e}}$ only likes going d$\overset{*}{\text{o}}$wn.

Likewise, new information is not always focused. An example of this is (14). Although *the car* is a new piece of information, in the sense that the word has not been mentioned yet, it is unfocused. Apparently, the utterance, or rather the speaker, assumes that the listener knows that a Roll Royce is a kind of car, and can infer that *the car* is just another way of referring to a previously mentioned discourse entity, without adding any important information.

(14) The Rolls Royce was severely damaged.
It $\underline{\text{took s}\overset{*}{\text{i}}\text{x w}\overset{*}{\text{e}}\text{eks to rep}\overset{*}{\text{a}}\text{ir}}$ the car.

Examples like these indicate that the focus distribution is not equivalent to or derivable from the difference between given and new information in any obvious way. They have motivated some researchers to propose a classification of *givenness* on a more refined scale (instead of the binary distinction focus versus background) with classes like *brand-new*, *inferable*, *situationally invoked*, etc. (Prince 1981; van Deemter and Odijk 1997). These are interesting options to explore, but at our current state of knowledge it is hard to relate these categories to the accent distribution (see however van Donzel (1999)).

There may be many (pragmatic) reasons for a speaker to chose a particular distribution of focus. For current purposes, however, we will leave questions about the reasons for a particular focus distribution to semantic and pragmatic research, and concentrate on the question how focus is *realized*, in particular by means of pitch accent placement. In other words, we will assume that focus is a theoretical primitive, and abstract away from the underlying motivations that the speaker may have for a particular focus distribution. This entails that the underlying motivation for a particular focus distribution is not allowed to affect the distribution or the realization of accents. Evidence for this is provided by a study that addressed (among other thing) the relation between informational categories and their prosodic realization in Dutch (van Donzel 1999, p. 152):

"Measurements of the actual realization of the pitch accents in fun-

> damental frequency revealed that the type of information on which the pitch accent is realized has no effect. That is, pitch accents realized on new information are not realized differently from those realized on inferable information or modifiers, for instance."

Likewise, Kruyt (1985, ch. 6) reported on a perception experiment showing that neither the particular type of pitch accent, nor the presence or absence of a subsequent prosodic boundary has any effect on the interpretation as 'given' or 'new'.

2.5.2 Argument structure

In Gussenhoven (1992b), argument structure is defined as follows:

> "[T]hree types of constituent are assumed: predicates, arguments (also 'inherent arguments') and modifiers (also 'noninherent arguments'). Every predicate constitutes a clause, which will have from zero to three arguments, and any number of modifiers. The distinction between arguments and modifiers corresponds with the distinction between 'inherent' roles and 'noninherent' roles, that is, between roles that are specified in the lexical entry of a predicate and those that are not. Expressions that instantiate listed roles are inherent arguments, or simply arguments, while expressions filling other roles are noninherent arguments, or modifiers."

The distinction between arguments and modifiers plays a role in morphology, syntax and phonology. For instance, it is important in NLG, because it constrains the word order in a clause (Section 6.3.7). Here, we will concentrate on its effect on accent placement. The following examples, which were taken from Gussenhoven (1992b), illustrate a systematic discrepancy between the focus distribution and the accent distribution, which occurs in English (15), as well as in Dutch (16). In (15-a), the verb *smoked* is focused and accented, whereas in (15-b), the verb *remained* is focused but *not* accented. The problem is to explain why in the (a) examples an accent on the verb is required to signal its focus, whereas in the (b) examples an accent on the verb can be omitted without affecting its focus.

(15) What did John do?

a. John $\underline{\text{sm}\overset{*}{\text{o}}\text{ked in the t}\overset{*}{\text{e}}\text{nt}}$

b. John $\underline{\text{remained in de t}\overset{*}{\text{e}}\text{nt}}$

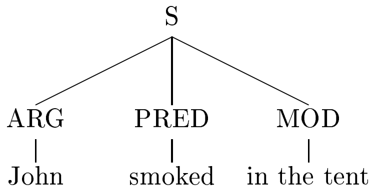
(16) Wat deed Jan?

a. Jan $\underline{\text{heeft in de t}\overset{*}{\text{e}}\text{nt ger}\overset{*}{\text{o}}\text{okt}}$

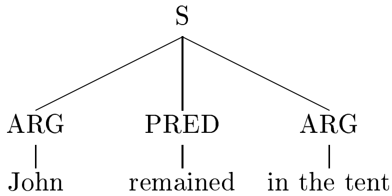
b. Jan $\underline{\text{is in de t}\overset{*}{\text{e}}\text{nt gebleven}}$

Gussenhoven's solution is to take the argument structure of a clause into account (Gussenhoven 1984; Gussenhoven 1992b). The argument structures corresponding to (15-a) and (15-b) are given in (17-a) and (17-b), respectively.

(17) a.



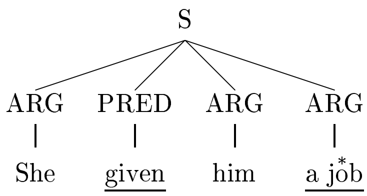
b.



The lack of an accent on the predicate is attributed to the fact that it is adjacent to a focused argument. Following Gussenhoven's terminology, we can say that the predicate *remained* and the argument *in the tent* merge into a single *focus domain*, which is satisfied with just one accent.

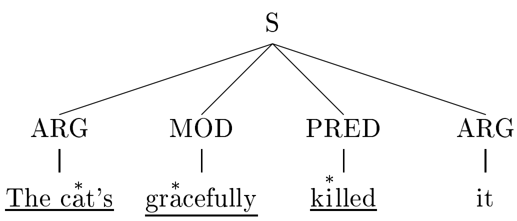
Actually, the situation is more complex, because unfocused elements are ignored. This is illustrated in (18), where the intervening unfocused argument does not prevent the predicate *given* from forming a single focus domain with its focused argument *a job*.

(18) Any news about Mary and John?



However, an intervening *focused* modifier does prevent a focused predicate from forming a single focus domain with its argument.

(19) Where's the canary?



Gussenhoven proposed to explain these patterns by the following generalisation.

(20) **Sentence Accent Assignment Rule** (SAAR)

If focused, every predicate, argument, and modifier must be accented, with the exception of a predicate that, discounting unfocused constituents, is adjacent to an argument

Experimental evidence in support of this generalisation is reported in Gussenhoven (1983b) and Birch and Clifton (1995). We will use SAAR as the basis for our accent placement algorithm. There are a number of exceptions to SAAR (Gussenhoven 1992b, Appendix), some of which will be addressed in Section 7.3.1.

2.5.3 Rhythm

Example (21) repeats another discrepancy between the focus distribution and the accent distribution.

(21) a. Where did Sue go?

b. She went to $\underline{\text{a } \overset{*}{\text{v}}\text{ery big } \overset{*}{\text{p}}\text{arty}}$

Although there is no reason to assume that *big* is not focused, it is unaccented. Again, the problem is to explain why the accent on *big* can be omitted without affecting the focus interpretation. Following Gussenhoven (1991a), we will assume that the accent on *big* may be deleted to obtain a more rhythmical distribution of pitch accents, i.e. a more regular pattern of accented and unaccented words. Responsible for this is an accent deletion rule called the *Rhythm Rule*.

(22) **Rhythm Rule** (RR)

$* \;\rightarrow\; \emptyset \;/\; [_\varphi \;\; * \;\; \_ \;\; * \;\;]$

This rule allows the deletion of an accent if it is surrounded by accents within the same phonological phrase, the prosodic constituent introduced in Section 2.2). Application of the rule is optional, and it can be applied more than once. The net effect of exhaustive application is that only the outermost accents within a phonological phrase are preserved. Empirical evidence for this rule in English is reported in Horne (1990); see also Grabe, Warren, and Nolan (1994).

The phonological phrases corresponding to (21-b) are as follows, where the former accent deleted by RR is indicated by ø.

(23) $[_\varphi$ She went $]$ $[_\varphi$ $\underline{\text{to a } \overset{*}{\text{v}}\text{ery } \overset{\text{ø}}{\text{b}}\text{ig } \overset{*}{\text{p}}\text{arty}}$ $]$

Notice that an alternative phonological phrasing, such as (24), would not lead to the desired result.

(24) [$_\varphi$ She went] [$_\varphi$ to a vèry] [$_\varphi$ bìg pàrty]

Moreover, incorrect phonological phrasing may trigger an accent deletion that gives rise to an unintended focus distribution, as in (25).

(25) a. Where did Sue go?

b. ?[$_\varphi$ She went] [$_\varphi$ to a bìg party in Àmsterdam]

Thus, in order to prevent incorrect accent deletions, it is crucial that the boundaries of phonological phrases are correctly determined. Therefore, phonological phrasing in Dutch will be investigated in Chapter 3.[11]

2.5.4 A grammar for pitch accent placement

To sum up, our preliminary grammar for pitch accent placement requires four types of input:

- the focus distribution
- the argument structure (predicates, their arguments, and optional modifiers)
- the prosodic structure, minimally consisting of the phonological phrases
- the surface string

Given this input, the grammar consists of just two rules:

1. the Sentence Accent Assignment Rule (SAAR) to account for the focus-accent relation
2. the Rhythm Rule (RR) to account for rhythmical adjustments to the pitch accent distribution

An example of its application is given in (26).

This simple grammar forms the basis for the elaborate pitch accent placement algorithm in our SLG system, which will be described in Chapter 7.

[11]There is another type of stress shift, or pitch accent retraction, that appears to serve rhetorical purposes, and is called *rhetorical stress shift* in (Gussenhoven 1983a). For example, *ongelòoflijk* 'unbelievable' can become *òngelooflijk* in a phrase like *wat een òngelooflijke fòut!* 'What an unbelievable mistake!'. We abstracted away from this phenomenon, as it did not occur in our application domains.

(26) *Context:* What did you do last evening?

S

Input: ARG PRED ARG

[$_\varphi$ we <u>vistited</u>] [$_\varphi$ <u>a very big party</u>]

SAAR: we v$\overset{\emptyset}{\text{i}}$sited a v$\overset{*}{\text{e}}$ry b$\overset{*}{\text{i}}$g p$\overset{*}{\text{a}}$rty

RR: we visited a v$\overset{*}{\text{e}}$ry b$\overset{\emptyset}{\text{i}}$g p$\overset{*}{\text{a}}$rty

Output: we visited a v$\overset{*}{\text{e}}$ry big p$\overset{*}{\text{a}}$rty

2.6 Dutch syntax

There are at least three reasons why we need syntactic structures in the current work. First, correct application of RR depends on correct phonological phrases, which are derived from the syntactic structure of an utterance by means of mapping rules (Section 2.2). Second, syntactic information is essential for intonational phrasing. Third, syntactic knowledge is required for generating syntactically well-formed output in our SLG system (see Chapter 6). However, there are many different approaches to the syntactic analysis of Dutch, ranging from traditional structuralistic grammars (Haeseryn et al. 1997; Rijpma and Schuringa 1978; van den Toorn 1984) to modern transformational (Model 1991; Bennis and Hoekstra 1989; Zwart 1993), categorial (Bouma and van Noord 1995), and unification-based grammars (Bouma 1996). Rather than adopting a particular syntactic theory, we made some minimal assumptions about the phrasal structure that can be satisfied by most theories. This keeps our results compatible with different syntactic views and guarantees a wide range of application.

The following two sections illustrate our assumptions on the basis of a some examples. Admittedly, this is rather simplistic and ignores complications; a more elaborate account can be found in Chapter 6 about syntactic realization.

2.6.1 S/VP structure

We assume clauses (S) of two types: main clauses (27) and subordinate clauses (28). Both minimally contain a verb phrase (VP). Main clauses have a finite verb at the syntactically second position, directly dominated by the S-node. Declarative main clauses (and wh-questions) have a preposed constituent at the first position (27-a), whereas polar questions (and imperatives) do not allow this (27-b). Extraposed prepositional phrases (PP), clauses and heavy noun phrases (NP) may occur after the VP (27-c).

(27) a. ($_S$ ($_{NP}$ u) ($_V$ kunt) ($_{VP}$ ($_{PP}$ in Amsterdam) ($_{NP}$ de
you can in Amsterdam the
stoptrein) ($_V$ nemen)))
slow-train take
'you can take the slow train in Amsterdam'

b. ($_S$ ($_V$ kan) ($_{VP}$ ($_{NP}$ ik) ($_{AdvP}$ daar) ($_{NP}$ de stoptrein)
can I there the slow-train
($_V$ nemen)))
take
'can I take the slow train there?'

c. ($_S$ u kunt ($_{VP}$ de stoptrein nemen) ($_{PP}$ om twaalf uur
you can the slow-train take at twelve hour
vijf))
five
'you can take the slow-train of five past twelve'

Subordinated clauses have all verbs at the end of the VP. Bound subordinated clauses start with a complementizer (C) (28-a), and relative subordinated clauses with a relative expression (28-b).

(28) a. ($_S$ ($_C$ dat) ($_{VP}$ u in Amsterdam de stoptrein kunt
that you in Amsterdam the slow-train can
nemen))
take
'that you can take the slow train in Amsterdam'

b. ($_{NP}$ de stoptrein ($_S$ ($_{NP}$ die) ($_{VP}$ u in Amsterdam
the slow-train that you in Amsterdam
neemt)))
take
'the slow train that you take in Amsterdam'

2.6.2 NP structure

Noun phrases are headed by (possibly empty) noun, may be preceded by a determiner (Det), a quantifier (Q), a numeral phrase (NumP)[12], and premodifying adjectival phrases (AP), a premodifying NP[13], and followed by postmodifying prepositional phrases (PP) or relative clauses (S).

[12] See Section 6.2.3.4
[13] See Section 7.3.2

(29) a. $(_{NP}$ $(_{Det}$ de) $(_{AP}$ $(_{A}$ gele)) $(_{N}$ bloemen) $(_{PP}$ $(_{P}$ in) $(_{NP}$
the yellow flowers in
het gras)))
the grass

b. $(_{NP}$ $(_{Q}$ alle) $(_{N}$ bloemen) $(_{S}$ die geel zijn))
all flowers that yellow are

c. $(_{NP}$ $(_{NP}$ station) $(_{N}$ Amsterdam-Centraal))
station Amsterdam-Central

2.6.3 Pronouns are determiners

In contrast to the above analyses, which are rather conventional, we assume that pronouns are actually determiners, following the analysis by Coppen (1991, p. 100-112), which is compatible with the DP analysis of Abney (1986). We repeat some of Coppen's arguments below.

First, a so-called demonstrative pronoun (Haeseryn et al. 1997, ch. 5) can precede a noun (30-a) in the same way a determiner can precede a noun (30-b), suggesting an identical syntactic structure. For independent demonstrative determiners, we assume the NP has an empty head (30-c).

(30) a. $(_{NP}$ $(_{Det}$ die) $(_{N}$ boeken))
those books

b. $(_{NP}$ $(_{Det}$ de) $(_{N}$ boeken))
the books

c. $(_{NP}$ $(_{Det}$ die) $(_{N}$ $\emptyset$))
those

Second, both so-called personal (31-b) and demonstratives pronouns (31-c) can be followed by a quantifier, and are similar to a determiner in this respect (31-a), whereas nominal heads do not have this possibility (31-d).[14]

(31) a. de twee boeken
the two books

b. wij twee(en) hebben dat gedaan
we two(+plur) have that done
'the two of us did that'

c. die twee hebben dat gedaan
those two have that done
'those two did that'

[14] Coppen remarks that this pattern occurs only with *plural* pronouns, and with numerals below ten.

d. *de kinderen twee hebben dat gedaan
the children two have that done

The third argument, which goes back to Postal (1969), is that pronouns can appear in the same position as determiners.[15]

(32) a. wij fietsenmakers
we bicycle-repairmen

b. jij kleine wijsneus
you little know-it-all

c. ik arme stumper
I poor wretch

See Coppen (1991) for additional arguments and a more extensive analysis.

2.7 Summary

This chapter presented the theoretical framework for our work on pitch accent placement and intonational phrasing in SLG, providing a more precise interpretation of these issues in terms of phonological theories, and a first suggestion as to how we may account for them.

In Section 2.2, prosody was defined as the hierarchical organisation of segments into higher units, the patterns of relative prominence among these units, and their phonetic reflection in terms of phonetic parameters. Prosodic phonology was presented as the study of prosody from a phonological point of view. We introduced its core concept: a prosodic structure consisting of seven prosodic categories derived by mapping rules, and serving as the interface from morphology, syntax and semantics to phonology. We discussed the well-formedness constraints imposed on prosodic structures by the prosodic hierarchy and the Strict Layer Hypothesis. We briefly reviewed two arguments for the existence of prosodic constituents. First, to explain segmental phenomena: certain segmental rules can only apply within – or at the boundary of – prosodic constituents of a certain rank. Second, to explain the distribution of prosodic boundaries in an utterance: the higher the rank of a predicted prosodic constituent, the more prominently its boundaries should be perceived as a result of phonetic cues such as pause and lengthening of the preboundary syllable (preboundary lengthening).

In Section 2.3, intonational phonology was introduced as the study of intonation from a phonological point of view, relying on a division of labour between phonological description and phonetic implementation. We summarised the intonational phonology of Dutch developed by Gussenhoven, consisting of a tone string of tonal events that is aligned with a segmental string by means of

[15] Coppen convincingly argues against the view that these constructions should be analysed as containing an appositional NP.

tiers. The tonal events are either pitch accents (H*L, L*H, or H*LH) or boundary tones (%H, %L, H%, or L%), both of which are composed of the primitive tones H and L. According to the association principles, every starred tone of a pitch accent is linked to a stressed syllable, and every boundary tone to an AD-boundary, while the remaining tones either spread, or move up to the next tone segment or AD-boundary.

The model is transformational in the sense that the underlying structure is rewritten to a surface structure. In the underlying representation, every utterance is exhaustively parsed into AD's, and every AD contains exactly one pitch accent. AD-restructuring allows for merging of AD's. Tonal rules like Tone Linking may modify the tone string, and tone morphemes like Downstep may be attached. The structure is mapped onto an actual pitch contour by the phonetic implementation module, taking into account tone morphemes like Downstep.

We also followed Gussenhoven in his view that the intonational phrase cannot be incorporated into the prosodic hierarchy, but must be analysed as a independent domain called the association domain (AD).

In Section 2.4, it was shown that the model provides a precise account for variation along three dimensions: (1) intonational phrasing (the partition of an utterance into one or more AD's); (2) pitch accent placement (the number of pitch accents and their association with stressed syllables in the utterance); (3) tune choice (the choice of the kind of pitch accents and boundary tones, and the application of tonal rules and tone morphemes). The challenge of *generating intonation* was defined as intonational phrasing, pitch accent placement and tune choice in such way that the resulting intonational form fits the linguistic structure and context of an utterance. The contribution of the intonational model in this respect is (1) that allows us to focus on generating intonational forms at a phonological level, abstracting away from the details of phonetic implementation, and (2) that it defines the well-formed intonational descriptions, ruling out those forms that are unacceptable on purely phonological grounds.

In Section 2.5, we reviewed three factors that affect pitch accent placement: focus, argument structure, and rhythm. We presented a preliminary grammar for pitch accent placement based on the work of Gussenhoven. It requires as input the argument structure (predicates, their arguments, and optional modifiers), the focus distribution, the prosodic structure (the phonological phrases), and the surface string. It consists of just two rules: (1) the Sentence Accent Assignment Rule (SAAR) to account for the focus-accent relation, in particular for focused yet unaccented predicates, and (2) the Rhythm Rule (RR) to account for rhythmical adjustments by deletion of pitch accents within phonological phrases. It was explained that in order to prevent incorrect accent deletions, it is crucial that the boundaries of phonological phrases are correctly determined. Finally, Section 2.6 described the syntactic analysis of Dutch used throughout our work, relying on minimal assumptions about the phrasal structure that are compatible with most syntactic theories.

Chapter 3

Phonological Phrasing

3.1 Introduction

Given the research objective to design, build, and test a linguistically-based SLG system for generating the most acceptable pitch accents and intonational boundaries, a preliminary grammar for pitch accent placement was presented in Section 2.5.4. Part of this grammar is an accent deletion rule called the *Rhythm Rule* (RR).

(1) **Rhythm Rule** (RR)

$$* \quad \rightarrow \quad \emptyset \; / \; [_{\varphi} \quad * \quad \_ \quad * \quad]$$

This rule deletes an accent occurring between two other accents if all of them are within the same phonological phrase (φ) (Section 2.2). It accounts for the observation that accentuation is influenced by rhythmic considerations, that is, some accents are deleted to obtain a more rhythmical pattern of accented and unaccented syllables. We will follow the strong hypothesis that the rule is applied both obligatory and exhaustively.

RR crucially depends on starting from the right φ's. Consider the example in (2). We assume that the whole phrase is focused, which will be the default assumption for all examples in this chapter. As a result, all content words are initially accented. Given this accent distribution, RR correctly deletes the accent on *afkomstige* (as indicated by ø) under the assumption that the whole phrase is a φ (2-a). However, if we were to assume that *afkomstige* is followed by a φ-boundary, then RR would have been blocked, and would have failed to delete the accent (2-b). This type of failure will be called an *undershoot error*.

(2) a. [$_\varphi$ een uit N$\overset{*}{\text{e}}$derland afk$\overset{ø}{\text{o}}$mstige pl$\overset{*}{\text{a}}$nt]
a from Netherlands originating plant
'a plant originating from the Netherlands'

b. ?[$_\varphi$ een uit N$\overset{*}{\text{e}}$derland afk$\overset{*}{\text{o}}$mstige] [$_\varphi$ pl$\overset{*}{\text{a}}$nt]

Analogously, the accent on *Nederlandse* is correctly preserved in (3-a) as long as a φ-boundary after *zeldzame* is assumed. Without this boundary, RR would have deleted the accent (3-b), and *Nederlandse* could only have been interpreted as unfocused, which contradicts our default assumption that the whole phrase is focused. This will be called an *overshoot error*.

(3) a. [$_\varphi$ een z$\overset{*}{\text{e}}$ldzame] [$_\varphi$ N$\overset{*}{\text{e}}$derlandse pl$\overset{*}{\text{a}}$nt]
a rare Dutch plant

b. ?[$_\varphi$ een z$\overset{*}{\text{e}}$ldzame N$\overset{ø}{\text{e}}$derlandse pl$\overset{*}{\text{a}}$nt]

Evidently, our algorithm for pitch accent placement using RR requires the φ-boundaries to be correctly determined. In this chapter, therefore, we will investigate phonological phrasing in Dutch. We will start in Section 3.2 with reviewing some of the better-known proposals for φ-phrasing, arguing that these are inadequate for Dutch. On the basis of a careful examination of examples of φ-phrasing in a number of syntactic structures, we will propose a new φ-phrasing algorithm (for Dutch) that is observationally adequate in Section 3.3. Next, we will report on a corpus-based evaluation of this algorithm in Section 3.4. We will conclude with a summary in Section 3.5.

3.2 A review of phonological phrase definitions

There are quite a few proposals for the derivation of φ's. Some are intended to be language-specific, e.g. Ferreira (1993), Gee and Grosjean (1983), Hayes (1986), Selkirk (1984, 1996), van Wijk (1987a), Zec and Inkelas (1990) for English and Neijt (1984) for Dutch, whereas others are claimed to be universal (Nespor and Vogel 1986).[1] We will first discuss some preliminary considerations with respect to the syntactic input and the evidence for φ-phrasing. Next, we will review two representative proposals, by Nespor and Vogel (1986) and by Hayes (1986), and show that these are inadequate in the case of Dutch.

3.2.1 Preliminaries

Recall from Section 2.2 that prosodic constituents are derived from the morpho-syntactic and semantic structure of an utterance by means of mapping rules. Nearly all φ-phrasing algorithms require as input a syntactic tree based on some version of $\overline{X}$-theory (Jackendoff 1977); semantic information appears to

[1] See Inkelas and Zec (1990) for phonological phrasing in some other languages.

be unnecessary for φ-phrasing. Although different authors may assume slightly different syntactic structures for certain inputs, most algorithms are insensitive to this variation in analysis, since they refer only to noncontroversial categories, like lexical categories (X), phrases (XP), and their dominance relations. The syntactic structures assumed in the current work (Section 2.6) provide sufficient information for all algorithms discussed here.

There are two types of evidence for prosodic constituents. First, there are phonological rules that are sensitive to the edges of a certain prosodic category (Section 2.2). Nespor and Vogel (1986) mention a number of segmental rules – for different languages – that apply within a φ, but not across a φ-boundary. For Dutch, however, no segmental rules are known that apply *exclusively* within φ's. To cite (Booij 1995, p. 146) in his survey of Dutch phonology: "For Dutch, the issue of the prosodic domains of rules above the level of the prosodic word is an underresearched area. [... T]he application of P-rules [EM phonological rules conditioned by phonological conditions only] above the word level decreases as the relevant prosodic domain of application becomes larger. In other words, I do not assign a particular prosodic domain to P-rules [...]". Hence, we will not use this type of evidence.[2]

Second, prosodic phonology predicts that the higher the rank of a prosodic category, the larger the amount of preboundary lengthening and the likelihood of a pause. Phonetic evidence for φ's in Dutch in the form of preboundary lengthening can be found in Hofhuis, Gussenhoven, and Rietveld (1995), and Gussenhoven and Rietveld (1992a), although this work addresses only a few (uncontroversial) φ-boundaries. Phonetic evidence may become stronger with the advent of Dutch speech corpora.

In the absence of reliable segmental evidence or easily obtainable phonetic evidence, the only clear evidence we have is the applicability of RR. This then, brings us to some remarks about using RR as a diagnostic tool for φ-phrasing. First, its application depends to a certain degree on speaking style and speaking rate. In slow and careful speech, it may apply optionally, while in fast and sloppy speech, it may at times even cross φ-boundaries (Hayes 1986, p. 215). Second, application may be to be sensitive to the spacing between accented syllables. Hayes (1984) claims the existence of *eurhythmy*, the tendency to produce equal spacings of accents. Gussenhoven (1991a) acknowledges the existence of eurhythmy, but argues that its role is rather limited. Bearing the above in mind, we will assume that all examples are spoken at an average speaking rate, abstracting away from the possible effect of very fast or slow speaking. Furthermore, we will as much as possible compare alternative φ-phrasings and the resulting accentuations, using the symbol '?' to the mark the one that is comparatively less acceptable. In other words, the claim is not that the questionable examples are wrong, but rather that they are less acceptable compared with the preferred ones. Finally, unless mentioned otherwise, all examples are assumed to have broad focus.

[2] However, see Loots (1983) and Menert (1994).

3.2.2 Phonological phrasing according to Nespor & Vogel

Nespor and Vogel (1986) propose the following algorithm to derive φ's.

(4) **Phonological phrase formation**

a. *φ-domain*
The domain of φ consists of a C which contains a lexical head (X) and all C's on its nonrecursive side up to the C that contains another head outside of the maximal projection of X.

b. *φ-construction*
Join into an n-ary branching φ all C's included in a string delimited by the definition of φ.

The letter 'C' stands for *Clitic Group*, which is a content word with optionally attached clitic words. It is irrelevant for the current discussion[3], so it can be interpreted as 'word'.

When we try to apply this algorithm to Dutch, we encounter a number of problems. The first problem is also recognised by Nespor and Vogel (1986, p. 186, footnote 2): Dutch syntax shows both left-recursive and right-recursive structures, so it is not clear what the nonrecursive side should be. Similar to an English NP, a Dutch NP can have an arbitrary number of embedded PP's and relative clauses to the right of its head, just as in English. Assuming that Dutch is left-recursive predicts that all these PP's and S's belong to one and the same φ, which is of course an unintended result. We are therefore forced to assume that the left side is the nonrecursive one.

Another minor issue involves the definition of lexical head. According to Nespor & Vogel, only V, N and A are lexical heads. Even if we are prepared to categorise adverbs as A, there remain other other content words that trigger a φ-boundary. Example (5) contains two numerals, both of which are capable of carrying two pitch accents. Without counting numerals as lexical heads, we get the huge φ in (5-b), in which RR can apply three times, resulting in a totally unacceptable accent distribution.[4] In contrast, counting numerals as lexical heads results in an acceptable accent distribution (5-c).

(5) a. van de ($_{Num}$ vijfennegentig) hebben we er ($_{Num}$
of the five-and-ninety have we there
tweeenveertig) afgewezen
two-and-forty rejected
'out of the 95, we rejected 42'

b. *[$_\varphi$ van de $\overset{*}{\text{vi}}$jfenn$\overset{\emptyset}{\text{e}}$gentig hebben we er tw$\overset{\emptyset}{\text{ee}}env\overset{\emptyset}{\text{ee}}$rtig $\overset{*}{\text{a}}$fgewezen]

---

[3]However, the relation between clitics and φ's will be discussed in Section 3.2.3.

[4]It is possible to claim that both numerals modify a nominal head (N) that is empty, and that empty heads still count as lexical heads for the purpose of phonological phrasing. However, the assumption that empty elements (traces) affect the phonological structure is explicitly rejected by Nespor and Vogel (1986, ch. 2).

c. [$_\varphi$ van de v$\overset{*}{\text{i}}$jfenn$\overset{*}{\text{e}}$gentig] [$_\varphi$ hebben we er tw$\overset{*}{\text{e}}$eenv$\overset{*}{\text{e}}$ertig]
[$_\varphi$ $\overset{*}{\text{a}}$fgewezen]

Yet another problem involves words that are not included in a φ domain and consequently are not dominated by a φ node at all. In Dutch – an SOV language – auxiliary verbs can occur in the sentence-final verb cluster; see *zou*, *moeten* en *hebben* in (6). Since auxiliaries do not count as lexical heads, they do not trigger a φ-domain of their own. Neither are they included in the φ-domain of the main verb *gezien*, because they reside at its recursive side. As a consequence, the auxiliaries remain unparsed (6-b). This then, violates the exhaustive parsing condition on prosodic constituent structures (Nespor and Vogel 1986, p. 7, Principle 2).

(6) a. ... dat ik dat ($_N$ probleem) ($_V$ gezien) zou moeten
that I that problem seen should must
hebben
have
'... that I should have seen that problem'

b. ... dat ik dat probleem] [$_\varphi$ gezien] zou moeten hebben

The most serious problem, however, is one of indeterminacy. Consider (7), consisting of a lexical head V, followed by an NP containing a lexical head A and a lexical head N. Deriving the φ-domain for *rode* results in (7-b), while deriving the φ-domain for *bloem* gives (7-c). Clearly, these φ-domains are incompatible with the Strict Layer Hypothesis (Section 2.2), which prevents recursive embedding of prosodic constituents of the same category.

(7) a. de plant ($_V$ draagt) ($_{NP}$ ($_{Det}$ een) ($_{AP}$ ($_A$ rode)) ($_N$
the plant carries a red
bloem))
flower

b. de plant draagt [$_\varphi$ een rode] bloem

c. de plant draagt [$_\varphi$ een rode bloem]

Now assume, for the sake of argument, that we always give priority the *largest* φ-domain. So in (7), we take (7-c) rather than (7-b). This implies, however, that the φ-domain for a lexical N includes *all* of its premodifiers. For example, the NP in (8) would be one big φ. Moreover, as the number of premodifiers for a noun is in theory unlimited, it follows that the length of a φ is in theory unlimited as well. This conclusion is absurd, as it predicts that accents on *voltallige* and *voorzitter* would suffice to focus the whole NP. A similar argument can be made for the premodifiers of V.

(8) *[$_\varphi$ de door de voltallige raad van bestuur met
the by the entire board of management with
meerderheid van stemmen gekozen nieuwe voorzitter]
majority of votes elected new chairman

To sum up so far, the φ-algorithm by Nespor & Vogel is inadequate for Dutch because it is not evident what the nonrecursive side should be, the class of lexical heads is to narrow, and sentence final auxiliaries are left unparsed. Moreover, the algorithm is non-deterministic and cannot be easily repaired.

In addition to algorithm (4), Nespor & Vogel postulate an optional restructuring rule.

(9) **φ-restructuring (optional)**

A nonbranching φ which is the first complement of X on its recursive side is joined into the φ that contains X.

Restructuring accounts for observations of stress shift (Nespor and Vogel 1986, p. 178). Stress shift is assumed to result from a φ-bound rule that moves the main stress position of a word backwards in case of a *stress clash*, i.e. when the main stress of a word is too close to the main stress of the next word. In (10), for example, *quickly* forms a nonbranching φ that is the first complement of *reproduce* on its recursive side, and is therefore joined in a single φ with *reproduce*. Moreover, the normal main stress of *reproduce* clashes with the main stress of *quickly*. As a consequence of these two circumstances, stress shifts from the penultimate to the initial syllable of *reproduce*.[5]

(10) a. Given the chance, rabbits
($_{VP}$ ($_V$ reproduce) ($_{AdvP}$ ($_{Adv}$ quickly)))

b. Given the chance, rabbits [$_\varphi$ réproduce quíckly]

In contrast, the complement *very quickly* in (11) is branching, which prevents restructuring, and therefore stress shift is blocked.

(11) a. Given the chance, rabbits
($_{VP}$ ($_V$ reproduce) ($_{AdvP}$ ($_{Adv}$ very) ($_{Adv}$ quickly)))

b. Given the chance, rabbits [$_\varphi$ reprodúce] [$_\varphi$ véry quickly]

A similar pair of examples, involving a branching coordination of two adverbs, is given in (12) and (13).

(12) a. John ($_V$ perseveres) ($_{AdvP}$ ($_{Adv}$ gladly))

b. John [$_\varphi$ pérseveres gládly]

[5] Nespor & Vogel use an accentmark (´) to indicate primary stress.

(13) a. John ($_V$ perseveres) ($_{AdvP}$ ($_{Adv}$ gladly) and ($_{Adv}$ diligently))

b. John [$_\varphi$ persevéres] [$_\varphi$ gládly] [$_\varphi$ and diligently]

However, the restructuring option gives the wrong predictions for Dutch. First, observe that an adjective like *razendsnel* can carry two accents.

(14) die auto is r$\overset{*}{\text{a}}$zendsn$\overset{*}{\text{e}}$l
that car is super-fast

Second, if *razendsnel* directly precedes the accented verb *gereden*, its second accent may be deleted (15). Our explanation is that both words belong to the same φ, and that the second accent of *razendsnel* is deleted by RR.

(15) a. we hebben ($_A$ razendsnel) ($_V$ gereden)
we have super-fast driven

b. [$_\varphi$ we hebben r$\overset{*}{\text{a}}$zendsn$\overset{\emptyset}{\text{e}}$l ger$\overset{*}{\text{e}}$den]

Now consider (16), where the verb precedes *razendsnel* (cf. *reproduce quickly*). According to (9), *razendsnel* is a nonbranching φ which is the first complement of the verb on its recursive (right) side, and is therefore joined into the φ that contains the verb. This allows RR to delete the first accent of *razendsnel*. However, (16-b) is far less acceptable than (16-c), if not unacceptable. Apparently, the φ-boundary between between the verb and the adjective is obligatory, and cannot be subject to φ-restructuring.

(16) a. we ($_V$ rijden) ($_A$ razendsnel)
we drive super-fast

b. ?[$_\varphi$ we r$\overset{*}{\text{i}}$jden r$\overset{\emptyset}{\text{a}}$zendsn$\overset{*}{\text{e}}$l]

c. [$_\varphi$ we r$\overset{*}{\text{i}}$jden] [$_\varphi$ r$\overset{*}{\text{a}}$zendsn$\overset{*}{\text{e}}$l]

In conclusion, both the initial algorithm for φ-phrasing and the φ-restructuring rule proposed by Nespor & Vogel show serious defects when applied to Dutch.

3.2.3 Phonological phrasing according to Hayes

A somewhat different definition of the phonological phrase for English is proposed by Hayes (1986), where the formation of Clitic Groups (17) precedes the construction of Phonological Phrases (18).

(17) **Clitic Group Formation**

a. Every content word (lexical category) belongs to a separate Clitic Group.
b. Definition: The *host* of a Clitic Group is the content word it contains.
c. Definition: X and Y *share category membership* in C if C dominates both X and Y.
d. Rule: Clitic words are incorporated leftward or rightward into an adjacent Clitic Group. The group selected is the one in which the clitic shares more category membership with the host.

(18) **Phonological Phrase Construction**[6] (English)

In the configuration ($_{XP}$ ...X YP ...)
a. The sequence (...X) obligatory occupies the same φ,
b. YP may optionally adjoin to the φ of X if it contains only one Clitic Group, and
c. all Clitic Groups unaffected by rules (a) and (b) form φ's.

Applying these definitions to the syntactic structure in Figure 3.1 gives the C's in (19-a) and the φ's in (19-b).

(19) a. [$_{C}$ he kept it] [$_{C}$ in a large] [$_{C}$ jar]

b. [$_{\varphi}$ he kept it] [$_{\varphi}$ in a large jar]

Hayes' definition of clitics is much broader than that of Nespor & Vogel: every function word counts as a clitic word. As a result, the problem with the unparsed auxiliaries in (6) does not occur, because they are incorporated in the Clitic Group of the content verb. At the same time, however, the notion of shared category membership introduces new problems with regard to the direction of incorporation. Consider the sentence in (20-a). It is important to notice that the syntactic structure above the PP is irrelevant with respect to Clitic Group Formation: whatever the VP and S structure may be, the preposition *in* will *always* share more category membership with the host *regen* than with *fietste*. Therefore, *fietste* and *in* must belong to different C's (20-b). However, this is contradicted by phonological evidence. First, the schwa of *fietste* can be deleted by the rule *Prevocalic Schwa Deletion* (20-c), which cannot apply across prosodic word (ω) boundaries (Booij 1995, p. 49), and by implication, not across a C boundary either. Second, resyllabification is likely, which can only occur within a ω.

[6] The rendering is slightly different from the original statement: φ instead of *P-phrase*, parenthesis instead of square brackets, XP and YP instead on X'' and Y'', and X instead of X^0. These notational variations do not alter its meaning.

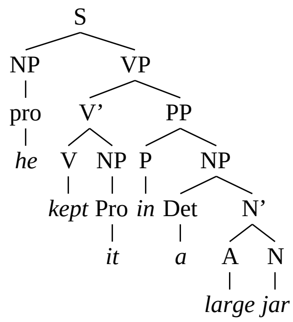


Figure 3.1: Syntactic tree for examples in (19).

(20) a. ze ($_V$ fietste) ($_{PP}$ ($_P$ in) ($_{NP}$ de regen))
she cycled in the rain

b. [$_C$ ze fietste] [$_C$ in de regen]

c. fietst in
/fitstə in/ ⟶ /fit-stin/

Analogously, the predicted C's for sentence (21-a) are given in (21-b), but are contradicted by the fact that resyllabification is certainly allowed here (21-c).[7]

(21) a. ik ($_V$ zie) ($_{NP}$ ($_{Det}$ het) ($_N$ water)) verdampen
I see the water evaporate

b. [$_C$ ik zie] [$_C$ het verdampen [

c. zie het
/si ət/ ⟶ /si-jət/

In conclusion, Hayes' algorithm for Clitic Group Formation based on maximal shared membership leads to false predictions for Dutch. Since Phonological Phrase Construction crucially depends on Clitic Group Formation, it follows that Phonological Phrase Construction makes false predictions as well.[8]

In support of his φ-construction algorithm, Hayes gives the following examples of stress shift.

(22) a. ($_{NP}$ ($_{AP}$ Jápanese) ($_N$ connéctions))

[7] The /j/ is inserted by the rule *Homorganic Glide Insertion*, which is normally applied within ω's, but appears to be across constituents higher as well in casual speech (Booij 1995, p. 150).

[8] This, however, is difficult to prove by means of RR, because clitics are virtually always unaccented.

b. ($_{NP}$ ($_{NP}$ Tóscanini's) ($_{N}$ íce cream))

c. ($_{AP}$ ($_{Adv}$ évidently) ($_{A}$ trúe))

d. he'll ($_{VP}$ ($_{Adv}$ ábsolutely) ($_{V}$ flíp))

Clause (18-a) is very close, but not identical to Nespor & Vogel's algorithm. The difference is that Hayes' formulation does not suffer from indeterminacy. Consider example (23): substituting *true* for X in (18), we can deduce that the sequence *evidently true* must occupy the same φ; substituting *statement* instead, entails that the sequence *an evidently true statement* must occupy that same φ. Both results are compatible without the need for embedded φ's (23-b).

(23) a. ($_{NP}$ an ($_{AP}$ ($_{Adv}$ evidently) ($_{A}$ true)) ($_{N}$ statement))

b. [$_{\varphi}$ an évidently true státement]

As far as Dutch is concerned, however, (18-a) is still inadequate, because it predicts that all modifiers preceding the head of an NP occupy the same φ. As mentioned earlier in relation to the Nespor & Vogel algorithm, this results in improbably long φ's (cf. example (8)) and overapplication of RR.[9] The same problem occurs with respect to a VP: according to (18-a), the sequence ($_{VP}$ ...V) in (24) obligatory occupies the same φ.

(24) a. ($_{S}$ dat hij ($_{VP}$ ($_{PP}$ door de voltallige raad van
that he by the entire board of
bestuur) ($_{PP}$ met meerderheid van stemmen) ($_{PP}$ tot
management with majority of votes to
nieuwe voorzitter) is ($_{V}$ gekozen))$_{VP}$)$_{S}$
new chairman is elected

b. *[$_{\varphi}$ dat hij door de voltallige raad van bestuur met meerderheid van stemmen tot nieuwe voorzitter is gekozen]

Clause (18-b) is in effect identical to Nespor & Vogel's optional restructuring rule (9).[10] Hayes gives the English examples in (25), which are similar to (10) and (12).

(25) a. ($_{VP}$ ($_{V}$ cómprehending) ($_{NP}$ éverything))

b. it'll ($_{V}$ ínterfere) ($_{PP}$ with télevision)

c. to ($_{V}$ íntervene) ($_{AdvP}$ intélligently)

[9] In Hayes' view, the spacing between two accented syllables is also relevant for the application of RR. Thus, a large φ does not necessarily imply overapplication of RR. We will show in the next section, however, that even in a short sequence of premodifiers, RR is often not applicable.

[10] Notice, however, that Hayes' formulation is more precise, as it avoids the notion *non-branching*. In (25-b), for instance, the PP *with television* is – strictly speaking – binary branching.

d. ($_{NP}$ the ($_{N}$ jápanese) ($_{PP}$ of Hónshu))

We have already argued that the Dutch equivalent of (25-c) is not a single φ (cf. *rijden razendsnel* in (10)). In addition, the Dutch equivalent of (25-d) is almost never a single φ either. Consider (26), for example, where the accent on *porselein* cannot be deleted without altering the focus interpretation.

(26) ($_{NP}$ orgin$\overset{*}{\text{e}}$el ($_{N}$ porsel$\overset{*}{\text{e}}$in) ($_{PP}$ uit D$\overset{*}{\text{e}}$lft))
original porcelain from Delft

This appears to hold for nearly all postmodifying PP's in Dutch, with the exception of a some (idiomatic) expressions like (27), where *lagen* is deaccented.

(27) ($_{PP}$ in ($_{NP}$ br$\overset{*}{\text{e}}$de ($_{N}$ l$\overset{\varnothing}{\text{a}}$gen) ($_{PP}$ van de bev$\overset{*}{\text{o}}$lking)))
in broad layers of the population
'in large sections of the population'

In conclusion, the φ-algorithm by Hayes is not adequate for Dutch either. The problems appear to be caused by the syntactic differences between English and Dutch. First, Dutch offers more complex possibilities for premodifying a noun, spanning more than one φ, and has a right-headed VP structure. Second, nonbranching – or single C – postmodifiers are almost never merged with the preceding φ. Third, C-formation based on shared membership contradicts phonological evidence.

3.3 The phonological phrase in Dutch

In this section, a better φ-algorithm for Dutch will be proposed. We will first present a preliminary version of the algorithm. Next, we will show that it accounts well for φ-phrasing in many different syntactic configurations, but is not yet entirely adequate. This motivates the subsequent revision of the algorithm.

3.3.1 Preliminary algorithm

We start with a preliminary formulation of φ-phrasing.

(28) **Phonological phrase construction for Dutch** (1st version)

A φ-boundary occurs after every content word that is not an adjacent modifier.

(29) **Adjacent modification** (1st version)

X is an *adjacent modifier* if and only if there is a head such that both

a. X is left-adjacent to Y, and
b. X is dominated by YP.

The class of content words excludes function words belonging to a closed class, such as determiners, complementizers, prepositions, auxiliaries, pronouns, conjunctions, etc. The notion of adjacent modification can be explained by means of example (30). In (30-a), the adjective *gemetselde* is both adjacent to the noun *schoorsteen* and dominated by the maximal projection of *schoorsteen* (NP), and is therefore an adjacent modifier. In contrast, the adjective *onhandige* is adjacent to the adjective *gemetselde*, but not dominated by the maximal projection of *gemetselde* (AP), and therefore *not* an adjacent modifier. We can contrast this with (30-b), where *gemetselde* is again an adjacent modifier of *schoorsteen*, but in addition the adjective *onhandig* is an adjacent modifier of *gemetselde*.[11]

(30) a. $(_{NP}$ $(_{Det}$ een) $(_{AP}$ $(_{A}$ onhandige)) $(_{AP}$ $(_{A}$ gemetselde))
a awkward bricked
$(_{N}$ schoorsteen))
chimney
'a chimney that is awkward and made out of bricks'

b. $(_{NP}$ $(_{Det}$ een) $(_{AP}$ $(_{A}$ onhandig) $(_{A}$ gemetselde)) $(_{N}$
a clumsy bricked
schoorsteen))
chimney
'a chimney that has been clumsily bricked'

The relation of adjacent modification is related to the inflection of adjectives. In Dutch, attributive adjectives are inflected with the suffix *-e*, except when modifying an indefinite neuter singular noun.[12] In (30-a), both *onhandige* and *gemetselde* are inflected. In (30-b), only *gemetselde* is inflected, as *onhandig* is adverbially used, in which case it is never inflected. In the particular case of a noun that is indefinite, neuter and singular, however, a modifying adjective is not inflected either and an NP can become syntactically ambiguous as in (31).

(31) a. $(_{NP}$ $(_{Det}$ een) $(_{AP}$ $(_{A}$ onhandig)) $(_{AP}$ $(_{A}$ gemetseld)) $(_{N}$
a awkward brick
schoorsteentje))
chimney + DIM
'a small chimney that is awkward and made out of bricks'

[11] Most adjectives in Dutch can also be used adverbially to modify a verb or another adjective.

[12] Although this is certainly the most frequent pattern, it is an oversimplification. See Haeseryn et al. (1997, ch. 6.4) for exceptions.

b. $(_{NP}$ $(_{Det}$ een) $(_{AP}$ $(_{A}$ onhandig) $(_{A}$ gemetseld)) $(_{N}$
a clumsily bricked
schoorsteentje))
chimney + DIM
'a small chimney that has been clumsily bricked'

Interestingly, the two readings correspond to different phonological phrasings, resulting in different accent distributions. Although the adjective *onhandig* is a content word, it is not an adjacent modifier in (31-a), and is followed by a φ-boundary according to definition (28). Therefore, its accent cannot be deleted by RR (32-a). In contrast, *onhandig* is both a content word and an adjacent modifier in (31-b), thus no φ-boundary is predicted. Hence, RR deletes the accent on *gemetseld* (32-b). In short, the two sentences in (32) form a minimal pair, where the relation between accentuation and interpretation can be explained in terms of φ-phrasing and RR.

(32) a. $[_{\varphi}$ een onh$\overset{*}{\text{a}}$ndig] $[_{\varphi}$ gem$\overset{*}{\text{e}}$tseld scho$\overset{*}{\text{o}}$rsteentje]

b. $[_{\varphi}$ een onh$\overset{*}{\text{a}}$ndig gem$\overset{\emptyset}{\text{e}}$tseld scho$\overset{*}{\text{o}}$rsteentje]

This contrast cannot be explained by the algorithms reviewed earlier. Additionally, we will to show that our proposal can cope with the types of modification that the reviewed algorithms could not. In order to demonstrate that it yields all and only correct φ's, we need to do two things. First, we have to investigate all syntactic structures with an adjacent modification relation, i.e. of the type $(_{YP} \ldots (_{X}\ )\ (_{Y}\ ) \ldots)$, and show that the RR can apply freely, i.e. across X and Y. Second, we have to show that in all syntactic structures where a content word is *not* an adjacent modifier, application of RR is blocked. Of course, investigating all relevant syntactic structures is practically impossible, for one because the number of structures that are distinguished is to some extent dependent on the particular syntactic theory adopted. Nevertheless, we will show in the next two sections that our preliminary algorithm based on the notion of adjacent modification makes the correct predictions in many different syntactic structures.

3.3.2 Phonological phrasing in NP, AP and AdvP

3.3.2.1 Adjacent modifiers preventing a φ-boundary

We will start with adjacent modifiers before a noun. Example (33) shows an adjective premodifying a noun: (33-a) shows the syntactic structure; (33-b) shows the predicted φ-phrasing and the accent distribution after RR has applied; (33-c) shows that the compound adjective *ijskoud* can indeed carry two accents in an appropriate context. Example (34) does the same for a noun premodifying another noun.[13]

[13]The structure in which an adverb premodifies a noun, i.e. $(_{NP} \ldots (_{Adv}\ )\ (_{N}\ ) \ldots)$ appears to be non-existent in Dutch. A phrase like *jouw keuze is een altijd gelukkige* 'your

(33) $(_{NP} \ldots (_{A}\ )\ (_{N}\ ) \ldots)$

a. $(_{NP}\ (_{AP}\ (_{A}$ ijskoud $)\ )\ (_{N}$ bier $)\ )$
ice-cold beer

b. $[_{\varphi}$ $\overset{*}{\text{i}}$jsk$\overset{ø}{\text{o}}$ud b$\overset{*}{\text{i}}$er]

c. het b$\overset{*}{\text{i}}$er is $\overset{*}{\text{i}}$jsk$\overset{*}{\text{o}}$ud
the beer is ice-cold

(34) $(_{NP} \ldots (_{N}\ )\ (_{N} \ldots\ )$

a. $(_{NP}\ (_{NP}$ een halve $(_{N}$ kilo $)\ )\ (_{N}$ kaas $)\ )$
a half kilo cheese

b. $[_{\varphi}$ een h$\overset{*}{\text{a}}$lve k$\overset{ø}{\text{i}}$lo k$\overset{*}{\text{a}}$as]

c. graag een h$\overset{*}{\text{a}}$lve k$\overset{*}{\text{i}}$lo
please a half kilo

The next three examples explore the predictions with respect to an adjacent modifier before an adjective.

(35) $(_{AP} \ldots (_{A}\ )\ (_{A}\ ) \ldots)$

a. $(_{NP}$ een $(_{AP}\ (_{AP}\ (_{A}$ lichamelijk $)\ )\ (_{A}$ gehandicapt $)\ )\ (_{N}$
a physically handicapped
kind $)\ )$
child

b. $[_{\varphi}$ een lich$\overset{*}{\text{a}}$melijk geh$\overset{ø}{\text{a}}$ndicapt k$\overset{*}{\text{i}}$nd]

c. het k$\overset{*}{\text{i}}$nd is lich$\overset{*}{\text{a}}$melijk geh$\overset{*}{\text{a}}$ndicapt
the child is physically handicapped

(36) $(_{AP} \ldots (_{Adv}\ )\ (_{A}\ ) \ldots)$

a. $(_{NP}$ een $(_{AP}\ (_{Adv}$ erg $)\ (_{A}$ mooie $)\ )\ (_{N}$ tuin $)\ )$
a very beautiful garden

b. $[_{\varphi}$ een $\overset{*}{\text{e}}$rg mo$\overset{ø}{\text{o}}$ie t$\overset{*}{\text{u}}$in]

c. hun tuin is $\overset{*}{\text{e}}$rg mo$\overset{*}{\text{o}}$i
their garden is very beautiful

choice is an always lucky one' might appear to fit this pattern. However, we would argue that *gelukkkige* is an adjective, as indicated by its inflection, modifying an empty N.

(37) $(_{AP} \ldots (_{N}\ ) (_{A}\ ) \ldots)$

a. $(_{NP}$ de $(_{AP}$ $(_{NP}$ $(_{N}$ maandag)) $(_{A}$ aangekondigde)) $(_{N}$
the Monday announced
maatregelen))
measures

b. $[_{\varphi}$ de m$\overset{*}{a}$andag aangek$\overset{\varnothing}{o}$ndigde m$\overset{*}{a}$atregelen]

c. m$\overset{*}{a}$atregelen werden m$\overset{*}{a}$andag aangek$\overset{*}{o}$ndigd
measures were Monday announced
'measures were announced on Monday'

What remains is the case of an adjacent modifier before an adverb. Again, one of the logical possibilities, the structure $(_{AdvP}\ (_{A}\ )\ (_{Adv}\ )\ \ldots\ )$, would appear to be lacking in Dutch.

(38) $(_{AdvP} \ldots (_{Adv}\ ) (_{Adv}\ ) \ldots\ )$

a. $(_{AP}$ $(_{AdvP}$ $(_{AdvP}$ $(_{Adv}$ erg)) $(_{Adv}$ vaak)) $(_{A}$ ziek))
very often ill

b. $[_{\varphi}$ $\overset{*}{e}$rg v$\overset{\varnothing}{a}$ak zi$\overset{*}{e}$k]

c. hij sliep $\overset{*}{e}$rg v$\overset{*}{a}$ak
he slept very often

(39) $(_{AdvP} \ldots (_{N}\ ) (_{Adv}\ ) \ldots\ )$

a. $(_{AdvP}$ $(_{NP}$ een $(_{A}$ heel) $(_{N}$ eind)) $(_{Adv}$ verder))
a whole piece further
'a whole lot further'

b. $[_{\varphi}$ een h$\overset{*}{e}$el $\overset{\varnothing}{e}$ind v$\overset{*}{e}$rder]

c. naar de top bleek een h$\overset{*}{e}$el $\overset{*}{e}$ind
to the top proved a whole piece
'to the top turned out to be a long way'

So far, we have looked at relatively simple syntactic structures. More complex examples are given in (40) and (41). Notice that in the latter RR applies twice.

(40) a. $(_{NP}$ de $(_{AP}$ $(_{PP}$ van $(_{NP}$ $(_{N}$ water))) $(_{A}$ afhankelijke))
the on water dependent
$(_{N}$ eenden))
ducks
'the ducks which depend on water'

b. $[_{\varphi}$ de van w$\overset{*}{\text{a}}$ter afh$\overset{\emptyset}{\text{a}}$nkelijke $\overset{*}{\text{e}}$enden]

(41) a. $(_{NP}$ de $(_{AP}$ $(_{PP}$ op $(_{NP}$ $(_{AP}$ $(_{A}$ goed)) $(_{N}$ geluk)))) $(_{A}$
the at good luck
gekozen)) $(_{N}$ deelnemers))
chosen participants
'the participants (who were) chosen hoping for the best'

b. $[_{\varphi}$ de op go$\overset{*}{\text{e}}$d gel$\overset{\emptyset}{\text{u}}$k gek$\overset{\emptyset}{\text{o}}$zen d$\overset{*}{\text{e}}$elnemers]

3.3.2.2 Non-adjacent modifiers triggering a φ-boundary

We now turn to syntactic configurations where a content word is *not* an adjacent modifier and therefore triggers the insertion of a φ-boundary. As far as NP's are concerned, these involve coordinations and postmodifiers. Example (42) has two coordinated AP's, showing that deletion of the second accent is incompatible with a full focus interpretation (42-c). Example (43) has two coordinated AP's of which the first contains the compound adjective *peperduur*, which can carry two accents. Deletion of its second accent is less unacceptable (43-c).

(42) a. $(_{NP}$ $(_{AP}$ $(_{A}$ eindeloze)) $(_{AP}$ $(_{A}$ saaie)) $(_{N}$ vlaktes))
endless boring plains

b. $[_{\varphi}$ $\overset{*}{\text{e}}$indeloze] $[_{\varphi}$ sa$\overset{*}{\text{a}}$ie vl$\overset{*}{\text{a}}$ktes]

c. ?$[_{\varphi}$ $\overset{*}{\text{e}}$indeloze sa$\overset{\emptyset}{\text{a}}$ie vl$\overset{*}{\text{a}}$ktes]

(43) a. $(_{NP}$ een $(_{AP}$ $(_{A}$ peperduur)) $(_{AP}$ $(_{A}$ nieuw)) $(_{N}$
a very-expensive new
speeltje))
toy

b. $[_{\varphi}$ een p$\overset{*}{\text{e}}$perd$\overset{*}{\text{u}}$ur] $[_{\varphi}$ ni$\overset{*}{\text{e}}$uw sp$\overset{*}{\text{e}}$eltje]

c. ?$[_{\varphi}$ een p$\overset{*}{\text{e}}$perd$\overset{\emptyset}{\text{u}}$ur ni$\overset{\emptyset}{\text{e}}$uw sp$\overset{*}{\text{e}}$eltje]

A case of two coordinated NP's is given in (44). While the noun *maandag* is an adjacent modifier of the adjective *gesloten*, the noun *zondag* is not, and is therefore followed by a φ-boundary. This correctly predicts that the accent on *gesloten* can be deleted, whereas that on *maandag* must be preserved.

(44) a. ($_{NP}$ een ($_{AP}$ ($_{NP}$ ($_{NP}$ ($_{N}$ zondag)) ($_{Conj}$ en) ($_{NP}$ ($_{N}$
a Sunday and
maandag))) ($_{A}$ gesloten)) ($_{N}$ winkel))
Monday closed shop
'a shop (which is) closed on Sundays and Mondays'

b. [$_{\varphi}$ een z$\overset{*}{\text{o}}$ndag] [$_{\varphi}$ en m$\overset{*}{\text{a}}$andag gesl$\overset{\text{ø}}{\text{o}}$ten w$\overset{*}{\text{i}}$nkel]

c. ?[$_{\varphi}$ een z$\overset{*}{\text{o}}$ndag en m$\overset{\text{ø}}{\text{a}}$andag gesl$\overset{\text{ø}}{\text{o}}$ten w$\overset{*}{\text{i}}$nkel]

The data in (45) concerns two coordinated AP's of which the first is complex in the sense that it contains a PP modifier. Since *onsnapte* is not an adjacent modifier and must thus be followed by a φ-boundary, none of the accents in (45) can be deleted by RR.

(45) a. ($_{NP}$ de ($_{AP}$ ($_{PP}$ op ($_{NP}$ ($_{N}$ maandag))) ($_{A}$ ontsnapte))
the on Monday escaped
($_{AP}$ ($_{A}$ gevaarlijke)) ($_{N}$ criminelen))
dangerous criminals
'the dangerous criminals (who) escaped on Monday'

b. [$_{\varphi}$ de op m$\overset{*}{\text{a}}$andag ontsn$\overset{*}{\text{a}}$pte] [$_{\varphi}$ gev$\overset{*}{\text{a}}$arlijke crimin$\overset{*}{\text{e}}$len]

c. ?[$_{\varphi}$ de op m$\overset{*}{\text{a}}$andag ontsn$\overset{\text{ø}}{\text{a}}$pte gev$\overset{\text{ø}}{\text{a}}$arlijke crimin$\overset{*}{\text{e}}$len]

The last example involving coordination is a rather complex structure (46). The noun *huis* is adjacently modified by *verborgen*, which is itself adjacently modified by the PP *achter ... bomen*. Within this PP, the noun *bomen* is modified by the two coordinated adjectives *hoge* and *groene*. Since only *groene* is an adjacent modifier, we predict a φ-boundary after *hoge*. As result, only the accents on *bomen* and *verborgen* are in the scope of RR.

(46) a. ($_{NP}$ een ($_{AP}$ ($_{PP}$ achter ($_{NP}$ ($_{AP}$ ($_{A}$ hoge)) ($_{AP}$ ($_{A}$
a behind high
groene)) ($_{N}$ bomen))) ($_{A}$ verborgen)) ($_{N}$ huis))
green trees hidden huis
'a house (that was) hidden behind high green trees'

b. [$_{\varphi}$ een achter h$\overset{*}{\text{o}}$ge] [$_{\varphi}$ gro$\overset{*}{\text{e}}$ne b$\overset{\text{ø}}{\text{o}}$men verb$\overset{\text{ø}}{\text{o}}$rgen hu$\overset{*}{\text{i}}$s]

c. ?[$_{\varphi}$ een achter h$\overset{*}{\text{o}}$ge gro$\overset{\text{ø}}{\text{e}}$ne b$\overset{\text{ø}}{\text{o}}$men verb$\overset{\text{ø}}{\text{o}}$rgen hu$\overset{*}{\text{i}}$s]

Postmodifiers of N's most frequently occur in the form of PP's and relative S's. These are shown to trigger a φ-boundary in (47) and (48) respectively.

(47) a. ($_{NP}$ een ($_{N}$ tent) ($_{PP}$ met lichtgewicht stokken))
a tent with lightweight poles

b. [$_\varphi$ een t$\overset{*}{\text{e}}$nt] [$_\varphi$ met l$\overset{*}{\text{i}}$chtgewicht st$\overset{*}{\text{o}}$kken]

c. ?[$_\varphi$ een t$\overset{*}{\text{e}}$nt met l$\overset{ø}{\text{i}}$chtgewicht st$\overset{*}{\text{o}}$kken]

(48) a. ($_{NP}$ een ($_N$ tent)) ($_S$ die ($_{NP}$ lichtgewicht stokken)
a tent that lightweight poles
heeft))
has

b. [$_\varphi$ een t$\overset{*}{\text{e}}$nt] [$_\varphi$ die l$\overset{*}{\text{i}}$chtgewicht st$\overset{*}{\text{o}}$kken heeft]

c. ?[$_\varphi$ een t$\overset{*}{\text{e}}$nt die l$\overset{ø}{\text{i}}$chtgewicht st$\overset{*}{\text{o}}$kken heeft]

3.3.3 Phonological phrasing in VP and S

It was already shown in example (15) that φ-phrasing and RR also occur at the level of VP's and clauses. Another example is given in (49). The compound adjective *ijskoud* can carry two accents (cf. example (33)). However, if it directly precedes a verb, its second accent can be deleted. This is in line with our expectations, as *ijskoud* adjacently modifies *drinken*, and does therefore not trigger a φ-boundary.

(49) a. bier moet men ($_{VP}$ ($_A$ ijskoud) ($_V$ drinken))
beer should one ice-cold drink
'one should drink beer cold as ice'

b. bier [$_\varphi$ moet men $\overset{*}{\text{i}}$jsk$\overset{ø}{\text{o}}$ud dr$\overset{*}{\text{i}}$nken]

c. bier [$_\varphi$ moet men $\overset{*}{\text{i}}$jsk$\overset{*}{\text{o}}$ud] [$_\varphi$ dr$\overset{*}{\text{i}}$nken]

Analogously to φ-phrasing in NP's, two subsequent modifiers are separated by a φ-boundary when both modify the verb (50), but are merged into the same φ when the first is an adjacent modifier of the second (51).

(50) a. koffie kan men ($_{VP}$ ($_A$ koud) ($_A$ onmogelijk) ($_V$
coffee can one cold impossibly
drinken))
drink
'it is impossible to drink coffee cold'

b. koffie [$_\varphi$ kan men k$\overset{*}{\text{o}}$ud] [$_\varphi$ onm$\overset{*}{\text{o}}$gelijk dr$\overset{*}{\text{i}}$nken]

c. ?koffie [$_\varphi$ kan men k$\overset{*}{\text{o}}$ud onm$\overset{ø}{\text{o}}$gelijk dr$\overset{*}{\text{i}}$nken]

(51) a. koffie moet men $(_{VP}$ $(_{AP}$ $(_{Adv}$ zeer) $(_{A}$ heet)) $(_{V}$
coffee must one very hot
drinken))
drink
'one should drink coffee very hot'

b. koffie $[_{\varphi}$ moet men z$\overset{*}{\text{e}}$er h$\overset{\emptyset}{\text{e}}$et dr$\overset{*}{\text{i}}$nken]

c. ?koffie $[_{\varphi}$ moet men z$\overset{*}{\text{e}}$er h$\overset{*}{\text{e}}$et] $[_{\varphi}$ dr$\overset{*}{\text{i}}$nken]

Another example of this contrast is given in (52).

(52) a. ?onze software kunt u sn$\overset{*}{\text{e}}$l doeltr$\overset{\emptyset}{\text{e}}$ffend gebr$\overset{*}{\text{u}}$iken
our software can you quickly effectively use
'you will quickly be able to use our software effectively'

b. onze software laat zich z$\overset{*}{\text{e}}$er doeltr$\overset{\emptyset}{\text{e}}$ffend gebr$\overset{*}{\text{u}}$iken
our software lets itself very effectively use
'our software can be used very effectively'

These examples suggest that φ-phrasing in clauses is governed by the same principles as φ-phrasing in NP's. Nevertheless, there are some exceptions. To begin with, we can insert an auxiliary between an adjective and a main verb without affecting the φ-phrasing (53). This might be explained by assuming that the adjective is an adjacent modifier of the auxiliary, but since this is not the only problematic example, we will not further explore this option.

(53) a. bier moet $(_{VP}$ $(_{A}$ ijskoud) $(_{Aux}$ worden) $(_{V}$
beer should ice-cold be
gedronken))
drunk
'serve beer cold as ice'

b. bier $[_{\varphi}$ moet $\overset{*}{\text{i}}$jsk$\overset{\emptyset}{\text{o}}$ud worden gedr$\overset{*}{\text{o}}$nken]

In addition, a noun that adjacently modifies a verb appears to resist incorporation in the same φ, even though its nominal head is an adjacent modifier of the verb; see the NP in (54) [14] and the PP in (55) respectively.

(54) a. goede voorbereiding kan $(_{VP}$ $(_{NP}$ nare verrassingen) $(_{V}$
good preparation can dismal surprises
voorkomen))
prevent
'good preparation can prevent nasty surprises'

[14] One might expect the predicate *voorkomen* to be unaccented as result of applying SAAR. However, even though it has an adjacent focused argument, the utterance is contingent rather than eventive, and therefore an exception to SAAR (Gussenhoven 1992b, Appendix).

b. goede voorbereiding [$_\varphi$ kan n$\overset{*}{\text{a}}$re verr$\overset{*}{\text{a}}$ssingen] [$_\varphi$ voork$\overset{*}{\text{o}}$men]

c. ?goede voorbereiding [$_\varphi$ kan n$\overset{*}{\text{a}}$re verr$\overset{\text{ø}}{\text{a}}$ssingen voork$\overset{*}{\text{o}}$men]

(55) a. Frankrijk wordt ($_{VP}$ ($_{PP}$ door Nederlandse toeristen) ($_V$
France is by Dutch tourists
gemeden))
avoided
'France is avoided by Dutch tourists'

b. Frankrijk [$_\varphi$ wordt door N$\overset{*}{\text{e}}$derlandse toer$\overset{*}{\text{i}}$sten] [$_\varphi$ gem$\overset{*}{\text{e}}$den]

c. ?Frankrijk [$_\varphi$ wordt door N$\overset{*}{\text{e}}$derlandse t$\overset{\text{ø}}{\text{o}}$eristen gem$\overset{*}{\text{e}}$den]

While all preceding examples had the finite verb in sentence-final position, its *base* or *deep-structure* position in the terminology of transformational grammar, it can also occur in the (syntactically) second position of a declarative main clause. The question arises if the changed position affects the φ-phrasing behaviour of the verb. The following sentences show the verb in second position preceded by an AP (56), an NP (57), and a PP (58) respectively. The resulting accent patterns suggest that the changed position is irrelevant to φ-phrasing: only an A merges with a following V into a single φ.

(56) a. ($_{AP}$ jarenlang) ($_V$ meed) hij haar
years-long avoided he her
'for years, he avoided her'

b. [$_\varphi$ j$\overset{*}{\text{a}}$renl$\overset{\text{ø}}{\text{a}}$ng m$\overset{*}{\text{e}}$ed hij haar]

(57) a. ($_{NP}$ vele jaren) ($_V$ sleet) hij daar
many years spent he there
'many a year he spent there'

b. ?[$_\varphi$ v$\overset{*}{\text{e}}$le j$\overset{\text{ø}}{\text{a}}$ren sl$\overset{*}{\text{e}}$et hij daar]

(58) a. ($_{PP}$ na vele jaren) ($_V$ schreef) hij haar
after many years wrote he her
'after many years, he wrote her'

b. ?[$_\varphi$ na v$\overset{*}{\text{e}}$le j$\overset{\text{ø}}{\text{a}}$ren schr$\overset{*}{\text{e}}$ef hij haar]

There is an additional complication here, however. Since the moved V is no longer the *head* of a VP, the preceding A is not dominated by the VP in (56), and therefore does not match the definition of an adjacent modifier.

Besides movement of the verb, Dutch syntax allows extraposition – rightward movement across the verb cluster – of S's, PP's and heavy NP's. Without exploring all combinations in depth, the examples in (59) and (60) show that our φ-algorithm correctly predicts that a φ-boundary is obligatory between a content verb and an extraposed constituent.

(59) a. voor een stoplicht moet men ($_{VP}$ ($_V$ wachten)) ($_{PP}$ op groen licht)
before a traffic-light must one wait for green light
'before a traffic light, one has to wait for the green light'

b. voor een stoplicht [$_\varphi$ moet men wa*chten] [$_\varphi$ op groe*n li*cht]

c. ?voor een stoplicht [$_\varphi$ moet men wa*chten op groøen li*cht]

(60) a. ze zijn ($_{VP}$ ($_V$ vertrokken)) ($_{PP}$ met al hun bezittingen)
they are left with all their possessions
'they left with all their possessions'

b. [$_\varphi$ ze zijn vertro*kken] [$_\varphi$ met a*l hun bezi*ttingen]

c. ?[$_\varphi$ ze zijn vertro*kken met aøl hun bezi*ttingen]

We finish this discussion of φ-phrasing in S's and VP's by signalling a few problems that also plagued the other algorithms. First, our algorithm suffers from the problem of unparsed auxiliaries (cf. example (6)), because it fails to predict a sentence-final φ-boundary when the sentence ends with a function word. Moreover, without the notion of a Clitic Group, a similar problem occurs with clitics (61).

(61) a. ik ($_{VP}$ ($_V$ geef) 't aan 'm)
I give it to him

b. [$_\varphi$ ik geef] 't aan 'm

In general, as observed in (20) and (21), clitics may end up in the wrong φ because their host triggers a φ-boundary.

3.3.3.1 Revised algorithm

To sum up so far, we have shown that our preliminary proposal for φ-phrasing in Dutch based on the notion of adjacent modification works well in NP's, AP's and AdvP's. However, it proved to be not entirely adequate in VP's and S's, because it falsely predicts that an NP/PP and a following verb are merged into

Table 3.1: Syntactic structures in which no φ-boundary is inserted after a content word.

Left context:	*Word:*	*Right context:*	
$(_{NP} \ldots$	*A*	*N*	$\ldots)$
$(_{NP} \ldots$	*N*	*N*	$\ldots)$
$(_{AP} \ldots$	*A*	*A*	$\ldots)$
$(_{AP} \ldots$	*Adv*	*A*	$\ldots)$
$(_{AP} \ldots$	*N*	*A*	$\ldots)$
$(_{AdvP} \ldots$	*Adv*	*Adv*	$\ldots)$
$(_{AdvP} \ldots$	*N*	*Adv*	$\ldots)$
	A	Aux^{*} V	
	Adv	Aux^{*} V	
	V	Aux^{+}	
	V	$Clitic^{+}$	

one φ. In addition, it suffers from the unparsed auxiliaries problem, and from clitics that are either unparsed or in the wrong φ. Table 3.1 summarises our current state of knowledge.[15]

Our preliminary proposal needs to be revised. The problem with the unparsed auxiliaries and clitics can be solved by incorporating them into the φ of the preceding V, as will be proposed in Section 7.2.1. The observation that a N adjacently modifying a V still triggers a φ-boundary can be accounted for by assuming that a V cannot be adjacently modified at all, that is, by removing the option Y=V from the definition of adjacent modification (62). Interestingly, the observation that V differs from other lexical heads with respect to φ-phrasing is not unique to Dutch (Hayes and Lahiri 1991).

(62) **Adjacent modification** (final version)

For every $X, Y \in \{N, A, Adv\}$:
X is an *adjacent modifier* of Y if and only if

a. X is left-adjacent to Y, and
b. X is dominated by YP.

As a result, the fact that an A/Adv before a V does not trigger a boundary can no longer be accounted for in terms of adjacent modification, but must be enforced explicitly (63-b). However, this stipulation is needed anyway for an A/Adv preceding either an auxiliary (cf. example (53)) or a fronted verb (cf. example (56)).

[15] Using regular expressions, the superscript star (*) means zero or more occurrences of the superscripted category, and superscript plus (+) means one or more occurrences.

(63) **Phonological phrase construction for Dutch** (final version)

A φ-boundary occurs after every content word W unless

a. W is an adjacent modifier, or
b. W is an A or Adv directly preceding a verb.

Although there is room for improvement, (63) provides an observationally adequate definition of φ-construction for Dutch.

A few issues have remained unexplored. In Section 7.3.5, we will show that our φ-phrasing algorithm handles numeral expressions as well. We leave φ-phrasing with respect to quantifiers and idiomatic expressions to future research.

3.4 Evaluation

In this section, we will report on an evaluation of the proposed φ-algorithm. Recall that the reason for investigating phonological phrasing is that the grammar for pitch accent placement using RR crucially depends on the φ-boundaries being correctly determined. It is from this perspective that the algorithm was evaluated. That is, rather than looking for phonemic or phonetic evidence (Section 3.2.1) for the φ-boundaries, we checked for a substantial number of utterances whether applying RR within the predicted φ's led to the desired accentuation. Below, we will describe the method, material, and results of this evaluation.

3.4.1 Method

As explained in Section 1.1.3, acceptability judgements regarding the accentuation may vary, because (1) they may be gradual rather than categorical, (2) some variants may be equally acceptable, and (3) intersubjective differences may occur. To cope with this variation, we have chosen to aim for the intonation that is most acceptable according to the judgements of intonational experts, ignoring differences between equally acceptable forms. In addition, we argued in Section 1.1.4 that the disadvantage of a perceptual evaluation is that other aspects of the wave form, such as segmental errors, inadequate phonetic realization, or speaker characteristics, may interfere with judgement of the intonation proper. We therefore decided to rely on a symbolic assessment of the acceptability judgements of the experts, asking them to prescribe the preferred intonational form for a given texts, and comparing this with the prediction by our algorithms. The method that we used for the evaluation of our phonological phrasing algorithm is a practical implementation of these methodological choices. It encompasses the following steps.

1. **Construction of the corpus**
   Assemble a corpus of sentences. For each sentence, add its syntactic structure and its preferred distribution of pitch accents.

2. **Implementation of accentuation algorithm**
   Implement the accentuation algorithm that includes RR.

3. **Evaluation of accentuation algorithm**

   (a) *Automatic comparison*
   Apply the algorithm to the sentences in the corpus. For each sentence, automatically compare the predicted accentuation with the preferred accentuation. Report all differences and generate a confusion matrix.

   (b) *Error analysis*
   On the basis of the results of the automatic comparison, determine which accentuation errors are due to φ-phrasing errors.

Notice that a complete accentuation algorithm is used rather than only RR. The reasons for this will be explained further on (Section 3.4.3). The above steps will described in detail in the next three sections.

3.4.2 Corpus

3.4.2.1 Text

It is evident that evaluating φ-phrasing on text material in which RR rarely or never applies makes no sense. The text material has to be relevant in the sense that it must contain sentences in which words are frequently deaccented for rhythmic reasons. We found botanical texts to be an appropriate challenge, since most of their sentences are relatively long and have a rather complex syntactic structure, giving rise to a large number of accents and many opportunities to apply RR. Our corpus consisted of 10 descriptions of flowers, which were taken from a botanical handbook about wild flowers in Europe (Forey 1992). Each description consisted of 4 to 8 sentences. The total number of sentences was 52, the total number of orthographic words 907, an average of 17.4 words per sentence. Some words, like the noun *Zuid-Europa* 'Southern-Europe' or the compound adjective *eirond* 'egg-round', can receive two accents. Counting these as two words, which reflects the actual number of potential accent locations, the total number of words is 935. An example sentence is given in (64); a number of full texts are in given in Appendix A and Appendix B.

(64)
De	Reuzenbalsemien	is	uit	de	Himalaya	naar	Europa
The	Impatiens-Glandulifera	is	from	the	Himalayas	to	Europe

gebracht	en	koloniseert	hier	waterkanten	in	vele	gebieden	ook
brought	and	colonises	here	watersides	in	many	areas	also

plaatselijk	in	Nederland	en	Belgie.
locally	in	the-Netherlands	and	Belgium

3.4.2.2 Accent distribution

A panel of three intonationally experienced, native speakers of Dutch was asked to prescribe the preferred accentuation for these texts. They worked on a version of the texts in which all abbreviations and numbers were expanded to their full lexical form. They were asked to specify the locations of the pitch accents to be used in a competent reading of the text. To accommodate for the possibility that alternative accentuations are equally acceptable (Section 1.1.3), a distinction was allowed between obligatory and optional accents. The panel discussed disagreements in their accentuations until a consensus was reached. In total, they assigned 435 obligatory accents and 10 optional accents. The prescription produced for (64) is given in (65); see Appendix A and Appendix B for other sentences.

(65) De R$\overset{*}{\text{e}}$uzenbalsemien is uit de Himal$\overset{*}{\text{a}}$ya naar Eur$\overset{*}{\text{o}}$pa gebracht en kolonis$\overset{*}{\text{e}}$ert hier w$\overset{*}{\text{a}}$terkanten in v$\overset{*}{\text{e}}$le gebi$\overset{*}{\text{e}}$den ook pla$\overset{*}{\text{a}}$tselijk in N$\overset{*}{\text{e}}$derland en B$\overset{*}{\text{e}}$lgie.

3.4.2.3 Syntactic structure

The sentences were manually parsed by the author according to the syntactic analysis outlined in Section 2.6.[16] The syntactic structures provide all information that is required by the φ-phrasing algorithm, and by our accent placement algorithm in general (to be described in the next section). This amounts to an analysis in terms of major constituents (S, VP, PP, AP, AdvP), their heads (V, N, P, A, Adv) and several minor categories (Determiner, Quantifier, Complementizer, Conjunction). A number of word level features was used to mark pronouns, auxiliaries, copula, and the verbal part of compound verbs. Empty nodes or traces were not included. Figure 3.2 shows the syntactic structure corresponding to (64).

In addition to this purely syntactic information, the accentuation algorithm needs two other types of input. First, it requires the focus distribution. The default assumption is that all content word are focused (N, V, A, Adv, Q), whereas function words are not (determiners, prepositions, auxiliaries, copula, complementizers, dislocated verbal parts of compound verbs, conjunctions). Only non-default focus is explicitly indicated by means of focus and background features. This is rare, however, and only occurs on anaphoric expressions, i.e. some of the pronouns, definite noun phrases and pronominal adverbs. Second, the predicates and their corresponding arguments are labelled to enable application of SAAR.

[16] We experimented with syntactic parsers, but they either failed to handle the level of complexity of these sentences, or did not provide the required level of detail.

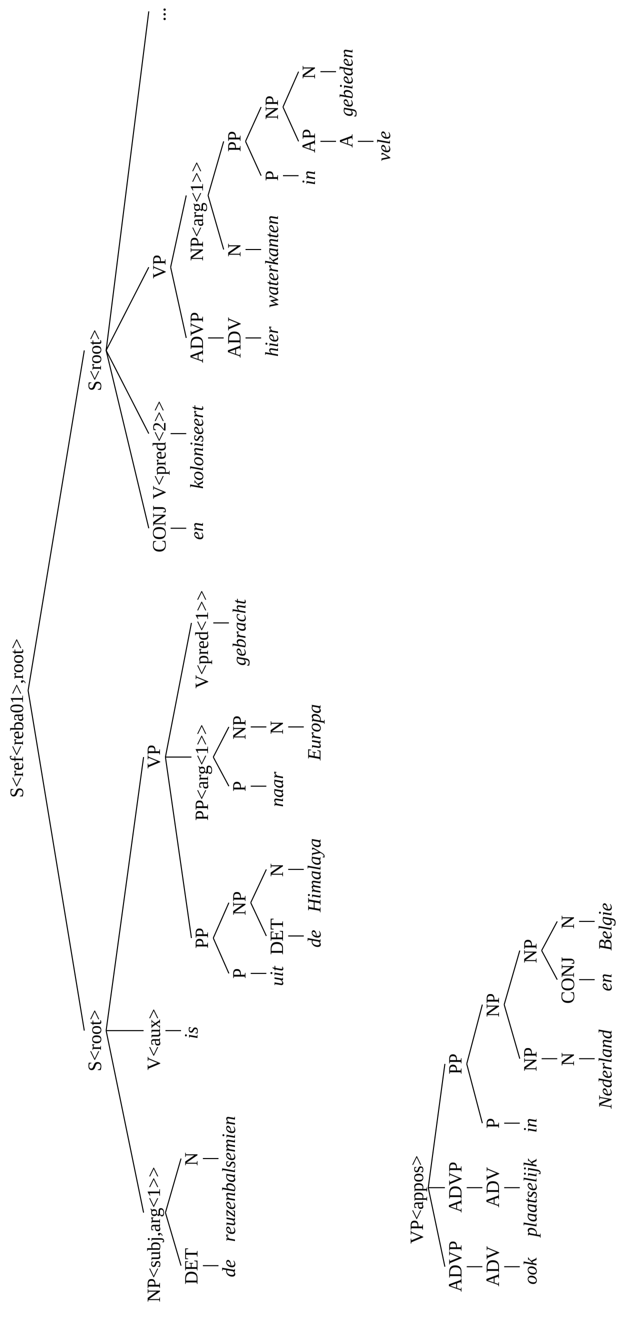


Figure 3.2: Syntactic tree for the sentence in (64).

3.4.3 Accentuation algorithm

In order to test RR, an initial accent distribution is needed. If the initial accentuation makes no sense, it is impossible to judge whether RR applied correctly. Therefore, the first step is accent assignment based on the focus distribution: every focused word is accented; compound words capable of bearing two accents receive two accents (e.g. *donkerbruin* 'dark brown').

However, this is not sufficient yet. Suppose we accent all focused content words in (66-a), and derive the φ's, then RR incorrectly deletes the accent on *later* (66-b). The error is not due to incorrect φ-phrasing, but to the fact that the predicate *verschijnt* should be unaccented. According to SAAR (Section 2.5.2), the predicate is adjacent to its focused argument *een witte bloem*, and must therefore be unaccented. That is, only by applying SAAR before RR can the correct accentuation in (66-c) be obtained .

(66) a. ($_{AdvP}$ ($_{NP}$ een maand) ($_{Adv}$ later)) ($_{V}$ verschijnt) ($_{NP}$
a month later occurs
een witte bloem)
a white flower

b. ?[$_{\varphi}$ een m$\overset{*}{\text{a}}$and l$\overset{\text{ø}}{\text{a}}$ter versch$\overset{*}{\text{i}}$jnt] [$_{\varphi}$ een w$\overset{*}{\text{i}}$tte bl$\overset{*}{\text{o}}$em]

c. [$_{\varphi}$ een m$\overset{*}{\text{a}}$and l$\overset{*}{\text{a}}$ter versch$\overset{\text{ø}}{\text{i}}$jnt] [$_{\varphi}$ een w$\overset{*}{\text{i}}$tte bl$\overset{*}{\text{o}}$em]

In conclusion, evaluating φ-phrasing requires an accentuation algorithm including initial accent assignment and SAAR in order to avoid unintended applications of RR. The algorithm in (67) is in fact identical to that proposed earlier in Section 2.5.4.[17]

(67) **Accentuation algorithm**

1. Assign initial accents
2. Apply SAAR
3. Derive φ's
4. Apply RR

3.4.4 Results and discussion

We distinguished two types of errors: *undershoot errors* are words[18] that are unaccented in the reference, but not deleted by the algorithm; *overshoot errors* are words are accented in the reference, but are deaccented by the algorithm. We assumed that the algorithm, which itself never predicts optional accents, can never make a wrong prediction with respects to an optional accent in the reference. That is, whether the algorithm predicts an accent that is optional

[17] It is implemented in GRAMTSY (Coppen 1990; Coppen 1991), a program to transform trees by means of transformational rules. See Section 6.3.1.

[18] This includes parts of a compound words

Table 3.2: Comparison of the predicted accentuation with the reference accentuation in the form of a confusion matrix.

		Predicted		
		Accented	*Unaccented*	*Total*
Reference	*Accented*	414 (44.76%)	13 (1.41%)	427
	Unaccented	42 (4.54%)	456 (49.30%)	498
	Total	456	469	925

Table 3.3: The number of correct and incorrect applications of the deaccentuation rules.

Rule:	*Correct:*	*Incorrect:*
SAAR	31	7
RR	39	5
Total:	70	12

according to the reference, or fails to predict this accent, either way is considered correct.

The results for one text as produced during automatic comparison are given in Appendix A. The overall results are summarised in Table 3.2 in the form of a confusion matrix. There were 42 undershoot errors and 13 overshoot errors. Since the ten words with optional accents were ignored, the total number of words was 925 (i.e. 935 minus 10). This means that the algorithm correctly predicted the accentuation of 94.05% of all words.

3.4.4.1 RR overshoot errors

Of course, not all overshoot errors are due to RR. Table 3.3 shows the contribution of SAAR and RR to the total number of overshoot errors.[19]

With only five incorrect applications against 39 correct applications, RR appears to perform reasonably well. Two of these errors are shown in (68). The adverb *ook* and the adjective *zeldzame* are both adjacent modifiers, while the pronoun *dit* and the copular verb *is* are function words, so the first φ-boundary

[19]One overshoot error is due to the fact that a word was not initially accented.

is as late as after the noun *soort*. After initial accent assignment[20], RR applies twice (68-b). Compared to the reference (68-c), we see one overshoot error (*ook*) and two undershoot errors (*dit* and *zeldzame*). The deaccentuation of *dit* would not have occurred without the overshoot error, i.e. the initial assignment of an accent to *ook*. However, the incorrect deaccentuation of *zeldzame* can only be attribted to the lack of a φ-boundary after *dit*. We may hypothesise that a focused function word gains the status equivalent to a content word, trigging a φ-boundary .

(68) a. ($_S$ ($_{NP}$ ($_{AdvP}$ ($_{Adv}$ ook)) ($_N$ dit)) ($_V$ is) ($_{VP}$ ($_{NP}$ een
also this is a
zeldzame soort)) . . .)
rare species
'this is a rare species as well'

b. [$_\varphi$ $\overset{*}{\text{o}}$ok d$\overset{\emptyset}{\text{i}}$t is een z$\overset{\emptyset}{\text{e}}$ldzame so$\overset{*}{\text{o}}$rt]

c. ook d$\overset{*}{\text{i}}$t is een z$\overset{*}{\text{e}}$ldzame so$\overset{*}{\text{o}}$rt

The third incorrect application of the RR (69) appears to be caused again by initial accent assignment: the adverb *vaak* should be left unaccented.[21]

(69) a. Kruipend stalkruid heeft houtige vertakte ($_{NP}$ ($_{AP}$
Ononis-Repens has woody branching
($_{Adv}$ vaak) ($_A$ liggende)) ($_N$ stengels)) . . .
often recumbent stems

b. [$_\varphi$ va$\overset{*}{\text{a}}$k l$\overset{\emptyset}{\text{i}}$ggende st$\overset{*}{\text{e}}$ngels]

c. vaak l$\overset{*}{\text{i}}$ggende st$\overset{*}{\text{e}}$ngels

The fourth error is deletion of the accent on *stelen* in (70). We have no explanation here, except that the syntactic structure is rather complex, which may have caused the panel to add an extra accent. The same goes for the fifth and final erroneous application of the RR, not shown here.

(70) a. . . . ($_{AP}$ ($_{PP}$ in ($_{NP}$ ($_N$ kluwens) ($_{PP}$ op ($_{NP}$ ($_A$
in tangles on
overhangende) ($_N$ stelen))))) ($_A$ bijeenstaand)) . . .
overhanging stems gathered
'gathered in tangles on overhanging stems'

b. [$_\varphi$ in kl$\overset{*}{\text{u}}$wens] [$_\varphi$ op $\overset{*}{\text{o}}$verhangende st$\overset{\emptyset}{\text{e}}$len bije$\overset{*}{\text{e}}$nstaand]

c. in kl$\overset{*}{\text{u}}$wens op $\overset{*}{\text{o}}$verhangende st$\overset{*}{\text{e}}$len bije$\overset{*}{\text{e}}$nstaand

[20] The pronoun *dit* is explicitly focused and therefore receives an accent by initial accent assignment.

[21] An alternative interpretation, in which *vaak* is a sentential adverb rather than a modifier of the following adjective, would at least have prevented the deletion of the accent on *liggende*.

3.4.4.2 RR undershoot errors

We encountered 42 undershoot errors. The question is to what extent we can blame the combination of φ-phrasing and RR for the failure to delete accents. This question is considerably harder to answer, because it means that for each error, we have to consider alternative φ-phrasings that would have preserved the accent.

To begin with, at least six errors involved accented content verbs. The accent on a verb can never be deleted by RR, because there is always a φ-boundary after a content verb or after a verb cluster, which means that it can never be followed by another accent in the same φ.

As many as 17 errors stem from accented adverbs. Here, the problem is the initial accent assignment: how to decide whether a given adverb can be accented at all. We have no solution to offer, but will further discuss this issue in Section 7.3.6.3.

The remaining 19 errors may indeed be attributed to φ-phrasing. For example, the fact that (71) has no accent on *soort* could be taken to indicate that *de bekendste soort van de familie* is in fact one φ.[22] Example (72) is structurally identical, and occurs five times in the corpus. These examples suggest, contrary to our position in Section 3.3.2, that a postmodifying PP does not always force a φ-boundary.[23]

(71) a. ... ($_{NP}$ de ($_{AP}$ bekendste) ($_{N}$ soort) ($_{PP}$ van de familie))
the best-known species of the family

b. [$_{\varphi}$ de bek*endste so*ort] [$_{\varphi}$ van de fam*ilie]

c. de bek*endste soort van de fam*ilie

(72) a. ... ($_{PP}$ in ($_{NP}$ ($_{AP}$ grote) ($_{N}$ delen) ($_{PP}$ van Europa)))
in large parts of Europe

b. [$_{\varphi}$ in gr*ote d*elen] [$_{\varphi}$ van Eur*opa]

c. in gr*ote delen van Eur*opa

Another class of undershoot errors consists of those in which the second accent of a compound adjective is not deleted. Example (73) contains the compound adjective *kalkhoudende*, which may carry two accents when it is in sentence-final position. The predicted φ-phrasing causes it to carry two accents (73-b), but the reference prescribes just one (73-c).

[22] Alternatively, we can assume that – for some reason – words like *soort* 'type', *deel* 'part', *grond* 'ground', *gebied* 'area' etc. are never accented when postmodified.

[23] As a matter of fact, we examined the effect of incorporating all postmodifying PP's in the φ of the preceding noun, but found that this results in far too many deaccentuation errors.

(73) a. ... ($_{PP}$ op ($_{AP}$ droge) ($_{AP}$ kalkhoudende) of ($_{AP}$
on dry calcareous or
basische) ($_{N}$ bodems))
alkaline soils

b. [$_{\varphi}$ op dro⃰ge] [$_{\varphi}$ ka⃰lkhou⃰dende]] of ba⃰sische bo⃰dems]

c. op dro⃰ge ka⃰lkhoudende of ba⃰sische bo⃰dems

The same goes for all of the compound adjectives listed in (75).

(74) *losbloemig* loose-flowered
kalkhoudend calcareous
langwerpig elongated
eenjarig annual
kransstandig verticillate
veerlobbig feather-lobate
eironde egg-round
purperroze purple-pink

It should be noted that whether a compound adjective has one or two accents is not always clear. While many adjectives have clearly either one (75-a) or two potential accents (75-b), some appear to allow both possibilities, or at least show variation among speakers (cf. Gussenhoven (1983a, p. 314)).

(75) a. langdra⃰dig
long-winded

b. la⃰ngbe⃰nig
long-legged

c. la⃰ngwe⃰rpig / langwe⃰rpig
long-shaped
'elongated'

Recall that the reference accentuation as produced by the panel did not specify if a compound can potentially have two accents.[24] Rather, this decision was made by the author while manually parsing the sentences. Thus, some of the deviations in (74) may be due to doubt or disagreement with respect to the accent potential of a compound adjective.

In conclusion, the general outcome of this evaluation is that pitch accent placement using a combination of SAAR, φ-phrasing, and RR correctly predicted 94.05% of all 925 accent decisions (ignoring optional accents). Considering the length and complexity of the sentences, this is a rather good result. There were more than three times as many undershoot errors as there were overshoot errors. In other words, we predicted too many rather than too few accents. This suggests that our grammar is too generous, and that additional rules for deaccentuation are needed.

[24]Unless, of course, two accents were prescribed.

With respect to RR, it was found that it applies correctly in 39 cases and incorrectly in only 5 cases. Two of these overshoot errors can be attributed to an incorrect initial accent assignment, whereas the remaining three may be real errors, requiring further research. Furthermore, 19 out of the 42 undershoot errors may in principle be blamed on either RR and/or on φ-phrasing. It would appear that a postmodifying PP is not always preceded by a φ-boundary, as predicted by our φ-phrasing algorithm, although the conditions for integration are unclear. It was suggested that undershoot errors in compound adjectives can be partly explained by variability in their accent potential (one or two accents). Yet, the undershoot errors do not provide us with compelling reasons to modify our φ-phrasing algorithm. At a minimum, we can conclude that the proposed φ-algorithm for Dutch is fairly accurate.

3.5 Summary

In this chapter, we investigated φ-phrasing in Dutch. The reason for this exercise is that our algorithm for pitch accent placement using RR requires that the φ-boundaries are correctly determined. We started by reviewing the φ-phrasing algorithms proposed by Nespor & Vogel and by Hayes, and argued that these are inadequate for Dutch. We then proposed a new φ-algorithm for Dutch based on the notion of adjacent modification. A careful examination of φ-phrasing in a number of syntactic structures revealed that it was not entirely adequate in the case of VP's and S's, which called for a minor revision of the definitions of φ-phrasing and adjacent modification. Although further generalisation is desirable, it provides us with an observationally adequate definition of the Dutch φ.

In the second part of this chapter, the proposed algorithm was evaluated by checking for a substantial number of utterances if applying RR within φ's leads to the preferred accentuation. Rather than a perceptual evaluation, we opted for an evaluation at the symbolic level, using a partly automatic, corpus-based method. The corpus consisted of flower descriptions containing 52 relatively long and complex sentences, with a total of 935 words. The text material was manually parsed and a preferred accentuation was prescribed by a panel of three experts. The predictions of the algorithm for pitch accent placement, which included φ-phrasing and RR application, were compared with the reference. Prediction and reference corresponded in 94.05% of all 925 accent placement decisions. A detailed analysis of the errors of RR showed no compelling reasons to revise the φ-phrasing algorithm. At a minimum, we can conclude that the proposed φ-algorithm for Dutch is fairly accurate. This result may contributes to a better description of the prosodic phonology of Dutch.

Chapter 4

Intonational Phrasing

4.1 Introduction

Our research objective is to design, build, and test a linguistically-based SLG system that generates speech with pitch accents and intonational phrases that are most acceptable to experts (Section 1.3). This chapter addresses intonational phrasing, i.e. the way the flow of speech is segmented into chunks by intonational means. These chunks have been referred to by terms such as *intonational domains*, *intonational phrases*, *tone units*, *tone groups*, and *association domains*. For current purposes, we will use the general term intonational phrase, which is defined as a stretch of speech covered by a perceptually coherent intonation contour. The boundaries of an intonational phrase correspond to a perceived discontinuity in the intonation contour, resulting from of a change in pitch level. In addition, they may be, and usually are, signalled by durational means such as pause or preboundary lengthening. In our view, however, a prosodic boundary marked by durational cues only does not qualify as an intonational boundary (Section 2.3.7).

The work described in this chapter does not directly concern SLG itself. However, since intonational phrasing is an essential task in any SLG system, an adequate approach to intonational phrasing is a prerequisite. Our goal here is to obtain an intonational phrasing algorithm for Dutch that can generate the preferred intonational boundaries for an utterance. This algorithm can then be incorporated in our SLG system, which will be the topic of the following chapters. An additional question in this respect is what linguistic information the algorithm requires as input, because this is the information that the NLG part of the SLG system will have to supply.

Although a considerable amount of linguistic theory about intonational phrasing is available in the literature, most work is based on a small set of crucial observations, and has neither been tested against larger sets of data (Gussenhoven 1992a; Hirst 1993; Ladd 1986; Nespor and Vogel 1986; Selkirk 1984), nor systematically compared with competing approaches (Altenberg

1990; Bachenko and Fitzpatrick 1990; Croft 1995). In the first part of this chapter, therefore, three intonational phrasing algorithms are systematically compared using a substantial set of reference data. The corpus-based method for evaluating these algorithms will be described in Section 4.2; the corpus itself in Section 4.3. In Section 4.4 to Section 4.6, we will successively describe each algorithm and analyse its errors. An intermediary discussion and conclusion will be provided in Section 4.7. Next, a new algorithm will be proposed and tested in Section 4.8. We will end this chapter with a summary.

4.2 Method

The evaluation method used here is essentially the same as that used in the previous chapter (Section 3.4.1) to evaluate phonological phrasing, and is a practical implementation of several methodological choices. As explained in Section 1.1.3, acceptability judgements regarding intonational phrasing may vary, because (1) they may be gradual rather than categorical, (2) alternative phrasings may be equally acceptable, and (3) intersubjective differences may occur. To cope with this variation, we have chosen to aim for the intonation that is most acceptable according to the judgements of intonational experts, ignoring differences between equally acceptable forms. In addition, we argued in Section 1.1.4 that the disadvantage of a perceptual evaluation is that other aspects of the wave form, such as segmental errors, inadequate phonetic realization, or speaker characteristics, may interfere with the judgement of the intonation. We therefore decided to rely on a symbolic assessment of the acceptability judgements of the experts, asking them to prescribe the preferred intonational form for a given texts and comparing this with the prediction by our algorithms. Accordingly, the evaluation method used here encompasses the following steps.

1. **Construction of the corpus**
   Assemble a corpus of sentences. For each sentence, add its syntactic/semantic structure and its preferred intonational structure, i.e. pitch accents[1] and intonational phrase boundaries.

2. **Implementation of intonational phrasing algorithms**
   Implement the intonational phrasing algorithms.

3. **Evaluation of intonational phrasing algorithms**

   (a) *Automatic comparison*
   Apply each algorithm to the sentences in the corpus. For each sentence, automatically compare the predicted boundaries to the preferred boundaries. Report all differences and generate a confusion matrix.

[1] Some of the intonational phrasing algorithms make use of the pitch accent distribution.

(b) *Error analysis*
On the basis of the results of the automatic comparison, trace the origin of systematic errors in the algorithms. Find the advantages and disadvantages of each algorithm, and determine which algorithm has the best overall performance.

The implemented algorithms were selected on the basis of the following criteria. First, given our goal of building a linguistically-based SLG system, purely probabilistic approaches to intonational phrasing (Hirschberg 1991; Ostendorf and Veilleux 1994; Sanders and Taylor 1995; Taylor and Black 1998) were not taken into account.[2]. Second, the algorithm had to be applicable to Dutch, as this is the language we are primarily interested in. The majority of the algorithms address English, and are either not intended for other languages, or would require significant modifications when used for other languages. This excludes for example Altenberg (1990), and Bachenko and Fitzpatrick (1990). Third, the algorithm had to be explicitly described and sufficiently formalised in order to be implemented, which rules out proposals such as those by Dirksen and Quené (1993), Hirst (1993), and Selkirk (1984, 1995). Using these criteria, two algorithms were chosen that represent quite different views on intonational phrasing. The first is by Nespor and Vogel (1986, ch. 7), which is probably the best known algorithm for intonational phrasing, and the second by Gussenhoven (1990, 1992a, 1992a). We also looked at a revised version of the first algorithm as proposed by Caspers (1994, ch. 2). In addition, we tested a new algorithm.

All implementations were written in GRAMTSY (*Grammatical Transformational System*), a programming system that supports the use of transformational rules and grammars to manipulate trees in the form of labelled bracket structures (Coppen 1990). As far as the first three algorithms are concerned, lack of space makes it impossible to discuss all of them in depth, so we refer to the original publications for details. However, some of the problems that arose during implementation are mentioned in the footnotes.

4.3 Corpus

4.3.1 Text

The text material was the same as that used for the evaluation of phonological phrasing (Section 3.4.2), and consisted of 10 descriptions of wild flowers taken from a botanical handbook (Forey 1992). Each description consisted of 4 to 8 sentences. The total number of sentences was 52, the total numbers of words 907, an average of 17.4 words per sentence. An example is given in (1); a number of complete flower descriptions are given in Appendix A and Appendix B. Since most sentences were relatively long and have a rather complex syntactic structure, they formed an appropriate challenge for intonational phrasing

[2] However, these will be addressed in Section 8.5.3

algorithms. Their length and complexity gave rise to a large number of non-trivial instances of intonational phrasing. In contrast, the discourse structure of the texts was fairly simple, and essentially consisted of the introduction of a particular flower and an elaboration on its properties, minimising the potential influence of discourse structure on intonational phrasing was minimal.

(1) De Reuzenbalsemien is uit de Himalaya naar
The Impatiens-Glandulifera has-been from the Himalayas to
Europa gebracht en koloniseert hier waterkanten in vele
Europe brought and colonises here watersides in many
gebieden ook plaatselijk in Nederland en Belgie.
areas also locally in the-Netherlands and Belgium

4.3.2 Intonational structure

The reference intonational phrases were produced by the same panel that was responsible for the accent distribution used in the evaluation of phonological phrasing, and consisted of three intonationally experienced, native speakers of Dutch. They worked on a preprocessed version of the texts, in which commas were removed, semicolons[3] were changed to full stops, and abbreviations and numbers were expanded to their full lexical forms. They were asked to prescribe the intonational structure to be used in a competent reading of the text. This included the *accent distribution* and the *boundary distribution*, i.e. the locations of pitch accents and intonational phrase boundaries. As expected, boundaries turned out to always coincide with word junctures. A distinction was made between *obligatory* and *optional* boundaries, indicated by a '#' and a '|' respectively. This distinction was introduced to accommodate equally acceptable alternative phrasings (Section 1.1.3). The panel discussed disagreements in their phrasings until a consensus was reached. In total, they assigned 109 obligatory boundaries[4] and 42 optional boundaries, against 704 word junctures where no boundary was specified. The prescribed phrasing for (1) is given in (2); see Appendix A and Appendix B for some other sentences.

(2) De R*euzenbalsemien | is uit de Himal*aya naar Eur*opa gebracht #
en kolonis*eert hier w*aterkanten in v*ele gebi*eden # ook pl*aatselijk in
N*ederland en B*elgie. #

4.3.3 Syntactic structure

The sentences were manually parsed by the author according to the syntactic analysis outlined in Section 2.6.[5] These syntactic structures provide all infor-

[3] There were eight of these, each connecting a pair of main clauses, i.e. clauses with the finite verb in second position.

[4] We ignore the 52 trivial obligatory boundaries at the end of a sentence.

[5] We experimented with using syntactic parsers, but they either failed to handle the level of complexity of these sentences, or did not provide the required level of detail.

mation that is required by the intonational phrasing algorithms. The syntactic structure corresponding to the sentence in (2) was shown in Figure 3.2. A number of extra features were added to the parse trees to satisfy the needs of particular phrasing algorithms. For instance, to distinguish arguments from modifiers, arguments were linked to their corresponding predicates by means of a numerical index (cf. arg_1 and $pred_1$ in Figure 3.2). Furthermore, subjects, root sentences[6] (cf. *root* in Figure 3.2), non-restrictive relative clauses and appositional elements (cf. *appos* in Figure 3.2) were indicated by special features. Following Nespor and Vogel (1986), we used a broad definition of *apposition*, one which includes parenthetical expressions, non-restrictive relative clauses, tag questions, vocatives, expletives, and certain moved elements, not all of which necessarily occurred in the corpus.

4.3.4 Prosodic structure

All three intonational phrasing algorithms require information about the prosodic structure. $\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$ both need a parse of the utterance in terms of phonological phrases (φ's). For this, we used the φ-phrasing algorithm for Dutch as proposed in Section 3.3.3.1. In addition, $\mathcal{GUS}$ assumes as input a *recursive* φ-structure. The φ-structure on top of the base-level φ's is constructed in such way that the dominance relations among the φ's mirrors the dominance relations among the syntactic constituents they are derived from. Figure 4.1 shows the recursive prosodic structure derived for sentence (1).[7]

4.4 Intonational phrasing according to Nespor & Vogel

4.4.1 $\mathcal{N}\&\mathcal{V}$ algorithm

The intonational phrasing algorithm by Nespor and Vogel (1986, ch. 7) ($\mathcal{N}\&\mathcal{V}$) is the mapping rule for π's[8], which constitutes a part of their version of prosodic phonology (Section 2.2). In general, such rules receive input from two sources. The first is the morpho-syntactic structure of an utterance. As morphological distinctions are irrelevant for deriving π's, $\mathcal{N}\&\mathcal{V}$ only requires the syntactic structure in which root sentences and appositional elements can be identified. Second, a partially-derived prosodic structure is required, because it is assumed that the mapping into prosodic constituents that are lower on the prosodic hierarchy has already been accomplished. Therefore, $\mathcal{N}\&\mathcal{V}$ requires a parse of

[6]A *root sentence* is an S node that is dominated by S nodes only. The notion of *root sentence* stems from a period in transformational generative syntax in which *root transformations*, such as *Verb Second* and *Topicalisation* in Dutch, only applied to certain cycles. These were called *root transformations* and *root sentences*, respectively.

[7]The π's in this structure are required by $\mathcal{GUS}$ in order to derive AD's. This point will be discussed in Section 4.6.1.

[8]Nespor & Vogel use the symbol 'I' to designate an intonational phrase. We use π for reasons explained in Section 2.3.7.

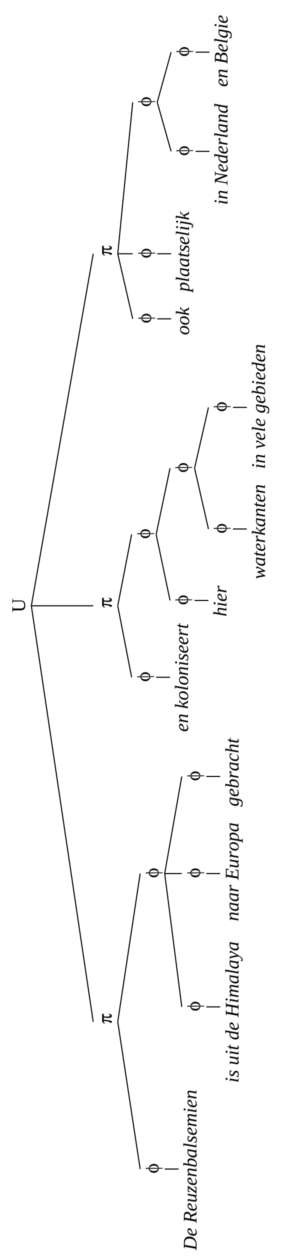


Figure 4.1: Prosodic tree for the sentence in (1); the prosodic categories are Utterance (U), Intonational Phrase (π), and Phonological Phrase (φ).

the utterance in terms of φ's as well. Its output consists of the locations of optional and obligatory π-boundaries.

The phrasing algorithm proceeds in two steps. In the first step, called *π-formation*, the initial π-boundaries are established.

(3) **π-formation**

Join into an n-ary branching π:

a. all the φ's in an appositional string;

b. any remaining sequence of adjacent φ's in a root sentence.

The next step, called *π-restructuring*, involves optionally dividing π's into smaller π's by adding boundaries. π's may be restructured due to factors such as the length of a π, speaking rate, style or contrastive prominence. Although the presence of such boundaries may be optional, their location is highly restricted by the syntax. According to the following definition, restructuring can only occur at certain syntactically defined positions.

(4) **π-restructuring** (optional)

a. Restructure after an NP if this does not interrupt another NP and does not separate an internal argument from its verb.[9]

b. Restructure before an S if this does not interrupt another NP.

The first part inserts optional boundaries after each argument of a verb, except the one that is adjacent to the verb. The second part will insert optional boundaries before an argument S, but not before a restrictive relative clause.

In addition, there is a special provision for restructuring lists.

(5) **List restructuring** (optional):

In a sequence of more than two constituents of the same type, i.e. $x_1, x_2, \ldots x_n$, restructure before each repetition of the node X, i.e. before $x_2, x_3, \ldots x_n$.

Applying $\mathcal{N}\&\mathcal{V}$ to the sentence in (1) gives the phrasing in (6), where ' # ' stands for an obligatory boundary and ' | ' for an optional boundary. Notice that there is no optional boundary after the NP *Europa*, as this would separate the verb *gebracht* from its internal argument *naar Europa*. Also, no boundary occurs after the NP *Nederland*, as this would interrupt the dominating NP.

(6) De R$\overset{*}{\mathrm{e}}$uzenbalsemien | is uit de Himal$\overset{*}{\mathrm{a}}$ya | naar Eur$\overset{*}{\mathrm{o}}$pa gebracht # en kolonis$\overset{*}{\mathrm{e}}$ert hier w$\overset{*}{\mathrm{a}}$terkanten in v$\overset{*}{\mathrm{e}}$le gebi$\overset{*}{\mathrm{e}}$den # ook pl$\overset{*}{\mathrm{a}}$atselijk in N$\overset{*}{\mathrm{e}}$derland en B$\overset{*}{\mathrm{e}}$lgie. #

[9] As noted in Caspers (1994, p. 181), it is not clear whether a subject is an internal or obligatory argument. We follow Caspers who, based on example (18a) in Nespor and Vogel (1986, p. 197), concludes that subjects are not internal arguments. We will also assume that the end of an NP embedded in another NP does not constitute an optional boundary, unless an intervening S node occurs (Caspers 1994, p. 181).

Table 4.1: Confusion matrices for $\mathcal{N}$&$\mathcal{V}$ algorithm. The top matrix contains the absolute number of confusions. The bottom matrix contains the percentage of confusions relative to the total number of boundaries in a column.

Reference

		Obl	*Opt*	*Non*	*Tot*
	Obl	39	0	0	39
N& V	*Opt*	42	34	50	126
	Non	28	8	654	690
	Tot	109	42	704	855

Reference

		Obl	*Opt*	*Non*
	Obl	35.78	0	0
N& V	*Opt*	38.53	80.95	7.10
	Non	25.69	19.05	92.90
	Tot	100	100	100

4.4.2 $\mathcal{N}$&$\mathcal{V}$ results

4.4.2.1 General

Table 4.1 summarises the results for $\mathcal{N}$&$\mathcal{V}$. The kappa value calculated for the matrix (without the optional boundaries) indicates that the agreement between the prediction and the reference is very low, and not significant at the 1%-level ($\kappa = 0.022$, $p = 0.045$). The most salient finding is that the algorithm predicts no incorrect obligatory boundaries. Apparently, all appositional elements and non-restrictive relative clauses are indeed delimited by an obligatory boundary in the reference. The other side of the coin is that, due to this conservative strategy for obligatory π's, 42 (38.53%) of the obligatory boundaries in the reference are predicted to be merely optional, whereas 28 of them (25.69%) are not predicted at all.

Looking at the predicted optional boundaries, we see that $\mathcal{N}$&$\mathcal{V}$ generates a rather large number of incorrect optional boundaries. There are 50 optional boundaries at locations where no boundary should occur, and 42 optional boundaries where instead an obligatory one is required.

In the next subsections, we will look in more detail at the these errors in order to identify their causes. The reference prescribes both optional and obligatory boundaries, the two types that are predicted by the phrasing algorithms as well. Including the category 'absence of a boundary', this means

Table 4.2: Locations that were obligatory boundaries in the reference, but no boundaries according to $\mathcal{N}$&$\mathcal{V}$.

Syntactic configuration:	*Number:*
between AP's	8
before a PP	
before an attributive PP	6
before a non-attributive PP	3
between conjuncts	
between two NP conjuncts	6
between two VP conjuncts	2
before a restrictive relative clause	3
Total:	28

there are nine possible combinations of reference and prediction. To facilitate the discussion, boundary locations will be indicated as 'Reference/Prediction', where both can take the value of '#' (an obligatory boundary), '|' (an optional boundary) or '∅' (no boundary). Thus for example #/ | denotes a location that is an obligatory boundary according to the reference, but an optional boundary according to the algorithm. Whenever a comparison between the two is irrelevant for the discussion, only the prediction will be mentioned.

If the reference prescribes an optional boundary, this is interpreted as: a boundary is acceptable here, but no boundary is equally acceptable. At such positions, therefore, whether the algorithm predicts a boundary, an optional boundary, or no boundary at all is irrelevant. In other words, where an optional boundary is allowed, the prediction can never be wrong. As a consequence, we will ignore incorrect predictions of optional boundaries of the types | / ∅ and | /# , in the error analysis. Instead, we focus on important deviations of the types #/ ∅ , #/ | , ∅/# , and ∅/ | .

4.4.2.2 Obligatory in the reference, but none according to $\mathcal{N}$&$\mathcal{V}$ (#/ ∅)

Table 4.2 lists all syntactic configurations in which a boundary was obligatory according to the reference, but $\mathcal{N}$&$\mathcal{V}$ failed to predict one. Eight of these boundaries occurred between AP's. Although in general we find no boundary between two attributive AP's, a boundary becomes obligatory if the AP's are sufficiently long and/or complex. For example, the reference requires an obligatory boundary between the two AP's in (7), *groenachtig witte* and *netvormig geaderde*. However, $\mathcal{N}$&$\mathcal{V}$ is incapable of predicting a boundary at this position, since it is not the end of a NP, neither the start of an S, nor located between two listed items.

(7) ... # en vijf groenachtig witte #/∅ netvormig geaderde
and five greenish white clathrate veined
kroonbladen.
petals

Nine obligatory boundaries in the reference occurred before a PP, which was used either attributively or non-attributively.[10] According to $\mathcal{N}$&$\mathcal{V}$, however, no boundary can occur between a noun and an attributive PP, as this would break up an NP. In (8), for instance, the algorithm fails to predict a boundary between the noun *klimplant* and the attributive PP *met lange kantige en vertakte stengels*. The absence of a boundary after *klimplant* in combination with the presence of a boundary after *lange* – caused by list restructuring – would result in an odd kind of phrasing. The fact that a boundary is required here may again be explained by the length and/or complexity of the premodifier *borstelig behaarde* in combination with that of the postmodifier *met lange kantige en vertakte stengels*.

(8) Heggerank | is een borstelig behaarde klimplant #/∅ met
Bryony is a setaceously hairy climber with
lange | kantige | en vertakte stengels | ...
long angular and branching stalks

A different problem occurs with non-attributive PP's. Since $\mathcal{N}$&$\mathcal{V}$ only provides for restructuring after an NP or an S, it follows that only PP's preceded by one of these categories can begin a π. In Dutch, however, so called *extraposed* PP's are preceded by the verb(s) of the verb cluster. Consequently, $\mathcal{N}$&$\mathcal{V}$ is unable to account for a boundary before an extraposed PP. An example of this is (9), for which the reference has an obligatory boundary between the verb *plaats* and the extraposed PP *met losbloemige bebladerde trossen van purperroze bloemen*.[11]

(9) Bloei vindt in de zomer plaats #/∅ met losbloemige
bloom does in the summer occur with polysepalous
bebladerde trossen van purperroze bloemen #
foliate bunches of purple-pink flowers

The third type of discrepancy is the context of two coordinated constituents. Example (10) shows coordinated NP's, and (11) two coordinated VP's. In both cases, the conjuncts are separated by an obligatory boundary in the reference, but $\mathcal{N}$&$\mathcal{V}$ cannot account for such a boundary. In (10), a boundary would break up the dominating NP, whereas in (11), the boundary is after the verb,

[10] An *attributive* PP modifies a noun, whereas a *non-attributive* PP modifies a verbal or adjectival predicate.

[11] As a matter of fact, *plaats* is only the first part of the separable verb *plaatsvindt* 'takes place', lit. 'place-takes'. The final part, *vindt*, has moved to the second position of the root sentence, directly after the subject NP.

a position that the algorithm does not accommodate for.

(10) Ze bezitten afstaande kelkbladen #/∅ en vijf groenachtig
they possess protruding sepals and five greenish
witte netvormig geaderde kroonbladen. #
white clathrate veined petals

(11) Hij wordt wel | in tuinen gekweekt #/∅ en in
it is sometimes in gardens grown and into
muntsaus verwerkt. #
mint-sauce processed

The last category of errors concerns boundaries before *restrictive* relative clauses. Since such boundaries interrupt the dominating NP, $\mathcal{N}\&\mathcal{V}$ fails to account for them. However, they occur repeatedly in the reference. An example of this is (12). In fact, the only case in which a boundary before a restrictive relative clause does *not* appear is when the clause is very short, as with the clause *die erop landen* in (13).

(12) ... # en dragen halverwege een bloeikolf #/∅ die schuin
and carry halfway a bloom-cob which slanted
afstaat en volgepakt is met onaangenaam geurende |
stands and packed is with unpleasant smelling
geelachtige | tweeslachtige bloemen. #
yellowish androgynous flowers

(13) insekten ∅ die erop landen | blijven steken # ...
insects that on-it land get stuck

4.4.2.3 Obligatory in the reference, but optional according to $\mathcal{N}\&\mathcal{V}$ (#/|)

As can be seen from Table 4.3, the majority of the boundaries that are obligatory in the reference but merely optional according to $\mathcal{N}\&\mathcal{V}$ occur between the items of a list. The general pattern appears to be that if the items are sufficiently complex, which may be defined as 'containing more than one lexical head', list restructuring becomes obligatory. Alternatively, the complexity of a conjunct may be defined in terms of the number of accents that it contains. Example (14) contains a list with moderately complex items, which are separated by obligatory boundaries in the reference. As matter of fact, there are sentences containing lists with much more complex items, even with items that are lists themselves.

Table 4.3: Locations that were obligatory boundaries in the reference, but optional boundaries according to $\mathcal{N}\&\mathcal{V}$.

Syntactic configuration:	*Number:*
between list items	27
before a non-attributive PP	9
between two conjuncts	
between two PP conjuncts	2
between two VP conjuncts	1
after a subject NP	3
Total:	42

(14) Aarmunt groeit op vochtige plaatsen #/ | langs
Mentha-Spicata grows in humid places along
wegen #/ | en op ruderale terreinen in grote delen van
roads and on ruderal grounds in large parts of
Europa # ...
Europe

Another class of problems are obligatory boundaries before non-attributive PP's. Although these PP's are often preceded by an NP, and consequently by an optional boundary, the reference frequently requires an obligatory boundary before them. For instance, $\mathcal{N}\&\mathcal{V}$ inserts an optional boundary after the NP *smaak en geurmiddel* in (15), while this ought to be an obligatory boundary in order to separate it from the long non-attributive PP *in snoepgoed, likeuren, medicijnen en tandpasta.*

(15) Deze olie | wordt gebruikt als smaak en geurmiddel #/ | in
this oil is used as flavour and aromatic-substance in
snoepgoed | likeuren | medicijnen | en tandpasta. #
candy liqueurs medicine and toothpaste

Yet another type of discrepancy can be found in contexts of the coordination of two constituents. In (16), for instance, the optional boundary between the two PP conjuncts should be an obligatory one according to the reference.

(16) De plant | groeit in venen en natte heide #/ | of op natte
the plant grows in peat and wet heath or on wet
rotsen ...
rocks

Table 4.4: Locations that had no boundaries in the reference, but were optional boundaries according to $\mathcal{N}\&\mathcal{V}$.

Syntactic configuration:	*Number:*
after a subject NP	34
after a non-attributive PP	9
between two PP conjuncts	5
after a non-subject NP	1
before an S	1
Total:	50

Finally, there are a number of sentences in the corpus where a subject NP must be followed by an obligatory boundary, as in (17). Again, this may be attributed to the complexity of the constituent.

(17) De geelgroene bovenzijde van de bladschijf #/ | is bedekt
the yellow-green top of the leaf-slice is covered
door een weke | slijmige laag # ...
by a pappy slimy film

4.4.2.4 None in the reference, but optional according to $\mathcal{N}\&\mathcal{V}$ ($\emptyset$/ |)

The bulk of the predicted optional boundaries that do not correspond to a boundary in the reference occur after a subject NP (see Table 4.4). Since a lot of subject NP's are rather short and uncomplicated – typically only a determiner and a noun – it makes sense that they are not followed by an optional boundary. According to $\mathcal{N}\&\mathcal{V}$, however, a subject is not an internal argument, and therefore insensitive to the constraint that forbids separating an internal argument from its verb. As a matter of fact, $\mathcal{N}\&\mathcal{V}$ would perform significantly better if this constraint were to extend to subjects as well. However, this would in turn introduce errors of another type, albeit less than the present 34. For instance, it would prevent the boundary after the subject NP in (17).

Some of the predicted optional boundaries are wrong because they create optional π's that lack a pitch accent. This is the case in (18), where an optional boundary after the subject NP *Deze olie* creates an unaccented π, which are generally considered to be unacceptable.[12]

(18) Deze olie $\emptyset$/ | wordt gebruikt als sm$\overset{*}{a}$ak en ge$\overset{*}{u}$rmiddel | ...
This oil is used as flavour and aromatic-substance

[12] There are a few exceptions to the rule; see Tone Copy in Gussenhoven (1992a, p. 96).

Other incorrectly predicted optional boundaries occur after NP's that function as complements in non-attributive PP's. Such a boundary occurs in (19) after the PP *in het wild.* For the same reasons, an optional boundary is predicted after the PP *om de pepermuntolie.* Since the verb *gekweekt* carries no pitch accent, the predicted boundary would give rise to an unaccented π.

(19) Hij groeit in het w*ild $\emptyset$/ | op v*ochtige pla*atsen # en wordt
it grows in the wild in humid places and is
o*ok om de peperm*untolie $\emptyset$/ | gekweekt. #
also for the peppermint-oil grown

Finally, we find inappropriate optional boundaries between coordinated PP's. For example, (20) has incorrect boundaries between *in akkers* and *op braakliggende gronden*, as well as between *in wegbermen* and *langs rivieren.*

(20) ... # en is op veel plaatsen | ingeburgerd in akkers $\emptyset$/ | en
and is in many places established in fields and
op braakliggende grond | in wegbermen $\emptyset$/ | en langs
on fallow ground at roadsides and along
rivieren | ...
rivers

4.5 Intonational phrasing according to Caspers

4.5.1 $\mathcal{CAS}$ algorithm

Caspers performed a series of experiments about how Dutch speakers adapt their accent-lending and boundary-marking pitch movements during fast speech (Caspers 1994). As part of one of these experiments, she took three short texts (1,327 words, 52 sentences) read aloud by a professional speaker and transcribed all boundary-marking pitch movements. She also located all obligatory and optional π-boundaries as predicted by $\mathcal{N}\&\mathcal{V}$. Comparing the actual boundaries with the predicted boundaries, she found that 44% of the boundary-marking pitch movements occurred at places where no π-boundary was predicted (p. 37). Unpredicted boundaries were found before a PP (27%), before a restrictive relative clause (9%), after a conjunction (11%), between the items of a list of two (8%), and in indeterminate configurations (45%) (p. 186). In a follow-up experiment, she did the same for a short text (6 sentences) read aloud by six non-professional readers, although this time the beginning of a restrictive relative clause and the beginning of a PP counted as predicted optional boundaries as well. The number of unpredicted boundaries, at least for this text, dropped to 7% (p. 51). We will refer to this extension of $\mathcal{N}\&\mathcal{V}$ as $\mathcal{CAS}$. *π-formation* and *List restructuring* are identical, but *π-restructuring* is as follows.

(21) **π-restructuring** (optional):

a. Restructure after an NP if this does not interrupt another NP and does not separate and internal argument from its verb.

b. Restructure before an S.

c. Restructure before a PP.

Applying $\mathcal{CAS}$ to the sentence (1) gives (22). The difference with (21) consists of two extra optional boundaries: before the PP *in vele gebieden* and before the PP *in Nederland en Belgie.*

(22) De R$\overset{*}{\text{e}}$uzenbalsemien | is uit de Himal$\overset{*}{\text{a}}$ya | naar Eur$\overset{*}{\text{o}}$pa gebracht # en kolonis$\overset{*}{\text{e}}$ert hier w$\overset{*}{\text{a}}$terkanten | in v$\overset{*}{\text{e}}$le gebi$\overset{*}{\text{e}}$den # ook pl$\overset{*}{\text{a}}$atselijk | in N$\overset{*}{\text{e}}$derland en B$\overset{*}{\text{e}}$lgie. #

4.5.2 $\mathcal{CAS}$ results

4.5.2.1 General

The overall results for $\mathcal{CAS}$ are presented in Table 4.5. The kappa value calculated for the matrix (without the optional boundaries) is identical to that for $\mathcal{N}\&\mathcal{V}$: the agreement between the prediction and the reference is very low and not significant at the 1%-level ($\kappa = 0.022$, $p = 0.045$). Recall that the part of the algorithm that decides on obligatory boundaries is the same for $\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$. Therefore, the predicted obligatory boundaries are identical to those for $\mathcal{N}\&\mathcal{V}$. However, in contrast to $\mathcal{N}\&\mathcal{V}$, $\mathcal{CAS}$ allows optional boundaries before PP's and restrictive relative clauses as well. As a consequence, it predicts more optional boundaries: sixteen obligatory boundaries that were not predicted at all by $\mathcal{N}\&\mathcal{V}$ are now predicted as optional by $\mathcal{CAS}$.

Contrary to what might be expected, the coverage of optional boundaries does not improve. From this, we can deduce two things. First, that there are no optional boundaries in the reference before a PP or a restrictive relative clause that were not already covered by $\mathcal{N}\&\mathcal{V}$. Second, there are still optional boundaries in the reference which are not located at the end of a NP, the beginning of an S, or the beginning of a PP. Thus, the extension of the $\mathcal{N}\&\mathcal{V}$ algorithm with respect to optional boundaries appears not to have the intended result.

In addition, the number of predicted optional boundaries where no boundary was allowed, which was 50 in $\mathcal{N}\&\mathcal{V}$, radically increases to 129 in $\mathcal{CAS}$.

In other words, a small improvement in the coverage of obligatory boundaries (realized as optional by $\mathcal{CAS}$ instead of not at all by $\mathcal{N}\&\mathcal{V}$) must be paid for with an excessive number of incorrect optional boundaries. This conclusion in fact agrees with the result of Caspers' experiments, which primarily addressed 'undershoot', i.e. actual boundaries in unpredicted locations; 'overshoot', i.e. predicted boundaries that are not actually realized, was of no direct concern. She observes, however, that one third of the predicted boundaries

Table 4.5: Confusion matrices for $\mathcal{CAS}$ algorithm. The top matrix contains the absolute number of confusions. The bottom matrix contains the percentage of confusions relative to the total number of boundaries in a column.

		Reference *Obl*	*Opt*	*Non*	*Tot*
$\mathcal{CAS}$	*Obl*	39	0	0	39
	Opt	58	34	129	221
	Non	12	8	575	595
	Tot	109	42	704	855

		Reference *Obl*	*Opt*	*Non*
$\mathcal{CAS}$	*Obl*	35.78	0	0
	Opt	53.21	80.95	18.32
	Non	11.01	19.05	81.68
	Tot	100	100	100

before a restrictive clause and almost three quarters of the boundaries before a PP did not receive a boundary-marking pitch movement (Caspers 1994, p. 51).

4.5.2.2 Obligatory in the reference, but none according to $\mathcal{CAS}$ ($\#/\emptyset$)

When we compare Table 4.6 to Table 4.2, we can conclude that the most important difference with $\mathcal{N}\&\mathcal{V}$ is that there are no missed obligatory boundaries before a PP or a restrictive relative clause. Moreover, the number of missed obligatory boundaries between AP's is halved. This is because some AP's are premodified by a PP, and consequently $\mathcal{CAS}$ places an optional boundary before them. Nevertheless, this result is not entirely satisfactory, for boundaries that $\mathcal{N}\&\mathcal{V}$ missed altogether are only predicted to be optional by $\mathcal{CAS}$, instead of obligatory.

4.5.2.3 Obligatory in the reference, but optional according to $\mathcal{CAS}$ ($\#/|$)

Errors of the type $\#/|$ are listed in Table 4.7. For several locations, the results are the same as for $\mathcal{N}\&\mathcal{V}$: between list items, between conjuncts and after a subject NP. However, the three obligatory boundaries before a restrictive relative clause, which $\mathcal{N}\&\mathcal{V}$ missed altogether (cf. Table 4.2), now appear as

Table 4.6: Locations that were obligatory boundaries in the reference, but no boundaries according to $\mathcal{CAS}$.

Syntactic configuration:	*Number:*
between AP's	4
between conjuncts	
between two NP conjuncts	6
between two VP conjuncts	2
Total:	12

Table 4.7: Locations that were obligatory boundaries in the reference, but optional boundaries according to $\mathcal{CAS}$.

Syntactic configuration:	*Number:*
between list items	27
before a PP	
before an attributive PP	6
before a non-attributive PP	12
before a PP modifying an A	4
between two conjuncts	
between two PP conjuncts	2
between two VP conjuncts	1
before a restrictive relative clause	3
after a subject NP	3
Total:	58

Table 4.8: Locations that had no boundaries in the reference, but were optional boundaries according to $\mathcal{CAS}$.

Syntactic configuration:	*Number:*
before a PP	
before an attributive PP	36
before a non-attributive PP	42
before a PP modifying an A	1
after a non-attributive PP	9
after a subject NP	34
between two PP conjuncts	5
after a non-subject NP	1
before an S	1
Total:	129

optional boundaries. The same goes for the six obligatory boundaries before an attributive PP, as well as for the four between AP's premodified by a PP.

4.5.2.4 None in the reference, but optional according to $\mathcal{CAS}$ ($\emptyset / \mid$)

The errors in Table 4.8 are to a large extent identical to the errors by $\mathcal{N}\&\mathcal{V}$ (cf. Table 4.4). As will be clear from a comparison of the two tables, the excessive number of incorrect optional boundaries is entirely due to those before a PP.

4.6 Intonational phrasing according to Gussenhoven

4.6.1 $\mathcal{GUS}$ algorithm

The previous two algorithms stem from prosodic phonology, which is primarily interested in prosodic phrasing at different levels and the way this affects the behaviour of segmental rules. It acknowledges the intonational phrase as one of these levels, but it tends to abstract away from intonational aspects such as pitch accent placement and the shape of the pitch contour, aspects of intonation that are addressed in the body of work that has recently become known as *intonational phonology.* Consequently, intonational phonology is in a position to criticise this abstraction by showing that the accent distribution and the pitch contour do affect intonational phrasing.

This is exactly the type of criticism that can be found in Gussenhoven (1992a, 1992a). Gussenhoven acknowledges that the morpho-syntactic structure is relevant to the derivation of intonational phrases, but shows that the pitch accent distribution is just as relevant. In fact, Gussenhoven takes the pitch accent distribution as the point of departure for intonational phrasing. In his view, every pitch accent needs to be realized in a certain part of the utterance. In autosegmental terms: the individual tones of a pitch accent need to be associated within a particular stretch of speech, which is called the pitch accent's *association domain* (AD).

AD's are derived by means of an algorithm that links them to both the accent distribution and the prosodic constituent structure of an utterance. Thus, the input to this phrasing algorithm consists of an accent distribution and a prosodic structure. As already mentioned in Section 4.3.4, this prosodic structure is somewhat different from the one assumed by $\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$. It does not conform to the *Strict Layer Hypothesis*, since recursion is allowed at the φ level.[13] Furthermore, its π's remain as derived by π-formation; there is no subsequent π-restructuring. Given these assumptions regarding the prosodic structure, the output of the algorithm consists of the locations of optional and obligatory AD boundaries.

It is important to keep apart the notions of π and AD. In Gussenhoven's view (cf. Section 2.3.7), π is still a part of the prosodic structure, but is no longer the domain of a single, coherent intonation contour; it is the AD which has taken over this property. The reason to preserve the π is that it constitutes a level of the prosodic structure, and is therefore relevant to the derivation of AD's and as the domain of application of certain segmental rules, just like the level of φ's is.[14]

As in the previous algorithms, phrasing proceeds in two steps. The first step, known as *AD derivation*, derives for every accent a corresponding unique boundary.

(23) **AD derivation**

For every accent, place an AD boundary at the end of the *highest* prosodic constituent that dominates it, but which does *not* dominate the next accent.

For the sentence in Figure 4.1, AD derivation results in the following AD's.

(24) De R$\overset{*}{\text{e}}$uzenbalsemien # is uit de Himal$\overset{*}{\text{a}}$ya # naar Eur$\overset{*}{\text{o}}$pa gebracht # en kolonis$\overset{*}{\text{e}}$ert # hier w$\overset{*}{\text{a}}$terkanten # in v$\overset{*}{\text{e}}$le # geb$\overset{*}{\text{i}}$eden # ook pl$\overset{*}{\text{a}}$atselijk # in N$\overset{*}{\text{e}}$derland # en B$\overset{*}{\text{e}}$lgie. #

The second step, called *AD restructuring*, joins one or more AD's into a

[13] See example (5) in Gussenhoven (1992a, p. 93). A strictly layered prosodic structure would in fact pose serious problems to AD-restructuring (Marsi, Coppen, Gussenhoven, and Rietveld 1997).

[14] Given this state of affairs, it is actually rather confusing to call an π an *intonational phrase*, and a different name would be more appropriate.

single one by deleting boundaries. This accounts for the observation that often several accents share a common AD. Restructuring is in part optional. Also, there is a constraint on the order of restructuring: precedence must be given to those boundaries that coincide with lower ranking prosodic constituents only (Gussenhoven and Rietveld 1992a, p. 288). This implies, for example, that φ-final AD boundaries must be deleted before π-final AD boundaries. Still, this formulation of restructuring is not specific enough to build an implementation on, because in principle every boundary can be deleted. Also, since there is no clear distinction between optional and obligatory boundaries, it difficult to compare the results to those of the two other algorithms. What is needed is some sort of principle to determine which boundaries will be deleted, and which boundaries will be turned into optional ones. For the purpose of evaluation, the following implementation of restructuring was used, which relies on both accent distribution and prosodic constituency.[15]

(25) **AD restructuring**

a. Delete all boundaries within a prosodic constituent that contains at most three accents.

b. Change all boundaries within a prosodic constituent that contains four accents into optional boundaries.

Restructuring the boundaries in (24) gives (26).

(26) De R$\overset{*}{\text{e}}$uzenbalsemien is uit de Himal$\overset{*}{\text{a}}$ya naar Eur$\overset{*}{\text{o}}$pa gebracht # en kolonis$\overset{*}{\text{e}}$ert | hier w$\overset{*}{\text{a}}$terkanten in v$\overset{*}{\text{e}}$le gebi$\overset{*}{\text{e}}$den # ook pl$\overset{*}{\text{a}}$atselijk in N$\overset{*}{\text{e}}$derland en B$\overset{*}{\text{e}}$lgie. #

4.6.2 $\mathcal{GUS}$ results

4.6.2.1 General

The results for $\mathcal{GUS}$ are in Table 4.9. The kappa value calculated for the matrix (without the optional boundaries) indicates that the agreement between the prediction and the reference is rather high, and significant at the 1%-level ($\kappa = 0.876$, $p = 0.000$). In contrast to the other two algorithms, it predicts a considerable number of incorrect obligatory boundaries (33). On the other hand, it covers 86.24% of the obligatory boundaries in the reference. As for optional boundaries, $\mathcal{GUS}$ predicts only 40.48% of them. In exchange for this, the overshoot of optional boundaries is fairly limited in comparison to the other two algorithms.

Errors tend to fall in two categories. Recall that $\mathcal{GUS}$ requires a prosodic structure that is recursive at the φ-level. Accordingly, a number of the errors can be traced back to an inadequate prosodic structure. The majority of the

[15] It was found that using these numbers, i.e. three and four accents, gave the best results. However, there is no theoretical motivation for it.

Table 4.9: Confusion matrices for $\mathcal{GUS}$ algorithm. The top matrix contains the absolute number of confusions. The bottom matrix contains the percentage of confusions relative to the total number of boundaries in a column.

		Reference			
		Obl	*Opt*	*Non*	*Tot*
$\mathcal{GUS}$	*Obl*	94	12	21	127
	Opt	13	17	12	42
	Non	2	13	671	686
	Tot	109	42	704	855

		Reference		
		Obl	*Opt*	*Non*
$\mathcal{GUS}$	*Obl*	86.24	28.57	2.98
	Opt	11.93	40.48	1.70
	Non	1.83	30.95	95.31
	Tot	100	100	100

errors, however, is caused by an accent distribution that either incorrectly triggers restructuring or fails to trigger restructuring.

Some of the errors of $\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$ return in $\mathcal{GUS}$. For example, failure to predict obligatory boundaries between list items, and between two AP's. Other errors, like incorrectly predicted boundaries between conjuncts, do not recur. Also, due to the role of the accent distribution, creating intonational phrases without accents is avoided.

4.6.2.2 Obligatory in the reference, but none according to $\mathcal{GUS}$ ($\#/\emptyset$)

There were only two instances where $\mathcal{GUS}$ failed to predict an obligatory boundary (see Table 4.10). These two errors are in (27) and (28), where the b-parts show their prosodic structure. Recall that AD restructuring deletes all boundaries within a prosodic constituent that contains at most three accents. In (27), there should be an obligatory boundary after the subject NP *insekten die erop landen*, but since the dominating π contains only three accents, all boundaries inside it are deleted. In (28), the PP modifiers *in de bladoksels* and *aan verschillende planten* belong to a single φ that has just three accents. As a consequence, $\mathcal{GUS}$ is incapable of predicting the obligatory boundary that separates them. Interestingly, this boundary contributes to the disambiguation

Table 4.10: Locations that were obligatory boundaries in the reference, but no boundaries according to $\mathcal{GUS}$.

Syntactic configuration:	*Number:*
after subject NP	1
after non-attributive PP	1
Total:	2

of the sentence: it prevents an attributive interpretation of the final PP.

(27) a. Ins$\overset{*}{\text{e}}$kten die erop l$\overset{*}{\text{a}}$nden #/ ∅ blijven st$\overset{*}{\text{e}}$ken # ...
insects that on-it land get stuck

b. [$_\pi$
  [$_\varphi$ Ins$\overset{*}{\text{e}}$kten]
  [$_\varphi$ die erop l$\overset{*}{\text{a}}$nden] #/ ∅
  [$_\varphi$ blijven st$\overset{*}{\text{e}}$ken]]

(28) a. De m$\overset{*}{\text{a}}$nnelijke en vr$\overset{*}{\text{o}}$uwelijke bl$\overset{*}{\text{o}}$emen # groeien in de
the male and female flowers grow in the

bl$\overset{*}{\text{a}}$doksels #/ ∅ aan versch$\overset{*}{\text{i}}$llende pl$\overset{*}{\text{a}}$nten. #
leaf-axils on different plants

b. [$_\pi$
  [$_\varphi$
    [$_\varphi$ de m$\overset{*}{\text{a}}$nnelijke]
    [$_\varphi$ en vr$\overset{*}{\text{o}}$uwelijke bl$\overset{*}{\text{o}}$emen]]
  [$_\varphi$ groeien]
  [$_\varphi$
    [$_\varphi$ in de bl$\overset{*}{\text{a}}$doksels] #/ ∅
    [$_\varphi$ aan versch$\overset{*}{\text{i}}$llende pl$\overset{*}{\text{a}}$nten]]]

4.6.2.3 Obligatory in the reference, but optional according to $\mathcal{GUS}$ (#/ |)

Table 4.11 shows that a number of optional boundaries that should be obligatory occur in lists. For example, the list items in (29) constitute three separate φ's, which are in turn dominated by a higher φ. The latter contains four accents, which causes the boundaries separating the list items to become optional (cf. the principle of AD restructuring in Section 4.6.1). In other words,

Table 4.11: Locations that were obligatory boundaries in the reference, but optional boundaries according to $\mathcal{GUS}$.

Syntactic configuration:	*Number:*
between list items	6
between two AP's	4
between two VP conjuncts	2
before a non-attributive PP	1
Total:	13

whether the boundaries between list items are correctly predicted depends – – quite arbitrarily – on the number of accents in the list.

(29) a. ... op recht$\overset{*}{\text{o}}$pstaande | / ∅ $\overset{*}{\text{o}}$nbebladerde | / ∅ tot vi$\overset{*}{\text{j}}$ftien
at erect leaf-less up-to fifteen
centimeter hoge st$\overset{*}{\text{e}}$ngels. #
centimetres high stalks

b. [$_\varphi$
[$_\varphi$ op recht$\overset{*}{\text{o}}$pstaande] | / ∅
[$_\varphi$ $\overset{*}{\text{o}}$nbebladerde] | / ∅
[$_\varphi$ tot vi$\overset{*}{\text{j}}$ftien centimeter hoge st$\overset{*}{\text{e}}$ngels]]

For sequences of two AP's or VP's, the problem is essentially the same. Most of these do not contain enough accents to retain the obligatory boundary that, according to the reference, should separate their constituents.

4.6.2.4 None in the reference, but obligatory according to $\mathcal{GUS}$ (∅/#)

Table 4.12 lists all locations that were no boundaries in the reference, but obligatory boundaries according to $\mathcal{GUS}$. Recall that the dominance relations between φ's mirror the dominance relations between the syntactic constituents they are derived from (cf. Section 4.3.4). In particular, information about the level of attachment of an attributive PP remains encoded in the φ-structure. Consequently, a certain syntactic structure may lead to an inadequate φ-structure, which may in the case of $\mathcal{GUS}$ ultimately cause erroneous intonational boundaries. At least two of the incorrect boundaries before an attributive PP may in fact be traced back to the syntactic analysis.

To begin with, the boundary before the final PP in (30) – *in Europa* – would not appear if the prosodic constituents were as in (30-c) instead of (30-b),

Table 4.12: Locations that had no boundaries in the reference, but were obligatory boundaries according to $\mathcal{GUS}$.

Syntactic configuration:	*Number:*
before a PP	
before an attributive PP	6
before a non-attributive PP	6
after a subject NP	6
after a non-subject NP	2
after a verb	1
Total:	21

because all boundaries within a constituent containing at most three accents are deleted. This structure would arise if the PP were syntactically analysed as a modifier of *krijt en kalksteenbodems* rather than of *graslanden*. Thus, the question of whether this boundary is correct or not appears to depend on the syntactic analysis, and therefore ultimately on one's interpretation of the sentence.

(30) a. ... # in k$\overset{*}{\text{o}}$rte gr$\overset{*}{\text{a}}$slanden # op kr$\overset{*}{\text{ij}}$t en
in short grassland at chalk and
k$\overset{*}{\text{a}}$lksteenbodems $\emptyset$/# in Eur$\overset{*}{\text{o}}$pa. #
limestone-grounds in Europe

b. [$_\varphi$
[$_\varphi$ in k$\overset{*}{\text{o}}$rte gr$\overset{*}{\text{a}}$slanden] #
[$_\varphi$ op kr$\overset{*}{\text{ij}}$t en k$\overset{*}{\text{a}}$lksteenbodems] $\emptyset$/#
[$_\varphi$ in Eur$\overset{*}{\text{o}}$pa]]

c. [$_\varphi$
[$_\varphi$ in k$\overset{*}{\text{o}}$rte gr$\overset{*}{\text{a}}$slanden] #
[$_\varphi$
[$_\varphi$ op kr$\overset{*}{\text{ij}}$t en k$\overset{*}{\text{a}}$lksteenbodems] $\emptyset$/ $\emptyset$
[$_\varphi$ in Eur$\overset{*}{\text{o}}$pa]]]

The mirror image of this appears in (31). The boundary before the PP *van elliptische bladen* would be deleted with the prosodic analysis in (31-c) instead of that in (31-b), which would in turn require a syntactic analysis in which the PP *met een tot twintig centimeter hoge stevige bloeistengel* has scope over *een rozet van elliptische bladen*.

(31) a. … een roz$\overset{*}{e}$t $\emptyset/\#$ van ell$\overset{*}{i}$ptische bl$\overset{*}{a}$den # met een tot
a rosette of elliptical leaves with a up-to

tw$\overset{*}{i}$ntig centimeter h$\overset{*}{o}$ge | st$\overset{*}{e}$vige blo$\overset{*}{e}$istengel #
twenty centimeter high firm flower-stalk

b. $[_\varphi$
$[_\varphi$ een roz$\overset{*}{e}$t] $\emptyset/\#$
$[_\varphi$ van ell$\overset{*}{i}$ptische bl$\overset{*}{a}$den] #
$[_\varphi$
$[_\varphi$ met een tot tw$\overset{*}{i}$ntig centimeter h$\overset{*}{o}$ge] |
$[_\varphi$ st$\overset{*}{e}$vige blo$\overset{*}{e}$istengel]]] #

c. $[_\varphi$
$[_\varphi$
$[_\varphi$ een roz$\overset{*}{e}$t] $\emptyset/\emptyset$
$[_\varphi$ van ell$\overset{*}{i}$ptische bl$\overset{*}{a}$den]] #
$[_\varphi$
$[_\varphi$ met een tot tw$\overset{*}{i}$ntig centimeter h$\overset{*}{o}$ge] |
$[_\varphi$ st$\overset{*}{e}$vige blo$\overset{*}{e}$istengel]]] #

As for all other incorrect boundaries before an attributive PP, these are again a matter of too many accents, which trigger the deletion of the boundary.

Incorrect obligatory boundaries before non-attributive PP's invariably involve an extraposed PP, i.e. a PP after the verbal cluster containing the verbal or adjectival predicate. An example of this is (32). An alternative prosodic structure, where the φ's *is bedekt* and *door een weke slijmige laag* are joined into a higher φ, would solve the problem. Still, some of the extraposed PP's contain more than four accents, and will therefore be separated anyway.

(32) a. De g$\overset{*}{e}$elgroene b$\overset{*}{o}$venzijde van de bl$\overset{*}{a}$dschijf # is
the yellow-green top of the leaf is

bed$\overset{*}{e}$kt $\emptyset/\#$ door een w$\overset{*}{e}$ke sl$\overset{*}{i}$jmige la$\overset{*}{a}$g # …
covered by pappy slimy layer

b. $[_\pi$
$[_\varphi$
$[_\varphi$ De g$\overset{*}{e}$elgroene b$\overset{*}{o}$venzijde]
$[_\varphi$ van de bl$\overset{*}{a}$dschijf]] #
$[_\varphi$ is bed$\overset{*}{e}$kt] $\emptyset/\#$
$[_\varphi$
$[_\varphi$ door een w$\overset{*}{e}$ke]
$[_\varphi$ sl$\overset{*}{i}$jmige la$\overset{*}{a}$g]]] #

Table 4.13: Locations that had no boundaries in the reference, but were optional boundaries according to $\mathcal{GUS}$.

Syntactic configuration:	*Number:*
before a PP	
before an attributive PP	5
before a non-attributive PP	2
after a subject NP	2
rest	3
Total:	21

Finally, incorrect obligatory boundaries occur after subject NP's because the following φ contains too many accents. This happens when one of the arguments is a list, as in (33).

(33) a. De bl$\overset{*}{\text{a}}$den $\emptyset/\#$ zijn gr$\overset{*}{\text{o}}$ot | l$\overset{*}{\text{a}}$ngwerpig | l$\overset{*}{\text{a}}$nggesteeld |
the leaves are big elongated long-stalked
en get$\overset{*}{\text{a}}$nd. #
and serrated

b. $[_\pi$
$[_\varphi$ de bl$\overset{*}{\text{a}}$den] $\emptyset/\#$
$[_\varphi$
$[_\varphi$ zijn gr$\overset{*}{\text{o}}$ot] |
$[_\varphi$ l$\overset{*}{\text{a}}$ngwerpig] |
$[_\varphi$ l$\overset{*}{\text{a}}$nggesteeld] |
$[_\varphi$ en get$\overset{*}{\text{a}}$nd]]] #

4.6.2.5 None in the reference, but optional according to $\mathcal{GUS}$ ($\emptyset$/ |)

These errors (see Table 4.13) are similar to those described in the previous section and their origin is essentially the same. The difference is that just three accents are involved, instead of four, so that the incorrectly predicted boundaries are merely optional instead of obligatory.

4.7 Intermediary discussion and summary

So far, three intonational phrasing algorithms have been evaluated. Although all of them perform reasonably well for the domain investigated, they show a number of systematic errors. $\mathcal{N}\&\mathcal{V}$ fails to predict boundaries, both obligatory and optional, because the syntactically defined set of possible boundary locations is too restricted. Furthermore, it predicts a considerable number of incorrect optional boundaries. $\mathcal{CAS}$ extends the set of possible optional boundary locations with the positions before a PP and before a non-restrictive S. Yet, this extension does not improve the coverage of optional boundaries, although it has the effect that a number of obligatory boundaries are now realized as optional instead of not at all. The price for this is a large number of incorrect optional boundaries.

In defence of Nespor & Vogel and Caspers, it should be noted that they never claimed that *all* their optional boundaries are actually valid. In their opinion, this depends on factors like speaking rate, length, style and prominence. For the same reason, some of their optional boundaries can in fact become obligatory. Such semantic and performance factors have not been formalised, and consequently could not be implemented. In general, intonational phrasing algorithms are claimed to predict *acceptable* boundaries, whereas our evaluation of them was directed toward predicting the *most acceptable* (preferred) boundaries. Still, the fact that the set of possible boundary locations is too restricted remains a serious problem. Neither $\mathcal{N}\&\mathcal{V}$ nor $\mathcal{CAS}$ can account for boundaries between AP's or conjuncts. Also, there is no provision to avoid empty intonational domains.

The main flaws of $\mathcal{GUS}$ are that it predicts too many obligatory boundaries and that it is not very good at predicting optional boundaries. The main reason for this is the lack of an adequate account of restructuring. The process depends on counting accents within a prosodic constituent, but the number of accents that triggers restructuring, i.e. three for obligatory restructuring and four for optional restructuring, is arbitrary. Yet, it is hard to think of a better account of restructuring, because the prosodic structure and the accent distribution are the only things available as input to $\mathcal{GUS}$. This makes it impossible to detect certain configurations – syntactic or otherwise – that require a particular intonational phrasing, as for lists, sequences of complex AP's, or coordinations. In other words, the prosodic structure is not *informative* enough for the assignment of intonational phrasing.

This finding is in line with the results of an experiment on AD restructuring in Marsi, Coppen, Gussenhoven, and Rietveld (1997), which showed that a prosodic structure without recursion at the φ-level[16] is not informative enough to account for the different preferences in AD-restructuring immediately before *attributive* and *non-attributive* PP's.

[16] That is, a prosodic structure that conforms to the Strict Layer Hypothesis, which states that a prosodic constituent of a certain rank cannot be dominated by prosodic constituent of the same rank, only by one of the next higher rank.

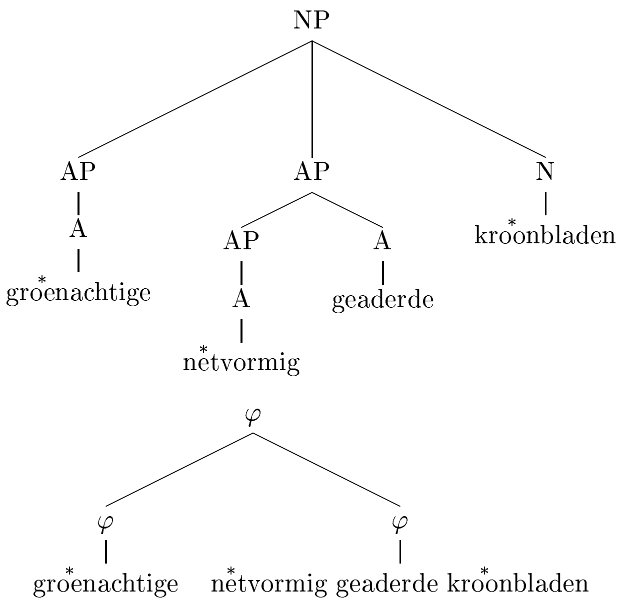


Figure 4.2: Syntactic structure and derived φ-structure for the phrase greenish clathrate veined petals.

With the current results, we can take this line of reasoning one step further, and argue that even a *recursive* prosodic structure is not informative enough for the purposes of intonational phrasing. In support of this position, consider the examples in Figure 4.2 and Figure 4.3: two different syntactic structures, which are nevertheless mapped to identical prosodic structures, i.e. with the same φ-structure and the same number of accents in each φ.[17] The former, however, requires an obligatory boundary between the two lower φ's according to the reference corpus, whereas the latter does not. Clearly, there is no way to explain this difference in intonational phrasing in terms of the prosodic structure and accent distribution only.

If we accept that our current prosodic structure is insufficient for the purpose of intonational phrasing, there are basically two ways to proceed. First, we can try to further extend our prosodic structure in such way that it incorporates the required information. This amounts to modifying the mapping rules so that even more semantic and syntactic details are preserved in the proso-

[17]The first example is derived from the following phrase in the reference corpus: ... # *en v$\overset{*}{i}$jf gr$\overset{*}{o}$enachtig witte # n$\overset{*}{e}$tvormig geaderde kr$\overset{*}{o}$onbladen*; lit. 'and five greenish white clathrate veined petals'. The second phrase does not occur literally in the corpus. However, of all the phrases in the corpus with an identical syntactic and prosodic structure, not one has an obligatory boundary between the two lower φ's.

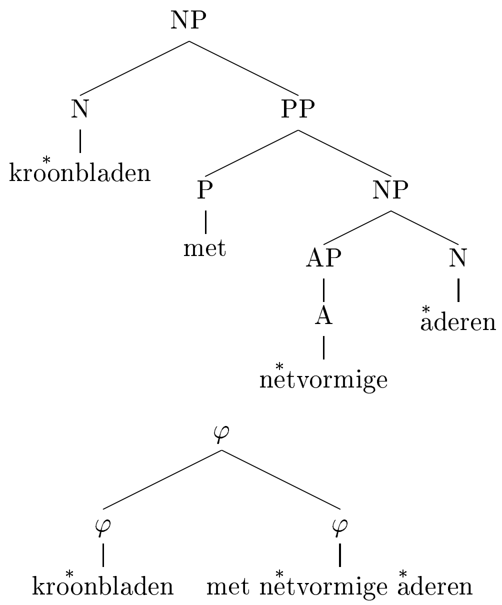


Figure 4.3: Syntactic structure and derived φ-structure for the phrase petals with clathrate veins.

dic structure. However, this raises a number of problems. To begin with, we would have to find a way to extend the prosodic structure, such as adding additional prosodic categories or allowing prosodic constituents to carry features. Moreover, these elaborations would require support by independent evidence, that is, we have to show that there are phonological rules that are sensitive to these extensions. Apart from these questions, this solution seems to defeat the whole purpose of the prosodic structure as an abstraction capturing only those semantic and syntactic aspects that are phonologically relevant. The prosodic structure would resemble the syntactic structure to such a degree that it would be hard to justify its existence. In conclusion, extending the prosodic structure is not an attractive solution.

The alternative way to proceed is to abandon the idea that intonational phrases are derived from the prosodic structure. Instead, we assume that they are derived directly from the syntactic structure. The rules for their derivation and restructuring have access to all the syntactic details that were shown to be relevant in the previous sections. At the same time, we can keep the mapping rules and the prosodic structure simple, in accordance with the Strict Layer Hypothesis as defended by Nespor & Vogel. As a drawback, we will probably lose the theoretical elegance of Gussenhoven's algorithm. However, arguments of theoretical economy have to give way to coverage of the empirical facts.

This second approach is clearly preferable to the first option outlined above. After all, since intonational phrasing – as one of the aspects of intonation – is inherently more closely related to *meaning* than purely *formal* phonological rules, it is a plausible assumption that it depends on input that contains a fair amount of meaning, i.e. fine-grained syntactic information.

So what we end up with is a combination of the best properties of the three algorithms. We agree with Nespor & Vogel and Caspers that intonational phrases are determined on the basis of the semantic and syntactic structure. However, the phrasing algorithm as proposed by Nespor & Vogel is in need of substantial modifications in order to increase its coverage and accuracy. Moreover, we preserve the two key insights of Gussenhoven, which may very well turn out to make a significant contribution to increasing the precision.

First, we reject the idea that a stretch of speech that corresponds to a single coherent intonation contour can be defined as a prosodic constituent with a fixed rank in the prosodic hierarchy (Section 2.3.7). Instead, we assume that it must be analysed as a separate entity of a different order. Following Gussenhoven, we may call it an *association domain*. Although the name of π, i.e. the prosodic category called *intonational* phrase, has thereby become somewhat misleading, we nevertheless preserve it as a prosodic category, because it is still relevant as the domain of application for certain phonological rules and for the overall temporal structure of the utterance (Gussenhoven and Rietveld 1992a). However, when there is no risk of confusion, we will use the terms *intonational phrase* and *association domain* indiscriminately. The same goes for *intonational phrasing* and *derivation of AD's*.

Second, we agree with Gussenhoven that the accent distribution plays a crucial role in intonational phrasing. In this way, phrases without accents can be avoided, and the number of accents can affect restructuring.

Finally, it is likely that intonational phrasing depends to some extent on speaking rate (Caspers 1994) and the length or weight of an intonational phrases (Marsi, Coppen, Gussenhoven, and Rietveld 1997) as well.

In conclusion, the results point in the direction of a syntax-driven algorithm for intonational phrasing, which is sensitive to a number of particular syntactic configurations, the accent distribution, and perhaps the speaking rate and weight of intonational phrases. An algorithm adhering to these principles will be presented in the next Section.

4.8 A new approach to intonational phrasing

4.8.1 $\mathcal{NEW}$ algorithm

This section proposes a new intonational phrasing algorithm called $\mathcal{NEW}$. It takes a syntactic tree and an accent distribution as input, and inserts intonational boundaries in the tree. It consists of eight rules for inserting intonational boundaries. Each rule describes a certain syntactic pattern plus conditions and the place to insert the boundary. There are dedicated rules for inserting a boundary in an apposition, a sentence, a verb phrase, a noun phrase, and a coordination (coordinated structure). An overview of all rules is given in Table 4.14. The rules apply in the order in which they are presented.[18] Each rule is applied obligatory and exhaustively before the next one is tried.

For all rules described below, it is assumed that intonational boundaries within coordinations are handled exclusively by the rule for this type of structures. This means, for instance, that the rules for phrasing in NP's do not match to a coordinated NP structure, as this would lead to unintended results. In addition, it is assumed that both the start and the end of an utterance constitute a trivial AD-boundary.

The rules are described and motivated with the help of examples. They are based on exemplary data, a considerable part of which is inspired by, or directly stems from the corpus of flower descriptions used in the above evaluation. The algorithm can thus to some extent be considered the result of manual modelling on the corpus of flower descriptions. It is therefore to be expected that a similar evaluation of $\mathcal{NEW}$ as described in Section 4.8.2, showed that it performs significantly better than two of the other algorithms. We will return to the question to what extent the algorithm is domain-independent Section 4.8.2.2.

4.8.1.1 Appositions

The class of appositional elements includes non-restrictive relative clauses, parenthetical expressions, tag questions, vocatives and expletives. There is no consensus about how appositions should be syntactically analysed. We will assume that they can take any syntactic category and can be attached anywhere in the syntactic tree, as long as they can be identified by the feature `appos`. With respect to the intonational phrasing of these elements, linguistic intuition appears to be unambiguous: appositional elements must be enclosed by intonational boundaries. This is corroborated by the results in Section 4.4.2, which clearly demonstrated that appositions were indeed always accompanied by intonational boundaries in the reference phrasing. We may explain this observation by the fact that appositional elements are typically signalled by lowering of the register, which by definition constitutes an intonational boundary.[19] Therefore, intonational phrasing for appositions is relatively simple, and is formalised in the following rule.

(34) **AD-to-Apposition**

For every apposition:
insert an intonational boundary before and after it.

[18] Only a few rules crucially dependent on their mutual ordering; these dependencies will be mentioned below. The rest of them is order independent in the sense that changing the order of the rules in the grammar will not affect the result.

[19] Although register lowering is not obligatory. For instance, tag questions can very well go without it.

Table 4.14: Rules of the $\mathcal{NEW}$ algorithm for AD-derivation.

Name:	*Definition:*
AD-to-Apposition	For every apposition: insert an intonational boundary before and after it.
AD-in-S-1	For every topicalised constituent: insert an intonational boundary after it, if (a) it has at least three accents, and (b) the subsequent AD has at least two accents.
AD-in-S-2	For every extraposed constituent: insert a boundary before it, if (a) it has at least two accents, and (b) the preceding AD has at least two accents.
AD-in-VP	For every daughter of a VP: insert a boundary after it, if (a) it has at least two accents, and (b) the subsequent AD has at least two accents.
AD-in-NP-1	For every premodifier in an NP: insert a AD-boundary after it, if (a) it carries at least one accent, and (b) its end coincides with a φ-boundary, and (c) the subsequent AD has at least one accent.
AD-in-NP-2	For every NP: insert an AD-boundary between its head and a postmodifier, if (a) the postmodifier has at least three accents, and (b) the preceding AD has at least one accent.
AD-in-NP-3	For every NP: insert an intonational phrase boundary between its postmodifiers, if (a) the preceding AD has at least one accent, and (b) the subsequent AD has at least one accent.
AD-in-coordination	For every pair of conjuncts in a complex coordination: insert an intonational phrase boundary between them, if (a) the preceding AD has at least one accent, and (b) the subsequent AD has at least one accent.

4.8.1.2 Sentences

At the sentence level, there are two structural positions at which an intonational phrase boundary may be inserted. The first is *after* a topicalised constituent, or – put differently – before the finite verb in a declarative main clause. The second is *before* an extraposed constituent, which is after the VP for the first extraposed constituent, and between two extraposed constituents for all others.

Boundary insertion at these positions appears to be primarily determined by the *length* or *weight* of the AD's that are created. For example, the boundary after the topicalised constituent *de mannelijke en vrouwelijke bloemen* in (35-a) is perfectly acceptable. However, if the topicalised constituent is shorter and has only one accent (35-b), or the remaining part of the sentence is shorter (35-c), inserting a boundary is far less appropriate, as this would results in relatively short AD's.

(35) a.

de	m*annelijke	en	vr*ouwelijke	bl*oemen	#	verschijnen
the	male	and	female	flowers		appear

d*oorgaans	aan	de	st*engeltop
normally	at	the	stalk-top

b.

?de	bl*oemen	#	verschijnen	d*oorgaans	aan	de	st*engeltop
the	flowers		appear	normally	at	the	stalk-top

c.

?de	m*annelijke	en	vr*ouwelijke	bl*oemen	#	zijn	hetz*elfde
the	male	and	female	flowers		are	identical

It would appear, therefore, that both the weight of the AD *preceding* the boundary and the weight of the AD *following* the boundary have to be taken into account. The same type of *balancing* will return in most of the other rules described below. The main difficulty is to formalise the intuitive notion of an AD's *weight* by expressing it in terms of some unit of measure. We found that counting the number of pitch accents in a AD provides a reasonable approximation, and is relatively easy to implement.

Using this measure, the following rule inserts an AD boundary after a topicalised constituent.

(36) **AD-in-S-1**

For every topicalised constituent:
insert an intonational boundary after it, if

a. it has at least three accents, and
b. the subsequent AD has at least two accents.

With the *subsequent AD*, we mean the AD following the boundary, which comes into existence as a result of inserting this boundary. In (36-a), for example, the subsequent AD is *verschijnen doorgaans aan de stengeltop*. The condition that the topicalised constituent must have at least three accents is not derived from some theoretical principle; rather, it is a tendency observed in our corpus of flower descriptions. That is, choosing this value gives the best result when the algorithm is applied to the corpus. The same goes for the condition that the subsequent AD must have two or more accents.

Extraposed constituents show a behaviour that mirrors that of topicalised constituents in the sense that the decision to insert a boundary is affected by the weight of the extraposed constituent, as well as that of the constituent preceding it. In (37-a), for instance, both the VP and the extraposed PP contain two accents, and are therefore separated by a boundary. In contrast, the PP in (37-b) has only one accent, which prevents insertion of a boundary. The same goes for the VP in (37-c).

(37) a. hij komt in het *algemeen v*oor # in d*uinen en z*anderige w*egbermen
he occurs in the general `prt` in dunes and sandy roadsides
'it normally occurs in dunes and sandy roadsides'

b. ?hij komt in het *algemeen v*oor # in d*uinen
he occurs in the general `prt` in dunes

c. ?hij is gev*onden # in d*uinen en z*anderige w*egbermen
he is found in dunes and sandy roadsides

On the basis of these observations, we propose the following rule.

(38) **AD-in-S-2**

For every extraposed constituent:
insert a boundary before it, if

a. it has at least two accents, and
b. the preceding AD has at least two accents.

Analogously to our use of the term *subsequent AD*, the *preceding AD* means the AD that is created to the left of the inserted boundary. In (37-a), the preceding AD is *hij komt in het algemeen voor*.

The rule for extraposed constituents is very similar to that for topicalised constituents. The difference is that the former requires the constituent to have at least three accents, whereas the latter requires at least two accents. The difference again stems from comparing the relevant instances in the corpus of flower descriptions.

In the implementation, the rules AD-S-1 and AD-S-2 are ordered. As a consequence, the former may destroy the context for the latter. In (39), for example, AD-in-S-1 inserts a boundary after the topicalised constituent, and therefore, AD-in-S-2 cannot insert a boundary before the extraposed constituent, as the preceding AD of AD-in-S-2 lacks the minimal number of accents.

(39) in N*ederland, D*uitsland, en Fr*ankrijk # komt hij vo*or in
in Netherlands Germany and France occurs he `prt` in

d*uinen en z*anderige w*egbermen
dunes and sandy roadsides

However, switching around the rules gives acceptable results as well. With respect to (39), this means that there would only be a boundary before the extraposed constituent. As matter of fact, both boundaries occur when the number of accents in the intermediate part is increased (40).

(40) in N*ederland, D*uitsland, en Fr*ankrijk # komt hij *algemeen
in Netherlands Germany and France occurs he generally

vo*or # in d*uinen en z*anderige w*egbermen
`prt` in dunes and sandy roadsides

It is not uncommon to have more than one extraposed constituent in a sentence. AD-in-S-2 will insert a boundary between them provided that both carry at least two accents. This is true for virtually every extraposed constituent, since they are S's, or NP's or PP's that are 'heavy', which appears to be highly correlated with having multiple accents. The idea that the syntactic structure is affected by phonological considerations is also found in Zec and Inkelas (1990).

(41) hij komt *algemeen vo*or # in N*ederland, D*uitsland, en
he occurs generally `prt` in Netherlands Germany and

Fr*ankrijk # in d*uinen en z*anderige w*egbermen
France in dunes and sandy roadsides

Another interesting point is that the rules do not allow for an AD boundary after a fronted finite verb, i.e. a verb taking the second position in a declarative main clause or the first position in a polar question. Thus, an example as (42) is ruled out.

(42) *de m*annelijke en vro*uwelijke blo*emen verschijnen #
the male and female flowers appear

do*orgaans aan de st*engeltop
normally at the stalk-top

When a boundary does follow the verb, this is in a sense a coincidence, since the boundary was actually inserted before an extraposed constituent. The boundary in (43), for instance, is inserted before the extraposed S.

(43) ($_S$ ($_{PP}$ uit rec*ente t*ellingen) ($_V$ blijkt) ($_{VP}$) # ($_S$ dat de
from recent countings shows that the
plant in *aantal ter*ugloopt))
plant in number decreases

4.8.1.3 Verb Phrases

There are two general observations to make about intonational phrasing at the VP level. First, it is very unusual for an AD-boundary to occur before the verb cluster (the cluster at the end of the VP consisting of a main verb plus any verbal particles and auxiliaries). Second, a boundary may occur between phrasal daughters of the VP, depending on their weight. Again, the number of accents in a constituent appears to be a reasonably good indicator of its weight. Both observations are captured in the following rule.

(44) **AD-in-VP**

For every daughter of a VP:
insert a boundary after it, if

a. it has at least two accents, and
b. the subsequent AD has at least two accents.

Examples of boundaries due to AD-in-VP are given in (45). In (45-a), a boundary is inserted between the AdvP, which has two accents, and the PP, which has two accents as well. Example (45-b) is similar, except that the subsequent AD contains two daughters of the VP, each with only one accent. Example (45-c) shows that there is no boundary following the second PP *op veel plaatsen*, even though it contains two accents, because otherwise the subsequent AD would have just one accent. Finally, (45-d) has two boundaries due to AD-in-VP.

(45) a. die verandert ($_{VP}$ ($_{AdvP}$ tw*ee maanden l*ater) # ($_{PP}$ in
which changes two months later in
een plu*izige b*ol))
a fluffy ball

b. de plant komt ($_{VP}$ ($_{PP}$ in N*ederland en B*elgie) #
the plant occurs in Netherlands and Belgium
($_{AdvP}$ z*elden) v*oor)
rarely `prt`

c. de plant komt ($_{VP}$ ($_{PP}$ in N*ederland en B*elgie) #
the plant occurs in Netherlands and Belgium
($_{PP}$ op v*eel pl*aatsen) v*oor)
at many places `prt`

d. de plant komt ($_{VP}$ ($_{PP}$ in N*ederland en B*elgie) #
the plant occurs in Netherlands and Belgium
($_{PP}$ op z*onnige pla*atsen) # ($_{AdvP}$ regelm*atig) vo*or)
at sunny places regularly prt

All these examples comply with our first observation above that a boundary before the verb cluster is exceptional. This automatically follows from the second part of AD-in-VP. First, a verb cluster is usually followed by an AD-boundary, either because it is sentence-final, or because it is followed by an extraposed constituent. Second, the verb cluster will virtually always have at most one accented word, because auxiliaries are rarely accented. Taken together, this means that the condition in AD-in-VP that the subsequent AD must have at least two accents cannot be fulfilled. At the same time, this predicts that there will be two exceptions. The first occurs when the verb cluster does contain two accents, a type of accentuation that has strong pragmatic implications; see (46).

(46) dat een gro*otschalige versch*uiving # zo*u kunnen pla*atsvinden
that a large-scale shift may could take-place
is nog ge*en bewi*js
is yet no proof
'that a large-scale shift *might* occur is no proof yet'

The second exception occurs when the verb cluster is not closed by an AD-boundary, and the AD it is part of continues into an extraposed constituent. An example of this is (47-a). In contrast, insertion of a boundary is blocked in (47-b), because an intervening AD-boundary is present before the extraposed PP *als kleurstof*; a boundary which is due to the rule AD-in-S-2.

(47) a. de plant wordt in het z*uiden van Euro*pa # gebru*ikt als
the plant is in the south of Europe used as
kle*urstof
colouring

b. de plant wordt in het z*uiden van Euro*pa gebru*ikt # als
the plant is in the south of Europe used as
kle*urstof in vo*edsel
colouring in food

To sum up so far, we have argued that intonational phrasing at the level of S and VP can be accounted for by just three rules, and that their ordering neatly accounts for the frequently observed balancing of AD's according to their weight.

4.8.1.4 Noun Phrases

There are two positions at which intonational boundaries in an NP may be observed: after a premodifier, i.e. after a daughter preceding the head, and before a postmodifier, i.e. before a daughter following the head.

With respect to the premodifiers, it appears to be the case that only AP's and NumP's are capable of triggering an intonational break, whereas other premodifiers, e.g. determiners and quantifiers, lack this ability (48-a). Moreover, a boundary can only occur between two subsequent premodifiers; never between the last premodifier and the head of the NP (48-b). By implication, a single premodifier will never trigger a boundary.

(48) a.

?dit	geldt	voor	($_{NP}$	($_{Q}$	$\overset{*}{\text{a}}$lle	)	#	bek$\overset{*}{\text{e}}$nde	leden	van de
this	goes	for			all			known	members	of the

N$\overset{*}{\text{a}}$chtschadefamilie)
Nightshade-family

b.

*dit	geldt	voor	($_{NP}$	$\overset{*}{\text{a}}$lle	($_{AP}$	r$\overset{*}{\text{e}}$chtstreekse	)	#	leden
this	goes	for		all		known			members

van	de	N$\overset{*}{\text{a}}$chtschadefamilie)
of	the	Nightshade-family

Both restrictions can be captured by the condition that the intonational phrase boundary must coincide with a phonological phrase boundary. Since there will never be a φ-boundary between the last premodifier and the head, there will never be an intonational boundary either. In the same way, determiners and quantifiers, which are function words, will never bring about a φ-boundary, and therefore never an AD-boundary either.

There is one complication: inserting a boundary after a premodifier may create an empty AD. Therefore, both the premodifier and the subsequent AD must have at least one accent. Taking all conditions together, we arrive at the following formulation.

(49) **AD-in-NP-1**

For every premodifier in an NP:
insert a AD-boundary after it, if

a. it carries at least one accent, and
b. its end coincides with a φ-boundary, and
c. the subsequent AD has at least one accent.

Postmodifiers – PP's and restrictive relative S's – may be preceded by an intonational boundary if sufficiently heavy. By inspecting the corpus of flower descriptions, we found that if the head of the NP is followed by a postmodifier that has at least three accents, head and modifier are separated by an AD-boundary.[20] As with the previous rule, creating an empty AD must be avoided.

[20] Of course, we found exceptions to this generalisation, but this value gives the highest

This can be accomplished by checking that the preceding AD has at least one accent.

(50) **AD-in-NP-2**

For every NP:
insert an AD-boundary between its head and a postmodifier, if

a. the postmodifier has at least three accents, and
b. the preceding AD has at least two accent.

Two examples of boundaries due to (50) are given in (51).

(51) a. de plant vormt $(_{NP}$ flinke $(_{N}$ kolonies) # $(_{PP}$ in
the plant forms substantial colonies in

bossen weilanden en duinen))
forest meadows and dunes

b. $(_{NP}$ een eenjarige $(_{N}$ plant) # $(_{S}$ die hooguit
a annual plant which maximally

twintig centimeter hoog wordt))
twenty centimetre high becomes

So far, only the *first* postmodifier has been addressed. However, an NP can have more than one postmodifier; see (52). The fact that the postmodifying PP's can be switched around without altering the meaning of the full NP suggests that we have indeed two independent constituents, and that it is not the case that one of them is embedded in the other.

(52) $(_{NP}$ een $(_{N}$ rozet) $(_{PP}$ van elliptische bladen) $(_{PP}$ met een
a rosette of elliptical leaves with a

tot twintig centimeter hoge stevige bloeistengel))
up-to twenty centimetre high firm flower-stalk

With regard to intonational phrasing, the best solution appears to be to always insert an intonational boundary between subsequent postmodifiers. Again, care must be taken not to create AD's without accents.

(53) **AD-in-NP-3**

For every NP:
insert an intonational phrase boundary between its postmodifiers, if

a. the preceding AD has at least one accent, and
b. the subsequent AD has at least one accent.

number of correctly predicted boundaries when applying the algorithm to the corpus.

These three rules suffice to account for intonational phrasing in NP's. Notice that – as with the rules for phrasing in S's and VP's – they are concerned with intonational boundaries between *direct* daughters of the NP only. These daughters – AP's, NP's, PP's, S's – may contain intonational boundaries themselves, which are due to other phrasing rules. This includes a recursive application of the AD-in-NP rules, as in the case of PP-embedding.

4.8.1.5 Coordinations

There appear to be a number of ways to phrase a coordination. The first is to map the whole coordination onto a single AD and to apply the tonal operation *Downstep* (Section 2.3.6), which results in one type of 'list intonation' (54-a). We may hypothesise that – at least in the case of coordinations – downstep facilitates parsing of the syntactic structure in the sense that each downstepped accent serves to signal a separate conjunct. This would explain the observation that downstep is less likely to be used for a coordination in which one of the prefinal conjuncts has more than one pitch accent (54-b). For some reason, the last conjunct of a coordination forms an exception to the rule (here *het noorden van Luxemburg)*, as it may contain multiple accents even when downstep is applied (54-c).

(54) a. op na*tte gro*nd ! # in ($_{NP}$ ($_{NP}$ Ne*derland) ($_{NP}$
at wet ground in the-Netherlands
Be*lgie) en ($_{NP}$ Lu*xemburg))
Belgium and Luxembourg

b. ?op na*tte gro*nd ! # in ($_{NP}$ ($_{NP}$ het zu*iden van
at wet ground in the south of
Ne*derland) ($_{NP}$ hee*l Be*lgie) en ($_{NP}$ het noo*rden
the-Netherlands whole Belgium and the north
van Lu*xemburg))
of Luxembourg

c. op na*tte gro*nd ! # in ($_{NP}$ ($_{NP}$ Ne*derland) ($_{NP}$
at wet ground in the-Netherlands
Be*lgie) en ($_{NP}$ het noo*rden van Lu*xemburg))
Belgium and the north of Luxembourg

We will refer to a coordination in which none of the prefinal conjuncts carries more than one pitch accent as a *simple coordination.* No phrasing rule is required for simple coordinations; they remain free of internal AD-boundaries. Insertion of the Downstep operator before a simple coordination will be discussed in Section 7.4. The outer boundaries at the start and at the end of a coordination are taken care of by the other phrasing rules, depending on the context. For instance, the boundary before the coordination in (54) is due to

AD-in-NP-2.[21] In contrast, a coordination occurring directly before the verb cluster will not be closed by an AD-boundary.

The second approach to phrasing a coordination is to map each conjunct onto an AD of its own. This is appropriate when at least one of the prefinal conjunct has more than one accent. We will refer to these as *complex coordinations*. For this purpose, we introduce the following rule.

(55) **AD-in-coordination**

For every pair of conjuncts in a complex coordination:
insert an intonational phrase boundary between them, if

a. the preceding AD has at least one accent, and
b. the subsequent AD has at least one accent.

In theory, it is possible that one of the conjuncts is unaccented. Therefore, the rule checks that the created preceding and subsequent AD's are not empty. In practice, however, it is hard to find examples of this type. No examples of this type occurred in the corpus of flower descriptions. As for a simple coordination, the boundaries surrounding a complex coordination are due to independent phrasing rules.

Interestingly, the weight of a constituent, which is assumed to affect phrasing in most of the other syntactic contexts discussed above, does not play a part in the phrasing of coordinations. Instead, the decisive factor appears to be whether at least one of the prefinal conjuncts contains more than one accent.

With a view to the treatment of pitch accent placement in numerals in Section 7.3.5, we will finish this section with showing that AD-in-coordination is also responsible for intonational phrasing in complex numerals. In Section 6.2.3.4, we will argue that complex numerals have to be syntactically analysed as coordinated structures, even though the conjunction *en* 'and' can usually be omitted. An example of this is (56), of which the syntactic structure is given in Figure 4.4.

(56) honderd vier en vijftig duizend zeven honderd twee en dertig
hundred four and fifty thousand seven hundred two and thirty
kasten
cupboards
'154,732 cupboards'

There are two alternative accentuations of (56) which appear to be equally acceptable: with an accent on *duizend* and a following AD-boundary (57-a), or without either an accent or a following AD-boundary (57-b).

[21] Strictly speaking, the boundary is not inserted before the coordinated NP structure itself, but rather before the PP that dominates it. These are details of the implementation that will be ignored here.

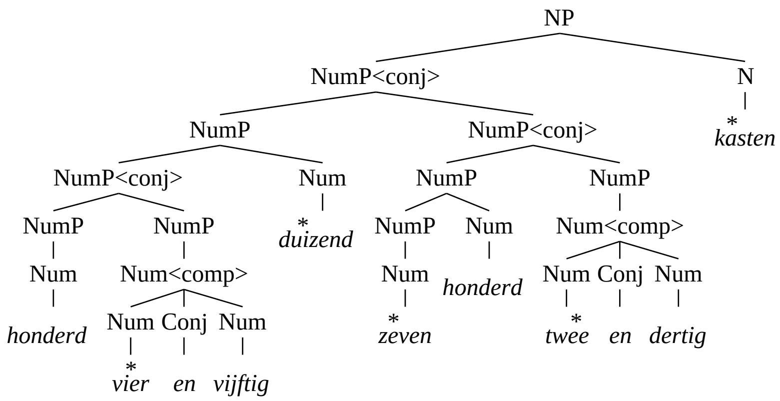


Figure 4.4: Syntactic structure of example (56).

(57) a. [AD honderd *vieren vijftig *duizend]
[AD *zeven honderd *twee en dertig *kasten]

b. [AD honderd *vieren vijftig duizend *zeven honderd *twee en dertig *kasten]

It will be clear that no AD boundaries are predicted in the lower coordinations, because the number of accents in their penultimate conjuncts is insufficient. However, an accent on *duizend* provides the penultimate conjunct of the highest coordination with the required number of accents to trigger AD-in-coordination. Thus, the rule AD-in-coordination, in combination with the syntactic analysis of complex numerals as involving coordination, correctly predicts dependency relations between accent and boundary as observed in (57). We will return to this issue in Section 7.3.5

4.8.2 $\mathcal{NEW}$ results

4.8.2.1 General

Earlier this chapter, the intonational phrasing algorithms $\mathcal{N}\&\mathcal{V}$, $\mathcal{CAS}$, and $\mathcal{GUS}$ were evaluated by applying them to a corpus of flower descriptions and comparing their output to a reference phrasing produced by human experts. Applying the same evaluation procedure to $\mathcal{NEW}$, we obtain the result summarised in Table 4.15, which shows the difference between the prediction and the reference in the form of a confusion matrix. The kappa value calculated for the matrix (without the optional boundaries) indicates that there is a high agreement between the prediction and the reference that is significant at the 1%-level

Table 4.15: Confusion matrices for $\mathcal{NEW}$ algorithm. The top matrix contains the absolute number of confusions. The bottom matrix contains the percentage of confusions relative to the total number of boundaries in a column.

Reference

$\mathcal{NEW}$		*Obl*	*Opt*	*Non*	*Tot*
	Obl	98	10	9	117
	Opt	0	0	0	0
	Non	11	32	695	738
	Tot	109	42	704	855

Reference

$\mathcal{NEW}$		*Obl*	*Opt*	*Non*
	Obl	89.91	23.81	1.28
	Opt	0	0	0
	Non	10.09	76.19	98.72
	Tot	100	100	100

($\kappa = 0.923$, $p = 0.000$). The second row contains only zeros, because $\mathcal{NEW}$ – in contrast to the other algorithms – does not assign optional boundaries. The second column can be ignored, because at locations where an optional boundary occurs in the reference, both a boundary and no boundary are acceptable, that is, the algorithm cannot be wrong. $\mathcal{NEW}$ assigns 9 erroneous boundaries and omits 11 boundaries. Another way to look at this is that of all 855 boundary decisions, only 20 (2.34%) are wrong. Considering the length and complexity of the sentences in the corpus, this is a highly accurate result.

It is interesting to compare these results with those obtained with the previously tested algorithms. Of these, $\mathcal{GUS}$ performed best; see Table 4.9. However, a comparison is not straightforward, because $\mathcal{GUS}$ assigns optional boundaries, whereas $\mathcal{NEW}$ does not. Therefore, the optional boundaries have to be transformed to either obligatory boundaries or non-boundaries. There are three options:

1. Promote all optional boundaries to obligatory boundaries, i.e. add the second row to the first in Table 4.9.

2. Reduce all optional boundaries to non-boundaries, i.e. add the second row to the third one in Table 4.9.

3. Divide all optional boundaries equally over the obligatory boundaries and the non-boundaries, i.e. add half of the second row to the first one, and add the other half to the third one in Table 4.9. This amounts to a random choice for each optional boundary to either realize or omit it.

The results for each of these three options are given in Table 4.16. The differences among the three options with respect to the total number of errors turn out to be negligible. The general picture that emerges is that on the one hand $\mathcal{NEW}$ misses a few more boundaries than $\mathcal{GUS}$, but on the other hand assigns considerably fewer incorrect boundaries. Still, the total number of errors for $\mathcal{NEW}$ is 20, whereas that for $\mathcal{GUS}$ is at least 35.

A comparison of $\mathcal{NEW}$ with the other two algorithms ($\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$) is less interesting, because the differences are considerably bigger, regardless of the option we choose for counting optional boundaries. For instance, the total number of errors for $\mathcal{N}\&\mathcal{V}$ is minimally 70 (8.19%).

This interpretation of the differences between the algorithms was confirmed by a statistical analysis. For every algorithm, we calculated its kappa value (discounting optional boundaries) for each of the ten texts. An analysis of variance (repeated measurements) carried out on these kappa values showed that the factor algorithm was significant at $F_{3,27} = 24.140$, $p < 0.01$ (Huynh-Feldt corrected). A subsequent post-hoc comparison procedure yielded two homogeneous subsets: $\{\mathcal{N}\&\mathcal{V}, \mathcal{CAS}\}$, and $\{\mathcal{GUS}, \mathcal{NEW}\}$ ($p < 0.05$).

4.8.2.2 Discussion

$\mathcal{NEW}$ is based on exemplary data, a considerable part of which is inspired by, or directly stems from the corpus of flower descriptions used in the above evaluation. The algorithm is in part the result of repeating the cycle of testing and improving until the point that – in our opinion – no further improvement was feasible. Although it is possible to improve the results for $\mathcal{NEW}$ with additional phrasing rules, these rules would then account for just one or two

Table 4.16: $\mathcal{NEW}$ compared to $\mathcal{GUS}$ with respect to undershoot (#/∅) and overshoot (∅/#) errors for obligatory boundaries. Percentages are calculated with respect to the total number of boundary decisions (855).

Algorithm:	*Undershoot:*		*Overshoot:*		*Total:*	
$\mathcal{NEW}$	11	(1.29%)	9	(1.05%)	20	(2.34%)
$\mathcal{GUS}_1$	2	(0.23%)	33	(3.86%)	35	(4.09%)
$\mathcal{GUS}_2$	15	(1.75%)	21	(2.46%)	36	(4.21%)
$\mathcal{GUS}_3$	8.5	(0.99%)	27	(3.16%)	35.5	(4.15%)

extra correct boundaries. Clearly, this sort of evidence is insufficient to justify such extensions. $\mathcal{NEW}$ is thus the result of balancing economy of description on the one hand and accurate prediction on the other.

Consequently, we cannot present an analysis of systematic errors produced by $\mathcal{NEW}$. After all, if we were able to find patterns or classes of errors, we would seize the opportunity to improve the algorithm. Instead, the errors are all rather individual cases that might be attributed to an inadequate measure of AD weight, the possibility of alternative syntactic interpretations, special pragmatic contexts, explicit attempts to avoid ambiguity, or an incidental case of inconsistent judgements of the panel that produced the reference phrasing. Whatever our speculations may be, these are cases for further research with the help of more data (see Appendix B).

Since the algorithm is to some extent the result of manual modelling on the corpus of flower descriptions, it is not surprising that evaluation of $\mathcal{NEW}$ against this corpus showed that it performs better than at least two of the other algorithms. The question is to what extent a corpus of ten short texts, all of the same genre, is representative of written texts in general. In other words: does $\mathcal{NEW}$ perform equally well when applied to other types of written text? Ultimately, this question can only be answered by repeatedly evaluating the algorithm against many different corpora, something which is beyond the scope of the current work. Large speech corpora becoming available nowadays are probably not going to be of any help with respect to generating the most acceptable intonation. To begin with, they are usually not prosodically annotated. In addition, they are typically designed for wide coverage, containing speech from competent ('good') as well as less competent ('bad') speakers. Apart from the problem that the necessary semantic and syntactic information are probably not available, extracting regularities from the corpus will inevitably lead to a model of the intonational behaviour of an average speaker, and the generation of *largely acceptable* rather than *good* intonation.

Although the evaluation of $\mathcal{NEW}$ has not been repeated with another corpus, the algorithm has been incorporated in our SLG system (to be described in the next chapters). In addition to flower descriptions (WFD), this system generates travel descriptions (PTD). Results on an evaluation of the generated travel descriptions will be reported in Chapter 8. Part of this concerns the intonational boundaries as predicted by $\mathcal{NEW}$.

4.9 Summary

The goal of this chapter was to obtain an accurate intonational phrasing algorithm, i.e. one that can account for the intonational boundaries that are most acceptable according to intonational experts. In addition, we intended to find out which types of information are required as input. Although a considerable amount of linguistic theory regarding intonational phrasing is available in the literature, most work is based on a small set of crucial observations and has neither been tested against larger sets of data nor systematically compared with competing approaches. In the first part of this chapter, therefore, the algorithms by Nespor & Vogel ($\mathcal{N}\&\mathcal{V}$), Capers ($\mathcal{CAS}$), and Gussenhoven ($\mathcal{GUS}$) were systematically compared using a substantial set of data. We used a corpus-based method in which the intonational boundaries predicted by the algorithm were compared with those prescribed by a panel of experts, where the text material consisted of flower descriptions. A detailed error analysis revealed systematic errors in each of the algorithms. It was concluded that a better algorithm would have to be syntax-driven, and sensitive to a number of particular syntactic configurations, the accent distribution, and the weight of intonational phrases. Accordingly, a new algorithm ($\mathcal{NEW}$) was proposed consisting of eight rules for inserting intonational boundaries in particular syntactic configurations. It correctly predicted 97.66% of the boundary decisions in the corpus of flower descriptions, which was shown to be significantly better than the predictions by $\mathcal{N}\&\mathcal{V}$ and $\mathcal{CAS}$. An additional evaluation of the $\mathcal{NEW}$ algorithm, as incorporated in our SLG system, will be described in Chapter 8.

Chapter 5

Spoken Language Generation

5.1 Introduction

Spoken language generation (SLG) concerns computer programs that transform non-linguistic input into intelligible speech, for a certain communicative goal, and using knowledge of the context and the language. Our research goal is to design, build, and test a linguistically-based SLG system that generates speech with pitch accents and intonational boundaries that most acceptable to experts (cf. Section 1.3). This objective gives rise to two types of research questions: those concerning the linguistic theory on which the system is based (how adequate is it, and how can it be improved?), and those concerning the architecture of an SLG system. The latter are the topic of the current chapter.

In Section 5.2, we discuss the general architecture of an SLG system. Since SLG is a combination of natural language generation (NLG) and speech synthesis, we start with summarising the architectures of conventional NLG systems (Section 5.2.1) and speech synthesisers (Section 5.2.2). Next, we address the questions of how to combine NLG and speech synthesis systems, and what additional information (besides text) must be generated by the NLG system to facilitate the control of intonation (Section 5.2.3). We continue with discussing the best place in the system for controlling the intonation (Section 5.2.4). Finally, we propose an architecture consisting of a domain-specific text planner and a domain-independent modular content-to-speech system called CONPAS, which forms the basis for our SLG systems.

In the next four sections, we describe the modules of our SLG system for text planning (Section 5.3), syntactic realization (Section 5.4), prosodic realization (Section 5.5), and speech synthesis (Section 5.6), and the intermediary representations that serve as the interface between them. The chapter ends with a summary (Section 5.7).

5.2 The architecture of an SLG system

5.2.1 Natural language generation

The first — and only so far — text book on natural language generation gives the following definition.

> "Natural Language Generation (NLG) is the subfield of artificial intelligence and computational linguistics that focuses on computer systems that can produce understandable texts in English or other human languages. Typically starting from some non-linguistic representation of information as input, NLG systems use knowledge about language and the application domain to automatically produce documents, reports, explanations, help messages, and other kinds of texts." (Reiter and Dale 2000, p. 1)

Traditionally, the process of language generation has been divided into two tasks: determining *what* to say (content), and determining *how* to say it (form) (Bateman and Hovy 1991; De Smedt, Horacek, and Zock 1996). The former became known as *strategic generation*, *deep generation*, *conceptualisation*, or *text planning*; the latter as *tactical generation*, *surface generation*, *formulation*, or *linguistic realization*. More recently, however, the most popular architecture for NLG systems involves three stages. We will first summarise this architecture, and later on investigate how it can be adapted and extended for current purposes.

Figure 5.1 shows the standard architecture of an NLG system following Reiter and Dale (1997, 2000), referred to as the 'consensus architecture' in Reiter (1994). Although it is generally acknowledged that it has certain drawbacks, it appears to be the best architecture available for a practical NLG system. The input is assumed to be a combination of four types of information:

1. **Knowledge source**
   The knowledge source is some structured form of information about the domain from which the basic content of the text is derived. Taking train travel descriptions as an example, the knowledge source is a table with information about time and place of departure, time and place of arrival, type of conveyance, etc.[1]

2. **Communicative goal**
   The communicative goal indicates the purpose of the text to be generated. For instance: to let the user know how to get from A to B by train.

[1] We assume that the NLG system receives only the relevant information from a preceding program. This is called *push* use of NLG technology in McDonald (1999). Alternatively, the NLG system may search a knowledge base itself, which is called *pull* use.

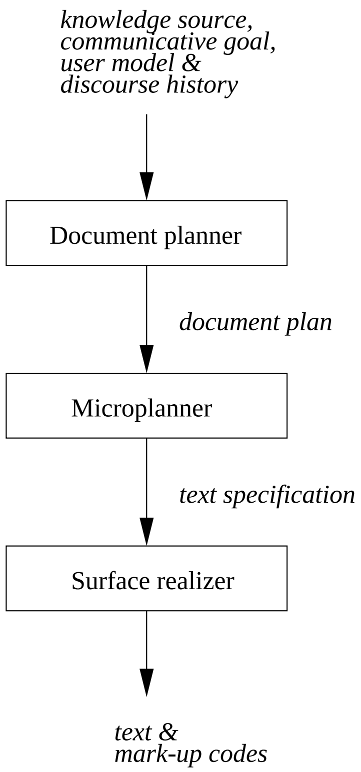


Figure 5.1: The standard architecture of an NLG system.

3. **User model**
   The user model contains a profile of the user. It may specify, for example, whether a particular user prefers a summary or extensive travel description.

4. **Discourse history**
   The discourse history is a structured record of what has been said so far in the discourse. It may inform the system, for example, that a particular station name has been recently mentioned.

The output of an NLG system can be either a plain text in electronic form, or a text enhanced with mark-up codes to be interpreted by a particular *document representation system* like a text formatting system (e.g. LaTeX) or a web browser.

The standard NLG system architecture consists of modules connected in a pipeline. Information flows from one module to the next, but feedback to a previous module is not allowed. There are three modules:

1. **Document planner**
The document planner generates a global plan for the final document, which is called a *document plan*. It selects the relevant pieces of information from the knowledge source. Continuing with our example of train travel descriptions, it may decide – depending on the preferences stored in the user model – to either include the platform numbers or to omit them. It may derive new information by means of reasoning. For instance, it may calculate the waiting time, or the difference in time between two alternative routes. These tasks are known as *content determination*. In addition, the document planner is responsible for *document structuring*. This involves – among other things – a hierarchical structuring of the information based on coherence. As far as travel descriptions are concerned, the natural structure is dictated by the chronological order of the events, i.e. a sequence of departure-arrival pairs.

2. **Microplanner**
The microplanner refines the global document plan, bringing it substantially closer to a linguistic form. It performs *lexical selection*, i.e. choosing content words, and *referring expression generation*, i.e. constructing appropriate expressions to refer to discourse entities. Here, for instance, the microplanner may choose to use the verb *depart*, and to refer to a station that was mentioned in the previous sentence by means of the anaphoric expression *there*. In addition, it performs *aggregation*, or parcelling of information that will be expressed in a single phrase or sentence. The result of microplanning is a *text specification* in the form of a tree. The non-terminal nodes are textual units like *section list*. The terminal elements are *phrase specifications*, which are detailed plans for the final form of phrases.

3. **Surface realizer**
The surface realizer generates the final form of the text, carrying out two tasks. First, it maps the phrase specifications onto grammatical expressions in the target language. This is called *linguistic realization*, and commonly relies on a syntactic grammar to enforce grammatical constraints. It involves the insertion of function words, the inflection of content words, and the determination of the word order. Often, adding orthographic symbols is looked upon as a part of linguistic realization as well. Second, it adds mark-up codes for the document representation system, which is called *structure realization*. For instance, a textual element like *list* may be mapped onto the HTML tags for an (un)ordered list.

For a software system of such complexity as NLG and SLG systems, a modular design is to be preferred over a monolithic design, because when the information is concentrated and encapsulated in modules, it becomes easier to design, implement, understand, document, maintain, and adapt the system. In addition, it provides the opportunity to experiment with different versions of a module, and to reuse one or more modules in a new system. The same considerations argue in favour of using a pipeline architecture, in which the flow of information is strictly sequential, and no feedback is allowed. Although pipeline architectures have been criticised for being too restrictive (Inui, Tokunaga, and Tanaka 1992; Rubinoff 1992), alternative architectures are significantly harder to implement. For the time being, therefore, the best thing is to elaborate on the three-stage pipeline architecture when designing an SLG system.

5.2.2 Speech synthesis

> "Speech synthesis is the process which allows the transformation of a string of phonetic and prosodic symbols into a synthetic speech signal." (d'Alessandro and Liénard 1995, p. 196)

Traditionally, *speech synthesis* (SS) has been developed and applied in the context of *text-to-speech* (TTS). A TTS system takes as input a text consisting of words and punctuation marks, and attempts to read it aloud as well as possible.[2] Although SS and TTS are often used as synonyms, the former is actually a part of the latter. In general, TTS involves two tasks (see e.g. Klatt (1987), Dutoit (1997), and Sproat (1997)):

1. A linguistic analysis of the input, which involves subtasks such as text preprocessing, grapheme-to-phoneme conversion, and determination of the prosody, of which the intonation is an important part.[3] The result of this is usually an abstract specification of an utterance in terms of phonetic symbols (phonemes) and prosodic annotations (e.g. accent and pause markers).

2. Speech synthesis, which involves mapping this specification into a synthetic speech signal.

Admittedly, the precise boundary between the two tasks is somewhat arbitrary, and differs from one TTS system to another. However, all systems use one or more intermediate levels of representation between the input (text) and the output (speech). This means that, in principle, it is possible to specify input to the synthesiser in terms of one of these intermediate representations as well.

[2] Recent approaches investigate the use of HTML or XML documents as input, which allow for annotations in the form of mark-up codes that guide the conversion of text to speech (Sproat et al. 1998).

[3] *Prosody* is defined in Section 2.2.1; *intonation* in Section 1.1.

5.2.3 Combining NLG and SS

The trivial way to combine NLG and SS systems is by using the text output of an NLG system as input to a TTS system. As might be expected, this is not the best approach. By using text as the interface, we throw away valuable linguistic information produced by the NLG system in the process of language generation, and introduce a lot of redundant processing. One example of this concerns the syntactic structure. As explained above, NLG systems usually have a syntactic realizer, a module which takes care of the syntactic form of sentences. Typically, a syntactic grammar is used for building syntactic structures, which guarantees that only acceptable strings are generated. Then why should a TTS system attempt to parse a potentially ambiguous sentence, if it can get its syntactic structure for free from the generator? A much better alternative, therefore, is to have the NLG system output not only the text, but also the accompanying syntactic structure.

The same goes for other types of linguistic information. In order to produce coherent pieces of text, the document planner in an NLG system relies on a discourse model that keeps track of the informational status of discourse referents. For instance, to use a pronoun, the generator needs to know that its antecedent is old, i.e. has been recently mentioned. This type of information, which is hard to extract automatically from a bare text, plays an important role in pitch accent placement (see Section 2.5.1). Likewise, generators often employ semantic categories such as predicates, arguments and modifiers. This kind of information is very useful for the correct accentuation of verbs (see Section 2.5.2). In conclusion, bare text is not the appropriate interface between NLG and SS, and a more elaborate representation is preferable. The same point has been made by a number of other researchers, e.g. Bateman (1997), Hess (1997), McKeown and Moore (1995), and Sproat (1997); in fact, the idea goes back as far as Young and Fallside (1979).

The interface proposed in the current work consists of three elements:

1. a phrase, i.e. a string of graphemic words;
2. its intonational structure, i.e. a phonological representation of the pitch accents, intonational boundaries, and tune;
3. part of its prosodic structure, i.e. a tree consisting of the higher-level prosodic constituent (U, π, φ).

This means that most of the tasks carried out during the linguistic analysis of the input in a conventional speech synthesis system, that is, a TTS system, become superfluous, as the desired information is already available in the input. The remaining tasks for the speech synthesiser involve grapheme-to-phoneme conversion, deriving the lower-level prosodic constituents, phonetic realization in terms of intonation and duration, and actual speech synthesis. These tasks will be described in Section 5.6. At the same time, the intonational and (part of the) prosodic structure is assumed to be generated before the speech synthesiser is reached. This raises the question where to generate prosody.

5.2.4 Where to generate prosody

Where does the generation of a phonological description of the prosody – including the intonational form – fit into the architecture of an SLG system? In fact, there is more than one plausible location: it can be seen as belonging to the microplanner, the surface realizer, or a separate module.

Recall that accent placement is highly correlated with the information status of discourse entities (cf. Section 2.5.1), that is, new information is usually accented, whereas old information is normally left unaccented. A similar correlation holds with respect to a definite versus an indefinite description, and between a full description versus an anaphoric one. Therefore, accent placement is akin to referring expression generation. Likewise, the structuring of information into sentences is akin to collecting information into intonational phrases, that is, aggregation is similar to intonational phrasing. Since both referring expression generation and aggregation belong to microplanning, it follows that prosodic realization must belong there as well.

However, there is a problem with this conception: certain information required for prosodic realization is not available when microplanning takes place. For instance, accent deletion rules like SAAR (cf. rule (20)) and the Rhythm Rule (cf. rule (22)) require the word order to be known. However, the word order is not known until after the next module has applied, that is, the surface realizer. Similarly, intonational phrasing requires information about the syntactic structure of the utterance, which is to be generated by the next module. Therefore, prosodic realization cannot be a part of the microplanner – or at least not entirely – because at that stage in the generation process the form of the text has not yet been sufficiently determined.

Alternatively, prosodic realization may be looked upon as a form of structure realization, which thus belongs to the surface realizer. Indeed, there are many parallels between printed, electronic, and spoken documents. According to this view, a speech synthesiser is just another type of document representation system, akin to a text formatting system or a web browser, and prosodic annotations are similar to mark-up codes or tags. For instance, the annotation for a pitch accent may be compared to a mark-up code for emphasis, which will ultimately be realized by the document representation system with the typographic means defined in its style sheets (e.g. in italics or underlined). Along the same lines, symbols for prosodic phrase boundaries may be compared to punctuation symbols or paragraph formatting commands. In addition, certain intonational tunes – question, exclamation, parenthetical remarks – appear to have direct orthographic counterparts. Thus according to this view, prosodic structuring takes place at an abstract level in the document planner and the microplanner, and its subsequent realization in the surface realizer is merely a straightforward mapping onto the prosodic symbols that are interpreted by the speech synthesiser.

Although this approach may be adequate for punctuation, orthography and layout, it is far less appropriate for speech, because it implies that the important decisions regarding prosody are made in the planning modules, leaving for the surface realizer only a more or less trivial mapping onto prosodic symbols. However, as was argued above, many decisions about prosodic realization cannot be made at the level of micro planning, let alone document planning, because at that stage in the generation process the final form of the text has not yet been sufficiently determined. Moreover, this approach fails to do justice to the fact that prosodic realization is a process of an inherently linguistic nature, rather than a rigid mapping from one structure to another. Accent placement, for instance, depends in a complex way on the focus distribution, the semantic structure, and the prosodic structure (which in turn depends on the syntactic structure). Therefore, even though there is an pervasive parallel between the prosodic structure and the visual structure of a document, the surface realizer is not the appropriate place for prosodic realization either.

A third option is to consider prosodic realization a separate task that is on a par with syntactic realization. This assumes that it is a linguistic rather than a purely formal mapping, and that it can be modelled in the form of a grammar, just like morphology, syntax, and segmental phonology. In addition, this approach provides the advantages of a modular architecture (cf. Section 5.2.1). We will therefore assume that prosodic realization requires an independent module that connects the surface realizer to the speech synthesiser. This means that the syntactic structure of an utterance, including its word order, is fully determined before its prosodic realization is taken into account, and that the problems mentioned earlier are avoided.

To sum up, prosodic realization cannot be reduced to either microplanning or structure realization, but is a separate task that requires an independent module. The description of prosodic realization in Chapter 7 will further support this view by showing that it enables a satisfactory implementation of the process.

5.2.5 A reusable content-to-speech system

Surface realization, and syntactic realization in particular, is probably the most well established branch of NLG. At least for English, several linguistic realization modules have been developed that are capable of mapping phrase specifications onto grammatically correct English phrases, e.g. SURGE (Elhadad and Robin 1996; Elhadad and Robin, to appear), KPML (Bateman 1994; Bateman 1997), RealPro (Lavoie and Rambow 1997; CoGenTex, Inc 1998), and Multex (Matthiesen, Licheng, Cross, Kobayashi, Teruya, and Canzhong 1998). These are freely or commercially available, reusable software packages that can be used to build new NLG systems. There are also surface realizer for Dutch (Appelo, Leermakers, and Rous 1993; Degand 1993; Degand 1996; De Smedt 1990; Kempen and Hoenkamp 1987; Marsi 1998b; Teunissen 1997), although their coverage is significantly less extensive than that of their English counterparts (see also Section 6.5.3). They allow developers of an NLG system to ignore the low-level syntactic details of the target language, and instead to focus on issues of document structuring and microplanning. Similar packages for document planning are not yet common, primarily because planning is intrinsically bound to a certain domain. For example, whereas the syntax used in a software user manual and a fairy tale is essentially the same, the planning processes for these texts are likely to be very different in nature (at least for most software).

Interestingly, prosodic realization appears to be closer to syntactic realization in this respect. That is, prosodic realization is fairly domain-independent; hence the existence of general-purpose TTS systems. This suggests that it is possible – at least to a certain degree – to develop a *reusable* 'content-to-speech' module that serves as the front-end for an SLG system, analogous to the reusable surface realizer in an NLG system. To carry the comparison further, this allows the developers of an SLG system to ignore the prosodic idiosyncrasies of the target language as long as they make sure their planning components deliver the required input. As a matter of fact, reusability provides an additional argument against the view of prosodic realization as a part of microplanning: since microplanning, in particular lexical choice and aggregation, is a fairly domain-dependent task, it would prevent the construction of a reusable module.

Following this idea, our SLG systems have two major parts. The first is a *domain-specific* text planner (PlanR) that is responsible for document structuring and microplanning. It generates the *content* of the text in the form of a text specification that includes phrase specifications. The second is a *domain-independent* front-end that is shared by all systems. This part is called CONPAS, which stands for 'CONtent to Prosodically Adequate Speech'. It is responsible for the *form* of the text, or rather, the speech. It translates content into a speech signal, using knowledge about the morpho-syntactic and prosodic properties of the target language. Figure 5.2 gives an overview of CONPAS as part of an SLG system.

Internally, CONPAS consists of three modules:

1. SynR: a module for syntactic realization
2. ProsR: a module for prosodic realization
3. SpeechS: a module for speech synthesis

Each of these modules will be described in the following sections, although SynR and ProsR are just introduced here, because they will be described in detail in the next two chapters.

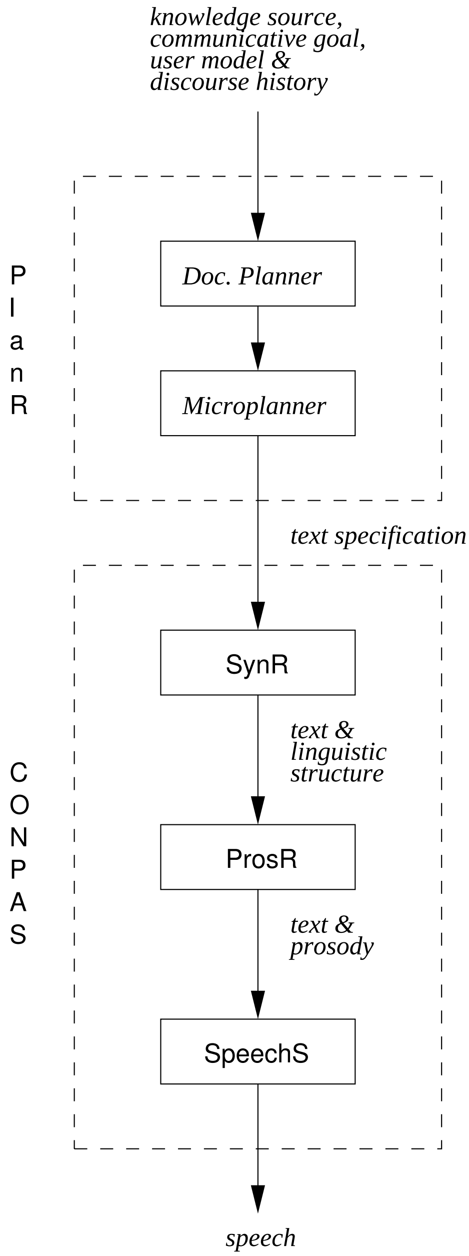


Figure 5.2: The architecture of an SLG system using CONPAS as its front-end.

5.3 Text Planning

5.3.1 Application domains

Unrestricted SLG is an achievement that is still beyond our reach, if only for the fact that a general-purpose knowledge representation formalism to model arbitrary domains is lacking. Restricting the domain of application of an SLG system is therefore unavoidable. Our work explored two application domains: the *Wild Flower Domain* and the *Public Transport Domain*.

The Wild Flower Domain (WFD) concerns botanical descriptions of wild flowers akin to those used in corpus-based work in the previous two chapters. These texts typically have long and complex utterances, which makes them challenging from a prosodic point of view. The choice for this domain allows us to reproduce the achieved results on phonological and intonational phrasing in an actual SLG system. The challenge is to show that the type of syntactic,semantic, and focus structures assumed, and manually assigned, in the corpus of flower descriptions can indeed be automatically generated by an SLG system. Example (1) is a short flower description generated by WFDGen, the CONPAS-based SLG system for this domain.

(1) a. aardoorn is een eenjarige plant # met korte # dikke #
aardoorn is an annual plant with short thick
donkerbruine stengels # en grote # donkergekleurde #
dark-brown stalks and big dark-colored
vlezige bladeren
fleshy leaves

b. de plant heeft rode # komvormige bloemen die op een
the plant has red bowl-shaped flowers that at a
lantaarn lijken
lantern resemble
'The plant has red, bowl-shaped flowers that look like a lantern.'

c. hij groeit in bossen en weilanden # in het noorden van
he grows in forests and meadows in the north of
europa
Europe
'It grows in forests and meadows in Northern Europe.'

The Public Transport Domain (PTD) concerns travel descriptions. The goal is to generate spoken travel descriptions that assist the user in travelling from one place to another using the public transport system, i.e. train, bus, or ferry.[4] In contrast to WFD, the utterances in this domain tend to be relatively

[4]Earlier work on PTD is reported in (Marsi 1995).

short and have a rather simple syntactic structure. For instance, subordinate clauses are rare, whereas complex noun phrases with multiple premodifiers do not occur at all. However, one of the interesting aspects of PTD is the occurrence of numbers and times. Also, unfocused information occurs more frequently than in WFD. We will refer to the CONPAS-based SLG system for this domain as PTDGen. Example (2) is a short travel description generated by PTDGen.

(2) a. eerst neemt u om *achtien uur dri*e # de sn*eltrein
first take you at eighteen hour three the fast-train
die richting *arnhem en den h*aag gaat
that towards arnhem and the hague goes
'First, you take the fast train to Arnhem and Den Haag at 18.03'

b. die vertrekt in nijmegen van perron *een *a
that departs in nijmegen from platform one a
'Which leaves in Nijmegen from platform 1a'

c. dan komt u om n*egentien *uur in *utrecht aan
then arrive you at nineteen hour in utrecht `prt`
'Which gets you to Utrecht at 19.00.'

d. vervolgens stapt u op de interc*ity richting amsterd*am en
next step you on the intercity towards amsterdam and
h*aarlem # die ger*eedstaat op perron z*even *a *b
haarlem that stands-ready at platform seven a b
'Next, take the intercity to Amsterdam and Haarlem, which is waiting at platform 7a/b.'

e. die vertr*ekt in utrecht om n*egentien uur tw*ee # en
that leaves from utrecht at nineteen hour two and
die komt in amsterdam *aan om n*egentien uur
that arrives in amsterdam `prt` at nineteen hour
v*ierend*ertig
thirty-four
'Which leaves from Utrecht at 19.02 and arrives in Amsterdam at 19.34.'

5.3.2 On the approach to text planning

Recall that the focus of the current research is on prosodic realization, in particular pitch accent placement and intonational phrasing. We argued earlier that this task should not be a part of microplanning, let alone document planning. As a consequence, both types of planning are less relevant for our purpose, allowing us to opt for a rather basic approach. The most important issue in planning is what information should be produced to facilitate syntactic and prosodic realization later on. In fact, in some respects it would be more appropriate to talk about our approach as *simulating* a real planning process.

To begin with, we will not maintain a distinction between document planning and microplanning. Although we fully subscribe to the arguments for distinguishing these two stages in NLG, there is no reason for it in our simple approach. Instead, we will refer to them collectively as *text planning*. As explained above, we assume that text planning is domain-specific, whereas the domain-independent realization is taken care of by CONPAS. As a result, there is a specialised text planner for each domain, i.e. a text planner for PTD called PTDPlanR, and one for WFD called WFDPlanR. Whenever the domain is evident or irrelevant, we use the term *text planner* (PlanR).

There are roughly two approaches to text planning found in NLG. *Relation-based planning*, which originates from the work of by Hovy (1988, 1991), relies on an AI-based planner that constructs hierarchical text specifications by combining rhetorical relations in order to satisfy communicative goals. *Schema-based planning*, which goes back to McKeown (1985b, 1985a), employs text schemas that describe the structure of a text, comparable to the way a syntactic grammar describes the syntax of a language. Although schema-based planning is less flexible, it it easier to implement, fast, and well-suited for generating the PTD and WFD type of texts. Hence, both PTDPlanR and WFDPlanR are schema-based text planners. They are discussed in more detail in the next section.

A more dynamic, constraint-based method of text planning for PTD is described in Marsi (1998a). In this method, the packaging and linear ordering of information, the focus distribution, the use of anaphoric expressions, and that of discourse markers result from explicit calculation. It is implemented as a functional unification grammar in FUF (Elhadad 1993). For reasons of space, we will not go into it here.

In contrast to surface realization modules, which are often implemented in a special-purpose grammar formalism or programming language, most text planners are coded directly in a general-purpose programming language, because the domain-dependent nature of the process has prevented the development of sufficiently general methods. The same goes for PTDPlanR and WFDPlanR, which are both implemented in Icon, a very high-level programming language with extensive facilities for processing strings and structures (Griswold and Griswold 1993).[5]

The purpose of our text planners is somewhat different than that of most text planners. Text planners are usually designed to generate a text specification that is optimally suited to achieve the communicative goal aimed for. This means that they are deterministic: with identical inputs, they will generate identical outputs. However, since we intend to use SLG systems as a test bed for prosodic theory, we are interested in generating as much variation as

[5]According to present terminology, Icon qualifies as a *scripting language*, akin to languages like Perl and Python.

possible. After all, the more varied the texts are, the better these theories can be put to the test. Hence, our text planners emphasise generating variation.

Recall from Section 5.2.1 that, according to the standard architecture, the input to an NLG system consists of a knowledge source, a communicative goal, a user model, and a discourse history. Actually, most systems take less input, because part of it is hard-wired in the program. For instance, many systems do not employ a user model, as assumptions about the user are implicit in the code. Of course, this precludes intelligent behaviour like tailoring output to different users (Paris 1988). However, since we are less interested in planning issues here, the input to our text planner is rather limited: PTDPlanR takes as input a knowledge source and a communicative goal, whereas WFDPlanR takes no input at all, as all of it is implicitly assumed in the code, which means that texts are randomly generated.

To sum up, text planning, which in our case covers both document planning and microplanning, is accomplished by a straightforward, schema-based approach, which is implemented in a general-purpose programming language. It aims at generating maximal variation on the basis of limited input.

5.3.3 Abstract Phrase Specifications

Prior to describing the details of the text planner, we have to address its output, which also is the input to CONPAS. According to the standard architecture of an NLG system described earlier, the result of text planning[6] is a tree-shaped text specification. The nonterminal nodes represent textual units – e.g. *section*, *paragraph* or *list* – which will ultimately be mapped to punctuation symbols and/or mark-up codes during structure realization. The terminal nodes are phrase specifications that tell the surface realizer what kind of phrase it should generate. Phrase specifications can be supplied at different levels of abstraction. They can take the form of canned text, which means that they are very close to the surface form, and the task of the surface realizer is limited to string concatenation and some orthographic processing. Close to the other extreme, they can be abstract specifications of the semantic structure in terms of semantic roles and features, often referred to as *lexicalised case frames* (Bateman and Hovy 1991; Reiter and Dale 2000). These require the surface realizer to have detailed knowledge about the syntax and semantics of the target language. The main advantage is that the more abstract a phrase specification is, the more freedom the planner has in constructing phrase specifications that optimally serve the communicative goal aimed for.

Clearly, canned text is not an option in an SLG system, because the corresponding surface realizer fails to provide the linguistic structure of phrases required for prosodic realization. In contrast, lexicalised case frames appear to hold too much information for our purpose. For instance, although the difference between arguments and modifiers is important for pitch accent placement, the subclassification of arguments into semantic roles like *agent*, *theme* and *ben-*

[6] Actually, the result of document planning followed by microplanning.

eficiary is assumed to be irrelevant (Gussenhoven 1992b). Therefore, the type of phrase specification used by CONPAS is somewhere in the middle between these extremes. It is called an *Abstract Phrase Specification* (APS), and is akin to a dependency structure. It describes a phrase in terms of content words with features, and their dominance relations. The format of APS's will be described in detail in the next chapter. For now, it is sufficient to have some idea about the form and meaning of APS's, so we can continue the discussion of text planning. Some examples with their corresponding strings are given in (3).

Example (3-a) is the most basic phrase specification for an NP. By default, the NP gets an indefinite determiner. This is a common thread in all APS's: the assumption of reasonable defaults allows for concise specifications. The feature def in (3-b) makes the NP definite. A modifier is added in (3-c), where P: indicates the syntactic category of the modifier's head, i.e. Preposition, which is different from the default, i.e. Noun. In addition, prop indicates a proper noun, which normally does not take a determiner. Finally, (3-d) shows the specification for a full sentence. Arguments of the verb are enclosed in square brackets instead of parentheses. [<listen>] generates a noun without content, i.e. a 'pronoun', where the feature for 'listener' (2nd person) helps select the correct form.

(3) a. APS: (sneltrein)
String: een sneltrein
a fast-train

b. APS: (sneltrein<def>)
String: de sneltrein
the fast-train

c. APS: (sneltrein<def>(P:naar(Amsterdam<prop>)))
String: de sneltrein naar Amsterdam
the fast-train to Amsterdam

d. APS: (V:nemen[<listen>]
[sneltrein<def>(P:naar(Amsterdam<prop>))])
String: u neemt de sneltrein naar Amsterdam
you take the fast-train to Amsterdam

The goal of text planning in PTD and WFD is to generate a list of APS's for a travel or flower descriptions respectively.

```
From Nijmegen to Amsterdam CS
10th of January 2000, Monday

Station          Time   Track  Type        Direction/Destination

Nijmegen      D  18:03  1a     Fast train  Arnhem, Den Haag CS
Utrecht CS    A  19:00
Utrecht CS    D  19:02  7a/b   Intercity   Amsterdam CS, Haarlem
Amsterdam CS  A  19:34

Travel time: 1:31, with 1 transfer
```

Figure 5.3: Output of the travel information system that serves as the knowledge source for PTDPlanR.

5.3.4 Text planning in PTD

To start PTDGen, the user queries the *NS Reisplanner*, a public transport information system, on how to get from one place to another at a certain date and time. Alternatively, a query may be generated randomly. The output of the information system is an ASCII file with tabular data regarding place and time of departure, place and time of arrival, and type of conveyance. Normally, the direction of the conveyance, its number, and the platform it leaves from are given as well. See Figure 5.3 for an example.[7]

This tabular information serves as the knowledge source for PTDPlanR. The file is parsed and for each *section* (a part from station to station) the travel parameters are stored in the form of a record. The record corresponding to the first section in Figure 5.3 is shown in Figure 5.4, where `sec` is the name of the record, and `sec.dp` refers to the field storing the departure place, `sec.dt` is the the departure time, etc.[8]

In addition, the planner accepts a communicative goal that specifies the type of description that is required. This can be BRIEF DESCRIPTION, NORMAL DESCRIPTION, or ELABORATE DESCRIPTION. For a BRIEF DESCRIPTION, it includes only the essential information, i.e. time of departure and destination, sometimes accompanied by the place of departure and type of conveyance, all in a single sentence. In a NORMAL DESCRIPTION, the place of arrival and the type of conveyance are always mentioned, while direction and platform are included when available. The information is divided over more than one sentence. Finally, an ELABORATE DESCRIPTION includes all information available

[7]The original output is in Dutch. For convenience, the English translation is given here. Moreover, there is an additional column called *Opmerkingen* 'Remarks' that is ignored by the planner.

[8]Sometimes the platform or direction are lacking, and the corresponding fields in the record remain empty.

```
sec.dp = "Nijmegen"
sec.dt = "18:03"
sec.ap = "Utrecht CS"
sec.at = "19:00"
sec.co = "Sneltrein"
sec.di = ["Arnhem", "Den Haag CS"]
sec.pf = "1a"
```

Figure 5.4: Record corresponding to the first section of the journey in Figure 5.3.

in the knowledge source. In addition, it will sometimes mention calculated waiting times as well. The information is divided over a number of sentences, and repetition of placenames is preferred over the use of anaphoric expressions. However, complex placenames are reduced when unfocused; for instance *Amsterdam* instead of *Amsterdam Centraal*.

As mentioned earlier, text planning is based on a text schema that describes the hierarchical structure of the texts. In the original work on schema-based planning (McKeown 1985b), this structure consists of rhetorical predicates that are claimed to capture general, domain-independent textual relation. Without intending to criticise this approach, we found it hard to apply these predicates to our domains, and instead opted for text schemas stated in terms of domain-dependent concepts. With respect to PTD, it is assumed that a description of a *route* consists of the descriptions of one or more *sections*, and that each section is in turn described by one or more APS's. Taking into account that there are brief, normal and elaborate descriptions, we arrive at the following text schema.[9]

(4) **Text schema for travel descriptions in PTD**

$TravelDescription \longrightarrow$
$\quad BriefTravelDescription^{+};$
$\quad NormalTravelDescription^{+};$
$\quad ElaborateTravelDescription^{+}.$

$BriefTravelDescription \longrightarrow$
$\quad BriefSectionDescription^{+}.$

$NormalTravelDescription \longrightarrow$
$\quad NormalSectionDescription^{+}.$

$ElaborateTravelDescription \longrightarrow$
$\quad ElaborateSectionDescription^{+}.$

[9] The superscript + in the regular expression means 'one or more times'.

$$BriefSectionDescription \longrightarrow$$
$$APS^{+}.$$

$$NormalSectionDescription \longrightarrow$$
$$APS^{+}.$$

$$ElaborateSectionDescription \longrightarrow$$
$$APS^{+}.$$

For each type of *SectionDescription*, there are several sets of APS templates available in order to promote variation in the output. The process of text planning is basically a matter of randomly choosing one of the appropriate sets, and substituting the travel parameters from the record in the slots of the template. This action is then repeated for each section of the route. The result is a list of APS's to be passed on to the syntactic realizer.

As an example, Figure 5.5 shows a text schema consisting of three templates implemented as an Icon procedure called `NormalDescription1`. It is called with three parameters: the record with the section's travel parameters (`sec`), the number of this section (`num`), and the total number of sections the route consists of (`tot`). It returns three APS's in the form of a list (`[...]`) of three strings (`"..."`). The templates themselves are simply concatenations of strings: everything between double quotes is a string, and `||` is the concatenation operator. A hash (`#`) indicates a comment, which is ignored by Icon.

The result of the call `Normal1(sec,1,2)`, where `sec` is the record in Figure 5.4, is the list of strings shown in Figure 5.6.[10] The lists resulting from planning subsequent sections are appended to obtain the full text specification. Its ultimate realization as a text was given in (2).

Actually, the travel parameters are not substituted directly, but are formatted by specialised functions first. These functions not only translate the knowledge source format to APS format, but also add significant pieces of content. For instance, the call `Place("Nijmegen","back")` in the first template returns the string `"(Nijmegen<back, prop>)"`, which specifies an NP with a backgrounded (unfocused) proper noun. Similarly, the call `Platform("1a")` returns `"(NUM:(1)(LET:a)(perron))"`.

The function `SectionDiscourseMarker` randomly generates an appropriate discourse marker on the basis of the number of the section and the total number of sections. With respect to Figure 5.6, the call `SectionDiscourse-Marker(1,2)` resulted in the adverb *eerst* 'first'.

The actual implementation of most schemas and templates is slightly more complicated. First, some checks are required to make sure that the application of a particular schema is allowed in the current context, which may be tested by means of preconditions. For instance, the template in Figure 5.5 can only be used if both the direction and platform number are available in the knowledge

[10] The layout is for convenience only. The actual output has no white space.

```
procedure NormalDescription1(sec,num,tot)
    return [

        # APS-TEMPLATE-1:
        # eerst/vervolgens neemt u om <DT> de <CO> richting <DI>
        # first/next take you at <DT> the <CO> towards <DI>
        "(V:nemen" ||
            "[<listen>]" ||
            "(P:om" || Time(sec.dt) || ")" ||
            "[" || Conveyance(sec.co) ||
                "(P:(N:richting<nonref,back>(P:" ||
                Direction(sec.di) || ")))]"
            SectionDiscourseMarker(num,tot) || ")",

        # APS-TEMPLATE-2:
        # die vertrekt in <dp> van <PF>
        # which leaves in <dp> from <PF>
        "(V:vertrekken" ||
            "[<spat>]" ||
            "(P:in(" || Place(sec.dp,"back") || "))" ||
            "(P:van" || Platform(sec.pf) || "))",

        # APS-TEMPLATE-3:
        # dan komt u in <AP> om <AT> aan
        # then arrive you in <AP> at <AT> +part
        "(V:aankomen" ||
            "[<listen>]" ||
            "[P:in(" || Place(sec.ap) || ")]" ||
            "(P:om" || Time(sec.at) || ")" ||
            "(ADV:dan<top,back>))"
        ]
end
```

Figure 5.5: A text schema for (a part of) a travel description implemented as an Icon procedure that returns a list of APS's.

```
[   "(V:nemen
        [<listen>]
        (P:om
            (TIME:(1)(8)(0)(3)))
        [sneltrein<def>
            (P:
                (N:richting<nonref,back>
                    (P:
                        (N:{(Arnhem<prop>)
                            (Den_Haag<prop>)}))))]
        (ADV:eerst<top,back>))",

    "(V:vertrekken
        [<spat>]
        (P:in
             (Nijmegen<prop,back>))
        (P:van
            (NUM:(1)(LET:a)(perron))))",

    "(V:aankomen
        [<listen>]
        [P:in
            (Utrecht<prop>)]
        (P:om
            (TIME:(1)(9)(0)(0)))
        (ADV:dan<top,back>))"
    ]
```

Figure 5.6: List of APS's that forms the the first part of the text specification for the travel description in (2). It is the result of applying the record in Figure 5.4 to the text schema in Figure 5.5.

source. Another example is a template using the verb *overstappen* 'transfer', which is appropriate only for non-initial sections. Second, some templates contain random choices to either include or omit pieces of information.

Notice that most of the tasks that belong to document planning and microplanning according to the standard architecture are not separately implemented in PTDPlanR. Instead, they are compiled in the schemas and templates. For example, the use of a pronoun is not the result of applying a referring expression algorithm, but is hard-wired in the template. Although this straight-forward, template-based approach may have its drawbacks (Reiter 1995), in our case there were no particular benefits to be obtained from using a dynamic approach.

5.3.5 Text planning in WFD

The goal of text planning in WFD is to deliver text specifications for the generation of flower descriptions similar to those found in popular botanical guidebooks. If we were to follow the standard architecture, this would require a text planner that is far from trivial. It has to start from a communicative goal supplied by the user, something like DESCRIBE(FLOWER X). Its knowledge source must be a large database about wild flowers. It has to select the relevant information, and possibly infer additional information. Next, it has to organise this information into coherent units, and present them in an order that makes sense. It might also need a user model to tailor the text to a layman or an expert, and a discourse history to make cross-references to previously generated flower descriptions. With respect to microplanning, it should be able to perform complex forms of lexical choice and aggregation.

Clearly, development of such a state-of-the-art text planner is beyond the the scope of the current work. Still, our goal is generating non-trivial text specifications to test CONPAS. The way we obtain them is by simulating the output of a text planner. For this purpose, we use a method of progressive expansion of APS templates. The approach is largely similar to that in PTD, with the difference that both the communicative goal and the knowledge source are hard-wired in the text planner.

Figure 5.7 shows a text schema for a flower description implemented as an Icon procedure. It returns a text specification as a list of three APS's (of which only the first APS is shown here). Each APS is responsible for describing one or more aspects of the plant such as:

- name;
- parts such as the flower, the leaves, and the stalk;
- geographical location, i.e. in which (parts of) countries it occurs (or does not occur);
- geological location, e.g, in which type of landscape it grows, on what type of soil, and with what density;

```
procedure Schema2()
    return [

        # APS-TEMPLATE-2a:
        # plant name, properties of stalk,
        # and properties of leave
        "(V:zijn" ||
            Plantname() ||
            "[plant" ||
                 "(" || PlantProperties() || ")
                 "(P:met" ||
                     "({" ||
                         "(stengel<plur>" ||
                              "(" || StalkProperties() || "))" ||
                         "(blad<plur>" ||
                             "(" LeaveProperties() || "))}))])",

        # APS-TEMPLATE-2b:
        # flower properties and flower relations
        ...

        # APS-TEMPLATE-2c:
        # geographical and geological location of occurrence
        ...

        ]
end
```

Figure 5.7: A text schema for a flower description implemented as a Icon procedure that returns a list of APS's (... indicate code that is not shown here).

- remarks, e.g., that it is useful for a certain purpose, or that it is similar to another flower.

The schema encodes the upper structure of the APS's, but relies on external procedures to fill in the details. For instance, APS template 2a in Figure 5.7 contains a call to the procedure `StalkProperties` in Figure 5.8, which generates an APS describing a flower's stalk. This procedure returns a randomly chosen element from a list of APS's that ranges from simple to complex ones.[11] The schemas were extracted from the flower descriptions in Forey (1992), and cover the most frequently observed patterns of presentation.

Procedures like `StalkProperties` may contain subsequent calls to other

[11] The question mark before the list, i.e. `?[ ...]`, is a unary Icon operator that produces a random element from the list.

```
procedure StalkProperties()
    return ? [

         # 1 property

            # 1a:
            # lang
            # long
            "A:lang",

            ...

        # 2 properties

            # 2a:
            # rechtopstaand, tot 40 centimeter hoog
            # erect, up to 10 centimeters high
            "A:{(V:rechtopstaan<part>)(A:hoog(P:tot
             (centimeter<nonref>(NUM:(4)(0)))))}",

            ...

        # 3 properties

            # 3a:
            # kort, dik , donkerbruin
            # short, thick, darkbrown
            "A:{(A:kort)(A:dik)(A:donkerbruin)}"

            ...
        ]
end
```

Figure 5.8: An Icon procedure that randomly generates an APS for describing a flower's stalk. It is invoked by the text schema in Figure 5.7.

```
[   "(V:zijn
        [aardoorn<prop>]
        [plant
            (A:eenjarig)
            (P:met
                ({
                    (stengel<plur>
                        (A:{
                            (A:kort)
                            (A:dik)
                            (A:donkerbruin)}))
                    (blad<plur>
                        (A:{
                            (A:groot)
                            (A:donkergekleurd)
                            (A:vlezig)}))}))])",

    ... ]
```

Figure 5.9: First part of the list of APS's that forms the text specification for the flower description in (1). It is the result of expanding the text schema in Figure 5.7.

ones. Of course, the more descriptions are chopped up into randomly chosen parts, the more varied they become. However, not all variation is interesting for our purpose. Most variation in lexical choice (e.g. with respect to a plant's name) is of limited interest with respect to prosodic realization, whereas variation in the syntactic structure and the length of phrases is far more interesting. Furthermore, concatenating smaller and smaller APS's increases the risk of generating nonsensical content. Even at the current level of granularity, the content of procedures must sometimes be chosen carefully. For instance, a geological and geographical location can be mutually exclusive: there is no way a flower is going to occur along the Swiss coast.

Figure 5.9 shows the first part of the list of APS's that forms the text specification for the flower description in (1). It is the result of expanding the text schema in Figure 5.7.

5.4 Syntactic realization

The APS's generated by the text planner have to be expanded to full-blown syntactic structures. This process of syntactic realization takes place in SynR, and will be the topic of the next chapter. Therefore, we only summarise the process here by means of an example. Figure 5.10 and Figure 5.11 show the syntactic tree corresponding to the first APS in Figure 5.6. The contribution of SynR covers the following areas:

- **Insertion of function words**
  Most function words do not possess an immediate semantic content of their own, but rather serve a grammatical function that depends on the context. SynR inserts all determiners, auxiliaries, pronouns[12], and pronominal adverbs (e.g. *daarvoor* 'to that end'). In addition, it inserts complementizers, prepositions and conjunctions not specified in an APS. Examples are the personal pronoun *u* 'you', the determiner *de* 'the' Figure 5.10, and the relative pronoun *die* 'that', and the conjunction *en* 'and' Figure 5.11.

- **Inflection of content words**
  Content words are usually supplied in the APS, but their morphological form depends on the context. For instance, *nemen* 'take' gets the form *neemt*, i.e. finite, singular, third person, and present tense. Likewise, nouns can be inflected as plural or possessive, and adjectives may receive the inflection suffix *-e*.

- **Distribution of focus**
  Unless something is explicitly focused or backgrounded by PlanR, SynR applies default rules to determine the focus distribution. For instance, it focuses the PP *om achttien uur drie*, but denies focus to the pronoun *u*. The focus distribution is instrumental in word ordering and pitch accent placement.

- **Expansion of numerals, times, dates, and abbreviations**
  An example of expansion the mapping of `(TIME:(1)(8)(0)(3)))` to the syntactic structure for *achttien uur drie*. Notice that whereas this type of expansion may be optional in NLG, it is obligatory in SLG.

- **Determination of word order**
  The word order is highly restricted by the syntax. For instance, the finite verb *neemt* is fronted in the main clause, whereas the finite verb *gaat* remains in final position in the embedded relative clause. Within the limits of syntax, the word order is also affected by the information structure, which includes the focus distribution, and by the weight or length of constituents.

[12]It was argued earlier that pronouns are actually determiners (Section 2.6.3).

The above tasks presuppose additional syntactic processing: syntactic roles like subject and object have to be determined, grammatical case must be assigned, and tense, mood, and voice have to be accounted for. In addition, mechanisms for feature percolation have to enforce subject-verb agreement and other syntactic dependencies. It will be clear that syntactic realization is a non-trivial task that involves much more than a few string manipulations: it requires the implementation of a fair amount of syntactic knowledge of the target language. Therefore, SynR is implemented as syntactic *grammar* that takes APS strings as input and produces labelled bracket strings as output.

Since CONPAS outputs speech rather than written text, it may seem superfluous to generate graphemic words. Nevertheless, it generates the graphemic forms first and leaves it to the speech synthesiser to convert these to their phonemic form. The alternative is more economical, and perhaps also more attractive from a theoretical point of view. However, the speech synthesiser has a large grapheme-to-phoneme lexicon at its disposal, as well as algorithms for deriving lower-level prosodic constituents (Section 5.6). Given the focus of the current project on the prosodic rather than the segmental level, we chose to take advantage of these existing resources. In addition, graphemic output allows SynR to be reused in future linguistically-based NLG systems.[13]

[13] Currently, SynR has a provisional implementation of structure realization. It can capitalise the first letter of sentence, as well as that of proper nouns. It can also insert commas and periods at the trivial locations.

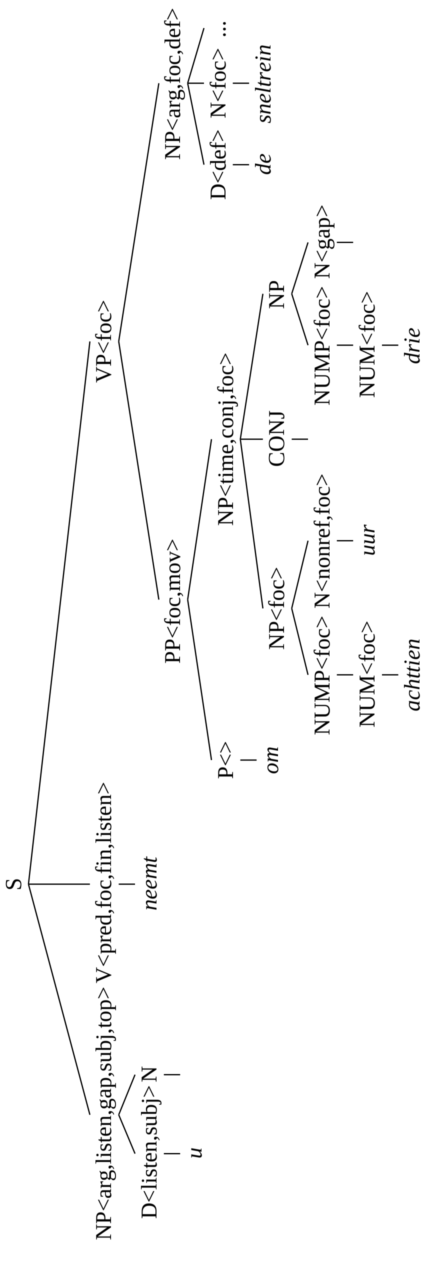


Figure 5.10: First part of the syntactic tree generated by SynR *corresponding to the first APS in Figure 5.6; the second part – shown in the next Figure – is attached at the three dots (...); see (2-a) for a gloss.*

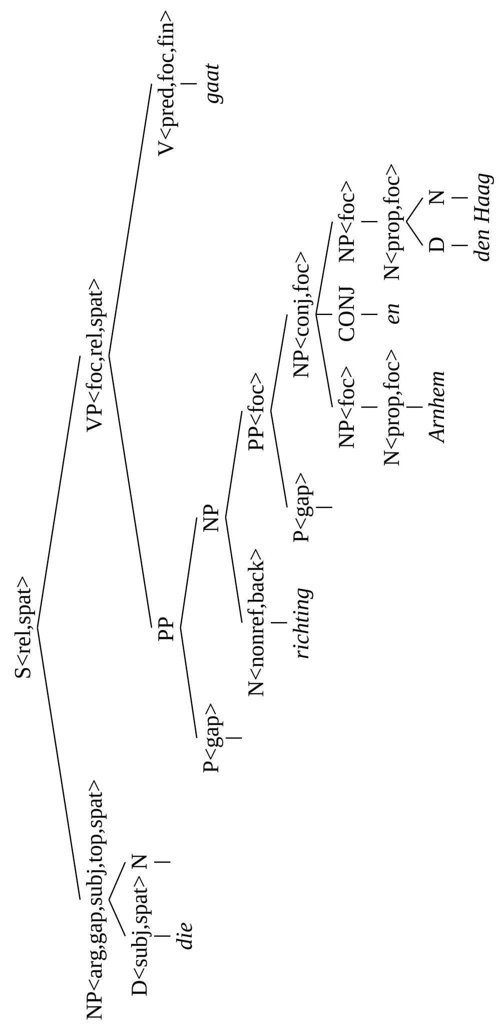


Figure 5.11: Second part of the syntactic tree generated by SynR *corresponding to the APS in Figure 5.6.*

5.5 Prosodic realization

The output of SynR is a list of utterances with their syntactic tree and focus distribution. On the basis of this information, ProsR has to generate an adequate prosodic form.

Recall that according to prosodic phonology, the prosodic structure of an utterance is a tree consisting of prosodic constituents (Section 2.2). The first part of prosodic realization in ProsR involves the derivation of the higher-level prosodic constituents, i.e. U, π, and φ. The prosodic constituents derived by ProsR for the tree in Figure 5.10 and Figure 5.11 is shown in Figure 5.12.

The second part of prosodic realization is generating the intonational form. As explained earlier (Section 2.4), this involves:

- pitch accent placement
- intonational phrasing (or the derivation of association domains)
- tune choice

The intonational form that ProsR generates for the tree in Figure 5.10 and Figure 5.11 is shown in Figure 5.13. Notice that there are two AD's, of which the second one has a Downstep tone morpheme (!).

The output of ProsR is a list of word strings interspersed with mark-up codes for pitch accents, AD-boundaries, tone attributes, and prosodic constituents. This phonological description is then translated into a phonetic description and a speech signal by SpeechS. Prosodic realization will be fully described in Chapter 7.

5.6 Speech synthesis

The final step in the generation process is speech synthesis in the module SpeechS. For this purpose, CONPAS incorporates the KUNTTS system developed at the University of Nijmegen (Kerkhoff and Rietveld 1994; Kerkhoff, Rietveld, and van Bergem 1998). Since a number of the tasks of a TTS system are superfluous when it is embedded in an SLG system, a stripped version of the KUNTTS system was used from which the modules for text preprocessing, tagging of word classes, parsing of the syntactic structure, deriving phonological and intonational phrases, assigning pitch accents, and choosing a tune were removed. The remaining tasks can be divided into three areas: phonology, phonetics, and actual synthesis.

5.6.1 Phonology

The process of mapping the graphemic words in the output of ProsR to their phonemic form is known as *grapheme-to-phoneme conversion*.[14] This is accom-

[14] KUNTTS uses combinations of ASCII characters to represent phonemes, similar to CPA (Lennig and Brassard 1984) and ESPRIT's SAMPA (ESPRIT 1992). See Kerkhoff, Rietveld,

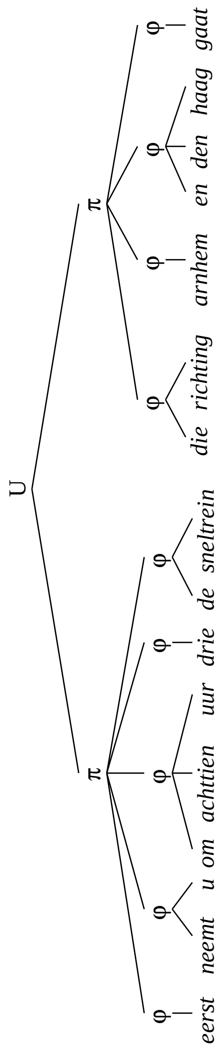


Figure 5.12: Higher-level prosodic constituents generated by ProsR *for the tree in Figure 5.10 and Figure 5.11; see (2-a) for a gloss.*

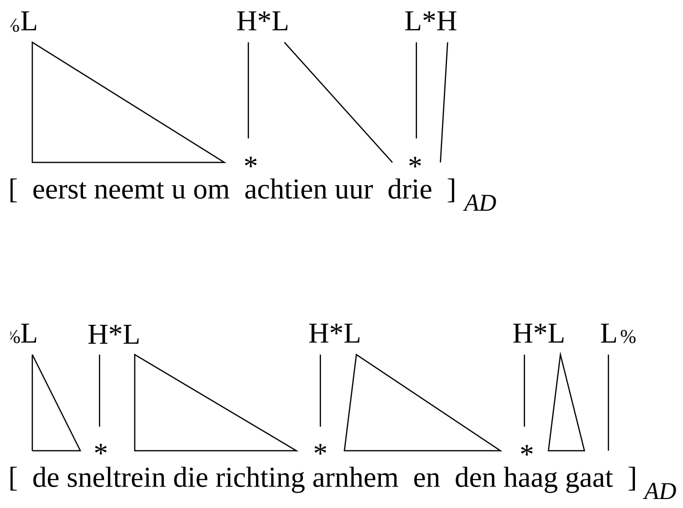


Figure 5.13: Intonational form generated by ProsR *for the tree in Figure 5.10 and Figure 5.11; see (2-a) for a gloss.*

plished by looking up the words in a lexicon (Gulikers and Willemse 1992). For instance, (5) shows the phonemic strings corresponding to the words *achttien* 'eighteen' and sneltrein 'fast-train'.

(5) a. `achttien` $\longrightarrow$ `'Axt-tin`
b. `sneltrein` $\longrightarrow$ `'snEl-+->trEIn`

Notice that the phonemic form retrieved from the lexicon does not only contain phonemes, but also information about the prosodic and morphological structure of the word: syllable boundaries are indicated by a single dash (`-`), compound boundaries by a dash-plus-dash sequence (`-+-`), the primary stressed syllable is preceded by a single right quote (`'`), and the secondary stressed syllable by greater-than sign (`>`). Thus, a word's syllabic structure and stress pattern are encoded in the lexicon.[15]

The information about stress, syllabification and compounding enables the derivation of the prosodic constituents below the φ: Prosodic Words (ω), Feet (Σ), and Syllables (σ) (cf. Table 2.1). The derivation of ω's is provisional: it

and van Bergem (1998, Appendix IV) for a translation table.

[15]KUNTTS allows for a user-defined lexicon, which overrules the main lexicon. Domain-dependent lexicons were defined for PTD and WFD to add words without affecting the main lexicon. When a word is not found in the lexicon, KUNTTS falls back on linguistic rules for grapheme-to-phoneme conversion and stress assignment. However, since all words generated in PTD and WFD are included in the domain-dependent lexicons, these rules are never used in our SLG systems.

is assumed that the ω's are identical to the graphemic words. Although this is a reasonable approximation, the relation between the morphological structure and prosodic structure of a word is often non-isomorphic – see e.g. Booij (1995, p.47). The implementation of ω-derivation in KUNTTS is the topic of current research.

In contrast, KUNTTS contains well-founded algorithm for Σ-derivation by Kerkhoff, Rietveld, and van Bergem (1998), based on the phonological description in Gussenhoven (1993), and supported by experimental evidence from Rietveld, Kerkhoff, and Gussenhoven (1999, to appear). A ω consists of one or more Σ's. The relative prominence of the Σ's is indicated by a *strong-weak* labelling. The Σ that contains the stressed syllable is marked as *strong* (Σ_s), and all others as *weak* (Σ_w). Each Σ consists in turn of one or more σ's, of which the first one is *strong* (σ_s) and all others are *weak* (σ_w).[16]

At this stage, a complete prosodic constituent structure of the utterance is available: the lower constituents – σ, Σ, and ω – are produced by KUNTTS, while the upper ones – U (Utterance), φ (Phonological Phrase), and π (Intonational Phrase) – are provided by ProsR. Figure 5.14 shows the prosodic structure of corresponding to the tree in Figure 5.10 and Figure 5.11.

Next, the phonemic string – the *underlying form* in the phonological sense – is modified by postlexical segmental rules that apply within – or at the juncture of – prosodic constituents of a certain rank.[17] As a matter of fact, these rules constitute a substantial part of the evidence for existence of prosodic constituents (cf. Section 2.2). For instance, the rule *Regressive Voice Assimilation* causes voiceless obstruents to become voiced before a voiced stop (Booij 1995). In (6), it changes the final /f/ of *elf* into /v/.

(6) [$_\pi$ die Amsterdam om v$\overset{*}{\text{e}}$ertien uur $\overset{*}{\text{e}}$lf bereikt]
that Amsterdam at fourteen hour eleven reaches
'which reaches Amsterdam at 14.11 hours'

However, the rule cannot apply across π-boundaries. In (7), where the non-restrictive relative clause forms a separate π[18], the final /f/ of *elf* is unaffected, even though it is followed by the voiced stop /d/.

[16] As a matter of fact, some words have an exceptional Σ-structure. In *perron* /pE-rOn/ 'platform', for instance, the first syllable is not dominated by a Σ, but dominated directly by a ω. The second syllable forms a monosyllabic Σ. This violation of the Strict Layer Hypothesis is assumed to be necessary for prosodic constituents below the φ (Booij 1995, p.144-145). See also (Gussenhoven 1993).

[17] In TTS, *lexical* rules, like *Final Devoicing*, apply to words that have been derived by grapheme-to-phoneme conversion. However, since we have chosen to include all words used in PTD and WFD in the lexicon, lexical rules are never applied.

[18] The derivation of π's will be described in Section 7.2.2.

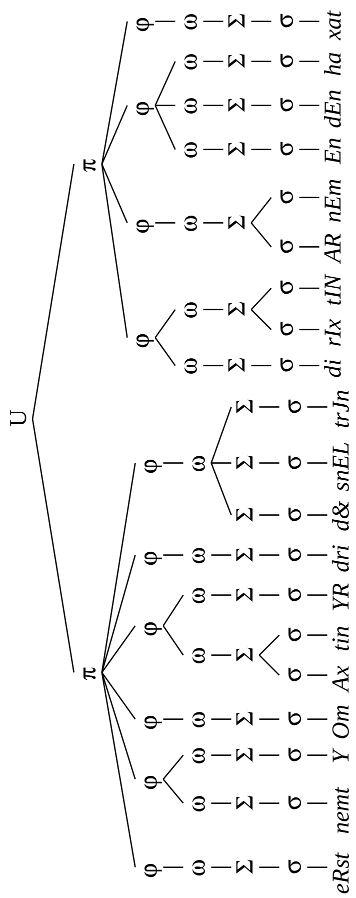


Figure 5.14: Prosodic structure derived by ProsR *and* SpeechS *corresponding to the tree in Figure 5.10 and Figure 5.11; see (2-a) for a gloss.*

(7) [$_\pi$ de trein van veertien uur elf] [$_\pi$ die naar Amsterdam
the train of fourteen hour eleven which to Amsterdam
gaat]
goes
'the 14.11 train, which goes to Amsterdam'

Although some interesting results for Dutch are available (Loots 1983; Menert 1994; Ernestus 2000), there is probably much more to be learned about the relation between segmental rules, the prosodic constituent structure, and performance factors like speaking rate and speech style. The current framework with KUNTTS embedded in CONPAS is well-suited for implementing, testing and improving our knowledge concerning this topic. Our informal experience so far indicates that the obtained agreement between the prosodic and the segmental structure of an utterance contributes to the perceived naturalness of its synthetic rendering.

5.6.2 Phonetics

The phonological structure consisting of a phoneme string, a tone string, and a prosodic constituent structure is passed on to the phonetic implementation module, which computes the phonetic details. Two dimensions of the phonetic structure can be distinguished: duration and intonation.

5.6.2.1 Duration

Creation of the durational structure involves two tasks. First, each phoneme must be assigned a duration in milliseconds. Second, pauses of various length must be inserted at appropriate positions.

Initially, vowels (with the exception of diphthongs) are assigned the same duration. No distinction is made between so-called 'long', 'half-long', and 'short' vowels. This basic duration corresponds to the mean duration found in vowels in the weakest prosodic position, that is, a weak syllable in a non-peripheral weak foot (Rietveld, Kerkhoff, and Gussenhoven 1999). The consonants are assigned specific, intrinsic values. Next, segments are lengthened according to their prosodic position in terms of headedness of foot and syllable, and vowel type. High lax vowels (/i, y, u/) are lengthened less than the other vowels. Finally, phoneme durations are modified by context-sensitive rules. The prosodic structure plays an important role here, as the right-side boundaries of prosodic constituents are subject to *preboundary lengthening*, i.e. the duration of the phonemes preceding the boundary is increased. It is assumed that the amount of lengthening is proportional to the rank of the prosodic constituent: the higher the rank, the more lengthening occurs (Hofhuis, Gussenhoven, and Rietveld 1995; Gussenhoven and Rietveld 1992a; Cambier-Langeveld 2000). In a similar way, strong Σ's and strong σ's are lengthened (Rietveld, Kerkhoff, and Gussenhoven 1999). Which phonemes are involved and how much they are lengthened depends on the phonemic context. For instance, the vowel in a strong σ that is the head of strong Σ is lengthened by 20%. These rules and their interactions form a delicate system for shaping the rhythmic structure of an utterance; for a more elaborate description, see Kerkhoff, Rietveld, and van Bergem (1998). The nett result is that the hierarchical and strong-weak relations in the prosodic structure of an utterance become mirrored in its durational structure.

Besides determining the length of phonemes, creating the durational structure also involves the insertion of pauses at appropriate positions. In analogy to preboundary lengthening, prosodic phonology predicts that the chance of pause insertion, as well as the length of the pause, is proportional to the rank of the prosodic constituent. In SpeechS, a φ-boundary triggers a small pause (by default 15ms), whereas a π-boundary triggers a larger pause (by default 50ms).

As might be expected, the durational structure is related to the intonational structure. Accented syllables receive additional lengthening. Similarly, the right-side boundaries of AD's force additional preboundary lengthening (Gussenhoven and Rietveld 1992a).

5.6.2.2 Intonation

The second aspect of the phonetic implementation concerns the details of the intonational structure. Recall that the phonological description of the intonation is a tone string that consists of pitch accents (H*L, L*H and H*LH), initial boundary tones (H% and L%), final boundary tones (%H and %L), and tone morphemes (e.g. ! for Downstep). Every starred tone is associated with a stressed syllable, and every boundary tone with an AD-boundary. This phonological description has to be translated to a sequence of F_0 targets describing the corresponding pitch contour. An F_0 target is a pitch value (in Hz) at a point in time (in ms). Determining its point in time is called *timing* (or *alignment*); determining its pitch value is called *scaling*. The pitch contour is obtained by interpolation through the targets. This *target-interpolation model* is described more extensively in van den Berg, Gussenhoven, and Rietveld (1992), Gussenhoven and Rietveld (1992b), and Kerkhoff and Rietveld (1995).

Timing Each H or L tone is phonetically realized by at least one F_0 target. We will take the tone string %L H*L H% as an example. A graphic representation of the timing of its targets is given in Figure 5.15. The appropriate domain for timing these targets is the AD. The timing of initial (t1) and final (t6) boundary tones is provided by, respectively, the (start of the) first and (end of the) last voiced phoneme of the AD. The starred tone of a pitch accent (t3) is timed at STARTIME. By default, it is timed at 30% of the duration of the vowel in the accented syllable, counting from the start of the vowel. However, the STARTIME is shifted as a function of the segmental content of a syllable (Rietveld and Gussenhoven 1995).

The remaining tones are timed relative to these reference points. The variables TOTIME and FROMTIME are used to locate them to the left or the

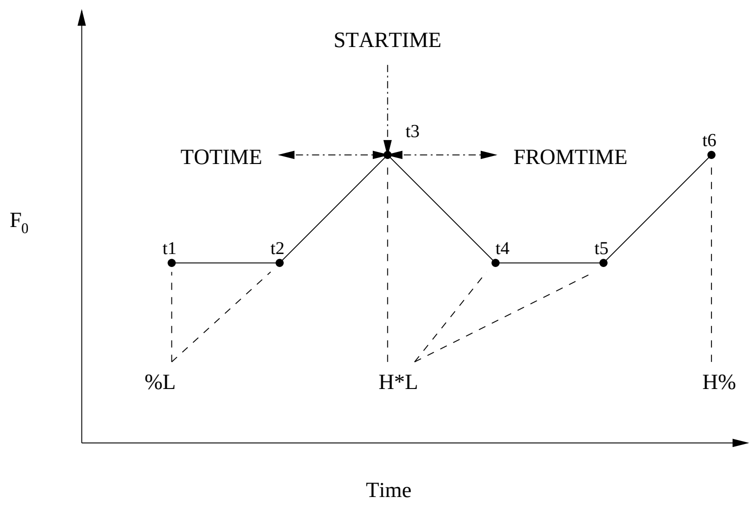


*Figure 5.15: Graphic representation of target timing for the tone string %L H*L H%. The initial (t1) and final (t6) boundary tones are timed at respectively the start and end of the AD. The starred tone of a pitch accent (t3) is timed at STARTIME, which is expressed as a percentage of the duration of the vowel in the accented syllable. The second target of the (spreading) initial boundary tone (t2) is located at TOTIME before the starred tone. The first target of the (spreading) non-starred tone (t4) is located at FROMTIME after the starred tone, while its second target (t5) is located at TOTIME before the final boundary tone.*

right of a reference point, respectively. Some tones control the pitch of longer stretches of speech, and yield two targets. This is called *spreading* of tones. An initial boundary tone always spreads. Its second target (t2) is located at TOTIME before the first starred tone. The second tone of an AD-final pitch accent spreads as well. Its first target (t4) is located at FROMTIME after the starred tone, while its second target (t5) is located at TOTIME before the final boundary tone.[19]

The time required by the variables FROMTIME and TOTIME is sometimes more than the time actually available in the phoneme string. To accommodate for such cases, overlap functions are used to merge targets.

Scaling The scaling of targets is based on the model described in Ladd (1987), Ladd (1990), and van den Berg, Gussenhoven, and Rietveld (1992). At its core is a scaling function $F(T, n, m)$ that yields a pitch value for the starred tone T (H* or L*) that is the nth accent in the mth AD. It has the following parameters:

1. *Frequency reference* (Fr) defines the bottom of the speaker's range. It represents speaker-dependent differences in range, e.g. between male and female speakers.
2. *Normalisation* (N) defines the degree of raising of the register. It models the overall emphasis or prominence of the pitch.
3. *Width* (W) defines the width of the register, i.e. the difference between the pitch value for L* and H* targets. It is related to the 'liveliness' of the pitch.
4. *da* (downstep accentual) defines the step size for lowering of H* targets in an AD with accentual downstep.
5. *dp* (downstep phrasal) defines the step size for lowering of AD's in an SD with phrasal downstep.

We will take the downstepped tone string !%L H*L H*L H*L L% as an example. A graphic representation of the scaling of its targets is given in Figure 5.16. The target for the first starred tone (t1) is scaled at frequency f1, which equals the reference frequency (Fr) plus the frequency normalisation (N) plus half of the register width (W). The second starred tone (t3) is scaled at f2, which equals the accentual downstep factor (da) times the frequency of the preceding starred tone (f1). Similarly, the third starred tone (t5 at f3) is scaled slightly lower again than the previous one. The low targets (t2, t4, and t6) are unaffected by downstep.

Scaling of nonstarred tones and boundary tones depends on the context. The pitch value for an H is somewhat lower than that of an H*, whereas the pitch value of an L is somewhat higher than that of an L* (cf. t2, t4, and

[19] Some tone attributes trigger the spreading of starred tones as well (Gussenhoven 1988).

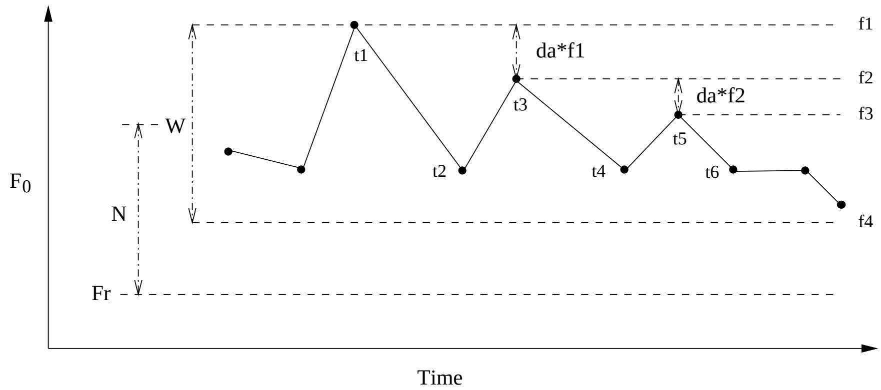


*Figure 5.16: Graphic representation of target scaling for the downstepped tone string !%L H*L H*L H*L L%. The target for the first starred tone (t1) is scaled at frequency f1, which equals the reference frequency (Fr) plus the frequency normalisation (N) plus half of the register width (W). The second starred tone (t3) is scaled at f2, which equals the accentual downstep factor (da) times the frequency of the preceding starred tone (f1). Similarly, the third starred tone (t5 at f3) is scaled slightly lower again than the previous one. The low targets (t2, t2, and t6) are unaffected by downstep.*

t6 in Figure 5.16); the difference is roughly one-third of the register width. The pitch value for a final boundary tone is obtained by multiplying the pitch value of the preceding tone by a factor less than one (for L%) or greater than one (for H%). Finally, utterance final H*L pitch accents are implemented with a certain amount of *final lowering* (Ladd 1984; Liberman and Pierrehumbert 1984; Pierrehumbert and Beckman 1988).

In addition to Accentual Downstep, there is another form of downstep that affects subsequent AD's, which is called *Phrasal Downstep*. It is similar to Accentual Downstep, but rather than lowering the next H*, the next AD is lowered, causing all the H* targets it contains to be lowered by same amount. This amount of phrasal downstep is controlled by the value of the parameter dp.

The pitch values of the targets depend on the values that are chosen for the scaling parameters mentioned above. It is assumed that these remain constant across a *scaling domain* (SD), which normally corresponds to a whole utterance. In principle, it is possible to vary the scaling parameters from one utterance to the next according to the global discourse structure. For instance, to obtain a gradual lowering of the average pitch level in a paragraph, or a temporary narrowing of the register in the case of a parenthetical remark. Although we experimented with this idea, it is has not been implemented in CONPAS, as it is beyond the scope of our research questions. We will further discuss this issue in Section 9.2.2.

5.6.3 Synthesis

With all phonetic details implemented, the only step left is the actual synthesis of speech. For this, SpeechS transfers the output of KUNTTS to the MBROLA speech synthesiser (Dutoit and Leich 1993; Dutoit, Pagel, Pierret, van der Vreken, and Bataille 1996). It makes use of *diphones*, units that start in the middle of a phoneme's steady state and end in the middle of the next one. Carefully selected diphones for all phoneme transitions occurring in the target language are stored in an efficiently coded database. We used the *nl2* diphone database developed by Arthur Dirksen.

The first step is to retrieve the diphones that match the phonemes in the input. Figure 5.17 shows the input format.[20] Each line starts with a phoneme symbol and its duration, optionally followed by one or more F_0 targets, which are specified in terms of a percentage of the phoneme's duration and an F_0 value in Hz.

Next, the pitch and duration of the diphones are modified to fit the parameters in the input. This is accomplished using a variant of the *pitch-synchronous overlap-add* (PSOLA) algorithm (Charpentier and Stella 1986). Finally, the diphones are concatenated to a speech signal, which can be played directly, or saved in some audio file format. As for any normal speech signal, we can

[20] These phonemes are slightly different from those used internally by KUNTTS, because they depend on the particular diphone database that is used.

```
_  20   0 109
I: 110  0 109
R  45
s  65
t  60
_  15
n  80
e  115
m  60
t  60
y  105
_  15
O  80   87 98
m  60
A  100  30 158
x  75
t  55
i  75
n  70
y  125
R  60
d  65
r  65   52 105  98 83
i  140  42 83  71 158
_  50   0 158
.
.
.
```

Figure 5.17: Input to the MBROLA synthesiser for the first part of utterance (2-a). The first and second column contain the phoneme labels and their duration in ms. The (optional) following columns specify F_0 targets as a percentage of the phoneme's duration and a frequency value in Hz. For instance, the 4th target occurs at 30% of the duration of the phoneme A, at 158 Hz.

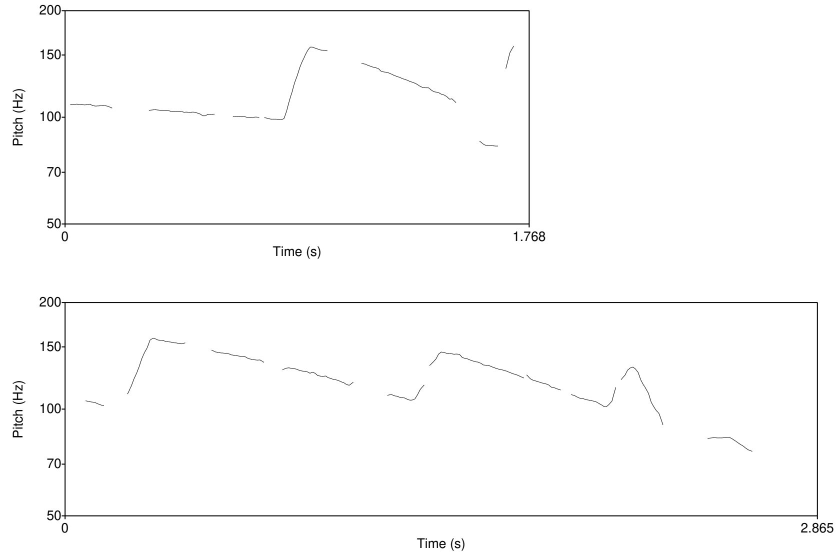


Figure 5.18: Pitch contour of the synthesised version of utterance (2-a).

extract the pitch contour with a signal processing program. Figure 5.18 shows the pitch contour for our example utterance.

5.7 Summary

This first part of this chapter addressed questions regarding the general architecture of an SLG system. We suggested to conceive an SLG systems as an extension of the standard three-module pipeline architecture for NLG systems. With regard to combining NLG and SS, it was argued that text is not a good interface. Instead, a phonological representation was proposed consisting of a sentence, its intonational structure (pitch accents, intonational boundaries, and tune), and its higher-level prosodic constituents. With respect to the proper place for controlling the intonation, it was argued that prosodic realization, which includes generating the intonational form, cannot be reduced to either microplanning or structure realization, but is a complex, inherently linguistic task that deserves an independent module.

We proposed to use a domain-independent, reusable content-to-speech system, analogous to the reusable surface realizers that are common in NLG. Accordingly, the architecture of our SLG systems has two major parts. First, a domain-specific text planner (PlanR) that generates the *content* as a text specification consisting of phrase specifications. Second, a domain-independent front-end called CONPAS ('CONtent to Prosodically Adequate Speech') that is responsible for the *form* of the spoken output. It translates the content to a speech signal, using knowledge about the morpho-syntactic, phonological, and phonetic properties of the target language. Internally, it has modules for syntactic realization (SynR), prosodic realization (ProsR), and speech synthesis (SpeechS).

Two application domains to explore prosody in SLG were introduced: the Public Transport Domain (PTD) and the Wild Flower Domain (WFD). An approach to text planning for these domains was described that is straightforward, schema-based, and directed towards generating maximal variation on the basis of limited input. We briefly introduced the abstract phrase specifications (APS's) that serves as the intermediary representation between PlanR and CONPAS, and argued that these representation have the desired level of abstraction. Next, domain-specific text planners for PTD and WFD were described. It was shown how text schemas and APS templates are implemented in Icon.

An overview of syntactic and prosodic realization was provided. The former involves the insertion of function words, the inflection of content words, the distribution of focus, the expansion of numerals, times dates, and abbreviations, and the determination of word order. The latter is concerned with pitch accent placement, intonational phrasing, tune choice, and derivation of higher-level prosodic constituents. Both forms of realization will be described extensively in the next two chapters.

Finally, speech synthesis was described in terms of three areas. The first area is phonology, which involves grapheme-to-phoneme conversion, the derivation of lower-level prosodic constituents, and the application of lexical and post-lexical segmental rules. The second area is phonetics, which deals with the durational and intonational details of the utterance. It was shown how the availability of a full prosodic constituent structure facilitates preboundary lengthening and pause insertion. It was also described how the phonological description of the intonation is translated into a sequence of F_0-targets for a pitch contour by a target-interpolation model that performs timing and scaling of targets. The third area is the actual synthesis, for which the MBROLA diphone speech synthesiser is used.

Chapter 6

Syntactic Realization

6.1 Introduction

Syntactic realization concerns the translation of phrase specifications into grammatical expressions in the target language. The complexity of this process mainly depends on the abstraction of the phrase specifications. In more concrete representations (e.g. strings of canned text), syntactic realization amounts to little more than string concatenation and some orthographic processing. If, on the other hand, the representation is very abstract (e.g. a predicate-logical expression), the process involves a substantial amount of work. Following the standard architecture for NLG systems (Section 5.2.1), we assume that lexical choice, aggregation, and referring expression generation belong to microplanning rather than to syntactic realization. As a consequence, our phrase specifications already contain the lemmas of content words, an abstract hierarchical specification of the syntactic form, and features indicating the type of referring expression. These Abstract Phrase Specifications (APS's) are somewhere in the middle on the scale of abstractness. Some examples of APS's and their realizations are given in (1).

(1) a. APS: (sneltrein)
String: een sneltrein
a fast-train

b. APS: (sneltrein<def>)
String: de sneltrein
the fast-train

c. APS: (sneltrein<def>(P:naar(Amsterdam<prop>)))
String: de sneltrein naar Amsterdam
the fast-train to Amsterdam

d. APS: (V:nemen[<listen>]
[sneltrein<def>(P:naar(Amsterdam<prop>))])
String: u neemt de sneltrein naar Amsterdam
you take the fast-train to Amsterdam

As explained in Section 5.3.1, our goals regarding prosodic realization require the use of this fairly abstract representation, and the generation of a complete syntactic structure for each utterance. Hence, the task of the syntactic realization module of CONPAS is far from trivial. Syntactic realization of APS's involves basically three tasks: determining the word order, inflecting content words, and inserting function words. In addition, and specific to SLG, it requires the distribution of focus, and the expansion of numerals, times, dates, and abbreviations. Moreover, not only strings of words are generated, but also their syntactic structures, which are to be used for prosodic realization. These syntactic structures conform to the analysis of Dutch syntax outlined in Section 2.6, which was also used for the syntactic annotation of the flower descriptions in the previous two chapters. These structures may be less intricate than those assumed in current syntactic models, but they proved to be sufficient for our purposes. That is, they serve as convenient data structures in the process of syntactic realization (e.g. to enforce agreement relations), and they provide the necessary information for prosodic realization. The results of the current work can therefore be incorporated in other syntactic frameworks without much difficulty.

We have chosen to split the realization process into three natural steps. Accordingly, the syntactic realization module SynR is divided into three components:

1. Base, which is primarily concerned with expanding an APS to a full-blown syntactic structure that adheres to the principles of $\overline{X}$-theory

2. Trans, which is responsible for enforcing grammatical constraints, determining the word order, and some additional tasks like distributing focus

3. Flex, which inserts function words and inflects content words

Like CONPAS itself, these components form a strict pipeline.

The three components are described below, including an introduction to the grammar formalisms used for their implementation. The goal is to provide a description that highlights interesting aspects, in particular with respect to SLG, at the expense of technical details of the implementation. The chapter closes with a discussion and summary of the results.

6.2 The Base component

6.2.1 The format of Abstract Phrase Specifications

An APS is a tree structure in which every node has the following form:

(2) **Format of an APS node**
(*category* : *lemma* < *features* > *daughters*)

The *category* is a major or a minor syntactic category. There are six major categories: N (Noun), V (Verb), A (Adjective), ADV (Adverb), P (Preposition), and NUM (Numeral). The minor categories are: D (Determiner), C (Complementizer), CONJ (Conjunction[1]), LET (Letter[2]). Although the major-minor distinction is irrelevant to the format of an APS itself, it is used during syntactic realization: following Gazdar, Klein, Pullum, and Sag (1985), only major categories project a phrase (e.g. N triggers the generation of an NP). This will be further explained in Section 6.2.3.1.

A *lemma* is the base form of a word, as used in a normal dictionary. In general, only the content words in an APS have a lemma, whereas function words have none (i.e. the empty string). However, it is sometimes necessary to overrule the default realization of a function word by explicitly prescribing its lemma. For instance, `(CONJ:alsmede)` forces the use of *alsmede* 'as well as' rather than the default *en* 'and'.

With respect to prepositions, a distinction is assumed between purely syntactic prepositions subcategorized for by a verb, and meaningful prepositions, e.g. indicating a spatial relation. The former are automatically inserted by SynR, the latter have to be supplied in the APS. A lemma in an APS may also be empty, which is often the case for nouns. Dependent on the features supplied, SynR generates an appropriate word form, e.g. a demonstrative determiner.

The *features* form a comma-separated list of semantic and syntactic features. In general, features mark exceptions to the default. For instance, nouns are assumed to be singular, unless they bear the feature `plur`; verbs are assumed to carry present tense, unless the feature `past` is supplied.

The *daughters* form a list of nodes, which makes the definition of a node recursive. Each node can have an arbitrary number of daughters, including zero. The mutual order of the daughters is irrelevant, with two exceptions. First, arguments are assumed to be ordered, in order to facilitate linking of arguments to syntactic functions (Section 6.3.3). Second, adjectives surface in the same order they occur in the APS (Section 6.3.7.5).

In an SLG environment primarily targeted at research, it is important that APS's can be easily read and written. This is partly accomplished by using defaults, which relieve the text planner (or the user) of the necessity to specify

[1]The syntactic category of all words that can be used to connect the conjuncts in a coordination. This includes a disjunction like *of* 'or'. We have no account yet for coordination with a 'split conjunction' like *zowel X als Y* 'both X and Y'.

[2]For example, the category of the letter *A* in *perron 1A* 'platform 1A'.

all features. In addition, there are some notational conventions ('syntactic sugar'):

- If the category and the semicolon are omitted, the node will interpreted as a noun. That is, (trein) is equivalent to (N:trein)
- If a noun starts with a capital, it will be interpreted as a proper noun, i.e. (Amsterdam) is equivalent to (Amsterdam<prop>).
- Rather than using arg to indicate that a daughter functions as an argument, it can be enclosed by square brackets ([...]) instead of parentheses.
- An empty list of features (<>) may be omitted.
- Braces ({})can be used to indicate a coordinate structure. For example, (N:{(N:stoptrein)(N:sneltrein)}) is equivalent to (N:<conj>(N:stoptrein)(N:sneltrein))

As a result, we can write [{(Utrecht)(Amsterdam)}] rather than (N:<arg,conj>(N:Utrecht<prop>)(N:Amsterdam<prop>)).

6.2.2 The AGFL system

APS's can be formally defined by a context-free grammar. We can build a parser that recognises all and only APS expressions, but this is not particularly interesting from an NLG viewpoint. However, if during the parsing process the input can be *transduced* to a new structure, we can use this to map APS's onto natural language expressions. This is exactly what the AGFL system (Koster 1991; Derksen and Potjer 2000) allows us to do.

AGFL stands for *Affix Grammars over Finite Lattices*. It is a member of the family of *two-level grammars*. The first level consists of the context-free rewriting rules familiar to any linguist.

(3) NP : Det, N.

The second level consists of context-free affix-rules, rewriting affix-nonterminals into affix-terminals or other affix-nonterminals.[3]

(4)
```
number ::  sing; plur.
definiteness ::  def; indef.
```

In contrast to the syntax-level, the affix-level does not allow recursion.

The two levels can be combined by using the affixes as *parameters* to nonterminals of the syntax rules. According to the *Consistent Substitution Rule*, all occurrences of an affix (with the same name) within a rule have the same value, which implies unification of the affix values. For instance, both occurrences of number will get the same value in the first rule the following grammar.

[3]The newest versions of AGFL use '—' rather than ';' to separate alternatives.

(5)
```
NP : Det(number,definiteness), N(number).

Det(sing,indef) :  "a".
Det(number,def) :  "the".
Det(plur,indef) :  .

N(sing) :  "flower".
N(plur) :  "flowers".
```

This grammar describes the noun phrases *a flower*, *the flower*, *flowers*, and *the flowers*, but rejects *a flowers* and *flower*.

The AGFL-system takes the grammar defined by (4) and (5), and generates a parser for it that produces a parse tree like (6) for the input *a flower*.

(6)
```
NP{Det(sing,indef){"a"},N(sing){"flower"}}
```

Of course, multiple trees will be generated if the input is ambiguous according to the grammar.

The affix mechanism is a powerful tool for making large grammars more compact. In addition, it allows a description that captures linguistic generalisations. It includes a number of special features that make it suitable for computational linguistic purposes. Particularly important for our purposes is that AGFL allows *transduction*: each rule can have a transduction part – starting with a forward slash (/) – that defines the string that is produced when the rule is successfully applied in the parsing process. Consider the grammar in (7).

(7)
```
NP : Det(number,definiteness), N(number) / Det, N.

Det(sing,indef) :  "a" / "een".
Det(number,def) :  "the" / "de".
Det(plur,indef) :  .

N(sing) :  "flower" / "bloem".
N(plur) :  "flowers" / "bloemen".
```

The AGFL-system can generate a transducer for this grammar. Given the inputs *a flower* and *the flowers*, the transducer will output the strings *een bloem* and *de bloemen* respectively. Thus, the transducer for grammar (7) is in fact a very primitive machine translation system. However, this approach is unlikely to be easily scaled up. In particular, the formalism is not very suitable for describing the complex word order phenomena in clauses (Section 6.3.7), or the long distance dependencies which may result from wh-movement. Moreover, since the descriptive power of an AGFL is that of a context-free rewrite grammar, accounting for certain types of dependencies may even prove to be theoretically impossible.

The same goes for using AGFL in a generation system: it is not suitable for mapping APS's onto grammatical expressions *directly*. However, this is not our goal, as we intend to build a modular generation system. We envision the generation process as naturally consisting of three steps. We therefore use the strengths of AGFL in the first and last steps, i.e. for mapping APS's onto basic syntactic structures in Base, and for inserting and inflecting words in Flex, but avoid its weakness by using a more powerful – but also slower and more complex – transformational grammar for the second step, i.e. for enforcing agreement relations and determining word order. Here, we continue with a description of the first step.

6.2.3 The tasks of Base

Base is the grammar that defines valid APS's.[4] At the same time, it defines the transduction of APS's into basic syntactic structures that can be processed by GRAMTSY in the next component. GRAMTSY, the interpreter for transformational grammars, will be described later on; for the moment, only the *format* of its syntactic structures is relevant. This is essentially the familiar labelled bracket structure enhanced with features.

(8) a. **Format of non-terminal node in GRAMTSY**
(*category* < *features* > - *daughters*)

b. **Format of a terminal node in GRAMTSY**
(*category* < *features* > - *lemma*)

Conceptually, mapping APS's onto basic syntactic structures involves the following tasks: (1) projecting the basic syntactic structure, (2) adding purely syntactic information, (3) adding the structure of compound words, and (4) expanding numbers and times.

6.2.3.1 Projecting the basic syntactic structure

Base projects a syntactic structure that adheres to a simplified version of $\overline{X}$ theory (Jackendoff 1977). It is assumed that every lexical item of a major category has a phrasal projection of the same category, i.e. an N is directly dominated by an NP, a V by a VP, etc. In contrast, minor categories have no phrasal projection. No intermediary levels of projection are assumed. The syntactic structures corresponding to the examples in (1) are given in (9).[5]

(9) a. APS: (sneltrein)
Syntax: (NP-(DET-een)(N-sneltrein))

b. APS: (sneltrein<def>)
Syntax: (NP-(DET-de)(N<def>-sneltrein))

c. APS: (sneltrein<def>(P:naar(Amsterdam<prop>)))
Syntax: (NP-(DET-de)(N<def>-sneltrein)
(PP-(P-naar)(NP-(N<prop>-Amsterdam))))

[4] More accurately, Base is the transducer generated by the AGFL system from the grammar that defines all and only APS's.

[5] Some syntactic feature are omitted for the sake of presentation.

d. APS: `(V:nemen[<listen>]`
`[sneltrein<def>(P:naar(Amsterdam<prop>))])`
Syntax: `(VP-(NP<arg>-(N<listen>-))`
`(NP<arg>-(DET-de)(N<def>-sneltrein)`
`(PP-(P-naar)(NP-(N<prop>-Amsterdam)))))`

The grammar fragment in Figure 6.1 shows how this can be accomplished in AGFL.[6] Notice how the square brackets of an APS node give rise to the insertion the feature `arg` at the phrasal GRAMTSY node. Of course, this grammar only illustrates the principle; the actual implementation employs the affix level for concise formulation, takes care of the notational conventions, and takes into account the additional tasks of Base described below.

6.2.3.2 Adding purely syntactic information

The second task of Base concerns adding purely syntactic information. Some properties of words are purely syntactic: they affect the syntactic structure, but make no semantic contribution. For example, the neuter/non-neuter distinction in common nouns affects the choice of the determiner: a singular, definite, *neuter* noun takes the determiner *het* (e.g. *het perron* 'the platform'), whereas other definiteness take *de* (e.g. *de trein* 'the train' *de perrons* 'the platforms'). It belongs to the task of Base to add this lexical information to the basic syntactic structure, for example, to attach the feature `neut` to neuter words.

Other purely syntactic information includes:

- the preposition of a verb's prepositional object
- the auxiliary that a main verb takes to form its perfect tense, either *zijn* 'be' or *hebben* 'have' (10)
- whether the complementizer of a finite sentential object of a verb is *of* rather than the default *dat* (11)
- whether an infinite sentential object does *not* take the default complementizer *om* (12)
- whether a noun is non-countable (mass noun)

(10) a. hij heeft gereden
he has driven

b. hij is overgestapt
he is transfered

[6]If a rule has no transduction part (cf. `MajorLemma`), the AGFL system assumes that its transduction is the concatenation of the transductions of its right-hand symbols, and that the transduction of a terminal is the terminal itself.

```
MajorCategoryNode :
    "[",
        MajorLabel,
            MajorLemma,
               Features,
                   Daughters,
                      "]" /
    "(",
        MajorLabel,
            "P<arg>-",
                Daughters,
                    "(",
                        MajorLabel,
                            Features,
                               "-",
                                  MajorLemma,
                                      ")",
                                          ")" ;
    ...

MajorLabel :
    "N:" / "N" ;
    "V:" / "V" ;
    ...

MajorLemma :
    "bloem" ;
    "trein" ;
    ...
```

Figure 6.1: AGFL grammar that shows the principle for projecting the basic syntactic structure.

(11) a. zij zegt dat de sneltrein gaat
she says that the fast-trein goes

b. zij vraagt of de sneltrein gaat
she asks if the fast-train goes

(12) a. zij belooft (om) te werken
she promises comp to work

b. zij beweert te werken
she declares to work

SynR assumes reasonable default values for unknown words. For instance, it is assumed that the default complementizer for a finite sentential object is *dat* (cf. (11-a)) rather than *of*, because this is the most frequent form. As a result, only exceptional words have to be added to the Base lexicon.

6.2.3.3 Adding the structure of compound words

For the purposes of pitch accent placement, the internal structure of compound words with two pitch accents ('even stress') needs to be known, as will be explained in Section 7.3.1. Moreover, the particle and the verbal part of a compound verb must be accessible for syntactic realization. Some examples are given in (13).

(13) a. APS: (N:Utrecht CS)
Syntax: (N<prop>-(N-Utrecht)(N-centraal))
String: Utrecht Centraal
Utrecht Central

b. APS: (V:aankomen)
Syntax: (V<zijn,pred>-(P-aan)(V-komen))
String: aankomen
arrive

c. APS: (A:donkerbruin)
Syntax: (A-(A-donker)(A-bruin))
String: donkerbruin
dark-brown

d. APS: (ADV:altijd)
Syntax: (ADV<lrr>-(ADV-al)(N-tijd))
String: altijd
always

The feature lrr in (13-d) is added to trigger the Lexical Rhythm Rule during pitch accent placement. This is a deaccentuation rule that, depending on the context, deletes one of the accents on a word like *altijd*, which belongs to a lexically marked class of exceptions. This will be further discussed in Section 7.3.4).

6.2.3.4 Expanding numbers and times

In order to be pronounced, Numerals have to be expanded to phrases first. Some examples are given in (14). There is quite some irregular behaviour in the mapping. Although we will not go into the implementation, all of it can be conveniently captured in an AGFL. Notice that two digit numbers may be expressed by a single word (14-b,d). How much structure is generated is determined by how much is needed for the purpose of SLG, to be precise, for accent placement. In (14-e), for instance, Base generates the internal structure of the compound, because it may carry two accents. In contrast, no internal structure is generated in (14-c), even though *vijftien* is a compound from a morphological point of view. The same goes *eenen* in (14-e).

(14) a. (NUM:1) ⟶ (NUMP-(NUM-een))

b. (NUM:11) ⟶ (NUMP-(NUM-elf))

c. (NUM:15) ⟶ (NUMP-(NUM-vijftien))

d. (NUM:50) ⟶ (NUMP-(NUM-vijftig))

e. (NUM:51) ⟶ (NUMP-(NUM-(NUM-eenen)(NUM-vijftig)))

A similar concern regarding structure generation is visible in the expansion of larger numbers. In (15-a), the feature urr is introduced, which will later play a role during pitch accent placement, and is added to *honderd* 'hundred', *duizend* 'thousand', *miljoen* 'million', etc. (15-b) shows that *twee* is taken to be a premodifier of *honderd*, comparable to an adjective. However, *honderd* and *vijftig* are assumed to be conjuncts in a coordinated numeral phrase in (15-c), because a conjunction can be inserted between them: we can say *honderd en vijftig*.[7]

(15) a. (NUM:100) ⟶ (NUMP-(NUM<urr>-honderd))

b. (NUM:200) ⟶
(NUMP-(NUMP-(NUM-twee))(NUM<urr>-honderd))

c. (NUM:150) ⟶
(NUMP<conj>-(NUMP-(NUM<urr>-honderd))
(NUMP-(NUM-vijftig)))

[7] The same analysis might be applied to (14-e): *een en vijftig*. However, there are some differences. First, the conjunction cannot be omitted. Second, the conjunction is subject to resyllabification, and reduction to schwa: /e-nən-vɛif-təx/. Applying the same phonological processes to (15-c) makes it sound Belgian Dutch rather than Standard Dutch: /hɔn-dər-tən-vɛif-təx/. Our phonological phrasing algorithm (Section 3.3.3.1) predicts a φ-boundary after *honderd* in (15-c) (but not in (14-e)). Since the domain for resyllabification is assumed to be the Phonological Word, a φ-boundary blocks resyllabification. Thus, the observed difference is in accordance with our predictions.

d. (NUM:151) $\longrightarrow$
(NUMP<conj>-(NUMP-(NUM<urr>-honderd))
(NUMP-(NUM-(NUM-eenen)(NUM-vijftig))))

e. (NUM:251) $\longrightarrow$
(NUMP<conj>-(NUMP-(NUMP-(NUM-twee))
(NUM<urr>-honderd))(NUMP-(NUM-(NUM-eenen)
(NUM-vijftig))))

A drawback of the above APS format for numerals is that we cannot indicate a non-default focus distribution, which is required in counting. Consider the series in (16).

(16)	twintig,	eenentwintig,	tweeentwintig,	drieentwintig,	...
	twenty	one-and-twenty	two-and-twenty	three-and-twenty	
	'20, 21, 22, 23, ...'				

Only the first occurrence of *twintig* is accented, presumably because it is unfocused (given/predictable) in the subsequent occurrences. To accommodate this fact, there is a more elaborate way to specify numbers in an APS using the feature `back` to indicate that a part of the numeral is backgrounded.

(17) (NUM:(D:2<back>)(D:1)) $\longrightarrow$
(NUMP-(NUM-(NUM-eenen)(NUM<back>-twintig)))

All APS's for numerals can be expressed in this format; the shorter format introduced earlier is actually only an abbreviation for this more elaborate specification when all parts are focused.

However, this allows some rather puzzling APS's. Should there be any difference in the realization of the following pair?

(18) a. (NUM:(D:1)(D:1<back>)) $\longrightarrow$ (NUMP-(NUM<back>-elf))

b. (NUM:(D:1<back>)(D:1)) $\longrightarrow$ (NUMP-(NUM<back>-elf))

Notice that although the focus distribution in the APS's is different, both result in exactly the same output, and there is no syntactic or prosodic way to express the difference in focus, if there is any. For practical purposes, we assume that both APS result in a backgrounded Numeral.

Finally, Base also expands expressions of time (19). Their syntactic structure naturally fits NP's on the assumption that (19-b) is a reduced form of *acht uur en dertig minuten* 'eight hours and thirty minutes'.

(19) a. APS: (TIME:0800)
Syntax: (NP<time>-(NUMP-(NUM-acht))(N-uur)))
String: acht uur
eight hour

b. APS: (TIME:0830) ⟶
Syntax: (NP<time,conj>-(NP-(NUMP-(NUM-acht))
(N-uur))(NP-(NUMP-(NUM-dertig))))
String: acht uur dertig
eight hour thirty

6.3 The **Trans** component

6.3.1 GRAMTSY

GRAMTSY (*GRAMmaticaal Transformationeel Systeem* 'Grammatical Transformational System') is a program to transform trees by means of transformational rules (Coppen 1990; Coppen 1991). The trees are labelled-bracket structures enhanced with feature structures (cf. (8)). An example of a transformational rule is the following.

```
*--------------------------------------------------------------*
%Rule:      Subject-verb-agreement
--> SD:     ... (VP (NP<subj>1 ... ) ... # (V<fin> === ) # )
--> COND:   f2 = (f1 intersection "<sing,plur>")
--> SC:     #1 #2 ==>
            #1
            #2<&f2>
*--------------------------------------------------------------*
```

The name of this rule is *Subject-verb-agreement*. Its *Structural Description* (SD) describes a node pattern: a VP that contains a subject NP and an finite V. The symbol ... matches an arbitrary substring, thus (NP ...) is the pattern for a NP containing any number of nodes.[8] The symbol === stands for a lemma, thus (V<fin> ===) describes a terminal V. The number 1 after the feature bundle subj causes the NP's features – including subj – to be assigned to the variable f1. Its intersection with the features sing and plur is assigned to the variable f2 in the *Condition* part (COND). If the SC matches the input and the COND succeeds, the *Structural Change* (SC) is applied. The SC refers to the tab strings in the SD delimited by hash symbols. The symbol #1 refers to the first tab string, i.e. ... (VP (NP<subj>1 ...) ...; #2 to the second, i.e. (V<fin> ===). The change involves adding the value of f2 to the second tab.

The effect of this rule is that the features for number occurring on the subject NP are copied to the finite verb. Applying the rule to (20-a) gives (20-b).

(20) a. (S-(VP-(NP<subj,sing>-(N-u))
(NP-(DET-de)(N-trein))(V<fin>-nemen)))

[8] GRAMTSY balances the brackets in an SD.

b. (S-(VP-(NP<subj,sing>-(N-u))
(NP-(DET-de)(N-trein))(V<fin,sing>-nemen)))

The next rule moves the verb from within the VP to become a direct daughter of the S node. Applying it to (20-b), we obtain (21).

```
*--------------------------------------------------------------*
%Rule:       Verb-second
--> SD:      (S # (VP ... # (V === ) # )
--> COND:    None
--> SC:      #1 #2 #3 ==>
             #1
             #3
             #2
*---------------------------------------------------------------*
```

(21) (S-(V<fin,sing>-nemen)(VP-(NP<subj,sing>-(N-u))
(NP-(DET-de)(N-trein))))

For each rule, we can specify whether it must be applied once or multiple times, whether it is optional or obligatory, and whether it may be applied onto its own output. Sets of rules can be collected in grammars and subgrammars, which can be applied cyclically, e.g. to S nodes, in a top-down or bottom-up manner.

The above rules are for expository reasons only, and show only the basic operations in GRAMTSY. The actual rules of Trans are considerably more complex, and we will not describe them in detail. However, it will be clear from this short introduction that GRAMTSY is very suitable for enforcing grammatical constraints and determining word order.

6.3.2 Focus distribution

The focus distribution of an utterance depends on its purpose and context (Section 2.5.1). Since the text planner in an NLG system has access to the communicative goals, the discourse history, and the user model, we assume that the focus distribution is determined by PlanR: it assigns the feature `foc` to every focused word in an APS. However, a default assumption regarding focus is used to reduce the size of APS's: by default, content words are focused, but function words are not. This default is implemented in Trans. As a result, PlanR needs to assigns only non-default focus using the features `foc` and `back`.

So far, only words are focused or backgrounded. However, to apply the accentuation rule SAAR in ProsR, it must be known whether an argument – or rather a syntactic phrase functioning as an argument – is focused. For this purpose, Trans implements the focus projection rules to be described in Section 7.3.1.2. Their effect can be summarised roughly as: a major phrase is focused if its head is a focused content word.

6.3.3 Linking arguments to syntactic functions

The grammatical function of subject is defined by both word order and word form phenomena. In a main clause, the subject occurs adjacent to the finite verb[9]; in a subordinate clause, directly after the complementizer. Word order will be described in Section 6.3.7. In addition, a finite verb must agree with its subject for number and person. This subject-verb agreement will be discussed in Section 6.3.6. Finally, the subject carries nominative case, which surfaces if it is a pronoun. Insertion of the pronoun with the right grammatical case belongs to the tasks of Flex, and will be described in Section 6.4. Here we will focus on the question how the subject is determined.

The argument structure of a main verb consists of up to three arguments, and is encoded in the lexicon. Following modern syntactic theories (Jackendoff 1990; Grimshaw 1990, ch. 11), we assume that the arguments are *linked* to syntactic functions. By default, the first argument is linked to the subject, i.e. it receives the feature `subj` triggering the corresponding word form and word order. Analogously, it is assumed that the second argument is linked to the grammatical function of direct-object when only two arguments are available, whereas the second argument is linked to the indirect-object, and the third to that of direct-object when three arguments are available. It will be clear that the mutual order of a verb's arguments in an APS is significant, because it determines the default mapping from arguments onto syntactic functions. However, certain grammatical alternations like passive voice or ergative may force another argument to be linked as subject, or even prevent subject linking.

6.3.3.1 Voice

Trans is capable of realizing grammatical voice, i.e. the alternation between the active (22) and the passive form (23). Passive formation is triggered by the feature `pas` on the verb. As a result, the first argument is degraded to a non-argument PP with preposition *door* 'by', whereas its second argument is linked to the subject function. The feature `pas` also triggers the insertion of a passive auxiliary by Flex (Section 6.3.4).

(22) APS: `(V:kweken[tuinder<plur>][plant<def>])`

String:	tuinders	kweken	de	plant
	market-gardeners	grow	the	plant

(23) APS: `(V:kweken<pas>[tuinder<plur>][plant<def>])`

String:	de	plant	wordt	door	tuinders	gekweekt
	the	plant	is	by	market-gardeners	grown

If the first argument has no lemma or features, the *door*-PP is omitted (24). Notice, however, that the empty argument must be present in the APS anyway,

[9]There are a few exceptions, which we will ignore here. See Haeseryn et al. (1997, Section 21.4).

because otherwise the *second* argument cannot be identified as such.

(24)	APS:	(V:kweken<pas>[][plant<def>])
	String:	de plant wordt gekweekt
		the plant is grown

6.3.3.2 Dative shift

Another alternation, often referred to as *dative shift*, involves switching around the second and third arguments. It is triggered by the feature ds, and causes the second argument to be transformed to a PP with default preposition *aan* 'to'[10].

(25)	APS:	(V:geven[nerven<def,plur>][blad<def>] [kleur(rood)])
	String:	de nerven geven het blad een rode kleur
		the veins give the leaf a red colour
(26)	APS:	(V:geven<ds>[nerven<def,plur>][blad<def>] [kleur(rood)])
	String:	de nerven geven een rode kleur aan het blad
		the veins give a red colour to the leaf

6.3.3.3 Imperative and non-finite clauses

The imperative has normally no overt subject (27). In terms of linking theory, this may be interpreted as the first argument not being linked to a grammatical function, but staying implicit in the interpretation of the clause. Under this analysis, the imperative is similar to a passive without a *door*-PP. In both cases, the first argument remains unlinked and the verb takes an appropriate conjugation. The difference is that the imperative has no auxiliary support, and the first argument cannot surface as a PP.[11]

(27)	APS:	(kweken<imp>[][plant<def>](P:in [aarde(A:vochtig)]))
	String:	kweek de plant in vochtige aarde
		grow the plant in moist soil

Non-finite clauses have no overt subject either (28). Whether this is because it is a phonetically empty pronoun (PRO) as in transformational grammar (Haegeman 1991, ch. 5), or because it is directly linked to an argument of the matrix clause as in HPSG (Pollard and Sag 1994) is not relevant with respect to generation. That is, the APS in (28-a) does not have to express the fact that the first argument of *proberen* and the first argument of *kweken* refer to the same person.

[10] If the verb carries the feature voor, the preposition *voor* 'for' is used.

[11] Although sometimes the subject may be linked as well, as in *Wees jij maar stil*, literally 'Be you better quiet' (You better keep quiet).

(28) APS: `(V:proberen[<speak>][V:kweken[][plant<def>]])`
String: ik probeer de plant te kweken
I try the plant to grow
'I try to grow the plant'

Of course, this coreference is an important fact during the *construction* of an APS, as we want to express the intended meaning. In (29), for example, the first argument of *kweken* is coreferential with the *second* argument in the matrix clause, i.e. it is the listener rather than the speaker who has to grow the plant. This is thus a lexical property of the controlling verb (*proberen* versus *adviseren*) that important during the construction of an APS, but is no longer relevant during the syntactic realization of the APS. Since the construction of APS's is a matter of lexical choice and aggregation, and therefore a part of the microplanning process, it is beyond the scope of the current work.

(29) APS: `(V:adviseren[<speak>][<listen,red>]`
`[V:kweken[][plant<def>]])`
String: ik adviseer je de plant te kweken
I advise you the plant to grow
'I advise you to grow the plant'

Therefore, Trans treats non-finite clauses in a simple yet adequate way. If it detects an embedded clause of which the matrix clause carries the feature `te` (added by Base), it deletes the first argument in the embedded clause.

6.3.4 Auxiliaries

The eight tenses of Dutch described in Haeseryn et al. (1997, p. 113) can be specified in terms of the features `past`, `fut` (future), and `perf` (perfect). Trans inserts the lemmas of the appropriate auxiliaries, a combination of *zijn* 'be', *hebben* 'have', and *zullen* 'will', before the content verb. Conjugation of auxiliaries and the main verb – if required – is left to Flex.[12]

(30) APS: `(V:werken<past>[<masc>])`
String: hij werkte
he worked

(31) APS: `(V:werken<perf,past,fut>[<masc>])`
String: hij zou hebben gewerkt
he would have worked

Besides auxiliaries of tense, the auxiliary *worden* is inserted in the case of a passive.[13]

[12] An alternative realization of 2 is *hij had zullen werken*. As there is no apparent difference in meaning, only the first variant is generated

[13] Unless perfect tense is specified, in which case *zijn* functions as the auxiliary of both passive voice and perfect tense. For example, *de plant is gekweekt* instead of **de plant is gekweekt geworden*.

The word *te* 'to' occurs in the verb cluster of non-finite clause, and has a certain similarity to an auxiliary. It is therefore inserted by the same set of rules that insert auxiliaries.[14]

6.3.5 Sentence projection

Recall that Base expands every V in an APS to a VP. Depending on its context, Trans may in turn projects the VP to a sentence. There are three types of sentences: root, bound, and relative. If used adjectivally, the VP is not projected to an S.

Root sentence If the VP is not embedded in another phrase, Trans assumes that a root clause must be generated, and inserts an S node directly dominating the VP node.[15] The word order of a root sentences will be discussed in Section 6.3.7.

Bound sentence If the VP is directly dominated by another VP, Trans generates a bound sentence by inserting an S node with feature `bound`. It checks whether the VP contains a C node, which stems from the explicit specification of a complementizer in the APS. If so, this C node becomes the leftmost daughter of the S node. Otherwise, Trans chooses a default complementizer according to the following logic.[16]

- If the verb of the dominating VP has the feature `noom` and the verb of the embedded VP has the feature `inf`, no complementizer is inserted (cf. example (12-b)).
- If the verb of the dominating VP lacks the feature `noom` and the verb of the embedded VP has the feature `inf`, the complementizer *om* 'to' is inserted (cf. example (12-a)).
- If the verb of the dominating VP has the feature `of` and the verb of the embedded VP lacks the feature `inf`, the complementizer *of* 'if/whether' is inserted (cf. example (11-b)).
- If the verb of the dominating VP lacks the feature `of` and the verb of the embedded VP lacks the feature `inf`, the complementizer *dat* 'that' is inserted (cf. example (11-a)).

The features `noom` ('no *om* complementizer') and `of` are lexically specified and added by Base (cf. Section 6.2.3.2).

[14] Modal auxiliaries may be accounted for in the same way as auxiliaries of tense, which means that they are inserted on the basis of a set of modality features. The same approach may be explored for aspect. This is not yet implemented in SynR.

[15] This default can be overruled by adding the feature `bound` to the VP's verb.

[16] In anticipation of a feature-based choice of all complementizers, Trans adds only the features for a certain complementizer, and leaves the insertion of the word form to Flex (Section 6.4.2)

Relative sentences If the VP is directly dominated by NP and its V does not bear the feature `part`, Trans generates a relative sentence by inserting an S node with feature `rel`. Insertion of relative determiners will be discussed in Section 6.3.7.2.

Adjectivally used verbs There is a strong parallel between the adjective *verzamelend(e)* 'collected' in the (32) examples and the verb *verzamelen* 'collect' in the relative clause examples in (33). The former is often regarded as an adjective, because the suffix *-e* can be attached to it. However, the unsuffixed form is identical to the present participle (*verzamelend*) or past participle (*verzameld*) of the corresponding verb.[17] This suggests that the participle form of most verbs can also be used as an adjective.

The feature `part` (participle) on the V's in (32) prevents projection of their VP's to sentences. By implication, they are not moved to the right of the NP's head, as only S's and PP's are (cf. Section 6.3.7.5). The feature also triggers insertion of the participle form by Flex. Attachment of the suffix *-e* follows from the same rules that account for the suffixation of real adjectives (see Section 6.3.6). As a result, the feature `part` triggers an alternation between verbs used relatively and adjectivally akin to the way `pas` triggers the active-passive alternation. As matter of fact, the two can be used simultaneously, as (32-c-d) shows.[18]

(32) a. APS: (man<def>(V:verzamelen<part>[] [plant<plur>]))
String: de planten verzamelende man
the plants collecting man

b. APS: (man<def>(V:verzamelen<part>[][]))
String: de verzamelende man
the collecting man

c. APS: (plant<def,plur>(V:verzamelen<part,pas> [man<def>][]))
String: de door de man verzamelde planten
the by the man collected plants

d. APS: (plant<def,plur>(V:verzamelen<part,pas> [][]))
String: de verzamelde planten
the collected plants

---

[17] The present participle occurs as a real verb in only a few constructions (Haeseryn et al. 1997, p. 107). The past participle is used in the passive voice and perfect tense.

[18] In the APS's, empty arguments at the right can be omitted. Thus, (plant<plur>(V:verzamelen<part,pas>[][])) can be shortened to (plant<plur>(V:verzamelen<part,pas>)).

(33) a. APS: (man<def>(V:verzamelen[][plant<plur>]))
String: de man die planten verzamelt
the man that plants collect
'the man that collects plants'

b. APS: (man<def>(V:verzamelen[][]))
String: de man die verzamelt
the man that collects

c. APS: (plant<def,plur>(V:verzamelen<pas> [man<def>][]))
String: de planten die door de man worden verzameld
the plants that by the man are collected

d. APS: (plant<def,plur>(V:verzamelen<pas>[][]))
String: de planten die worden verzameld
the plants that are collected

A problem with this approach is that some verbs do not allow a passive (34-a), but can nevertheless be used adjectivally (34-b)

(34) a. *In Amsterdam wordt aangekomen (door de trein)
In Amsterdam is arrived (by the train)

b. de aangekomen trein
the arrived train

To accommodate for these cases, we assume that (34-b) is related to the perfect tense form in (35).

(35) de trein is aangekomen
de train has arrived

With respect to an APS, this means that `part` can be combined with `perf`.

(36) APS: (trein<def>(V:aankomen<part,perf>[][]))
String: de aangekomen trein
the arrived train

The corresponding relative form is (37).

(37) APS: (trein<def>(V:aankomen<perf>[][]))
String: de trein die is aangekomen
the train that has arrived

It follows that <part,perf> can also be used with a verb that does allow a passive. In this case, however, the parallel between adjectivally and relatively used verbs no longer holds: the adjective form in (38) clearly means something different than the relative form in (39). We leave this exception for future research.

(38) APS: (man<def>(V:verzamelen<part,perf>[][]))
String: de verzamelde man
the collected man

(39) APS: (man<def>(V:verzamelen<perf>[][]))
String: de man die heeft verzameld
the man that has collected

6.3.6 Agreement

We will refer to the following (and similar) phenomena as *agreement*.

- *Determiner-noun agreement*
  In (40-a), the noun *gebied* is definite-plural, and therefore gets the determiner *de*. In (40-b), it is definite-singular-neuter, and therefore the determiner is *het*. In (40-c), it is indefinite-singular, and the determiner is *een*.

- *Adjective-noun agreement*
  In (40-b), the noun is indefinite-singular-neuter, and therefore the adjectives *groot* and *droog* receive no suffix. For all other combinations of these grammatical properties, the the suffix *-e* is attached (cf. (40-a,c)).[19]

- *Noun-relative determiner agreement*
  A similar relation exists between the noun and the demonstrative determiner introducing the relative clause, which has traditionally been called a 'relative pronoun'. Singular neuter referents take *dat* (40-b,c), all others take *die* (40-a).

- *Subject-verb agreement*
  Finally, the verb *liggen* is plural in (40-a), because its subject *die* is plural, but singular in (40-b,c), because *dat* is singular. Notice that since the subject of the relative clause depends on the noun, it mediates the relation between the noun and the verb.

(40) a. de grote en droge gebieden die onder een bergtop
the large and dry areas that under a mountain-top
liggen
lie

b. het grote en droge gebied dat onder een bergtop ligt
the large and dry area that under a mountain-top lies

[19] This is an oversimplification, as NP's like *een waarnemend minister* 'the acting minister' show. For an extensive discussion, see Haeseryn et al. (1997, ch. 6.4).

c.	een	groot	en	droog	gebied	dat	onder	een	bergtop	ligt
	a	large	and	dry	area	that	under	a	mountain-top	lies

It belongs to the tasks of Trans to enforce agreement of features between nodes, whereas the corresponding insertion of function words (e.g *de*) and inflection of content words (e.g. *ligt*) is the responsibility of Flex.

The APS for (40-a) is (41), and the corresponding tree generated by Base is given in the first part of Figure 6.2. Notice that Base adds `neut` to the features of *gebied*.

(41)
```
(gebied<def,plur>(A:{(A:groot)(A:droog)})
(V:liggen[<top>][P:onder[bergtop]]))
```

The corresponding tree generated by Trans is given in the second part of Figure 6.2. For the time being, we can ignore the addition of an S-node and the change in word order; we focus on the features here. The noun *gebied* has a feature `infl`, which indicates that its grammatical properties (`<def,neut,plur>`) require adjectives to be suffixed. This feature is copied onto its AP premodifier. This happens to be a coordinated structure, and the feature is therefore copied to each of its conjuncts. Finally, it is copied from each AP to its head A.

In a similar way, `<def,neut,plur>` is copied onto the first determiner, `<neut,plur>` is copied onto the second determiner, and `plur` onto the verb. The features resulting from intermediary steps were omitted for expository reasons. Since all features of the noun *bergtop* are default (i.e. singular, indefinite, non-neuter), there are no features to copy to its determiner either.

The enforcement of agreement is implemented as a grammar of transformational rules which traverse the tree in a top-down manner. The rules copy the relevant features from one node onto the next. They form an intricate system that accounts for agreement in all syntactic configurations that SynR can generate, including coordinated structures. Discussing their implementation is beyond the scope of the current work.

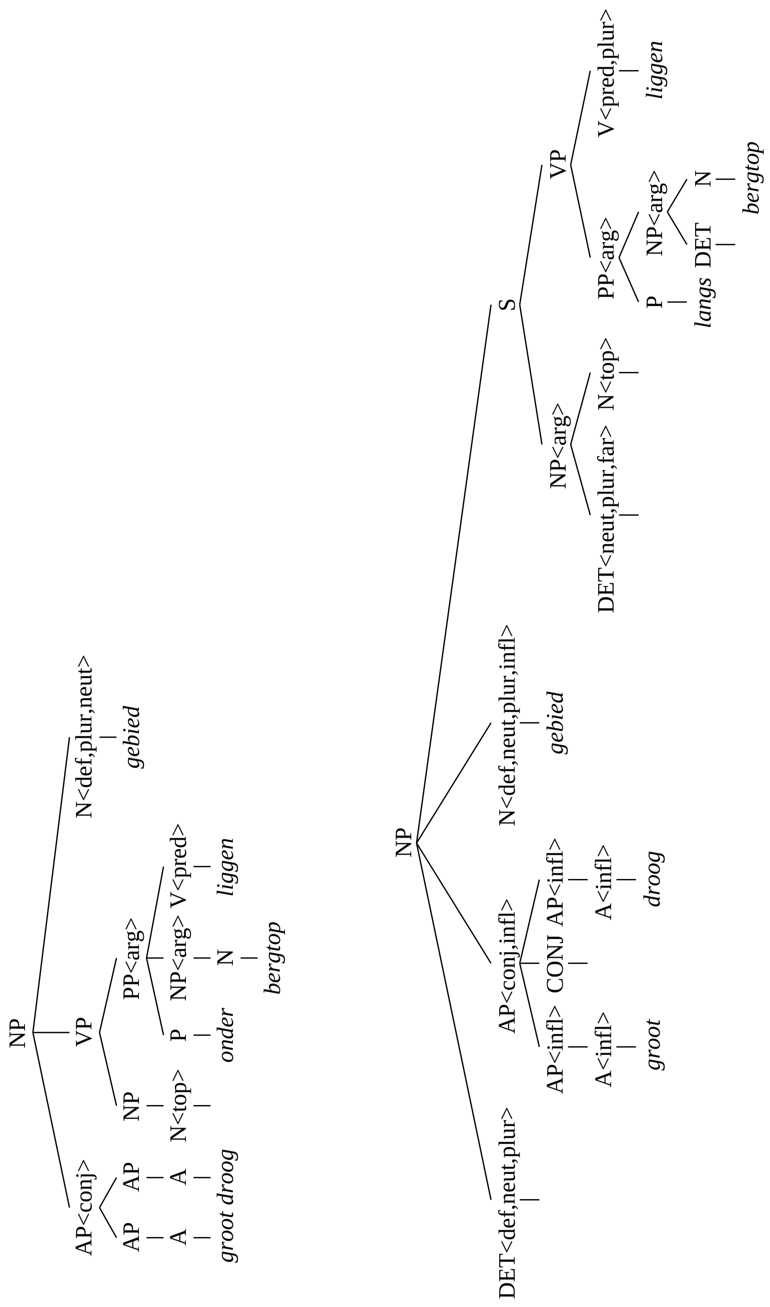


Figure 6.2: Output trees of Base *(top) and* Trans *(bottom) corresponding to the APS in (41) (irrelevant features have been suppressed).*

6.3.7 Word order

6.3.7.1 Verb second

The difference between a root clause and a bound clause is – among other things – reflected in the word order: the former requires movement of the finite verb to the second position in the clause, the so-called *verb second* (V2) transformation (a simplified implementation of this rule was given in Section 6.3.1). In terms of Trans: if an S node has no feature `bound` or `rel`, the leftmost verb in the VP is moved to become the leftmost daughter of the S node (42-a).[20]. In a bound or relative clause, the base position of a finite verb is unaffected (42-b).

(42) a. APS: `(V:nemen<perf>[<speak>][trein<def>])`
Syntax: $(_S$ $(_{NP}$ ik) $(_V$ heb) $(_{VP}$ de trein genomen))
I have the train taken

b. APS: `(V:nemen<perf>[<speak>][trein<def>])`
Syntax: $(_S$ $(_C$ dat) $(_{VP}$ ik de trein $(_V$ heb) genomen
that I the train have taken
))

6.3.7.2 Topicalisation

A root clause has also a grammatical mood, which is partly reflected in the word order. The default mood is declarative. It requires movement of one the VP's phrasal daughters to the first position of the sentence, which is known as *topicalisation*. By default, the subject will be topicalised (43-a), unless another constituent is explicitly marked with the feature `top` (43-b).

(43) a. APS: `(V:nemen[<listen>][trein<def>])`
String: u neemt de trein
you take the train

b. APS: `(V:nemen[<listen>][trein<def,top>])`
String: de trein neemt u
the train take you

Interestingly, recent syntactic theories claim that topicalisation of a non-subject is different from the fronting of a subject (Zwart 1993); see also Coppen (1991, p. 109) and van Eynde (1997). For instance, only subject clitics (reduced personal pronouns) can appear in fronted positions (44).

(44) a. je hebt me gezien
you have me seen

b. *je heb ik gezien
you have I seen

[20] Movement of *ik* is due to topicalisation; cf. Section 6.3.7.2

There are two other moods that are indicated by a feature on the main verb: `imp` 'imperative' (45-a), and `polar` 'polar question'/'yes-no question' (45-b). Neither of them allows topicalisation. A wh-question – often considered a mood as well (Halliday 1994) – is indicated by the feature `wh` on one of the VP's constituents (45-c-d), and triggers its topicalisation.[21]

(45) a. APS: `(V:nemen<imp>[][trein<def>])`
String: neem de trein
take the train

b. APS: `(V:nemen<polar>[<listen>][trein<def>])`
String: neemt u de trein
take you the train
'do you take the train?'

c. APS: `(V:nemen[<ani,wh>][trein<def>])`
String: wie neemt de trein
who takes the train

d. APS: `(V:nemen[<listen>][<wh>])`
String: wat neemt u
what take you
'what do you take'

Topicalisation also applies to relative clauses, where `top` marks the constituent bound by the noun of the matrix phrase.

(46) a. APS: `(man<def>(V:nemen[][trein<def>]))`
String: de man die de trein neemt
the man that the train takes
'the man who takes the train'

b. APS: `(trein<def>(V:nemen[man<def>][<top>]))`
String: de trein die de man neemt
the train that the man takes

c. APS: `(dag<def>(V:nemen[man<def>][trein<def>]`
`(P:op[<wh>])))`
String: de dag waarop de man de trein neemt
the day at-which the man the train takes
'the day at which the man takes the train'

6.3.7.3 Extraposition

Extraposition is rightward movement of a VP-internal constituent so that it becomes the rightmost daughter of the dominating S. It is normally restricted to PP's and S's (either a bound S or a non-restrictive relative S). Extraposition

[21] This is an oversimplification, as there are also forms like *Weet jij wie de trein neemt?* 'Do you know who takes the train?'.

of a PP is not obligatory, but is preferred if it is 'heavy'. We have made this operational in the following way: a PP is heavy if it contains another PP or an S. Alternatively, extraposition can be forced or blocked by means of the features `heavy` and `light` respectively. Of course, a topicalised PP cannot be extraposed.

Extraposition of S's is the default (47-a), but may be blocked for very light clauses (47-b).

(47) a. APS: `(V:nemen<polar,perf>[<listen>]`
`[stoptrein<def>](V:komen(P:in[Amsterdam])`
`(C:om)))`
String: heeft u de stoptrein genomen om in
have you the slow-train taken to in
Amsterdam te komen
Amsterdam to get
'did you take the slow train to get to Amsterdam?'

b. APS: `(V:nemen[<listen>][trein<def>(P:naar`
`[Amsterdam])](V:beginnen<light>[]`
`(C:om)))`
String: u neemt om te beginnen de stoptrein naar
you take `comp` to begin the slow-train to
Amsterdam
Amsterdam
'to begin with, take the slow train to Amsterdam'

An NP may be extraposed only if it is extremely heavy, a transformation that is known as *heavy NP shift*. In addition, extraposition can be applied to deeper embedded phrases, in particular to attributively used heavy PP's and to non-restrictive relative clauses.[22] In all these cases, extraposition is applied only if the APS explicitly specifies the feature `heavy`.

6.3.7.4 Scrambling

Determining the order of the remaining VP daughters is known as *scrambling* in transformational terms. This is a notoriously difficult problem, since the order is affected by many, sometimes contradictory requirements. We follow Haeseryn et al. (1997, ch. 21) and Verhagen (1986) in assuming that the word order is affected by syntactic category, syntactic function and focus. There are certainly other factors involved, particularly in the more marked word orders, but these three appear to be the most important. The implementation consists of transformational rules that move constituents until they fit the pattern in (48).

[22] In some cases, however, even restrictive relative clauses may be extraposed: *Ik heb alleen de trein gevonden die naar Arnhem gaat* 'I have only the train found that goes to Arnhem'.

(48) **Word order of VP daughters**

subject
unfocused indirect-NP-object
unfocused direct-NP-object
unfocused indirect-PP-object
unfocused modifiers*
focused modifiers*
focused indirect-NP-object
focused direct-object
focused indirect-PP-object
verbal cluster

Some examples are given in (49).

(49) a. APS: `(V:geven[nerf<plur,def>][][<far>] (ADV:indirect)(C:dat))`
String: dat de nerven het dat indirect geven
that the veins it that indirectly give

b. APS: `(V:geven<ds>[nerf<plur,def>][][<far>] (ADV:indirect)(C:dat))`
String: dat de nerven dat daaraan indirect geven
that the veins it to-that indirectly give

c. APS: `(V:geven[nerf<plur,def>][][kleur] (ADV:indirect)(C:dat))`
String: dat de nerven het indirect een kleur geven
that the veins it indirectly a colour give

d. APS: `(V:geven<ds>[nerf<plur,def>][blad] [<far>](ADV:indirect)(C:dat))`
String: dat de nerven dat indirect aan een blad
that the veins it indirectly a colour give
geven

e. APS: `(V:geven[nerf<plur,def>][blad][kleur] (ADV:indirect)(C:dat))`
String: dat de nerven indirect een blad een kleur
that the veins indirectly a leaf a colour
geven
give

f. APS: `(V:geven<ds>[nerf<plur,def>][blad][kleur] (ADV:indirect)(C:dat))`
String: dat de nerven indirect een kleur aan een blad
that the veins indirectly a colour to a leaf
geven
give

Although this simple approach works well for our application domains, there is room for improvement. For example, (50) is a more natural variant of (49-b) which we cannot account for.

(50) dat de nerven dat daar indirect aan geven
that the veins it that indirectly to give

Clitics violate the word order pattern in (48), because the direct object appears *before* the indirect object.

(51) ik heb 't 'm gegeven
I have it him given
'I have given it to him'

For another interesting problem, consider the pair in (52). In (52-a), the possessive determiner *zijn* 'its/his' refers to the preceding indirect object *een blad*. However, if *een blad* follows *zijn* as a result of dative shift, the coreference is broken. This is an example of what Postal (1971) called *crossover phenomena*. Apparently, some form of binding theory is required here to account for the interaction between word order and the interpretation of anaphora.

(52) a. dat de nerven indirect een blad zijn kleur geven
that the veins indirectly a leave its colour give
'that the veins indirectly give a leave its colour'

b. ?dat de nerven indirect zijn kleur aan een blad geven
that the nerves indirectly its colour to the leave give

The placement of particles and predicative elements, which appears to be strongly related to the verb cluster (van Dreumel 2000), has not been addressed in SynR either.

Correctly predicting the word order in complex VP's is probably the ultimate challenge for any syntactic realizer for Dutch.

6.3.7.5 Word order in NP's

Compared to the variation in S's and VP's, the word order in NP's is fairly rigid. Determiners, quantifiers, adjectives, verbs used adjectivally, and nominal premodifiers must all precede the head; attributive PP's and relative clauses must follow it. This is accomplished by a number of movement rules that we will not go into.

One interesting issue in relation to focus is the mutual order of adjectives, which appears to be determined by semantic criteria: the more 'inherent' (or 'objective' as opposed to 'subjective') the property expressed by an adjective, the closer the adjective occurs to the noun (van der Lubbe 1958). For instance, a material adjective is assumed to be more inherent than a colour adjective, which is in turn more inherent than a size adjective.

(53) grote groene houten stelen
big green wooden stalks

It is possible to deviate from this order if only one of the adjectives is focused, and therefore accented (Haeseryn et al. 1997, p. 837-840).

(54) gro*ene grote houten stelen
green big wooden stalks

In anticipation of a theoretical account of this matter, we assume that the microplanner determines the order of the adjectives. Therefore, SynR respects the order of the adjectives in the APS.

6.4 The Flex component

The Flex component is responsible for lexical and morphological realization: the insertion of function words and the inflection of content words. It maps the features of a terminal node, which have been collected in the previous two steps, onto an appropriate word. This is again accomplished by an AGFL. The idea is simple yet effective: we parse the syntactic structure generated by Trans and while parsing the terminal nodes, we use its features to transduce the node's content (a lemma or an empty string) to a word.

6.4.1 Inflection of content words

We will explain this process using the inflection of adjectives as an example. Adjectives marked with the feature `infl` normally receive the inflectional suffix *-e* (55-a). In some cases, this changes the spelling (55-b); in other cases (55-c), it affects the pronunciation of the stem as well (/xrɔf/ becomes /xrofə/). Moreover, some adjectives (e.g. most material adjectives) resist suffixation (55-d).

(55) a. groen ⟶ groene
green green+`infl`

b. groot ⟶ grote
big big+`infl`

c. grof ⟶ grove
coarse coarse+`infl`

d. houten ⟶ houten
wooden wooden+`infl`

In addition, adjectives can be marked for degree. We assume that an unmarked adjective is *positive*. The feature `cmp` marks a *comparative*, which normally receives the suffix *-er*, whereas `sup` marks a superlative, which is by default inflected with the suffix *-st* (56-a). However, deviations may occur (56-b,c) to the extent of complete irregularity (56-d).

(56) a. groen, groener, groenst
green, greener, greenest

b. groot, groter, grootst
big, bigger, biggest

c. duur, duurder, duurst
expensive, more expensive, most expensive

d. goed, beter, best
good, better, best

The problem to be solved by Flex is choosing the correct word form on the basis of an adjective's lemma and features.

Given the format of a GRAMTSY node (cf. (8)), it is not difficult to write a grammar that describes GRAMTSY output, and to generate a parser for it. The grammar rules for non-terminals are of no interest; their only function is for traversing the tree down to the terminals. It is the transduction rules for terminal nodes that do the real work. Figure 6.3 shows the rule for an Adjective terminal. It contains a call to another rule for parsing the feature set (AdjectiveFeatureSet), consisting of a list of comma-separated features between angled brackets, which is given in Figure 6.4. The key point to notice is that if relevant features are encountered, they are 'lifted' to the affix level. For instance, if the feature set contains infl, the value of the affix inflection becomes +infl. Likewise, if it contains cmp, the value of the affix degree becomes cmp. If no features are specified, the affixes remain unrestricted.[23]

Suppose the terminal node at hand is (A<infl>-groen). After its features have been parsed, inflection is restricted to +infl, and degree is still unrestricted. Next, the rule in Figure 6.3 attempts to parse the node's lemma by calling Adjective(+infl,degree). The rules for lemma parsing are given in Figure 6.5. For convenience, they are labelled 1a through 2c. According to the AGFL implementation, the rules are tried in the order in which they are given in the grammar. Rule 1a is not applicable, since it requires -infl. The left-hand side of rule 1b matches, but groen is not a member of any of its alternatives. Rule 1c matches, causing the subsequent call Adjective(degree). This matches rule 2a (degree is specialised to pos); the expression $MATCH("[a-z]*") is a regular expression that matches any string of lower case characters. The transduction returned by 2a is the original lemma groen. The transduction of rule 1c attaches suffix e, changing it to groene. This is in turn incorporated in the transduction of Figure 6.3, resulting in the string (A<infl>-groene). Thus, (A<infl>-groen) is transduced

[23] The feature set may contain other features, which are irrelevant for inflection. These are accounted for by a default rule not shown in Figure 6.4. Notice also that the order in which the features occur is irrelevant.

```
AdjectiveTerminalNode :
    "(",
        "A",
            AdjectiveFeatureSet(inflection,compare),
                "\-",
                    Adjective(inflection,compare),
                        ")" /
                            "(A",
                                AdjectiveFeatureSet,
                                    "-",
                                        Adjective,
                                            ")".
```

Figure 6.3: AGFL rule for transducing an adjective terminal node.

to (A<infl>-groene). Some other examples can be traced with the help of Table 6.1.

The same method is applied to inflect other content words:

- nouns are inflected for number (`plur`) and possession (`pos`)
- verbs are inflected for form (`fin`, `part`, or `imp`), number (`plur`), person (`speaker`, `listen`), tense (`past`), perfect (`perf`), and suffix (`infl`).[24]

Adverbs, which have no inflection, are simply ignored.

6.4.2 Insertion of function words

Insertion of function words is essentially the same as inflection, with the exception that there is usually no lemma. Flex inserts and inflects auxiliaries, complementizers, conjunctions, and pronominal adverbs. In addition, Flex inserts a wide and nearly complete range of determiners, including articles and those words that have traditionally been called 'pronouns' (Section 2.6.3). Insertion takes place according to features for number (`plur`), person (`speaker`, `listen`), case (`subj` 'subjective'), definiteness (`def`), neuter (`neut`), animacy and gender (`ani`, `fem`, and `masc`), formality (`inf` 'informal' and `red` 'reduced'), possession (`pos`), space (`near` and `far`), and 'wh-ness' (`wh`).

As explained earlier (Section 6.3.4), Trans inserts the lemmas of auxiliaries of tense and voice. Their inflection is left for Flex, and is accomplished in the same way as content verbs are inflected.

Insertion of complementizers is currently limited to those that are grammatically enforced and neutral in meaning, i.e *dat* 'that', *of* 'whether/if', and *om* (cf. Section 6.3.5). There are many more complementizers, such as those

[24] If used adjectivally; cf. Section 6.3.5

```
AdjectiveFeatureSet(inflection,compare) :
    "<",
       AdjectiveFeatureSequence(inflection,compare),
            ">" /
                "<",
                    AdjectiveFeatureSequence,
                        ">" ;
    .

AdjectiveFeatureSequence(inflection,compare) :
    AdjectiveFeature(inflection,compare),
        ",",
            AdjectiveFeatureSequence(inflection,compare) /
                AdjectiveFeature,
                    ",",
                        AdjectiveFeatureSequence ;
    AdjectiveFeature(inflection,compare) /
        AdjectiveFeature;
    .

AdjectiveFeature(+infl,compare) :
    "infl" /
        "infl".

AdjectiveFeature(inflection,cmp) :
    "cmp" /
        "cmp".

AdjectiveFeature(inflection,super) :
    "super" /
        "super".

# affix grammar
inflection :: +infl; -infl.
degree :: pos; cmp; sup.
```

Figure 6.4: AGFL rules for transducing an adjective's features.

```
# 1a
Adjective(-infl,degree) :
    Adjective(degree) /
        Adjective.

# 1b : irregular positive
Adjective(+infl,pos) :
    "groot" / "grote" ;
    "open" / "open" ;
    "roze" / "roze" ;
    ...

# 1c
Adjective(+infl,degree) :
    Adjective(degree) /
        Adjective,
            "e".

# 2a : regular positive
Adjective(pos) :
    $MATCH("[a-z]*") /
        $MATCH.

# 2b: comparative
Adjective(cmp) :
    "groot" / "groter" ;
    "donker" / "donkerder" ;
    "goed" / "beter" ;
    "kwaad" / "erger" ;
    ...
    $MATCH("[a-z]*") /
        $MATCH,
            "er".

# 2c: superlative
Adjective(sup) :
    "goed" / "best" ;
    "kwaad" / "ergst" ;
    ...
    $MATCH("[a-z]*") /
        $MATCH,
            "st".
```

Figure 6.5: AGFL rules for transducing an adjective's lemma, to its inflected form.

Table 6.1: Input-output samples of adjective inflection: Input *is a adjective terminal node;* Rules *are the rules in Figure 6.5 applied during the transduction;* Word *is the word form in the output node.*

Input:	*Rules:*	*Word:*
`(A<>-groen)`	1a & 2a	`groen`
`(A<infl>-groen)`	1c & 2a	`groene`
`(A<cmp>-groen)`	1a & 2b	`groener`
`(A<infl,sup>-groen)`	1c & 2c	`groenste`
`(A<infl>-groot)`	1b	`grote`
`(A<cmp>-groot)`	1a & 2b	`groter`
`(A<infl,sup>-groot)`	1c & 2c	`grootste`
`(A<>-goed)`	1a & 2a	`goed`
`(A<cmp>-goed)`	1a & 2b	`beter`
`(A<sup>-goed)`	1a & 2c	`best`

of time (*voordat* 'before'), of reason (*omdat* 'because'), of condition (*tenzij* 'unless'), of concession (*hoewel* 'although'), etc. Although rarely required in our application domains, these can be generated by explicitly including their lemma in an APS. However, if a suitable description of their meaning in terms of semantic and discourse features were available, this could easily be implemented in Flex.

The same goes for the insertion of conjunctions, which is currently limited to *en* and *of*. Since a description in terms of features is lacking, conjunctions other than these must be explicitly specified in the APS.

Finally, Flex also inserts the so-called pronominal adverbs: *er* 'there', *hier* 'here', *daar* 'there', *waar* 'where'. These can be a combined with a preposition: *door* 'through' plus *daar* 'there' becomes the adverb *daardoor* 'because of that' (lit. 'there-through'). In some cases, the combination changes the form of the preposition: *met* 'with' plus *daar* 'there' becomes *daarmee.* Discussing the implementation of these phenomena would lead us to far from the main topic. Suffice it to say that all of these are correctly generated by SynR.

6.5 Discussion

6.5.1 APS's versus templates

The easiest way to generate text is using string templates. In principle, we can generate travel descriptions by filling the slots of templates like (57) with travel parameters. All processing occurs at the string level, without any syntactic, semantic, or conceptual level of representation.

(57) `take the <conveyance> from <departure> to <arrival>`

Why then bother using abstract phrase specifications and a syntactic realizer? Below, we discuss three arguments in favour of using more abstract and general NLG techniques (Reiter 1995; Reiter and Dale 1997).

To begin with, the maintainability of template-based systems decreases if the required variation in the output increases. Since templates allow only limited abstraction and generalisation, generating many different texts requires adding many different templates. The resulting redundancy makes the system large and difficult to maintain. Altering the content or form of the output may involve changes at many points in the code, which is laborious and prone to error.

Another argument, which – to the best of our knowledge – has not yet been explicitly mentioned in the literature, is that a template-based realizer cannot provide the syntactic information that is required for prosodic realization. It is of course possible to annotate templates with prosodic mark-up codes, but this requires expert knowledge: not all programmers who implement string templates for grammatical expressions in his/her native language can confidently assign pitch accents and prosodic boundaries.

Finally, it is generally accepted that NLG systems using syntactic realizers can produce higher-quality output. This is not so much due to the fact that the output is 'more grammatical' than that accomplished with templates, but rather that the presence of an abstract semantic or syntactic level of representation enables the construction of better texts. To appreciate this point with respect to SynR, we will discuss the abstraction embodied in our APS's, and show how it facilitates microplanning.

As explained earlier, an APS is a representation that abstracts away from certain syntactic aspects, in particular, from (most) function words, the inflection of content words, and word order. From a somewhat different perspective, we can distinguish two types of abstraction. First, it abstracts away from purely syntactic phenomena that have no associated semantics. Examples of these are the form of a definite article (*de* or *het*), the difference in word order between root clauses and bounded clauses, agreement of a finite verb with the subject, and the suffixation of an attributive adjective. These may be considered formal properties of an utterance that facilitate its processing, but do not contribute to its meaning. Second, it abstracts away from syntactic alternations related to perspective and information structure. For instance, the difference between the active and passive form is no more than the feature `pas`. This represents a considerable abstraction of the process of linking the second argument to the subject, linking the first argument to a *by*-PP, inserting an auxiliary, and changing the main verb's conjugation.

These abstractions considerably simplify the task of microplanning, i.e. lexicalisation, referring expression generation, and aggregation. With respect to referring expression generation, the algorithm can concentrate on the question whether an anaphoric expression can be used, given the current discourse history; it does not need to be concerned with the grammatical case of a pronoun. Similarly, it can take definiteness decisions independently of the different syntactic realization of this property.

With respect to aggregation, it allows generalisations that cannot be expressed in terms of surface strings. Consider the example in (58), a standard example of aggregation by conjunction.

(58) a. boeren gebruikten de bladeren als thee
farmers used the leaves as tea

b. boeren gebruikten de bladeren als kleurstof
farmers used the leaves as colour

c. boeren gebruikten de bladeren als thee en als kleurstof
farmers used the leaves as tea and as colour

Suppose these are generated by instantiation of the template `x gebruikten y als z` 'x used y as z'. We can formulate the following quasi-formal aggregation rule to rewrite (58-a-b) to (58-c).

(59) Given (x_1 used y_1 as z_1) and (x_2 used y_2 as z_2),
if $x_1 = x_2$ and $y_1 = y_2$ then
replace with (x_1 used y_1 as z_1 and z_2)

The same type of aggregation occurs in (60).

(60) a. de bladeren werden door boeren als thee gebruikt
the leaves were by farmers as tea used

b. de bladeren werden door boeren als kleurstof gebruikt
the leaves were by farmers as colour used

c. de bladeren werden door boeren gebruikt als thee en als kleurstof
the leaves were by farmers used as tea and as colour

To account for this, however, we need another aggregation rule that matches the template `(x was used as y)`. Obviously, the two rules are closely related, but using string templates prevents the generalisation from being made.

This problem can be solved by increasing the abstraction of a phrase specification. The APS's corresponding to (58-a) and (60-a) are given in (61).[25]

[25]For the sake of the argument, we assume that *als* is a preposition here. It may as well be analysed as the complementizer of a clause with a gapped verb, i.e. *als thee zijnde* 'as tea being'.

(61) a. `(V:gebruiken<past>[boer<plur>][blad<plur,def>] [P:als[thee<nonspec>]])`

b. `(V:gebruiken<past,pas>[boer<plur>][blad<plur,def>] [P:als[thee<nonspec>]])`

At this level of abstraction, the only difference between them is the feature `pas`. The similarity of the APS's allows us to write a generalised aggregation rule.

(62) Given (V:gebruiken[x_1][y_1][z_1])
and (V:gebruiken[x_2][y_2][z_2]),
if $x_1 = x_2$ and $y_1 = y_2$ then
replace with (V:pred$_1$[x_1][y_1][{[z_1][z_2]}])

Admittedly, this rule can be improved by generalising over the predicate, and we can write another aggregation rule to factor out the preposition *als*. In fact, aggregation is also constrained by pragmatic and contextual factors (Elhadad, McKeown, and Robin 1997; Dalianis and Hovy 1993; Dalianis 1997; Dalianis and Hovy 1997; Shaw 1998). However, aggregation is not the topic of the current work; the point here is that the abstraction of an APS facilitates aggregation.

To sum up, it was argued that using APS's in combination with SynR provides significant advantages compared to string templates, which are essentially due to a higher level of abstraction. First, it is easier to maintain, especially when a system must be scaled up. Second, templates cannot provide the linguistic structure required for prosodic realization in SLG systems. Third, APS's facilitate microplanning – although not carried out in our systems – resulting in higher-quality output.

6.5.2 Syntactic realization and focus

The canonical word order in Dutch, as in many languages, is to have unfocused information precede focused information.[26] This relation between focus and word order is currently only exploited within the VP, but may be extended to clauses. Grammatical alternations like passive and dative shift may be interpreted as the means to obtain the preferred word order, and the same goes for transformations like topicalisation and extraposition. However, focus is clearly not the only reason for applying these operations. Passive, for instance, suggests abstraction and objectivity by suppressing the role of Agent, and is therefore often found in news reports or scientific texts ('It is commonly assumed that ...'). Topicalisation also occurs to front focused information, violating the canonical order in the second clause of the following piece of discourse.

[26] However, adjectives appear to form an exception (cf. Section 6.3.7.5.)

(63) u kunt de trein van 00.30 nemen
u can the train of 00.30 take
'you can take the 00.30 train'

l$\overset{*}{a}$tere verbindingen zijn er niet
later connection are there not
'there are no later connections'

As focus is not the only consideration in choosing these alternations, albeit an important one, we think that the choice should not be made by the syntactic realizer, but rather by the microplanner.

6.5.3 Comparison with other systems

Work on syntactic realization for Dutch is relatively rare. Probably the best known work is that by Kempen and Hoenkamp (1987) on Incremental Procedural Grammar (IPG), which has been continued by De Smedt (1990, 1996). In contrast to SynR, it aims at a psychologically plausible model of the syntactic aspects of human sentence production. IPG is implemented in Lisp as a set of procedures for generating sentence parts, and a stack that serves as the central data structure for communication. It has no clear distinction between the grammar (data part) and its implementation (procedural part), which is generally considered a disadvantage from a computational linguistic point of view. Other work on syntactic realization for Dutch includes Teunissen (1997), which focuses on a multilingual implementation of Semantic Syntax with limited coverage, and Degand (1993, Degand (1996), which is based on Systemic Functional Linguistics.

The multilingual text generation system KPML (Matthiessen and Bateman 1991), which is a continuation of the PENMAN system for English, has a grammar for Dutch as well (Degand 1993; Degand 1996). It profits from a long tradition of NLG using Systemic Functional Grammar (Halliday 1994), and has a thorough theoretical foundation as well an extensive grammar development environment. The coverage of the Dutch grammar appears to be comparable to that of SynR. The input structures have a slightly higher level of abstraction, closer to a semantic specification, but appear to be less suitable for manual input due to their complex structure.

Another well known surface generator for English is SURGE (Elhadad, McKeown, and Robin 1997; Elhadad, McKeown, and Robin 1997), which is actually a Functional Unification Grammar (FUG) implemented in the FUF system (Elhadad 1993). It is widely used, has a broad coverage, and allows input as case frames or as abstract syntactic structures. In earlier work on syntactic realization, we implemented a FUG for Dutch called SEM2SYN (Marsi 1998b), although with a substantially smaller coverage than SURGE. Its input has the form of a case frame consisting of a predicate, its participants, and a number of circumstances, which is slightly more abstract than SynR's input. In principle, there is a clear separation between the declarative grammar and its processing by FUF. Defaults for underspecified input follow naturally from the unification mechanism. However, a disadvantage of FUF is that it does not support Dutch inflection, which must therefore be coded in the grammar, causing a lot of overhead. In contrast to SynR, SEM2SYN – or rather FUF – uses backtracking during generation. We found that the use of advanced control mechanisms like goal-freezing becomes necessary when generating long sentences to prevent excessive backtracking. Unfortunately, this obscures the separation between grammar and processing, and complicates tracing. In addition, whereas transformational rules provide a natural way of changing word order, this proved to be less naturally expressed in a FUG. In particular, word ordering within VP's gave rise to rather cumbersome formulations.

Although for English, we also mention the commercial RealPro surface generator for English by CoGenTex (Lavoie and Rambow 1997; CoGenTex, Inc 1998), because it is closest to SynR in terms of input and design. The input is a Deep Syntactic Structure (DSynSs) similar to our APS.

(64)
```
help [aspect:cont,polarity:neg
(I John [class:proper-noun]
II Mary [class:proper-noun])
```

'John is not helping Mary'

RealPro has pipelined modules for inserting function words and default features (Deep Syntactic Component), for determining agreement and word order (Surface Syntactic Component), for morphological processing (Deep Morphological Component), and for orthographic processing (Graphical Component). In contrast to SynR, it is based on Meaning-Text Theory (MTT) (Mel′čuk 1988), a form of dependency grammar. It produces strings, but not their syntactic constituent structure.

In sum, none of the above systems offers precisely the kind of functionality that is required in the current work, which is the reason that SynR was built.

6.5.4 Orthography

In most NLG systems, syntactic realization includes generating orthographic properties such as punctuation symbols and capitalisations. In SLG, however, there is no need for orthographic processing, because the output is speech rather than text. Capitalisation of proper nouns, for example, has no equivalent in speech. Admittedly, some punctuation symbols would appear to have prosodic correlates. In read speech, for instance, a comma corresponds to an intonational phrase boundary, and a question mark to a tune ending in a high rise. However, this does not imply that the intonational form is *derived* from the orthographic form, thus requiring orthographic processing in an SLG system. Rather, both the orthographic form and the prosodic form encode aspects of the linguistic structure of a sentence/utterance, its function, and its relation to the context. For example: the writer's/speaker's intention to verify the truth of a proposition by means of a yes/no question results in a question mark in text and in question tune in speech.[27] Likewise, a piece of information that is not an essential contribution according to the writer/speaker may surface between parentheses in text and be pronounced in a lower register in speech. Therefore, the relation between orthographic form and prosodic form is only *indirect*. It may be exploited in TTS, by lack of access to the underlying information, but it should be ignored in SLG, where the relation between the underlying information and the prosodic form can be modelled directly.[28]

6.5.5 Coverage

SynR currently generates a reasonable subset of Dutch. The following lists are intended to give an impression of its coverage. With respect to clauses, it is capable of generating root clauses with different grammatical moods (declaratives, polar questions, wh-questions, imperative), finite and non-finite bound clauses, relative clauses, and expressing passive voice, dative shift, tense, topicalisation and extraposition. As far as NP's are concerned, it can handle determiners (articles, also personal, demonstrative, interrogative, and possessive 'pronouns'), and several types of modifications, such as positive, comparative, and superlative adjectives, attributive PP's, relative S's, numerals and times. In addition, it can process pronominal adverbs and expand numerals. Of course, there are many ways to extend its coverage, for example by generating negation, quantification, modal verbs, appositions, grammatical alternations like ergative, clefts, and pseudo-clefts, and reflexive and reciprocal determiners.

6.6 Summary

This chapter described SynR, a syntactic realizer for Dutch that maps abstract phrase specifications (APS's) into grammatical expressions and their syntactic structures, allowing a user/system to abstract from most of idiosyncrasies of syntax. It provides the functionality required for SLG not provided by any of the other systems currently available for Dutch. An overview of SynR is given in Table 6.2.

APS's allow for a concise specification of phrases thanks to a compact format, the notational conventions introduced, and the assumption of reasonable default values for features. It was argued that using APS's in combination with SynR provides a number of advantages compared to string templates. First, it is easier to maintain, especially when a system must be scaled up. Second, templates cannot provide the linguistic structure required for prosodic realization in SLG systems. Third, APS's facilitate microplanning, resulting in

[27] Of course, there may be additional cues, like word order.

[28] An SLG system may output both text and speech in parallel, so a user can read and listen at the same time. This requires explicit synchronisation between orthographic and prosodic realization.

Table 6.2: Overview of SynR.

Component:	*Tasks:*	*Formalism:*
Base	Projecting the basic syntactic structure Adding purely syntactic information Adding the structure of compound words Expanding numbers and times	AGFL
Trans	Distributing focus Linking arguments to syntactic functions Inserting auxiliaries Projecting sentences Enforcing agreement Arranging word order	GRAMTSY
Flex	Inflecting content words Inserting function words	AGFL

higher-quality output.

The architecture of SynR is highly modular and forms a strict pipeline. The use of the special-purpose grammar formalisms AGFL and GRAMTSY guarantees a high degree of separation between the linguistic grammar (data) and the procedural aspects of its implementation. The strengths of affix grammars were employed in the first and last step, i.e. for mapping APS's onto basic syntactic structures in Base, and for inserting and inflecting words in Flex. At the same time, its weakness was avoided by using a more powerful transformational grammar for enforcing agreement relations and determining word order. As a result, SynR is easy to trace, modify, and extend, is reliable and sufficiently fast, and can be reused in future laboratory NLG and SLG systems.

SynR currently generates a number of syntactic structures covering a substantial part of the core grammar of Dutch. However, more constructions can be added. One of the interesting challenges is the word order in VP's in relation to the focus distribution and other grammatical constraints.

Chapter 7

Prosodic Realization

7.1 Introduction

This chapter describes ProsR, the module of our SLG system CONPAS (described in Chapter 5) that is responsible for the prosodic realization of utterances. It receives its input from SynR, the syntactic realization module (described in Chapter 6), in the form of syntactic trees augmented with semantic and informational features. Its task involves, in chronological order, prosodic phrasing, pitch accent placement, intonational phrasing, and tune choice. Its output consists of strings of words with prosodic annotations, which are passed on to the speech synthesiser.

At least two aspects of prosodic realization have already received sufficient attention in the previous chapters. Phonological phrasing, which is the most important part of prosodic phrasing for our purposes, was investigated in depth in Chapter 3. The φ-algorithm implemented in ProsR is virtually identical to the one developed there, and will therefore only be summarised in Section 7.2. Similarly, ProsR uses the $\mathcal{NEW}$ algorithm for intonational phrasing developed in Chapter 4, and a description of it will not be repeated here.

The focus of the current chapter is on pitch accent placement. Following our research objective of designing, building and testing a linguistically-based SLG system that generates pitch accents most acceptable to experts, we present a rule-based generative grammar for pitch accent placement. The point of departure is the preliminary grammar that was presented in Section 2.5. Further development is directed by the research questions formulated in Section 1.3: how adequate is the theory, and how can it be improved? In Section 7.3, we argue for a number of modifications and extensions, part of which are motivated by inadequacies encountered during the development of PTDGen and WFDGen. In particular, we will pay attention to the correct accentuation of complex numerals (Section 7.3.5). The analyses in the current chapter rely on exemplary data. A formal evaluation of the accent placement algorithm (and of that for intonational phrasing as well) will be described in the next chapter.

Another aspect of prosodic realization, tune choice, is strictly speaking beyond the scope of our research objective. However, since an SLG system cannot function properly without it, a simple, provisional implementation of tune choice will be outlined in Section 7.4.

Prosodic realization is implemented in GRAMTSY (Section 6.3.1) as a grammar of transformational rules inserting features into labelled-bracket structures. Following the practice of the previous chapters, we will avoid discussing the implementation details, and describe prosodic realization in ProsR at a higher level of abstraction.

7.2 Prosodic phrasing

Prosodic phrasing in ProsR concerns the derivation of the higher-level prosodic constituents: the phonological phrase (φ) and the intonational phrase (π).[1] Recall that the prime motivation for deriving φ's is that they serve as the domain of application for the Rhythm Rule (RR), an accent deletion rule that modifies the accent distribution (Section 2.5.3). In addition, prosodic constituents are used as the domain of application for segmental postlexical rules (Section 5.6.1), and their boundaries may trigger preboundary lengthening and pause (Section 5.6.2).

7.2.1 Phonological phrase

In Chapter 3, several algorithms for phonological phrasing were evaluated, and it was found that none of them was fully adequate, and an improved algorithm for φ-phrasing was proposed, which has been implemented in ProsR. Since this subject has been amply investigated in Chapter 3, we will confine ourselves to a recapitulation of the algorithm (1).

(1) **Phonological phrase construction for Dutch**

A φ-boundary occurs after every content word W unless

a. W is an adjacent modifier, or
b. W is an A or Adv directly preceding a verb.

The algorithm inserts a φ-boundary after every content word, which may be a noun[2], verb[3], adjective, adverb[4]. The class of content words is extended with numerals and letters; for instance, the numeral *een* 'one' and the letter *A* in *platform one A*. However, the content word can neither be an adjacent modifier (2), nor an adjective or adverb directly preceding a verb. The definition of adjacent modification has been extended to numerals, in order to account for their accentuation (which will be discussed in Section 7.3.5).

[1] The lower-level prosodic constituents are derived by SpeechS (Section 5.6.1).

[2] Recall from Section 2.6.3 that 'pronouns' are syntactically analysed as determiners.

[3] This excludes functional verbs, i.e. auxiliaries and copula.

[4] This also excludes adverbs that are traditionally analysed as pronominal adverbs, such as *er* 'there' and *daarvoor* 'therefore'.

(2) **Adjacent modification**

For every $X, Y \in \{N, A, Adv, Num\}$:
X is an *adjacent modifier* of Y if and only if

a. X is left-adjacent to Y, and
b. X is dominated by YP.

Compounds are opaque with respect to φ-phrasing, that is, (1) will never insert a φ-boundary after a content word that is a non-final constituent of a compound word. Examples of φ-phrasing are repeated in (3) and (4). Notice that *erg* is an adjacent modifier of *mooie*, which is in turn an adjacent modifier of *tuin*; *ijskoud* is an adjective directly preceding a verb.

(3) a. $(_{NP}$ een $(_{AP}$ $(_{Adv}$ erg $)$ $(_{A}$ mooie $)$ $)$ $(_{N}$ tuin $)$ $)$
a very beautiful garden

b. $[_{\varphi}$ een $\overset{*}{\text{erg}}$ $\overset{\emptyset}{\text{mooie}}$ $\overset{*}{\text{tuin}}$ $]$

(4) a. bier moet $(_{VP}$ $(_{A}$ ijskoud $)$ $(_{Aux}$ worden $)$ $(_{V}$ gedronken $)$ $)$
beer should ice-cold be drunk
'serve beer cold as ice'

b. bier $[_{\varphi}$ moet $\overset{*}{\text{ijskoud}}$ worden $\overset{*}{\text{gedronken}}$ $]$

As mentioned at the end of Section 3.3.3, the algorithm still suffers from a problem with unparsed auxiliaries and clitics. In (5), the verb *neemt* /nemt/ and the pronoun *u* /y/ are subject to *resyllabification*, that is, they are pronounced as a word with two syllables: /nem-ty/. Since it is generally assumed that the domain for resyllabification is the prosodic word (ω), it follows that both words must belong to the one and the same ω. However, our algorithm predicts a φ-boundary between them. According to the principles of prosodic phonology, a ω cannot dominate a φ, and we thus end up with a contradiction. Moreover, the resulting lack of resyllabification in the synthesised version of (5) leads to a conspicuously unnatural pronunciation.

(5) $[_{\varphi}$ vervolgens $]$ $[_{\varphi}$ neemt $]$ $[_{\varphi}$ u de sneltrein $]$
next take you the fast-train

In fact, Dutch has a limited number of function words like *u* that behave like phonological *clitics*, which attach to an appropriate host to form a single ω(Booij 1995, ch. 8). Accordingly, ProsR contains a cliticization rule that moves a clitic to the preceding φ, which solves this problem. A similar rule moves auxiliaries at the end of a VP to the preceding φ (see Section 3.2.2 for a description of this problem). Admittedly, these are rather ad hoc rules, and recent approaches in Optimality Theory may allow a better solution (Truckenbrodt 1999; Post 2000; Selkirk 2000).

7.2.2 Intonational phrase (π)

In Section 2.3.7, we distinguished between two phonological interpretations of the 'theory-neutral' concept of an intonational phrase: (1) the *Intonational Phrase* (π) as a prosodic category in the theory of prosodic phonology, and (2) the *Association Domain* (AD) as an independent domain for realizing an intonation contour. Since the first interpretation was shown to give rise to inconsistencies, we adopted the second view. The π, although no longer associated with intonation, is nevertheless preserved, because it useful as the domain of application for postlexical rules, and because its boundaries would appear to trigger pause insertion and preboundary lengthening. ProsR uses the following mapping rule for π-phrasing, which is essentially the rule proposed in Nespor and Vogel (1986, ch. 7), the difference being that not only root sentences, but all sentences, including bound and relative, trigger an π-boundary in (6).

(6) **Intonational phrase construction**

A π-boundary occurs:
a. before and after an appositional string;
b. before and after a sentence.

7.3 Pitch accent placement

A linguistic solution to pitch accent placement – as aimed for in the current work – is a generative grammar that takes into account all relevant factors in order to produce a correct accent distribution. In Section 2.5.4, we outlined a concise grammar for this goal that required four types of information as input: the surface string, the phonological phrases, the argument structure, and the focus distribution. Given this input, the grammar itself consisted of only two rules. First, the Sentence Accent Assignment Rule (SAAR), which accounts for the focus-accent relation. Second, the Rhythm Rule (RR), which accounts for rhythmical adjustments to the pitch accent distribution. This grammar constitutes the core of the implementation of pitch accent placement in ProsR described below. It will be shown, however, that the formulation of SAAR needs to be somewhat refined. In addition, it will be argued that an extra rule called *NP Premodifier Deaccentuation* is necessary. Furthermore, a lexically-driven rule called the *Lexical Rhythm Rule* is added. Finally, it will be shown that with the help of just one extra rule, *Unit Rhythm Rule*, an adequate accentuation of Dutch numerals is obtained.

7.3.1 Sentence Accent Assignment Rule

The relation between focus and accent forms the point of departure for pitch accent placement. This is captured in the first part of SAAR (Gussenhoven 1992b) – repeated here as (7)– which requires every focused constituent to be accented. The second part makes an exception for predicates that meet the specified condition.

(7) **Sentence Accent Assignment Rule** (SAAR)

If focused, every predicate, argument, and modifier must be accented, with the exception of a predicate that, discounting unfocused constituents, is adjacent to an argument.

An example from WFD is given in (8-a), where focus is indicated by underlining. The focused predicate *gebruikt* remains unaccented, because its adjacent argument *als kleurstof* is focused. In (8-b), the same argument is not focused, but the predicate can remain unaccented by virtue of its focused argument *het sap van de bladeren*; notice that the intervening unfocused argument *daarvoor* is ignored. In contrast, intervention of the *focused* modifier *zelden* forces the predicate to be accented in (8-c).

(8) a. het sap van de bladeren wordt als kleurstof gebruikt
the juice of the leaves is as dye used
'the juice of the leaves is used as dye'
[ARG het sap van de bladeren] wordt [ARG als kleurstof]
[PRED gebruikt]

b. het sap van de bladeren wordt daarvoor gebruikt
the juice of the leaves is for-that used
'the juice of the leaves is used for that purpose'
[ARG het sap van de bladeren] wordt [ARG daarvoor]
[PRED gebruikt]

c. het sap van de bladeren wordt zelden gebruikt
the juice of the leaves is rarely used
[ARG het sap van de bladeren] wordt [MOD zelden]
[PRED gebruikt]

From a practical point of view, it turns out to be more convenient to implement SAAR in two steps: (1) accent all focused constituents; (2) deaccent predicates where appropriate.

7.3.1.1 Accenting focused constituents

Recall that the focus distribution is either explicitly specified by the document planner (in terms of the features `foc` and `back`) or determined by the default focus rules in SynR. The latter are based on the content-function word dichotomy, foregrounding content words and backgrounding function words (Section 6.3.2). The first step in ProsR is thus to accent every word that carries the `foc` feature, which amounts to adding the feature `acc` to it. There are two cases that deserve special attention: compounds capable of bearing more than one accent, and compound verbs.

Compounds A compound adjective like *roodbruin* 'red-brown' can have an accent on its first part or on both parts, depending on the context of the word. Therefore, it is not sufficient to attach the feature acc to the word *roodbruin*; ProsR must have access to the internal structure of the word, or more precisely, to those parts that can carry an accent.[5] This means that syntactic trees with orthographic words as terminals will not do as input to ProsR.

Yet, generating the morphological structure of every word would be an exaggeration, because most of it is irrelevant for the purposes of accentuation. To be precise, the syntactic structures generated by SynR must include the internal structure of compounds if and only if this is relevant for accent placement, i.e. for those words in which the place of the accent depends on the context or which can take multiple accents. This class of compounds is called *phrasal compounds* in Varga (1998), because with respect to their accentuation they are closer to a syntactic phrase than a word. For Dutch, this class includes many compound adjectives and adverbs, all compound verbs, most compound numerals, and some compound nouns.[6] Their internal structure is listed in SynR's lexicon, and consequently ends up in the syntactic structures passed on to ProsR. All other words that are compounds from a morphological point of view, however, have a fixed accent position and are treated by SynR and ProsR like ordinary non-compound words.

The accentuation of phrasal compounds initially depends on the focus distribution. If focused, however, the feature acc is not attached to the compound itself, but to one or more of its parts. In the case of adjectives, adverbs, and numerals, every part receives an accent. Within nouns, only those parts that represent content words receive an accent; not the function words (e.g. conjunctions or prepositions). For example, *aan* 'at' is not accented in *Zandvoort-aan-Zee* 'Zandvoort-at-sea' (the name of a city). Some of these initial accents will be subject to deletion by rhythmic rules later on.

Compound verbs The accentuation of focused compound verbs, which consist of a particle and a verbal part, requires special care. In contrast to the other phrasal compounds, compound verbs can carry only one accent, which always goes to the particle. However, either the verbal or the particle part may have been moved out of its original position for syntactic reasons. For

[5] One might consider introducing different features to indicate an accent on the first part, the second part, or both parts of a compound. These would then have to be translated to corresponding escape codes that instruct the speech synthesiser to realize the accent in the intended location. One of the drawbacks is that this requires the speech synthesiser to do a morphological analysis to detect the parts of the compound. Apart from the fact that KUNTTS is currently not capable of interpreting such escape codes, it is not a structural solution either, because some compounds have three or more members. Examples of this are *Zoetermeer-Centrum-West* 'Zoetermeer-Centre-West' (name of a railway station), *kat-en-muis-spel* 'cat-and-mouse-game', and *doe-het-zelf-winkel* 'do-it-yourself-shop'. To account for all these possibilities, even more accent features and corresponding escape codes would be required.

[6] For practical reasons, we included compounds with an internal structure that is so complex that the speech synthesiser cannot determine the position of the accent correctly.

example, both parts remain together in (9-a), but the verbal part is fronted in (9-b), whereas the particle is moved leftwards in (9-c). With respect to (9-b), we have to make sure that the verb ***stapt*** is recognised as the verbal part of a compound, preventing accentuation as a normal content verb. Similarly, *over* must be identified as a particle (not a preposition) in (9-c), which is accented only if the compound verb that it originates from is focused.

(9) a. dat u daar ($_V$ ($_P$ over) ($_V$ stapt))
that you there `prt` transfer
'that you transfer there'

b. u ($_V$ stapt$_i$) daar ($_V$ ($_P$ over) e_i)
you transfer there `prt`
'that's where you transfer'

c. om daar ($_P$ over$_i$) te ($_V$ e_i ($_V$ stappen))
to there `prt` `prt` transfer
'to transfer there'

Recall from the previous chapter that determining the focus distribution precedes determining the word order, and that the feature `foc` is attached to a compound verb, not to its constituents. Therefore, if the verbal part is moved, as in (9-b), it will not carry the feature `foc` with it, and thus cannot be accented. This automatically leads to the correct prediction that the verbal part of a compound verb is indeed always unaccented. In order to account for the accentuation of the particle, SynR copies the feature `foc` to the particle *before* it is moved. As a result, it will be focused if and only if the compound verb is focused, and thus correctly accented, even when it is moved to another position, as in (9-c).[7]

To sum up so far, we have described the implementation of the first part of SAAR, which tells us to accent all focused material. We have seen that for normal words, the focus-accent relation is simply one-to-one, whereas for phrasal compounds, the relation is more complex, but can be satisfactorily implemented if SynR carries out the preliminary work.

7.3.1.2 Deaccenting the predicate

To continue with the second step of SAAR: a focused predicate is deaccented if it is adjacent to a focused argument, ignoring any intervening *unfocused* material. Since SynR marks predicates and arguments with the features `pred` and `arg` respectively, this is relatively straightforward to implement. For each focused predicate, ProsR checks if it has a focused argument to its left or right, and if no other focused constituent intervenes between these. If so, the feature `acc` is removed from the predicate.

However, there are a few special circumstances. The first has to do with the special properties of compound verbs. Second, SAAR is stated in terms

[7] As a matter of fact, not only `foc`, but also the predicate-marking feature `pred` is copied onto the particle, to facilitate the deaccentuation step of SAAR later on.

of focused *semantic constituents*, whereas CONPAS assumes that focus is distributed over *words*; this requires some mechanism of focus projection from words onto constituents. Third, there are a number of exceptions to SAAR, which require extra conditions. These issues will be discussed below.

Compound verbs Recall from the previous section that the `foc` feature of a compound verb is copied onto its particle, because if the verb is focused, it is the particle that must be accented, even when it is extracted from the verb. For similar reasons, the `pred` feature is moved from the compound verb onto its particle: SAAR must be able to deaccent the particle if it stands in the required relation to an argument, just as it would in the case of an ordinary verb. An example of this is (10), where the particle *over* of the focused compound verb *overstappen* is adjacent to its focused argument *op de stoptrein*, which allows the particle to remain unaccented.

(10) u hebt $\overset{*}{\text{vijf}}$ $\overset{*}{\text{minuten}}$ om op de ($_{N<foc,arg>}$ $\overset{*}{\text{stoptrein}}$)
you have five minutes for on the slow-train
($_{P<foc,pred>}$ over$_i$) te ($_{V<foc>}$ e_i ($_V$ stappen))
`prt` `prt` transfer
'you have five minutes to transfer to the slow train'

Focus projection For SAAR to apply to a predicate, a focused argument must be adjacent. However, if a complex argument contains multiple content words of which some are focused and some are not, this raises the question what makes the argument as a whole count as focused. According to Gussenhoven (1992b, Appendix), the *head* of a phrase that functions as argument must be focused; it is not sufficient to have only a modifier focused. This explains why in the question-answer pair (11) the predicate *vertrokken* is focused but unaccented, whereas it must be accented in in (12).

(11) *Question:* waarom wachten we?
'why are we waiting?'

Answer: de $\overset{*}{\text{trein}}$ is al vertrokken
the train is already gone

(12) *Question:* waarom wachten we op de trein naar Amsterdam?
'why are we waiting for the train to Amsterdam?'

Answer: de trein naar $\overset{*}{\text{Rotterdam}}$ is al $\overset{*}{\text{vertrokken}}$
the train to Rotterdam is already gone

There appear to be at least two ways to implement this. The first approach is to make SAAR (actually, the GRAMTSY rules that implement SAAR) scan the inner structure of an argument in order to determine if its head is focused.

However, since an argument can be an NP, PP, AP, or S with a complicated syntactic structure, even a coordinated structure, this would make the implementation of SAAR rather complex. The alternative approach is to assume that the fact that an argument is focused is given in the input to ProsR. This implies that it belongs to the tasks of SynR to project focus from the head of an argument onto its maximal projection. This approach is more attractive, as the distribution of (agreement) features along constituents is one of the tasks in syntactic realization anyway. Moreover, it allows for a concise formulation of SAAR in ProsR, abstracting away from syntactic details.

SynR implements this approach by means of a set of rules akin to those proposed by Selkirk (1984, 1995) in her focus projection theory. However, since Selkirk's generalised rules can be shown to make incorrect predictions in some cases (Gussenhoven 1992b; Gussenhoven 1999), we prefer the more restricted formulation below.

(13) **Focus Projection Rules**

a. If the head of an NP is focused,
then the NP is focused as well.

b. If the head of an AP is focused,
then the AP is focused as well.

c. If the NP argument of a P is focused,
then the PP is focused as well.

d. If the main verb in an S is focused,
then the S is focused as well.

e. If all the conjuncts of a coordination are focused,
then the coordination is focused as well.[8]

Rule (13-a), for example, projects focus from the N *trein* onto the NP argument *de trein* in (11), allowing deaccentuation of the predicate *vertrokken*. In contrast, although rules (13-a) and (13-c) project focus from *Rotterdam* onto the PP *naar Rotterdam* in (12), there are no rules that license further projection. That is, the argument *de trein naar Rotterdam* remains unfocused, precluding deaccentuation of the predicate *vertrokken*.

Topicalised arguments According to Gussenhoven (1992b), an exception to SAAR is that topicalised arguments cannot cause the predicate to be deaccented. Consider the example in (14), where everything except *ik* is focused. Whereas the predicate *slaat* may remain unaccented in (14-a) (as predicted by SAAR), it must be accented in (14-b), even though the topicalised direct object *Jan* is adjacent to it.

[8] The set of conjuncts does not include the conjunction (e.g. *and*)

(14) a. Jan slaat me
John beats me

b. Jan sla ik
John beat I

For most types of text, topicalised arguments other than the subject appear to be rare; they are commonly assumed to be bound to certain types of discourse transitions (see e.g. Rats (1996)). Despite the fact that they rarely occurred in PTD or WFD, this restriction on SAAR is implemented in ProsR by checking that the argument to the left of the predicate is not topicalised (i.e. lacks the feature `top`).[9]

Extraposed arguments One of the observations resulting from the current work is that deaccentuation of the predicate often appears to be less appropriate with extraposed arguments. Consider the data in (15): whereas the predicate *overstappen* must remain unaccented by virtue of the adjacent argument *op de stoptrein naar Schiedam Centrum* in (15-a), it can very well be accented if the same argument occurs to its right as a result of extraposition (15-b). In fact, accentuation of the predicate is obligatory when followed by an intonational boundary (15-c), if it is not to be interpreted as unfocused.

(15) a. u hebt acht minuten # om op de stoptrein naar
you have eight minutes for to the slow-train to

Schiedam Centrum over te stappen
Schiedam Centre `prt` to transfer
'you have eight minutes to transfer to the slow train to Schiedam Centre'

b. u hebt acht minuten # om over te stappen op de
you have eight minutes for `prt` to transfer to the

stoptrein naar Schiedam Centrum
slow-train to Schiedam Centre

c. u hebt acht minuten # om rustig over te stappen #
you have eight minutes for quietly `prt` to transfer

op de stoptrein naar Schiedam Centrum
to the slow-train to Schiedam Centre

Example (15-c) thus suggests an interaction between pitch accent placement and intonational phrasing, challenging our assumption that pitch accent placement precedes intonational phrasing. We will return to this issue in Section 7.3.6.4.

[9] In our syntactic analysis, subjects can also be topicalised. However, this exception to SAAR does not apply to topicalised subjects.

In sum, the second part of SAAR (deaccenting of the predicate) requires focus projection rules, because we need to know when an argument counts as focused. The implementation of deaccenting a compound verb requires some extra precautions, as the particle may be separated from the verbal part. In addition, deaccentuation is not allowed if a topicalised argument other than the subject is involved, and is blocked by an intervening AD-boundary if an extraposed argument is involved.

7.3.2 Deaccentuation of NP premodifiers

In addition to the deaccentuation of predicates by SAAR, there appears to be at least one other type of deaccentuation. Examples of this type occur in the NP's below.

(16) a. station Nijmegen
station Nijmegen
'Nijmegen station'

b. minister-president Kok
prime-minister Kok

c. de wedstrijd Ajax-Feijenoord
the game Ajax-Feijenoord

d. perron vijf
platform five

e. bladzijde vierendertig, regel twaalf
page 34 line 12

f het woord 'trein'
the word train

g een glas wijn
a glass wine

h meneer de voorzitter
mister the chairman

All of these involve a sequence of two closely related nominal parts of which the first always has a common noun. The point is that although all these examples can be interpreted as fully focused, only the second part is accented, whereas the first remains unaccented (even though it is a content word).

The syntactic structure of these examples is unclear. The structuralistic tradition suggests they constitute 'double-headed structures' (Haeseryn et al. 1997, p. 875-878). We will assume they form two NP's of which the first one serves as a premodifier of the second, akin to an AP modifying an N. That is, given the abstract phrase specification in (17-a), SynR will generate the syntactic structure in (17-b).

(17) a. (Nijmegen<prop>(station<nonref>))

b. $(_{NP}\ (_{NP}\ (_{N}$ station)) $(_{N}$ Nijmegen))

It wil be clear that neither SAAR nor RR can explain the lack of an accent on the NP premodifier: the former is not applicable, as there is no obvious predicate-argument structure present; the latter needs at least three accents to delete one of them. This forces the introduction of an additional deaccentuation rule:

(18) **NP Premodifier Deaccentuation** (NPPD)

Delete the pitch accents on an NP that serves as a premodifier of another NP.

7.3.3 Rhythm Rule

After SAAR and NPPD have applied, RR (19) deletes the intermediary accents in phonological phrases containing three or more accents.[10] The net effect of exhaustive application is that only the outermost accents in a phonological phrase are preserved. In order to prevent incorrect accent deletions, it is crucial that the boundaries of phonological phrases are correctly determined. Since RR and phonological phrasing for Dutch have been extensively discussed and tested in Chapter 3, we will not elaborate on this topic here.

(19) **Rhythm Rule** (RR)

$$* \rightarrow \emptyset \;/\; [_{\varphi} \ldots [_{\omega} *] \ldots [_{\omega} \_] \ldots [_{\omega} *] \ldots]$$

7.3.4 Lexical rhythm

There is a very limited class of quantifiers and adverbs for which the location of the pitch accent depends on the context (Gussenhoven 1983a). Examples of this are *allemaal* 'all of them', *altijd* 'always', and *meestal* 'often'. The pitch accent goes to the first part of the word if another accent follows in the same utterance (20-a), but goes to the second part if it is the last accent in the utterance (20-b).

(20) a. die trein rijdt $\overset{*}{\text{a}}$ltijd om z$\overset{*}{\text{e}}$ven $\overset{*}{\text{u}}$ur
that train drives always at seven hour

b. die trein rijdt alt$\overset{*}{\text{ij}}$d om zeven uur

[10]The notation is somewhat different from that in Section 2.5.3 – more precise actually – in order to facilitate the formulation of two additional rhythm rules below.

It is assumed that words belonging to this class are all phrasal compounds (cf. Section 7.3.1.1), which means that, if focused, they initially receive two accents. In addition, it is assumed that they are lexically marked with the feature `lrr`, originating from SynR's lexicon. This feature is then recognised by a special rule for lexical rhythm that deletes either the first or the second accent.

(21) **Lexical Rhythm Rule** (LRR)

$$* \rightarrow \emptyset \ / \ \left\{ \begin{array}{l} [_{\mathrm{U}} \cdots [_{\omega} [_{\omega} *] \, [_{\omega} \_] \,]_{\mathtt{lrr}} \cdots [_{\omega} *] \cdots] \\ [_{\omega} [_{\omega} \_] \, [_{\omega} *] \,]_{\mathtt{lrr}} \end{array} \right\}$$

That is, for every word marked with the feature `lrr`, if the remainder of the utterance contains at least one accent, then the accent on its second constituent is deleted, otherwise the accent on its first constituent is deleted. Compared with RR, there are two differences: (1) the left context of the word is irrelevant; (2) the right side context is not limited to a φ, but extends to the end of the utterance.

7.3.5 Accentuation of Numerals

Numerals occur in PTD during the generation of times and platform numbers, and in WFD in expressions of length or height. As a matter of fact, numerals are likely to occur in almost any application domain for SLG. As a consequence, the accentuation of numerals is a topic that deserves to be addressed. Ideally, their accentuation should be covered by the rules proposed so far. As will explained below, however, an additional deaccentuation rule is needed to obtain the intended results.

A numeral qualifies as a content word, and will therefore by default be focused, and thus initially accented. Numbers from 21 to 29, 31 to 39, and so on up to 91 to 99, are numeral compounds, which may carry two accents (22-a), just like any other phrasal compound (Section 7.3.1.1). Whenever they serve as an adjacent modifier, they are included into a φ with the following head, and subject to RR (22-b).[11]

(22) a. het zijn er z$\overset{*}{\text{e}}$sen n$\overset{*}{\text{e}}$gentig
it are there six-and ninety
'there are 96 of them'

b. z$\overset{*}{\text{e}}$sen n$\overset{\emptyset}{\text{e}}$gentig k$\overset{*}{\text{a}}$sten
six-and ninety cupboards
'96 cupboards'

In Section 6.2.3.4, we argued that for larger numeral phrases as in (23), *zesen negentig* should syntactically be analysed as a premodifier of *honderd/duizend*,

[11] All examples in this section are assumed to be fully focused.

analogously to (22-b). As a consequence, it becomes an adjacent modifier of the following numeral, included in the same φ, and subject to RR. This elegantly explains the accent deletions in (23).

(23) a. z$\overset{*}{e}$sen n$\overset{\emptyset}{e}$gentig h$\overset{*}{o}$nderd
six-and ninety hundred
9,600

b. tw$\overset{*}{e}$een d$\overset{\emptyset}{e}$rtig d$\overset{*}{u}$izend
two-and thirty thousand
32,000

c. tw$\overset{*}{e}$e h$\overset{\emptyset}{o}$nderd k$\overset{*}{a}$sten
two hundred cupboard

While this successfully explains the accentuation of most numerals, certain types of numerals appear to pose a problem. Consider the data in (24). At first sight, it seems as if the lack of an accent on *honderd* is again due to RR, which would imply that the whole phrase constitute a single φ.

(24) tw$\overset{*}{e}$e h$\overset{\emptyset}{o}$nderd (en) z$\overset{*}{e}$s
two hundred (and) six

However, this analysis cannot be right. Consider example (25-a). From a comparison to (23-c), we can infer that there is no φ-boundary after *twee*. On the basis of its similarity to (24), we are forced to assume that there is no φ-boundary after *honderd* either. As shown in (22-b), there is also no φ-boundary after *zesen*. Therefore, the numeral phrase in (25-a) must be a single φ. As a result, RR deletes the accents on *honderd* and *zes*. However, this gives to an unintended interpretation: without an accent, *zes* can only be interpreted as unfocused.

(25) *tw$\overset{*}{e}$e h$\overset{\emptyset}{o}$nderd (en) z$\overset{\emptyset}{e}$sen n$\overset{*}{e}$gentig
two hundred (and) six-and ninety
'296'

Likewise, the analysis of (24) as a single φ would imply (26) is a single φ as well (cf. (23-b), again resulting in an undesired deletion of the accent on *zes*. In fact, a unlimited number of counterexamples of this type can be constructed.

(26) *tw$\overset{*}{e}$een d$\overset{\emptyset}{e}$rtig d$\overset{\emptyset}{u}$izend (en) z$\overset{\emptyset}{e}$sen n$\overset{*}{e}$gentig
two-and thirty thousand (and) six-and ninety
'32,096'

In addition, *zes* cannot be deaccented in (27), where the whole numeral phrase serves as an adjacent modifier to the noun *kasten*.

(27) *tw$\overset{*}{e}$e h$\overset{\varnothing}{o}$nderd (en) z$\overset{\varnothing}{e}$sen n$\overset{\varnothing}{e}$gentig k$\overset{*}{a}$sten
two hundred (and) six-and ninety cupboards
'296 cupboards'

Clearly, there must be a φ-boundary after *honderd* and *duizend* to prevent these undesired accent deletions. Actually, this is exactly what our φ-phrasing algorithm predicts. According to syntactic analysis of numerals proposed in Section 6.2.3.4, the last three examples all contain a coordination of two numeral phrases, of which the first is headed by *honderd/duizend*. See, for example, the syntactic structure of (25) in Figure 7.1. This syntactic analysis is supported by the fact that we can optionally insert the conjunction *en* 'and' after *honderd* in (25). Given this analysis, both *honderd* and the compound *zesennegentig* trigger a φ-boundary; see the predicted φ-phrases in (28).[12]

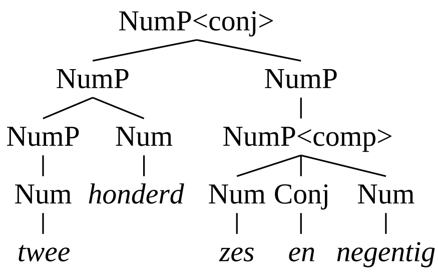


Figure 7.1: Syntactic structure of example (25).

(28) [$_\varphi$ tw$\overset{*}{e}$e h$\overset{*}{o}$nderd] [$_\varphi$ (en) z$\overset{*}{e}$s en n$\overset{*}{e}$gentig]

With respect to (27), the φ-phrases according to our algorithm are shown in (29). The accent on *negentig* is still deleted by RR, which is a correct prediction.

(29) [$_\varphi$ tw$\overset{*}{e}$e h$\overset{*}{o}$nderd] [$_\varphi$ (en) z$\overset{*}{e}$sen n$\overset{\varnothing}{e}$gentig k$\overset{*}{a}$sten]

To sum up, we have argued that there must be a φ-boundary after *honderd/duizend*, which is in accordance with the predictions of our φ-algorithm. However, returning to the example we started with (24), we now face the problem of explaining why the accent on *honderd* is deleted, as this cannot be attributed to RR.

[12] Notice that *twee*, although it is a content word, is an adjacent modifier of *honderd*, and therefore incapable of triggering a φ-boundary. *Zes* is part of a numeral compound, and therefore invisible for the φ-phrasing algorithm.

Gussenhoven (1991a) addresses the same problem for English. On his account, the syntactic structure of the phrase *hundred and two thousand men*, which is left-branching (30-a), is mapped onto a right-branching prosodic structure with recursion at the φ-level (30-b). It is assumed that the application of RR is optional and cyclic, starting with the lowest φ. Consequently, the accent on *thousand* is deleted first, while that on *two* may be preserved.

(30) a. (((hundred and two) thousand) men)

b. [$_{\varphi''}$ h$\overset{*}{\text{u}}$ndred and [$_{\varphi'}$ [$_{\varphi}$ tw$\overset{*}{\text{o}}$th$\overset{\emptyset}{\text{o}}$usand] m$\overset{*}{\text{e}}$n]]

Although this solution would solve the above cases, it has a few drawbacks. First, the reason for the reversed mapping of syntactic onto prosodic structure remains somewhat mysterious. Second, except for a general tendency towards eurhythmy, it is unclear why RR should not delete the accent on *two* as well when it is applied to the φ''. Third, it requires the prosodic structure to be recursive, violating the Strict Layer Hypothesis, which we tried to adhere to in the current work (Section 2.2). A more serious problem, however, is that this approach cannot account for the lack of an accent on *honderd* in the phrases below.

(31) a. h$\overset{\emptyset}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren d$\overset{*}{\text{e}}$rtig
hundred four and thirty
134

b. h$\overset{\emptyset}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren d$\overset{\emptyset}{\text{e}}$rtig d$\overset{*}{\text{u}}$izend
hundred four and thirty thousand
134,000

c. h$\overset{\emptyset}{\text{o}}$nderd $\overset{*}{\text{e}}$en bl$\overset{*}{\text{a}}$dzijden
hundred one pages
101 pages

In these examples, the *first* accent in the utterance is lacking without affecting a broad focus interpretation. Since RR requires at least one accent in its left-hand context, it is by definition incapable of deleting the first accent in an utterance.[13]

By now, we might suspect that the reason that *honderd* is unaccented is simply because it virtually always unfocused. However, as (32) shows, there are a number of instances where *honderd* does receive an accent.

(32) a. h$\overset{*}{\text{o}}$nderd
hundred

b. n$\overset{*}{\text{e}}$gen h$\overset{*}{\text{o}}$nderd
nine hundred

[13] Similar problems arise in English: *a hundred and* $\overset{*}{\text{o}}$*ne* p$\overset{*}{\text{a}}$*ges*

c. n$\overset{*}{\text{e}}$genen dertig h$\overset{*}{\text{o}}$nderd
nine thirty hundred
39,000

d. h$\overset{*}{\text{o}}$nderd d$\overset{*}{\text{u}}$izend
hundred thousand
100,000

e. h$\overset{*}{\text{o}}$nderd k$\overset{*}{\text{a}}$sten
hundred cupboards

In conclusion, deletion of the accent on *honderd* cannot be accounted for by RR, at least not in the case of initial deaccentuation, nor can it be attributed to a lack of focus. This motivates the introduction of additional accent deletion rule. In the remainder of this section, we will formulate this rule, and show that its use is not limited to the case of *honderd*, but includes a number of similar phenomena.

The numeral *honderd* differs from the basic numerals, i.e. *een* up to *twintig* (1-20) and *twintig* up to *negentig* (20,30,... 90), in the sense that it may function as a unit of measure. That is, (33) can paraphrased as *twee van honderd, en zes* 'two of hundred, and six', or *twee hondertallen en zes* 'two hundreds, and six'.

(33) tw$\overset{*}{\text{e}}$e h$\overset{\emptyset}{\text{o}}$nderd (en) z$\overset{*}{\text{e}}$s
two hundred (and) six

The same pattern is found for *duizend* 'thousand' and *miljoen* 'million' (34), and for other abstract units of measure when combined with a numeral (35).

(34) a. het zijn er tw$\overset{*}{\text{e}}$e d$\overset{\emptyset}{\text{u}}$izend $\overset{*}{\text{e}}$enen tw$\overset{*}{\text{i}}$ntig
it are there two thousand one-and twenty
'there are 2,021 (of them)'

b. het zijn er tw$\overset{*}{\text{e}}$e milj$\overset{\emptyset}{\text{o}}$en $\overset{*}{\text{e}}$enen tw$\overset{*}{\text{i}}$ntig
it are there two million one-and twenty
'there are 2,000,021 (of them)'

(35) a. tw$\overset{*}{\text{e}}$e $\overset{\emptyset}{\text{u}}$ur z$\overset{*}{\text{e}}$sen tw$\overset{*}{\text{i}}$ntig
two hour six-and twenty
'2:26'

b. tw$\overset{*}{\text{e}}$e m$\overset{\emptyset}{\text{e}}$ter v$\overset{*}{\text{i}}$jftig
two meter fifty
'two meters and fifty centimetres'

c. z$\overset{*}{\text{e}}$ven g$\overset{\emptyset}{\text{u}}$lden v$\overset{*}{\text{i}}$jfen n$\overset{*}{\text{e}}$gentig
seven guilder five-and ninety
'Fl 7.95'

d. $\overset{*}{\text{e}}$lf h$\overset{\emptyset}{\text{o}}$nderd na Chr$\overset{*}{\text{i}}$stus
eleven hundred after Christ
'1100 A.D.'

e. t$\overset{*}{\text{i}}$en gr$\overset{\emptyset}{\text{a}}$den onder n$\overset{*}{\text{u}}$l
ten degrees below zero

f. t$\overset{*}{\text{i}}$en m$\overset{\emptyset}{\text{i}}$nuten na $\overset{*}{\text{a}}$ankomst
ten minutes after arrival

Again, these deaccentuations cannot be accounted for by RR, assuming these phrases form single φ's, because this fails to explain why *een* in (34), *zesen* in (35-a), and *vijfen* in (35-c) must remain accented. As a matter of fact, our φ-algorithm predicts φ-boundaries after *duizend*, *uur*, and *gulden*.

However, it is not only the lexical form of these nouns that counts. As the examples in (36) show, deaccentuation is impossible if they are not used as units of measure. For the same reason, *honderd* must keep its accent in the last example of (32).

(36) a. ?de koers van de g$\overset{\emptyset}{\text{u}}$lden d$\overset{*}{\text{a}}$alde
the price of the guilder dropped

b. ?het $\overset{\emptyset}{\text{u}}$ur van de w$\overset{*}{\text{a}}$arheid is $\overset{*}{\text{a}}$angebroken
the hour of the truth has come
'the moment of truth is upon us'

In anticipation of a more thorough investigation, we will assume that words used in this way are marked by the feature `urr`, which originates either from the input or from SynR. The feature triggers (37), which requires that the word so marked is deaccented if it is followed by another accent. *Honderd* cannot be deaccented if it is utterance-final, as is clear from the first three examples in (32), and neither can any other unit without causing a contrastive interpretation.

(37) **Unit Rhythm Rule** (URR)

$$* \rightarrow \emptyset \;/\; [_{U} \cdots [_{\omega} \_\_]_{\mathbf{urr}} \cdots [_{\omega} *] \cdots]$$

However, there is one complication: the rule is blocked by an intervening AD boundary. In all of the following examples, deleting the accent on the unit preceding the AD-boundary results in a contrastive reading.

(38) a. [$_{AD}$ het zijn er z$\overset{*}{e}$ven honderd tw$\overset{*}{e}$een d$\overset{\varnothing}{e}$rtig d$\overset{*}{u}$izend]
it are there seven hundred two-and thirty thousand

[$_{AD}$ n$\overset{*}{e}$gen h$\overset{\varnothing}{o}$nderd $\overset{*}{e}$enen tw$\overset{*}{i}$ntig]
nine hundred one-and twenty
'there are 732,921 (of them)'

b. [$_{AD}$ tw$\overset{*}{e}$e honderd t$\overset{*}{a}$chtig m$\overset{*}{e}$ter] [$_{AD}$ en v$\overset{*}{i}$jfen vijftig
two hunderd eighty meter and five-and fifty

c$\overset{*}{e}$ntimeter]
centimetre

c. het huidige wereldrecord staat op [$_{AD}$ tw$\overset{*}{e}$e $\overset{*}{u}$ur] [$_{AD}$ z$\overset{*}{e}$s
the current world-record stands on two hours six

min$\overset{*}{u}$ten] [$_{AD}$ en tw$\overset{*}{a}$alf sec$\overset{*}{o}$nden]
minutes and twelve seconds

Since we assume that pitch accent placement *precedes* AD-derivation, AD-boundaries are not yet known when URR applies. Therefore, this condition cannot be included in the structural description of URR. We return to this problem in Section 7.3.6.4.

A final point that needs to be addressed is the mutual order of RR and URR. If URR precedes RR, URR can destroy the context for RR, and therefore prevent the deletion of accents. An example of this is (39). In contrast, changing the order, i.e. RR before URR, yields the correct output, as is shown in (40).

(39) Input: twee en dertig honderd$_{\texttt{ur}}$ twaalf
two and thirty hundred twelve
3,212

IA: tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig h$\overset{*}{o}$nderd$_{\texttt{urr}}$ tw$\overset{*}{a}$alf

φ's: [$_{\varphi}$ tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig h$\overset{*}{o}$nderd$_{\texttt{urr}}$] [$_{\varphi}$ tw$\overset{*}{a}$alf]

URR: [$_{\varphi}$ tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig h$\overset{\varnothing}{o}$nderd$_{\texttt{urr}}$] [$_{\varphi}$ tw$\overset{*}{a}$alf]

RR: n/a

Output: ? tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig honderd tw$\overset{*}{a}$alf

(40) Input: twee en dertig honderd$_{\texttt{urr}}$ twaalf

IA: tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig h$\overset{*}{o}$nderd$_{\texttt{urr}}$ tw$\overset{*}{a}$alf

φ's: [$_\varphi$ tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig h$\overset{*}{o}$nderd$_{\texttt{urr}}$] [$_\varphi$ tw$\overset{*}{a}$alf]

RR: [$_\varphi$ tw$\overset{*}{e}$e en d$\overset{\emptyset}{e}$rtig h$\overset{*}{o}$nderd$_{\texttt{urr}}$] [$_\varphi$ tw$\overset{*}{a}$alf]

URR: [$_\varphi$ tw$\overset{*}{e}$e en dertig h$\overset{\emptyset}{o}$nderd$_{\texttt{urr}}$] [$_\varphi$ tw$\overset{*}{a}$alf]

Output: tw$\overset{*}{e}$e en dertig honderd tw$\overset{*}{a}$alf

We finish this section on numerals with a more complex example (41), which shows that the combined operation of φ-phrasing, RR, and URR can account for accent distributions that are far from trivial.

(41) Input: honderd$_{\texttt{urr}}$ vier en vijftig duizend$_{\texttt{urr}}$ zeven
hundred four and fifty thousand seven
honderd$_{\texttt{urr}}$ tween en dertig kasten
hundred two and thirty cupboards
154,732 cupboards

IA: h$\overset{*}{o}$nderd$_{\texttt{urr}}$ v$\overset{*}{i}$eren v$\overset{*}{i}$jftig d$\overset{*}{u}$izend$_{\texttt{urr}}$
z$\overset{*}{e}$ven h$\overset{*}{o}$nderd$_{\texttt{urr}}$ tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig k$\overset{*}{a}$sten

φ's: [$_\varphi$ h$\overset{*}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ v$\overset{*}{i}$eren v$\overset{*}{i}$jftig d$\overset{*}{u}$izend$_{\texttt{urr}}$]
[$_\varphi$ z$\overset{*}{e}$ven h$\overset{*}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ tw$\overset{*}{e}$e en d$\overset{*}{e}$rtig k$\overset{*}{a}$sten]

RR: [$_\varphi$ h$\overset{*}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ v$\overset{*}{i}$eren v$\overset{\emptyset}{i}$jftig d$\overset{*}{u}$izend$_{\texttt{urr}}$]
[$_\varphi$ z$\overset{*}{e}$ven h$\overset{*}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ tw$\overset{*}{e}$e en d$\overset{\emptyset}{e}$rtig k$\overset{*}{a}$sten]

URR: [$_\varphi$ h$\overset{\emptyset}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ v$\overset{*}{i}$eren vijftig d$\overset{\emptyset}{u}$izend$_{\texttt{urr}}$]
[$_\varphi$ z$\overset{*}{e}$ven h$\overset{\emptyset}{o}$nderd$_{\texttt{urr}}$]
[$_\varphi$ tw$\overset{*}{e}$e en dertig k$\overset{*}{a}$sten]

Output: honderd v$\overset{*}{i}$eren vijftig duizend
z$\overset{*}{e}$ven honderd tw$\overset{*}{e}$e en dertig k$\overset{*}{a}$sten

Due to the length of the expression, an AD-boundary after *duizend* is possible, which would prevent deaccentuation of *duizend* by URR. Conversely, the presence of an accent on both *vieren* and *duizend* would mean that the first conjunct in (41), which forms a coordinated structure, contains two accents, and would therefore trigger the insertion of an AD-boundary by the phrasing rule AD-in-coordination (cf. the end of Section 4.8.1.5). We thus correctly predict the coincidence of an accent on *duizend* and a subsequent AD-boundary. However, as already mentioned, AD-boundaries are not yet known when URR applies. We will further discuss this paradox in Section 7.3.6.4.

7.3.6 Discussion

7.3.6.1 Distribution of focus over words

The text planner indicates non-default focus by attaching the features `foc` (and `back`) to the lemmas in an APS. Since the level of APS is intended to abstract away from syntactic details, function words are normally not specified, because these are assumed to be inserted by SynR. As a result of this abstraction, function words are automatically unfocused and unaccented. For many function words, this prediction is indeed correct. In particular, pure function words, i.e. those that serve a only grammatical function and have no semantic content of their own, cannot be focused and accented.[14] In (42-a), for example, the verb *rekent* subcategorizes for a prepositional object with the preposition *op*. This preposition is purely functional, not contributing any semantic content of its own. In (42-b), we find a passive form of an intransitive verb. Since there is no semantic role available for the syntactic function of subject, its place is filled with the semantically empty element *er*. Example (42-c) shows the verb *vergist*, which obligatory subcategorizes for the reflexive pronoun *zich*. In all three cases, focus on the purely functional element is very unlikely.

(42) a. ?De organisatie rekent $\overset{*}{\text{op}}$ goed weer.
the organisation counts on good weather

b. ?$\overset{*}{\text{Er}}$ wordt gestaakt.
there is striked.
'A strike is going on.'

c. ?De minister heeft $\overset{*}{\text{zich}}$ vergist.
the minister has himself mistaken
'The minister was mistaken.'

The assumption that focus is distributed over the lemmas in an APS goes counter to linguistic theories which assume that focus is distributed over seman-

[14] Barring metalinguistic uses. It is virtually impossible to claim that a certain accent distribution is *ungrammatical*, because for every odd accentuation, one can imagine a context in which it would be appropriate. For example, *g$\overset{*}{e}$loven* 'belief' has an accent that is not on the main-stressed syllable of the word. Moreover, the accented syllable contains a schwa, which normally cannot be accented. However, this accentuation is appropriate to correct a misunderstanding, as in: *Moet ik dat beloven?* 'Do I have to promise that?'; *Nee, g$\overset{*}{e}$loven* 'No, believe'.

tic terms (Schmerling 1976; Gussenhoven 1984), and may not be entirely appropriate. At least some function words, like the proximal determiner *die* in (43-b), can be accented, which implies that they can be focused. However, since an APS abstracts away from such function words, there is no appropriate lemma to attach the feature `foc` to. A partial solution is to specify the function word anyway (43-a).

(43)	APS:	`(bloem<near,back>(DET:<foc>))`	
	Output:	* die	bloem
		that	flower

This is impossible in (44), where it is not clear whether an additional `(V:<foc>)` refers to the auxiliary *zou* or to *hebben*. Adding the auxiliary's lemma (i.e. `(V:<foc>zullen)`) defeats the whole purpose of an APS, that is, abstracting away from purely syntactic matters.

(44)	APS:	`(V:doen<past,fut,perf>[<speak>][<far,top>])`						
	Output:	dat	* zou	ik	hebben	gedaan	(ware het niet dat ...)	
		that	would	I	done	have	(if only ...)	
		I would have done that (if only ...)'						

In general, the possibility to focus on proximal/distal relations, modality, negation or other aspects appears to require a more semantic form of representation than an APS can offer; see also (Bolinger 1983). We leave this issue for future research.

7.3.6.2 Argument or modifier

As far as deaccentuation of the predicate is concerned, correct application of SAAR crucially depends on the identification of arguments. Clearly, SLG has an advantage over TTS here, as the relevant information is automatically generated, relying on a lexicon that encodes the number and type of arguments of a verb. Still, in a number of situations it is not that obvious how many arguments the lexical entry of a verb should have. If a constituent cannot be omitted, or the verb subcategorizes for its form (e.g. a prepositional object), we can safely assume that it is an argument. Similarly, there are some syntactic tests that may help in diagnosing arguments. However, sometimes we can only rely on semantic intuitions about the 'strong relation' between a verb and a supposed argument. As a consequence, when a predicate is incorrectly accented, we are forced to choose between attributing the error to malfunctioning of SAAR or to an incorrect specification of the argument structure.

Consider example (45-a), where everything except the subject is assumed to be focused. The predicate *gaat* ought to be unaccented. However, because the instrument *met de stoptrein* is supposed to be a modifier, and is focused as well, it prevents deaccentuation of *gaan*. Thus, either this is an exception to SAAR, or the instrument is in fact an argument. Notice, however, that the predicate cannot be accented either with other modifiers (45-b-c), which suggests that some other factor is responsible for the deaccentuation of the predicate. A possible explanation is that *gaan*, which also serves as an auxiliary of future tense in Dutch, is a semantically light verb, as most of the meaning of 'going to' is expressed by the preposition of its argument (*naar* 'to'), rather than by the verb itself. Alternatively, we may conjecture that in the context of train travel descriptions, predicates like *gaan* are contextually activated to such a degree that they are backgrounded.

(45) a. ?we gaan met de stoptrein naar Amsterdam
we go with the fast-train to Amsterdam

b. ?we gaan met de kinderen naar Amsterdam
we go with the children to Amsterdam

c. ?we gaan met een slakkegangetje naar Amsterdam
we go with a snail's-pace to Amsterdam

Another example concerns the verb *groeien* 'grow', which occurs seven times in the corpus of flower descriptions, but is never accented. In (46), for instance, the subject *de plant* is assumed to be unfocused, and the locative *in venen en natte heide* is taken to be a modifier. According to SAAR, therefore, the predicate must be accented, which is incorrect.

(46) ?de plant groeit in venen en natte heide
the plant grows in peats and wet heath

However, it may be argued that the predicate 'grow' in (46) is not used in its literal sense of 'becoming bigger', but rather as a synonym for 'occurring at'. The latter is much more likely to have a locative argument; compare (47).

(47) de plant komt voor in venen en natte heide
the plant occurs `prt` in peats and wet heath

Moreover, the predicate does have an accent in (48), where it can only be interpreted in the sense of 'becoming bigger'.

(48) de plant groeit snel
the plant grows fast

So with respect to (46), we have reason to believe that it is not an exception to SAAR, but an example of how a predicate's argument structure can depend on its interpretation.

7.3.6.3 Accentuation of adverbs

As for any other word, ProsR accents an adverb only if it is focused, which it is either by explicit specification from the text planner or by the default focus rules in SynR. These defaults rely on the content/function word distinction: only content words receive focus. While this distinction is relatively clear for verbs (auxiliaries and copula versus main verbs) and nouns (common and proper nouns versus pronouns, demonstratives, etc.), the situation is not so clear for adverbs. Some adverbs, like *gisteren* 'yesterday' and *rechtdoor* 'straight on', have a semantic content similar to that of common nouns and main verbs. As a matter of fact, it can be argued for many adverbs that they are adverbially used adjectives. Adverbs of this type can thus be considered content words. Others have an anaphoric function akin to that of pronouns, e.g. *hier* 'here', *daar* 'there' or *waarna* 'where-after', suggesting a classification as function word. For many other adverbs, a classification as either function or content word is difficult. Examples are:

- discourse markers (e.g. *echter* 'however' and *vervolgens* 'next')
- intensifiers (e.g. *zeer* 'very' and *nogal* 'quite')
- negations (e.g. *niet* 'niet' and *nooit* 'never')
- particles (e.g. *ook* 'also' and *al* 'already')

In spite of these difficulties, default focusing of adverbs by SynR is rather simplistic: those that are clearly anaphoric are classified as function words, all others as content words. As a result, adverbs are sometimes incorrectly accented. Some examples from WFD are given in (49): *wel* in (49-a), *ook* in (49-b), and *vaak* in (49-c).[15]

(49) a. ?hij wordt wel anderhalve meter hoog
it becomes as-much-as one-and-a-halve meter high

b. ?ook dit is een zeldzame soort
also this is a rare species

c. ?de plant heeft vertakte # vaak liggende stengels
the plant has branching often recumbent stems

In PTDGen, the default focus is overruled by the text planner, which backgrounds all adverbs. This appears to be the best strategy, as virtually all adverbs in this domain are discourse markers, such as *eerst* 'first', *vervolgens* 'next', and *tenslotte* 'finally'. This type of adverbs was found to be rarely accented (van Donzel 1999, p. 154).[16]

Needless to say, there is scope for improvement here. Possibly, a more refined classification of adverbs will help to improve their default accentuation;

[15] The accent on *vaak* causes RR to erroneously delete the accent on *liggende*.

[16] Interestingly, however, discourse markers were often followed by a pause.

see also Hirschberg (1993, p. 321). However, little appears to be known about what actually causes different types of adverbs to be focused/accented (but see e.g. Verhagen (1986) and Gussenhoven (1984, p.125)).

On the one hand, since the problem is one of focus distribution rather than of pitch accent placement, a solution should be expected from discourse theory. In terms of SLG, this amounts to saying that, rather than relying on defaults, the text planner must take responsibility for the focusing of adverbs. On the other hand, there appear to be systematic relations between the interpretation of focus particles (a special class of adverbs) and their accentuation (Verhagen 1986; Elffers 1992; Heinzel 1999). This suggest that the accentuation of adverbs is not 'simply' a matter of correctly predicting their focus, but requires a semantically driven accentuation rule in the spirit of SAAR.

7.3.6.4 Interaction between pitch accent placement and intonational phrasing

We encountered at least two cases that suggest an interaction between pitch accent placement and intonational phrasing. First, we found that an AD-boundary between a predicate and an extraposed argument blocks a predicate's deaccentuation by SAAR (cf. example (15-c)). Second, we found that URR deletes the accent on a unit term followed by another accent, unless an intervening AD-boundary occurs (cf. example (38)). The problem with these cases is that they contradict the assumption that pitch accent placement *precedes* intonational phrasing, on the assumption that correct intonational phrasing partly depends on the pitch accent distribution, as illustrated by the rule AD-in-Coordination in Section 4.8.1.5, where intonational phrasing in coordinated structures was argued to depend on the number of accents in the conjuncts. Other rules use the number of pitch accents as an indication for the weight of an AD. Moreover, the pitch accent distribution must prevent the creation of ill-formed AD's containing no accents, which was found to be one of the flaws in Nespor & Vogel's intonational phrasing algorithm (Section 4.4.2.4). Therefore, reversing the order of pitch accent placement and intonational phrasing is not an attractive option.

The interaction is a rule-order paradox: phrasing requires accentuation, but accentuation requires phrasing. In the case of URR, a possible solution is provided by *precompilation* as proposed by Hayes (1990) and used for the accentuation of numerals in Hungarian in Varga (1998). This implies that, for each unit word, the lexicon generates two forms, one accented and one unaccented, together with their appropriate contexts, and that the choice of the form is delayed until the required context can be checked. Precompilation appears to be intended to handle syntax-dependent phonological effects. Accounting for SAAR by precompilation would force us to assume that the lexicon generates at least two precompiled forms (accented and unaccented) for *every* predicate. An alternative direction to explore is Optimality Theory (Prince and Smolensky 1993), as it abandons the concept of rule ordering altogether (Selkirk 2000). For the time being, the issue is provisionally dealt with in the implementation of ProsRby undoing deletions due to SAAR or URR if an intervening AD-boundary is inserted later on.

7.4 Tune choice

When for each of the generated utterances the locations of pitch accents and intonational boundaries have been determined, the last task left in order to arrive at a complete intonational specification is choosing a tune. This requires two sorts of choices. First, for each pitch accent, we have to choose its type, i.e. H*L, L*H, or H*LH. Second, for each AD, tone morphemes like downstep may be added. Although investigating the whys and the wherefores of tune choice is beyond the scope of the current project, it is clear that some sort of provisional tune choice must be carried out if we are to synthesise our output. In this section, therefore, a simple mechanism for tune choice will be presented (see also the general discussion in Section 9.3.4).

7.4.1 Type of pitch accent

We assume that the default pitch accent is H*L. Deviation of the default only occurs for the last pitch accent in an AD, which may be either L*H or H*LH. The reason for this is that both L*H and H*LH end in an H tone, phonetically corresponding to an upwards pitch movement that effectively marks the end of a non-final AD. The choice between either of them is made dependent on the distance of the pitch accent to the following AD boundary. If the accent is on the last word in the AD, L*H will be used (50-a), but if the accent is on one of the prefinal words in the AD, H*LH is chosen (50-b).[17] This avoids long, high-pitched stretches of speech, which are hard to synthesise convincingly. There is some empirical evidence that the same strategy is used by human speakers (Terken 1993, p.39).

(50) a.

	L*H		H*L	H*L
de	grote	#	klokvormige	bloemen
the	large		bell-shaped	flowers

b.

	H*LH						H*L		H*L H*L
de	bloemen	van	de	plant	#	zijn	groot	en	klokvormig
the	flowers	of	the	plant		are	big	and	bell-shaped

If the utterance is a question, as indicated by the feature `polar` or `wh` in its syntactic structure, H*LH or L*H is chosen for its last accent, the choice between them being determined by the considerations of length given above. The algorithm for pitch accent selection can therefore be stated in the form of the following three rules.

[17]The intonational boundary in this particular location is for exemplification only, and would actually not be generated by $\mathcal{NEW}$.

(51) **AD-final-accent**

For each final pitch accent in a non-final AD:
if it is on the last word in the AD, select L*H, otherwise, H*LH.

(52) **Question-accent**

For the final pitch accent in a question:
if it is on the last word of the utterance, select L*H, otherwise, H*LH.

(53) **Default-accent**

Select H*L for all other pitch accents.

7.4.2 Boundary tones

Recall that every AD starts with an initial boundary tone. The default, unmarked choice is %L, whereas a %H tone has particular pragmatic implications. For instance, there is experimental evidence that it interacts with the subsequent pitch accent to signal friendliness, politeness, irritation, and aloofness (Grabe, Gussenhoven, Haan, Marsi, and Post 1998). Little is known about the meaning of this *high onset* in Dutch, but it is clear that it is only appropriate in particular situations (Keijsper 1984a; Keijsper 1984b; Pierrehumbert and Hirschberg 1990). In addition, it appears to be related to the discourse structure in the sense that a new discourse segment starts with a relatively high pitch (Wichmann, House, and Rietveld 2000). As far as our application domains are concerned, there is at least one suitable context. Recall that travel descriptions often consist of multiple paragraphs (e.g. from A to B, next from B to C, and finally from C to D). We found that in many cases a %H tone can be used at the start of a new paragraph in order to signal the paragraph break. It appears to highlight the textual structure and increase the liveliness of the synthetic speech.[18] This strategy is formalised in the two rules below.

(54) **Paragraph-marking-initial-boundary-tone**

Select %H for the first AD of a paragraph.

(55) **Default-initial-boundary-tone**

Select %L for all other AD's.

[18]However, if the distance to the first pitch accent is too large, the result is a long, high-pitched stretch of speech, which sounds less natural. Notice that this is similar to our observation regarding the use of an AD-final L*H (cf. Section 7.4.1). In addition, if the first syllable of the AD is accented, selecting a high boundary tone is useless, as there is no space for its phonetic realization, although it does no harm either.

Following Gussenhoven and Rietveld (1992a), we assume that an utterance-final H*L is combined with L% and an L*H with H%. No boundary tone is added if the last pitch accent is H*LH, as its final H tone serves as a boundary tone in AD-final position.

(56) **Utterance-final-boundary-tone**

If the last pitch accent in the utterance is H*L (L*H), then select L% (H%) for the last AD.

7.4.3 Tone morphemes

The only tone morpheme that we will take into account is downstep (see the paragraph on scaling in Section 5.6.2.2). Downstepped contours would appear to be typically used for coordinations. Recall from Section 4.8.1.5 that we hypothesised that there are basically two ways to phrase a coordination: either all conjuncts are taken together in a single AD, or each conjunct is assigned to a separate AD. It was assumed that the choice depends on how the accents are distributed over the coordination: the former is taken for simple coordinations, in which each of the non-final conjuncts carries at most one pitch accent, otherwise the latter is chosen. The same distinction between simple and complex coordinations also plays a role with respect to downstep. It would appear that downstep is appropriate when applied to a simple coordination (57-a), but odd when applied to a complex coordination, at least for non-final conjuncts (57-b). As in intonational phrasing, the final conjunct appears to form an exception, as it can carry multiple accents without blocking downstep (57-c).

(57) a. ![$_{AD}$ Fr$\overset{*}{a}$nkrijk D$\overset{*}{u}$itsland en B$\overset{*}{e}$lgie]
France Germany and Belgium

b. ? ![$_{AD}$ N$\overset{*}{o}$ord-Fr$\overset{*}{a}$nkrijk Z$\overset{*}{u}$id-D$\overset{*}{u}$itsland en $\overset{*}{O}$ost-B$\overset{*}{e}$lgie]
North-France South-Germany and Eastern-Belgium

c. ![$_{AD}$ Fr$\overset{*}{a}$nkrijk D$\overset{*}{u}$itsland en $\overset{*}{O}$ost-B$\overset{*}{e}$lgie]
France Germany and Eastern-Belgium

In sum, intonational phrasing and downstepping are taken to be complementary options as far as coordinations are concerned. A simple coordination has no internal phrasing, but allows downstep. In contrast, a complex coordination has internal AD-boundaries, but does not tolerate downstep. This is captured in the following rule.

(58) **Downstep**

For each AD containing a simple coordination, apply downstep.

The implementation of (58) requires some extra effort. Phonologically, downstep is a tone morpheme that is attached to an AD-boundary. The start and end of a simple coordination do not necessarily correspond to AD-boundaries. In (59), for instance, the AD starts with *ze*, whereas the actual coordination starts with *groot*. Therefore, the implementation of (58) requires a search for either an AD-boundary coinciding with the start of the coordination, or the first one preceding it.

(59)	![$_{\mathrm{AD}}$ ze	zijn	gr$\overset{*}{o}$ot	h$\overset{*}{o}$ekig	en	st$\overset{*}{e}$vig]
	they	are	big	angular	and	firm

As a matter of fact, some parts belonging to the same AD as the coordination may be accented as well. For instance, (60) has an accent on the verb *uitgestorven*, which is not a part of the preceding coordination, and downstep may be less appropriate here. Alternatively, an extra AD-boundary can be inserted after the coordination, or downstep can be omitted altogether. The former violates the constraint that no boundary should be inserted before the verb cluster; the latter fails to signal the structure of the simple coordination by intonational means. As intuitions are not clear-cut on this matter, we choose to implement the latter alternative in ProsR.

(60)	? ![$_{\mathrm{AD}}$ een	plant	die	in	Fr$\overset{*}{a}$nkrijk	D$\overset{*}{u}$itsland	en	N$\overset{*}{e}$derland	is
	a	plant	that	in	France	Germany	and	Netherlands	is

$\overset{*}{u}$itgestorven]
died-out
'a plant that has become extinct in France, Germany and the Netherlands'

Although the above approach to downstep in coordinations works satisfactorily, there is another way to render complex coordinations like (57-b), namely, by a special form of downstep that affects only the 'primary' accent in each conjunct. With respect to (57-b), this means that the H* targets for *Noord*, *Zuid*, and *Oost* are downstepped, while those for *Frankrijk*, *Duitsland*, and *Nederland* remain at the same level. This variant can be included into the phonological description as a new tone morpheme, with a corresponding phonetic implementation in the speech synthesiser.

In the same vein, it was found that appositions cannot be optimally accounted for within the current system. These are typically accompanied by a temporary lowering and compression of the register: the overall pitch level drops at the start of the apposition, but returns to the previous level when finished. This could be modelled by a tone morpheme for pitch reduction. As far as the implementation is concerned (cf. the scaling function in Figure 5.16), lowering of the register can be accomplished by decreasing parameter N, and compression of the register by decreasing W.

7.5 Summary

This chapter described ProsR, the module of CONPAS that is responsible for the prosodic realization of utterances on the basis of their syntactic structure, focus distribution, argument structure, and the surface string. The output is a string of words interspersed with prosodic annotations for prosodic constituents (φ's and π's), AD's, pitch accents, boundary tones, and tone morphemes. While producing this output, ProsR carries out the following four tasks:

1. **Derivation of prosodic constituents**
   ProsR implements the new algorithm for φ-phrasing in Dutch proposed in Chapter 3. Apart from the inclusion of the category Num in the definition of adjacent modification, and an extra rule for clitic movement, no substantial modifications were required, as the algorithm produced the intended results. In addition, ProsR implements a simple π-phrasing rule.

2. **Pitch accent placement**
   Starting from the simple algorithm presented in Chapter 2, several improvements were proposed.

   (a) With respect to SAAR, two steps were implemented: initial accentuation according to the focus distribution, followed by deaccentuation of the predicate where appropriate. It was argued that the internal structure of some compounds (phrasal compounds, which may carry two accents, and compound verbs) must be transparent for the accentuation algorithm. Furthermore, a set of focus projection rules was introduced to define the relation between a focused argument and the words it consists of. Finally, it was observed that deaccentuation of the predicate is blocked with topicalised arguments, and with extraposed arguments preceded by an AD-boundary.

   (b) An additional deaccentuation rule was introduced for NP's that serve as NP modifier (NPPD).

   (c) The Lexical Rhythm Rule (LRR) was implemented to account for an small class of exceptional quantifiers and adverbs.

   (d) An investigation of the accentuation of numerals motivated the introduction of the Unit Rhythm Rule (URR). It was shown that the combination of φ-phrasing plus rhythm rules correctly predicts the accent distribution in complex numerals. It was observed that URR is blocked by an intervening AD-boundary.

   The complete algorithm for pitch accent placement is summarised in Table 7.1.

   Some remaining problems were discussed.

   (a) The assumption that focus is distributed over the lemmas in an APS is too restricted, because it does not allow certain cases of narrow focus with an accented function words.

Table 7.1: Rules of the algorithm for pitch accent placement.

Name:	*Definition:*
Sentence Accent Assignment Rule (SAAR)	If focused, every predicate, argument, and modifier must be accented, with the exception of a predicate, that, discounting unfocused constituents, is adjacent to an argument. (Exceptions are possible for topicalised and extraposed arguments; cf. Section 7.3.1.2.)
NP Premodifier Deaccentuation (NPPD)	Delete the pitch accents on an NP that serves as a premodifier of another NP.
Rhythm Rule (RR)	$* \rightarrow \emptyset \;/\; [_{\varphi} \ldots [_{\omega}\, * \,] \ldots [_{\omega}\, \_\_ \,] \ldots [_{\omega}\, * \,] \ldots]$
Lexical Rhythm Rule (LRR)	$* \rightarrow \emptyset \;/\; \left\{ \begin{array}{l} [_{\mathrm{U}} \ldots [_{\omega} [_{\omega}\, * \,] [_{\omega}\, \_\_ \,]]_{\mathtt{lrr}} \ldots [_{\omega}\, * \,] \ldots] \\ \\ [_{\omega} [_{\omega}\, \_\_ \,] [_{\omega}\, * \,]]_{\mathtt{lrr}} \end{array} \right\}$
Unit Rhythm Rule (URR)	$* \rightarrow \emptyset \;/\; [_{\mathrm{U}} \ldots [_{\omega}\, \_\_ \,]_{\mathtt{urr}} \ldots [_{\omega}\, * \,] \ldots]$

(b) The distinction between argument and modifier is not always clear, and appears to be context-dependent for some predicates.

(c) The default accentuation of adverbs on the basis of the content versus function word distinction results in too many accented adverbs. A more refined categorisation of adverbs might be useful. It was additionally suggested that the accentuation of adverbs is not only a matter of correct prediction of their focus by the text planner, but may require a semantically driven accentuation rule.

(d) Examples of interaction between pitch accent placement and intonational phrasing were found with respect to SAAR and URR, leading to a rule-order paradox for which we have no sound solution.

3. **Intonational phrasing**
ProsR uses the $\mathcal{NEW}$ algorithm for intonational phrasing developed in Chapter 4.

4. **Tune choice**
A relatively simple algorithm for tune choice was described, which is

Table 7.2: Rules of the algorithm for tune choice.

Name:	*Definition:*
AD-final-accent	For each final pitch accent in a non-final AD: if it is on the last word in the AD, select L*H, otherwise, H*LH.
Question-accent	For the final pitch accent in a question: if it is on the last word, select L*H, otherwise, H*LH.
Default-accent	Select H*L for all other pitch accents.
Paragraph-marking-initial-boundary-tone	Select %H for the first AD of a paragraph.
Default-initial-boundary-tone	Select %L for all other AD's.
Utterance-final-boundary-tone	If the last pitch accent in the utterance is H*L (L*H), then select L% (H%) for the last AD.
Downstep	For each AD containing a simple coordination, apply downstep.

summarised in Table 7.2. It was suggested that a high initial boundary tone can be used to signal the start of a new paragraph, and that downstep is particularly appropriate for simple coordinations. In addition, it was observed that an alternative form of downstep exists, which cannot be generated with the current system, but is in principle not difficult to include in the phonological description and its phonetic implementation. The same goes for the pitch reduction accompanying appositions.

Chapter 8

Evaluation

8.1 Introduction

Our research objective has been to design, build, and test a linguistically-based SLG system that generates speech with pitch accents and intonational boundaries that are most acceptable to experts. The SLG system and the linguistic theory on which it is based were described in the previous chapters. The present chapter reports on an experimental evaluation in which the accents and boundaries generated by the system were compared with those assigned by a number of human experts, in order to determine to what extent our objective has been realized.

Some parts of the underlying linguistic theory were already evaluated. Phonologi1cal phrasing, an essential part of our approach to accent placement, was investigated in Chapter 3, and the newly proposed φ-algorithm was tested using a corpus of flower descriptions. However, the complete algorithm for pitch accent placement as described in the previous chapter, although supported by exemplary data, has not yet been formally evaluated. The $\mathcal{NEW}$ intonational phrasing algorithm proposed in Chapter 4 was also evaluated with respect to flower descriptions. Although this evaluation showed that it performed significantly better than at least two of the other algorithms, this result may to some extent be ascribed to domain-specificity. Thus, although parts of the underlying linguistic theory were tested, the output of the SLG system itself has not been formally evaluated. In addition, previous tests took place in the domain of flower descriptions, and it remains to be shown that the algorithms perform equally well when applied to other domains.

A point that has often been made in the literature is that SLG has an advantage over TTS, because it can exploit the detailed and reliable linguistic information generated by the NLG component for prosody assignment (Section 5.2.3). In order to get an impression of the performance difference between an SLG system and a general-purpose TTS system, the evaluation also included a comparison of output of our SLG system with that of KUNTTS, a representative TTS system for Dutch.

This chapter is organised as follows. We will first describe the method and material of the evaluation in Section 8.2. Next, we will present the results for pitch accent placement in Section 8.3, and those for intonational phrasing in Section 8.4. These results will then be discussed in Section 8.5, and compared to those reported in other work. We will conclude with a summary in Section 8.6.

8.2 Method and material

The purpose of this evaluation is to establish to what extent the pitch accents and intonational boundaries generated by the SLG system are in accordance with those preferred by human experts. One method is therefore to present the spoken output to a panel of (expert) listeners, and ask them to judge the acceptability of the accents and boundaries. As argued in Section 1.1.4, however, the disadvantage of a perceptual evaluation is that other aspects of the wave form, such as segmental errors, inadequate phonetic realization, or speaker characteristics, may interfere with judgement of the intonation. We therefore decided to rely on a symbolic assessment of the acceptability judgements of experts, asking them to prescribe the preferred accent placement and intonational phrasing for a given text, and comparing these with the predictions of our algorithms.

Below, we will describe the text material that was used (Section 8.2.1), and the assignment of pitch accents and intonational boundaries to these texts by the human experts (Section 8.2.2), our SLG system (Section 8.2.3), and the KUNTTS system (Section 8.2.4).

8.2.1 Text material

Evaluation at the symbolic level was applied earlier to investigate phonological phrasing and accent placement in Chapter 3, and intonational phrasing Chapter 4. We used a text corpus with flower descriptions that was syntactically analysed by hand, including the argument structure and the focus distribution. For this evaluation, we used *another* type of text material, in order to investigate whether the algorithms perform equally well when applied to other domains: travel descriptions and numbers (or expressions containing numbers) generated by our SLG system.[1] In contrast to the former evaluations, the text material did not have to be parsed manually, because all semantic and syntactic information was generated by the SLG system.

The travel descriptions were generated by PTDGen, the SLG system for the public transport domain (PTD). PTDGen can generate three types of descriptions: *concise*, *normal*, and *elaborate* (cf. Section 5.3.4). For each type of description, two texts were generated: a travel description involving only one

[1]For this reason, flower descriptions generated by WFDGen were not included in this evaluation.

transfer, and another involving three transfers. This resulted in six texts ranging from short (2 sentences) to long (14 sentences). In addition, the two output examples in Appendix E were included. Thus, 8 travel descriptions with a total of 56 sentences, and an average of 11.5 words per sentence, were used. Two example texts are given in Appendix C. In the remainder of this chapter, we will refer to this text material as TD (Travel Descriptions).

The second type of text material consisted of numerals, and phrases or sentences containing numerals, which were generated by our SLG system on the basis on manually supplied APS's. These are examples that occurred in Section 7.3.5, or variants with identical syntactic structures but different words. In total, there were 60 instances with an average of 4.6 words per instance. This text material will henceforth be referred to as NR (Numbers); a part of it is given in Appendix D. With the exception of a full stop at the end of sentences, the text material did not contain any punctuation symbols, digits, or abbreviations, as these are not generated by the system.

Some compounds, like the numeral in (1), can receive two or more accents.

(1) $\overset{*}{\text{e}}$enentw$\overset{*}{\text{i}}$ntig
one-and-twenty
'21'

For these phrasal compounds (see the paragraph Compounds in Section 7.3.1.1), the parts (i.e. *eenen* and *twintig*) were separated by a blank, and counted as a separate words. Accordingly, the total number of words in TD is 643, which represents the total number of accent decisions, while the total number of accent decisions in NR is 274.

8.2.2 Intonation by the human experts

The text material was presented to four human experts. All of them are native speakers of Dutch and have been involved in research on intonation; none of them participated in the previous experiment with the corpus of flower descriptions. After being informed of the purpose of the evaluation, they were asked to indicate the locations of the pitch accents and intonational boundaries to be used in an ideal reading of the text. To accommodate the possibility that alternative accentuations are equally acceptable, a distinction was made between obligatory accents (indicated by '*') and optional accents (indicated by '+'). Analogously, they could assign obligatory ('#') and optional boundaries ('|'). The result will be referred to as the *reference* or *prescription* (see Appendix C and Appendix D). In accordance with our research objective, tune was not taken into consideration.

The initial prescriptions were collected, formatted in a convenient way, and then presented again to the experts. They were asked to compare their own prescription with those of the three others, and to reconsider their choices in case of differences. Not only did this allow them to correct inconsistencies in their own judgements, it also served to increase the consensus. We postpone a presentation of the number of accents and boundaries assigned by the experts, and the agreement among them, to Section 8.3.1.

8.2.3 Intonation by the SLG system

The texts were generated by our SLG system, and accents and intonational boundaries were assigned by the prosodic realization module ProsR. Pitch accent placement was carried out by the rule-based algorithm described in Chapter 7 and summarised in Table 7.1. Intonational phrasing was done by the $\mathcal{NEW}$ algorithm presented in Section 4.8.1 and summarised in Table 4.14. In contrast to the human experts, ProsR did not assign optional accents or boundaries.

8.2.4 Intonation by the TTS system

As already mentioned, an additional point of interest in the evaluation was a comparison between SLG and TTS. Therefore, the text material was also processed by KUNTTS (Section 5.6), which is representative for the type of prosody assignment achieved by current TTS systems for Dutch (Kerkhoff and Rietveld 1994; Kerkhoff and Rietveld 1995; Kerkhoff, Rietveld, and van Bergem 1998). The accents and boundaries assigned by KUNTTS were extracted from the trace files, allowing us to compare TTS to SLG. KUNTTS did not assign optional accents or boundaries.

The *Syntax-To-Prosody module* (STP) of KUNTTS was developed as part of the multilingual TTS system developed in the ESPRIT-project POLYGLOT (Dirksen 1992; Dirksen and Quené 1993), and is based on metrical theory and the principles of focus-accent theory (Baart 1987). Its input consists of a syntactic parse by a robust wild card parser. The syntactic tree is translated to a binary-branching metrical tree, the nodes of which are labelled weak or strong according the syntactic relations between nodes. For example, the phrase structure rule in (2) produces a left-branching metrical tree in which the NP is strong (\) with respect to the V, and the DET is weak (/) with respect to the N. It is assumed that by default every major phrase receives a pitch accent, where location of the accent is determined by following the path of strong nodes until a terminal node is reached. The rules in (2) represent the 'normal' situation in which an accent within the NP assigns focus to the entire VP, as shown in in (3).

(2) $VP \longrightarrow NP \setminus V$
$NP \longrightarrow DET \ / \ N$

(3) hij heeft ($_{VP}$ ($_{NP}$ een bo*ek) ($_{V}$ gelezen))
he has a book read
'he read a book'

The lexicon specifies which words are typically *not* accented; by and large these are function words. If such a word appears in a position that is strong according to the phrase structure rules, a deaccenting rule is invoked which reverses the weak-strong labelling. As a result, an accent will shift to a sister node, as shown in (4).

(4) hij heeft ($_{VP}$ ($_{NP}$ het) ($_{V}$ gel*ezen))
he has it read
'he read it'

In addition, there are rules that reverse the weak-strong labelling in stress shift environments, akin to our RR.

As far as prosodic phrasing in concerned, STP places a major boundary between NP and VP, and a minor boundary between NP and PP, provided that the created prosodic phrase contains at least one accent. Some additional post-processing takes place to avoid phrases which are too short.

8.3 Results for pitch accent placement

We will first present the quantitative agreement with respect to pitch accent placement for the humans, ProsR and KUNTTS. Next, we will analyse differences between the experts and ProsR to see if these can be traced back to particular deaccentuation rules.

8.3.1 Agreement in accentuation

Table 8.1 shows the number of accents in TD and NR assigned by the four human experts and the two algorithms. As already mentioned, neither KUNTTS nor ProsR assigned optional accents. It can be seen that the experts assigned about the same number of obligatory and optional accents. The average number of obligatory pitch accents in NR was substantially higher than in TD, because most of the text material consisted of short expressions (numbers), containing few function words and lacking any unfocused material.

In order to estimate the agreement among the four human experts, Fleiss kappas were calculated using three categories (obligatory, optional, and no accent). For TD, kappa was 0.79 (643 objects, $p < 0.000$); for NR, kappa was 0.89 (274 objects, $p < 0.000$). Kappa values of 0.8 and higher are generally considered to indicate a high level of agreement (Rietveld and van Hout 1993).

The agreement between two accent assignments can also be expressed in the form of a 3×3 confusion matrix. For instance, Table 8.2 expresses the agreement in accent placement in TD between human expert CO and the ProsR algorithm. From this, we can calculate an agreement index (between zero and one) by taking the sum of the numbers on the diagonal, and dividing this by the total number of accent decisions:

$$agreement = \frac{233 + 0 + 322}{643} = 0.86$$

Table 8.1: Pitch accents in TD and NR assigned by the human experts and the algorithms, in both absolute numbers, and as a percentage of the total number of accent decisions in TD (n=643) and NR (n=274).

	TD:				NR:			
	Obligatory:		Optional:		Obligatory:		Optional:	
Humans:	N:	%:	N :	%:	N:	%:	N:	%:
CO:	260	40.44	37	5.75	172	62.77	3	1.09
JC:	284	44.17	40	6.22	169	61.68	7	2.55
JH:	246	38.26	45	7.00	164	59.85	10	3.65
MS:	250	38.88	55	8.55	167	60.95	7	2.55
$\overline{x}$:	260	40.44	44.25	6.88	168	61.31	6.75	2.46
sd:	17.05	2.65	7.89	1.23	3.37	1.23	2.87	1.05

	TD:				NR:			
	Obligatory:		Optional:		Obligatory:		Optional:	
Algorithms:	N:	%:	N:	%:	N:	%:	N:	%:
KUNTTS:	249	38.72	0	0	148	54.01	0	0
ProsR:	266	41.37	0	0	163	59.49	0	0

However, this method creates an unreasonable disadvantage for KUNTTS and ProsR, because neither of these assigned optional accents, which implies that all optional accents are interpreted as instances of disagreement. In fact, *optional* means that the accent is allowed, but may equally be omitted. Thus, whether the 37 optional accents assigned by CO are obligatory (9 cases) or none (28 cases) according to ProsR is actually not important, and we therefore chose to ignore optional accents (and optional boundaries for that matter). The same approach was taken in Chapter 3 and Chapter 4, where only clear cases (words that must be accented or cannot be accented) were taken into consideration, and variable cases (words that may be accented) were ignored. As a result, the agreement index for Table 8.2 was calculated in the following way:

$$agreement = \frac{233 + 322}{233 + 24 + 27 + 322} = 0.91$$

Table 8.3 shows the agreement indices for pitch accent assignment. It can be observed that the average agreement of ProsR is comparable to that of the human experts, whereas that of KUNTTS is substantially lower. This is clearly visible in the graphical representation of the same data in Figure 8.1, which

Table 8.2: Confusion matrix expressing the agreement in accent placement in TD between ProsR and human expert CO.

		CO:			
		Obligatory:	Optional:	None:	Total:
	Obligatory:	233	9	24	266
ProsR:	Optional:	0	0	0	0
	None:	27	28	322	377
	Total:	260	37	346	643

suggests that the accentuation of ProsR was close to that of the human experts, but there is still room for improvement.

Figure 8.2 shows the dendrograms resulting from an hierarchical cluster analysis (using complete linkage) on the agreement indices for pitch accent placement. As expected, the human experts were clustered first, closely followed by ProsR, whereas KUNTTS was at a large distance.

8.3.2 Disagreement in accentuation

Unlike the approach used in the evaluations reported in Chapter 3 and Chapter 4, the human experts were not asked to form a consensus accentuation, which may serve as a reference. Instead, there were four individual prescriptions, which showed high agreement. In order to facilitate a qualitative comparison between ProsR and the human experts, we made reference to a majority criterion. It was assumed that ProsR generated *too few accents* in either of the following situations:

1. all four experts assigned an obligatory accent, but ProsR failed to assign one;

2. three experts assigned an obligatory accent and one assigned an optional boundary, but ProsR failed to assign one.

Similarly, we assumed that ProsR generated *too many accents* if either:

1. none of the experts assigned an accent, but ProsR assigned one;

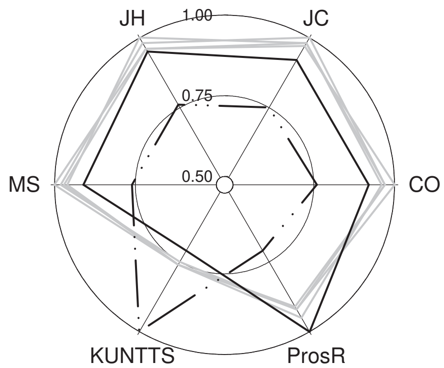
2. three experts assigned no accent and one assigned an optional accent, but ProsR assigned one.

Below, we will analyse both types of disagreement.

Table 8.3: Agreement indices in TD and NR among the human experts and the algorithms with respect to pitch accent placement.

TD:						
	CO:	JC:	JH:	MS:	KUNTTS:	ProsR:
CO:		0.97	0.96	0.97	0.76	0.92
JC:	0.97		0.98	0.98	0.75	0.92
JH:	0.96	0.98		0.96	0.76	0.95
MS:	0.97	0.98	0.96		0.76	0.91
$\overline{x}$:	0.97	0.98	0.97	0.97	0.76	0.93
sd:	0.01	0.01	0.01	0.01	0.00	0.02

NR:						
	CO:	JC:	JH:	MS:	KUNTTS:	ProsR:
CO:		0.97	0.96	0.96	0.76	0.92
JC:	0.97		1.00	0.99	0.79	0.95
JH:	0.96	1.00		0.99	0.79	0.96
MS:	0.96	0.99	0.99		0.79	0.96
$\overline{x}$:	0.96	0.99	0.98	0.98	0.78	0.95
sd:	0.01	0.02	0.02	0.02	0.02	0.02



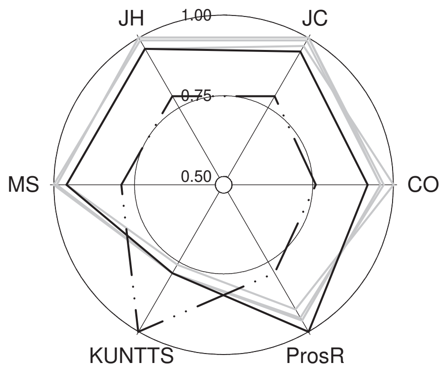


Figure 8.1: *A graphical representation of the data in Table 8.3, showing the agreement indices for TD (left) and NR (right) with respect to pitch accent placement. The centre of the circle represents an agreement of 0.5 and its outside represents an agreement of 1. ProsR is represented by the black line, KUNTTS by the dashed black line, and the four human experts by grey lines.*

```
                 Rescaled Distance Cluster Combine

        0         5         10        15        20        25
        +---------+---------+---------+---------+---------+

JC      --+
          |
MS      --+-+
          | |
CO      --+ +---------+
            |         |
JH      ----+         +-----------------------------------+
                      |                                   |
ProsR   --------------+                                   |-
                                                          |
KUNTTS  --------------------------------------------------+
```

```
                 Rescaled Distance Cluster Combine

        0         5         10        15        20        25
        +---------+---------+---------+---------+---------+

JC      --+
          +-+
JH      --+ |
            +-----+
            |     |
MS      ----+     +-------+
                  |       |
CO      ----------+       +-------------------------------+
                          |                               |
ProsR   ------------------+                               |-
                                                          |
KUNTTS  --------------------------------------------------+
```

Figure 8.2: Dendrograms resulting from an hierarchical cluster analysis on the agreement indices with respect to pitch accents (cf. Table 8.3) for TD (top) and NR (bottom).

Table 8.4: Differences involving ProsR assigning too few accents.

Rule:	TD:	NR:	Total:
Focus	15	0	15
URR	0	8	8
Total:	15	8	23

8.3.2.1 Too few accents

There were 23 cases in which ProsR assigned too few accents (see Table 8.4). These were caused either by lack of focus or overapplication of the Unit Rhythm Rule (Section 7.3.5); overapplication of SAAR, RR or any of the other deaccentuation rules did not occur.

As far as TD is concerned, 13 out of 15 differences involved focusing of repeated placenames. PTDGen backgrounded all placenames that were mentioned before in the current paragraph (i.e. the description of the current section of the journey). Apparently, this focus strategy was too restrictive, resulting in too few focused placenames and consequently too few accents. In contrast, the human experts appeared to accent every placename, unless it was mentioned in the directly preceding sentence. We will return to this issue in Section 8.5.1.

With respect to NR, there were four cases where an accent on *honderd* ‘hundred’ was incorrectly deleted by URR. Two of these are given in (5).

(5) a. ?h$\overset{\varnothing}{\text{o}}$nderd en z$\overset{*}{\text{e}}$ven
hundred and seven
107
b. ?h$\overset{\varnothing}{\text{o}}$nderd en tw$\overset{*}{\text{e}}$e
hundred and two
102

In contrast, there were many similar examples where *honderd* was correctly deaccented by URR, two of which occur in (6).

(6) a. h$\overset{\varnothing}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren d$\overset{*}{\text{e}}$rtig
hundred four-and thirty
134
b. h$\overset{\varnothing}{\text{o}}$nderd t$\overset{*}{\text{i}}$en h$\overset{*}{\text{o}}$ofdstukken
hundred ten chapters

This might be related to the fact that initially there are three accents in (6), but only two in (5), and that human speakers try to avoid a series of three or more accents.

In the two other cases (7), *honderd* was erroneously assumed to be a unit, due to shortcomings of the current implementation, causing it to be subject to URR deaccentuation.

(7) a. ?h$\overset{\emptyset}{\text{o}}$nderd d$\overset{*}{\text{u}}$izend
hundred thousand

b. ?h$\overset{\emptyset}{\text{o}}$nderd k$\overset{*}{\text{a}}$sten
hundred cupboards

There were three cases in which another unit, *duizend* 'thousand', was deaccented. All three are structurally identical to the one in (8). As explained at the end of Section 7.3.5, there appear to be two ways to accent these long numbers. The first (8-a) is without an accent on *duizend* and with no internal AD-boundary, which was the option taken by ProsR, whereas the second (8-b) is with *duizend* accented and followed by an AD-boundary, which was chosen by the experts. This analysis is supported by the fact that all experts (with the exception of MS, who assigned an optional boundary) preferred an obligatory boundary after accented *duizend*. If ProsR would have inserted this AD-boundary, this would have blocked deaccentuation of *duizend* by URR. This disagreement can thus be attributed to a phrasing decision, which we will further discuss in Section 8.4.2.

(8) a. [$_{\text{AD}}$ h$\overset{\emptyset}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren v$\overset{\emptyset}{\text{i}}$jftig d$\overset{\emptyset}{\text{u}}$izend z$\overset{*}{\text{e}}$ven h$\overset{\emptyset}{\text{o}}$nderd twe$\overset{*}{\text{e}}$en
hundred four-and fifty thousand seven hundred two-and
d$\overset{\emptyset}{\text{e}}$rtig k$\overset{*}{\text{a}}$sten]
thirty cupboards
'154,732 cupboards'

b. [$_{\text{AD}}$ h$\overset{\emptyset}{\text{o}}$nderd v$\overset{*}{\text{i}}$eren v$\overset{\emptyset}{\text{i}}$jftig d$\overset{*}{\text{u}}$izend] [$_{\text{AD}}$ z$\overset{*}{\text{e}}$ven h$\overset{\emptyset}{\text{o}}$nderd twe$\overset{*}{\text{e}}$en d$\overset{\emptyset}{\text{e}}$rtig k$\overset{*}{\text{a}}$sten]

8.3.2.2 Too many accents

The majority of the ten differences involving too many accents occurred in TD (see Table 8.5). Eight of these involved a predicate that should not have been accented, suggesting SAAR did not work satisfactorily. However, as many as five of these were due to the fact that an adjacent argument was a placename which was erroneously backgrounded, a type of error that was discussed in the previous section. In (9), for example, the argument *Amsterdam* is backgrounded, which prevents SAAR from deaccenting (the verbal particle of) the predicate *aankomen*. This is thus an error in focus distribution, rather than an error of SAAR.

(9) ?Dan komt u in Amsterdam $\overset{*}{\text{a}}$an om ...
then arrive you in Amsterdam `prt` at ...

Table 8.5: Differences involving ProsR assigning too many accents.

Context:	TD:	NR:	Total:
predicate	7	1	8
unit	1	1	2
Total:	8	2	10

One of the remaining three errors in relation to SAAR is given in (10). Since the time of departure is definitely not an argument, SAAR cannot apply here. A possible explanation is that *gaan*, which also serves as an auxiliary of future tense in Dutch, is a semantically light verb, as most of meaning of 'going to' is expressed by the preposition of its argument (*naar* 'to') rather than by the verb itself (see also Section 7.3.6.2). Alternatively, it may be related to the observation by Baart (1987) that sometimes initial verbs do not appear to behave according to SAAR (Gussenhoven 1992b, Appendix).

(10) ?G$\overset{*}{\text{a}}$ daarna om $\overset{*}{\text{a}}$cht uur z$\overset{*}{\text{e}}$ven tw$\overset{*}{\text{i}}$ntig naar Br$\overset{*}{\text{e}}$da
go next at eight hour seven twenty to Breda
'Next, go to Breda at 8.27'

Finally, (11-a) is one of the two errors involving a unit term (cf. Section 7.3.5. Application of URR to *minuten* 'minutes' was blocked because of the following boundary inserted by ProsR. However, none of the experts assigned this boundary. Given the lack of a boundary, ProsR would not have accented *minuten* either. In other words, this apparent accentuation error can again be traced back to a disagreement with respect to phrasing. The second error of this type, occurring on *uur* in (11-b), cannot be explained in this way.

(11) a. ?u hebt tw$\overset{*}{\text{e}}$e min$\overset{*}{\text{u}}$ten # om $\overset{*}{\text{o}}$ver te stappen op ...
you have two minutes to `prt` to transfer onto ...
'you have two minutes to transfer to ...'

b. ?die tr$\overset{*}{\text{e}}$in gaat om tw$\overset{*}{\text{e}}$e $\overset{*}{\text{u}}$ur # via ...
that train goes at two hour via

To sum up, our survey of differences in pitch accent placement between ProsR and the human experts employing a majority criterion showed that the number of cases involving too few accents (23) was more than twice of that involving too many accents (10), indicating that ProsR tended to assign too few rather than too many accents. About two-thirds of the missing accents involved placenames, and resulted from the focus strategy. The remainder was due to overapplication of URR, part of which could be explained by intonational phrasing differences between ProsR and the experts. The majority (8) of the superfluous accents could be attributed to overapplication of SAAR. However, five of these were actually caused by unaccented placenames, indicating interaction between accentuation errors.

8.4 Results for intonational phrasing

We will first present the quantitative agreement with respect to intonational phrasing for the human experts, ProsR and KUNTTS. Next, we will analyse differences between the experts and ProsR to see if these can be traced back to particular phrasing rules.

8.4.1 Agreement in phrasing

Table 8.6 shows the number of intonational boundaries in TD and NR assigned by the human experts and the algorithms. Assuming that intonational boundaries could only occur between accentable words (cf. Section 8.2.1), the total number of boundary decisions was 587 (= 643−56) and 214 (= 274−60), respectively. While the previous section showed that the experts assigned about the same number of accents, the difference with respect to boundaries turned out to be larger. One of the experts (MS) assigned relatively few obligatory boundaries in both domains, and another (OC) relatively few optional boundaries in TD. Both algorithms, and KUNTTS in particular, predicted substantially fewer boundaries than the experts (with the exception of MS).[2]

Fleiss kappas were again calculated in order to estimate the agreement among the four human experts, using three categories (obligatory, optional, and no boundary). For TD, kappa was 0.58 (587 objects, $p < 0.000$); for NR, kappa was 0.65 (214 objects, $p < 0.000$). Kappas between 0.6 and 0.8 are generally considered to indicate an acceptable level of agreement (Rietveld and van Hout 1993).

Table 8.7 presents the agreement with respect to intonational phrasing, where the agreement indices were calculated in the same way as those for accentuation, that is, without taking optional boundaries into account. A graphical representation of the same data is shown in Figure 8.1. It can be observed that the agreement between the algorithms and MS was relatively high. Part of the explanation is that both assigned relatively few boundaries. Furthermore, the differences between the human experts and the algorithms were smaller than for accentuation, in particular for KUNTTS. As will be discussed in Section 8.5.3, this may be be attributed to the fact that boundaries occurred less frequently. Still, the agreement among the human experts was higher. With respect to TD, ProsR appeared to be slightly better than KUNTTS.

The result of a hierarchical cluster analysis (using complete linkage) on the agreement indices is shown in Figure 8.4. As expected, the results were less

[2] This is in accordance with the strategy employed by KUNTTS that a wrong boundary is worse than a missing boundary.

Table 8.6: Intonational boundaries in TD and NR assigned by the human experts and the algorithms, in both absolute numbers, and as a percentage of the total number of boundary decisions in TD (n=587) and NR (n=214).

	TD:				NR:			
	Obligatory:		Optional:		Obligatory:		Optional:	
Humans:	N:	%:	N:	%:	N:	%:	N:	%:
CO:	55	9.37	14	2.39	13	6.07	20	9.35
JC:	48	8.18	57	9.71	9	4.21	25	11.68
JH:	45	7.67	43	7.33	15	7.01	20	9.35
MS:	2	0.34	46	7.84	2	0.93	26	12.15
$\overline{x}$:	37.5	6.39	40	6.81	9.75	4.56	22.75	10.63
sd:	24.03	4.09	18.35	3.13	5.74	2.68	3.2	1.5

	TD:				NR:			
	Obligatory:		Optional:		Obligatory:		Optional:	
Algorithms:	N:	%:	N:	%:	N:	%:	N:	%:
KUNTTS:	6	1.02	0	0	2	0.93	0	0
ProsR:	29	4.94	0	0	4	1.87	0	0

clear cut than for the accentuation data. As far as TD was concerned, three experts (JC, JH, and CO) were clustered, whereas MS was different and was clustered with KUNTTS. ProsR appeared to be in between the three human experts on the one hand and MS and KUNTTS on the other. In the NR data, the same three experts were clustered again, whereas MS, ProsR and KUNTTS were clustered in another group. This clustering may be explained by the fact that JC, JH and CO assigned substantially more boundaries than MS, ProsR, and KUNTTS (cf. Table 8.6).

Table 8.7: Agreement in TD and NR with respect to intonational boundaries.

TD:						
	CO:	JC:	JH:	MS:	KUNTTS:	ProsR:
CO:		1.00	0.99	0.97	0.90	0.94
JC:	1.00		1.00	0.98	0.91	0.95
JH:	0.99	1.00		0.98	0.91	0.96
MS:	0.97	0.98	0.98		0.99	0.98
$\overline{x}$:	0.99	0.99	0.99	0.98	0.93	0.96
sd:	0.02	0.01	0.01	0.01	0.04	0.02

NR:						
	CO:	JC:	JH:	MS:	KUNTTS:	ProsR:
CO:		1.00	0.99	0.99	0.94	0.95
JC:	1.00		1.00	0.99	0.96	0.97
JH:	0.99	1.00		0.99	0.93	0.94
MS:	0.99	0.99	0.99		0.99	0.99
$\overline{x}$:	0.99	1.00	0.99	0.99	0.96	0.96
sd:	0.01	0.01	0.01	0.00	0.03	0.02

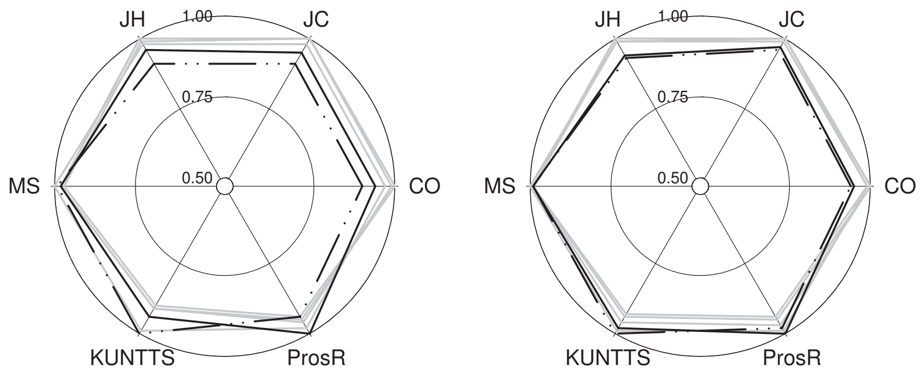


Figure 8.3: *A graphical representation of the data in Table 8.7, showing the agreement indices for TD (left) and NR (right) with respect to intonational phrasing. The centre of the circle represents an agreement of 0.5 and its outside represents an agreement of 1. ProsR is represented by the black line, KUNTTS by the dashed black line, and the four human experts by grey lines.*

```
                  Rescaled Distance Cluster Combine

          0         5        10        15        20        25
          +---------+---------+---------+---------+---------+

JC        --+
            +---+
JH        --+   |
                +-----------------------+
                |                       |
CO        ------+                       +-------------------+
                                        |                   |
ProsR     ------------------------------+                   |
                                                            |-
                                                            |
KUNTTS    ------+                                           |
                +-------------------------------------------+
MS        ------+
```

```
                  Rescaled Distance Cluster Combine

          0         5        10        15        20        25
          +---------+---------+---------+---------+---------+

JC        --+
            +-----+
JH        --+     |
                  +-----------------------------------------+
                  |                                         |
CO        --------+                                         |
                                                            |
MS        --------+                                         |-
                  +-------+                                 |
ProsR     --------+       +                                 |
                          +---------------------------------+
                          +
KUNTTS    ----------------+
```

Figure 8.4: Dendrograms resulting from an hierarchical cluster analysis on the agreement indices with respect to intonational boundaries in TD (top) and NR (bottom).

Table 8.8: Differences involving ProsR assigning too few boundaries.

Context:	TD:	NR:	Total:
in a coordination	0	5	5
before an extraposed PP	12	0	12
in a VP	2	0	2
after a topicalised AdvP	1	0	1
Total:	15	5	20

8.4.2 Disagreement in phrasing

For a qualitative comparison between ProsR and the human experts, we used the same type of majority criterion as we did for accentuation. That is, we assumed ProsR generated *too few boundaries* if either:

1. all four experts assigned an obligatory boundary, but ProsR failed to assign one;

2. three experts assigned an obligatory boundary and one assigned an optional boundary, but ProsR failed to assign one.

Similarly, ProsR generated *too many boundaries* if either:

1. none of the experts assigned a boundary, but ProsR assigned one;

2. three experts assigned no boundary and one assigned an optional boundary, but ProsR assigned one.

8.4.2.1 Too few boundaries

The differences involving too few boundaries are summarised in Table 8.8. Starting with TD, there were 12 differences due to the absence of a boundary before an extraposed PP. In these cases, the rule AD-in-S-2 (Section 4.8.1.2), which is responsible for inserting a boundary before an extraposed constituent, failed to apply. In (12), for example, there is only one accent in the first part of the sentence, which is not sufficient to allow AD-in-S-2 to insert the preferred boundary before the extraposed PP *om negen uur zesen veertig.*[3]

[3] In at least nine cases, this may seem to be caused by the non-accentuation of a placename, which is in turn due to focus error. For example, the placename *Amersfoort* was backgrounded and thus not accented, leaving only one accent in the first part of the sentence. However, if *Amersfoort* had been accented, then the verbal particle *aan* would have been deaccented by SAAR, and there would still be only one accent left.

(12) ?Die komt in Amersfoort aan om negen uur zesen veertig
which arrives in Amersfoort `prt` at nine hour six-and forty
'which arrives in Amersfoort at 9:46'

Two more differences were found within VP's. In (13-a), the experts preferred a boundary after *naar Delft.* However, AD-in-VP (Section 4.8.1.3) could not account for this boundary, because this VP daughter had only one accent. For the same reason, the rule could not provide a boundary after *Sneek* either in (13-b).

(13) a. ?u kunt van Apeldoorn naar Delft met een overstap
you can from Apeldoorn to Delft with one transfer

b. ?u kunt van Lelystad Centrum # naar Sneek met drie
you can from Lelystad Centre to Sneek with three
overstappen
transfers

The single case of a missing boundary after a topicalised constituent is given in (14). The topicalised AdvP *acht minuten later* received only two accents, which was not sufficient to trigger a following boundary by AD-in-S-1 (Section 4.8.1.2).

(14) ?acht minuten later gaat u met de intercity van Amsterdam
eight minutes later go you with the intercity from Amsterdam
naar Amersfoort
to Amersfoort

Turning to NR, missing boundaries occurred in the context of coordinated structures. For instance, the syntactic structure of the number 154,732 in (15) is a coordinated structure with *honderd vieren vijftig duizend* as the first conjunct and *zeven honderd tweeen dertig* as the second.

(15) ?honderd vieren vijftig duizend zeven honderd tweeen dertig
hundred four-and fifty thousand seven hundred two-and thirty
kasten
cupboards
'154,732 cupboards'

Since all but one of the initial accents in the first conjunct were deleted, the phrasing rule AD-in-coordination (Section 4.8.1.5) could not insert a boundary between the conjuncts. However, all four experts assigned an obligatory boundary in this position. We may explain this in terms of the complexity of the first conjunct: the first conjunct lacks the required number of accents, but it is sufficiently complex to trigger a boundary. There were two more cases that were very similar to (15). Notice also that this boundary would have been correctly inserted if the accent on *duizend* had not been deleted by URR, and vice versa, that deletion of the accent on *duizend* would have been blocked if this boundary had been present. This type of interaction between accentuation and phrasing was discussed in Section 7.3.6.4.

8.4.2.2 Too many boundaries

There was just one case with a superfluous boundary (16). It was due to AD-in-S-2, which inserted a boundary before extraposed S, as both AD's contained at least 2 accents.

			*	*			*					*
(16)	?u	hebt	twee	minuten	#	om	over	te	stappen	op	de	intercity
	you	have	two	minutes		for	`prt`	to	transfer	to	the	intercity
		*										
	naar	Leeuwarden										
	to	Leeuwarden										

To sum up, relative to the preferences of the human experts, ProsR tended to generate too few boundaries, but inserted almost no superfluous boundaries. For NR, an undershoot of AD-in-Coordination was observed, which could partly be explained by interaction with accentuation differences. For TD, the majority of the errors was due to undershoot of AD-in-S-2, which failed to insert a boundary before an extraposed PP.

8.5 Discussion

We will start with some general remarks regarding the evaluation. Next, we will compare our results for pitch accent placement with those of other approaches (Section 8.5.2), and similarly, compare our results for intonational phrasing with those of other approaches (Section 8.5.3).

8.5.1 General remarks

Comparison with earlier results In short, we have found that the accentuation by ProsR agreed with that by the human experts in 93% of the cases in TD and 95% in NR, and that its intonational phrasing agreed with that of the experts in 96% of the cases in both TD and NR (ignoring optional accents and boundaries, and taking the average agreement over all experts). In Chapter 3, we found that the simple version of the pitch accent placement algorithm (focus distribution, SAAR, and RR) resulted in an agreement of 94% with the consensus reference for the corpus of wild flower descriptions. This is comparable to the figures found here with the complete algorithm for pitch accent placement. It should be noted, however, that backgrounding was manually indicated in WFD, which prevented many accentuation errors related to the focus distribution. In addition, complex numbers did not occur in WFD either.

In Chapter 4, it was found that the $\mathcal{NEW}$ intonational phrasing agreed with the reference in 97.66% of the boundary decisions in the flower descriptions. This is just slightly higher than the 96% percent found here, suggesting that the possible effect of tuning on the corpus was relatively small.

Ideally, similar evaluations should be carried out in many different domains, or against a very large corpus that is representative of written text in general. Given the fact that large-scale prosodically-annotated corpora are not yet available for Dutch, such extensive evaluations are beyond the scope of the current work. Nevertheless, since the proposed algorithms for pitch accent placement and intonational phrasing show a similar performance when applied to three different text domains (flower descriptions, travel descriptions, and numbers), they appear to be fairly domain-independent.

Errors in focus distribution A number of accentuation differences (13 out of 23 involving too few accents) were caused by focus errors. PTDGen backgrounded, and therefore did not accent, placenames that were mentioned in the current paragraph (i.e. the description of the current section of the journey), whereas the experts preferred to accent every placename, unless it was mentioned in the directly preceding sentence. However, our objective has not been to predict focus distributions, but to account for their intonational implications (Section 2.5.1), which makes this type of error less relevant for our purposes.

Interaction For several differences, it was found that they were dependent on other differences. For instance, deaccenting of placenames caused failure of SAAR. Similar interactions were observed between URR and AD-in-Coordination. This suggests that decreasing one type of difference could simultaneously reduce other types of differences.

SLG versus TTS Comparing ProsR with KUNTTS, we found that the accentuation algorithm of KUNTTS performed considerably worse. This is to be expected given the advantages of SLG over TTS. We have not looked in detail at these errors, as evaluating KUNTTS was not the goal of this evaluation, but the majority of these can be attributed to the lack of discourse information (focus) in the input, and errors in word class assignment (POS tagging) and syntactic parsing. In addition, the TTS system is not tuned to accent long numerals in the form of words rather than digits.

As far as intonational phrasing is concerned, however, the differences turned out to be far less pronounced. A possible explanation is that the agreement index tends to favour an economical boundary insertion, as will be explained in Section 8.5.3, and virtually all the phrasing errors of KUNTTS involved too few boundaries. An additional disadvantage for the TTS system was that its intonational phrasing normally relies on punctuation, commas in particular, which was not present in TD.

Symbolic versus perceptual evaluation We chose to use a symbolic assessment of the acceptability judgements of our subjects because of the methodological problems associated with a perceptual evaluation (Section 1.1.4). This type of evaluation is similar to a production experiment: we compared the accents and boundaries generated by ProsR with those produced by the human experts. We assumed that there is a preferred intonational form, but allowed optional accents and boundaries in order to describe variants that are equally good. We also cited van Bezooijen (1989), who found a high correlation between a symbolic evaluation of the type conducted here and an additional auditory evaluation, which suggests a perception experiment would yield similar results.

Yet, from a more applied point of view, an additional perception experiment, in which the adequacy of the produced intonation in real speech is judged by listeners, would be desirable. After all, it may be that the lack (or presence) of an accent (or boundary) in one location is far more disturbing than in another. Consider the following pair repeated from Section 8.3.2.1: while the missing accent on *honderd* appears to be an acceptable variant in (17-a), it is a real error in (17-b), where it gives rise to a contrastive interpretation.

(17) a. ?h$\overset{ø}{\text{o}}$nderd en z$\overset{*}{\text{e}}$ven
hundred and seven
107

b. ?h$\overset{ø}{\text{o}}$nderd k$\overset{*}{\text{a}}$sten
hundred cupboards

Perhaps this difference is related to the reason for omitting the accent. In (17-a), *honderd* is deaccented for rhythmic reasons, whereas in (17-b), a deaccentuation can only be interpreted as lack of focus. A similar suggestion was made by Kruyt (1985, p. 81), where 'the effect of syntactic grouping' is roughly equivalent to the effect of RR:

> "These two kinds of leaving accentable words unaccented seem to be essentially different. The effect of syntactic grouping probably depends on such factors as speech rate and style of speech. In this case, leaving a word unaccented means the omission of an accent that could have been realized, and it does not necessarily make much difference to the listener whether the word is or is not accented. In contrast, the second kind of leaving accentable words unaccented, signalling 'givenness', comes closer to a conscious act of the speaker who has interpreted the text and is aware of the relations between the sentences. Probably, this kind of leaving words unaccented has greater relevance for communication."

In this connection, we make a final comment on the observation that ProsR produced too few accents, and too few boundaries in particular, rather than too many. As a trade-off exists between the two types of errors, we may choose to adapt the rules in such a way that more accents and boundaries are predicted, decreasing undershoot at the expense of overgeneration, but keeping the same

Table 8.9: Confusion matrices comparing the PROS accentuation with the accentuations by the subjects and the newsreader (Bezooijen 1989).

		Subjects:	
		+Accent:	–Accent:
PROS:	+Accent:	187	87
	–Accent:	11	303

		Newsreader:	
		+Accent:	–Accent:
PROS:	+Accent:	233	41
	–Accent:	28	286

overall agreement with the human experts. From a purely linguistic point of view, it is hard to decide which is better, but this may again be different from an applied point of view. For instance, there is experimental evidence that less accents are produced in fast speech (Caspers 1994, ch. 2). Therefore, the current model might fit a speech synthesiser that speaks fast, whereas the more generous model might be better suited to a synthesiser with a low speech rate.

8.5.2 Comparison with other approaches to accentuation

Van Bezooijen (1989) reports on a extensive evaluation of PROS (Kager and Quené 1987; Kager and Quené 1989; Quené and Kager 1989), a pitch accent placement algorithm for a Dutch TTS system that relies on heuristic rules. The test set consisted of 8 newspaper text (24 sentences) and 8 single sentences, with a total of 588 words. This material was presented in written form to 12 phonetically experienced subjects, who were asked to assign pitch accents according to their personal preferences. A reference accentuation was constructed by including accents assigned by at least 7 subjects. In addition, the material was read aloud by a professional newsreader, whose accents were transcribed by three intonational experts. A comparison of these two reference accentuations with the accentuation by PROS is given in Table 8.9. The agreement index between PROS and the subjects is 83.3%, whereas that between PROS and the newsreader is 88.3%. PROS assigned too many rather than too few accents, in particular in comparison to the subjects' accentuation. This behaviour is contrary to that of ProsR.

In addition, van Bezooijen conducted an auditory evaluation. For each sentence, three versions were synthesised according to the accentuations of PROS, the subjects and the newsreader, respectively. The same 12 subjects were asked to rate the adequacy of each utterance on a 10-point scale. The PROS accentuation received an average score of 6.0, whereas those of the subjects and the newsreader received scores of 7.7 and 7.4 respectively.[4] The difference between PROS and the two others turned out to be statistically significant at the 1% level. Interestingly, this suggests that even if an algorithm agrees with a human accentuation for 83-88% of the words, it can still be perceived as less adequate.

Nachtegaal (1997) describes a small-scale evaluation of accentuation in GoalGetter, an SLG system for Dutch that generates soccer reports (Theune 1997b; Theune 1997a; Theune, Klabbers, Odijk, and de Pijper 1997). The pitch accent placement algorithm of GoalGetter consists of a set of hand-crafted rules adapted from PROS-3 (Dirksen 1992; Dirksen and Quené 1993), which is in turn based on the linguistic theory developed by Baart (1987). In addition, GoalGetter includes a discourse-based approach to deaccentuation (van Deemter 1994; van Deemter and Odijk 1997). The system was used to generate three texts, comprising 22 sentences, or 211 words. These texts were read aloud by four non-professional speakers. Listening to the recordings, nine experts transcribed the accent locations. Next, the transcribed accents were compared with those predicted by the system. If 8 out of 9 expert listeners transcribed an accent, but the system failed to assign one, this was considered an error. Likewise, if 8 out of 9 listeners did *not* transcribe an accent, but the system assigned one anyway, this was counted as an error as well. This comparison resulted in 7 errors (an agreement of 96.68%) for the speaker that was best according to the listeners, and 10 (95.26%), 8 (96.21%) and 14 (93.36%) errors for the other speakers. The average agreement over all speakers is therefore 95.28%.[5] Most of the errors involved too few accents, indicating that GoalGetter is sparing with its accents. The fact that the agreement scores are considerably higher than those reported in van Bezooijen (1989) may be attributed to the benefits of a more advanced algorithm, and most importantly, to an SLG rather than a TTS context.

Hirschberg (1993) is an extensive study of pitch accent placement in American English using hand-crafted rules and automatic classification techniques. The input consisted of POS tags, the position of primary stress as indicated by a rule-based complex nominal analyser, and informational features resulting from modelling of the local and global focus structure and of contrastive contexts. The hand-crafted rules reached an accuracy of 82.4% on 2 radio news stories comprising a total of 9 minutes of speech from 9 different speakers. The

[4] Contrary to what might be expected, the scores for subjects and newsreader were not close to 10. Van Bezooijen attributed these relatively low values to, among other things, the poor quality of the synthesised speech.

[5] Although a kappa-coefficient of 0.827 is mentioned (Nachtegaal 1997, p. 30), indicating a rather high agreement among the transcribers, the actual agreement is not reported. That is, it is not known for how many of the 211 words 8 out of 9 transcribers agreed on either an accent or no accent. It may be the case, therefore, that many words were not taken into consideration during the comparison. In fact, comparing the data in Appendix 5.C shows that lowering the agreement criterion to 6 out of 9 listeners results in 14 errors (93.36%) for the best speaker, and 19 (91.00%), 14 (93,26%), and 24 (88.63%) errors for the others.

same rules scored a near-perfect 98.3% when applied to 302 short citation-form sentences/phrases (with an average length of 5.5 words, and a total of 1756 words), read by a single speaker under laboratory circumstances. This extremely high performance was attributed to the virtual lack of context for the sentences in this corpus.

In subsequent experiments, Classification and Regression Trees (CART) were used for automatic induction of accentuation models. When applied to the Audix corpus, which consists of 10 news stories read by a female professional newscaster (about 30 minutes of speech), the highest accuracy achieved was 80% (cross-validated). This assumed that intonational phrase boundaries were also available in the input. The same method was applied to a part of the ATIS corpus (about 24 minutes of spontaneous speech by 26 speakers, 298 sentences with an average of 11.5 words per sentence), which consists of speech of users who engaged in a task-oriented dialogue with a (simulated) travel agent. Again assuming the availability of intonational phrase boundaries, the cross-validated accuracy was 85.1%.

Pan and McKeown (1999) describe pitch accent assignment algorithms that employ Information Content (IC), an information theoretic measure for the relative informativeness of a word, which can be automatically derived from a large collection of texts. Their corpus contained 30 minutes of multi-speaker, spontaneous speech of doctors reporting on the post-operative status of a patient. The corpus was orthographically transcribed by a medical expert and intonationally labelled with pitch accents by a ToBI expert. Two methods were used to build pitch accent models on the basis of IC values and POS tags: a rule induction system called RIPPER, and Hidden Markov Models (HMM). Results were obtained using 10-fold cross-validation. RIPPER reached an accuracy of 73.71%, while the HMM accomplished 74.06%. It should be noted, however, that pitch accent placement was a difficult task here, as it concerned spontaneous speech. Pan and McKeown report that the Bell Labs TTS system, which normally scores around 80%, accomplished an accuracy of only 71.75% for this task.

Similar work is described in Pan and Hirschberg (2000), using uni-gram and bi-gram word predictability as a measure for word collocation, which is shown to be a useful predictor of pitch accent placement. The corpora used were in the same medical domain described above: a 50-minute, multi-speaker corpus of spontaneous speech and an 11- minute, single-speaker corpus of read speech. Again, RIPPER was used for automatic rule induction, and results were obtained using 5-fold cross-validation. The results, however, were limited to the prediction of accents on non-trivial base NP's, i.e. non-recursive noun phrases containing at least two elements. The best results were 85.03% accuracy for read speech, and 77.43% for spontaneous speech.

Finally, McKeown and Pan (to appear) describe results obtained with memory-based learning, another machine learning technique. The speech corpus consisted of recordings of a doctor reading aloud five reports that were generated by MAGIC (Medical Abstract Generation for Intensive Care), which is an

SLG system that automatically generates a post-operative status report for a patient who has undergone bypass surgery (McKeown, S.Pan, J.Shaw, Jordan, and Allen 1997).[6] Although at 60.41% the accuracy seemed to be rather low, McKeown and Pan argued that this was due to an unfair bias in their evaluation method. In fact, a subjective evaluation experiment showed that the memory-based system was judged significantly better than both the Bell Labs TTS system and the IC-based model described earlier.

To sum up, the accuracy of recent pitch accent placement algorithms ranges from about 85% for TTS, around 95% in an SLG context, up to 98% for citation-form sentences. Real comparison is difficult because of differences in language, text/speech material, and evaluation method. However, with an average agreement of 93% for TD and 95% for NR, the performance of ProsR is what may be expected for a moderately complex task in an SLG environment.

8.5.3 Comparison with other approaches to phrasing

With respect to intonational phrasing algorithms for Dutch, there appear to be no quantitative results available in the literature to compare the current results with. For English, some results on prosodic phrasing algorithms are available, but a comparison is not straightforward, because different assumptions are made regarding the type of prosodic breaks and the performance criteria.

Taylor and Black (1998) present algorithms for automatically assigning phrase breaks to unrestricted text, intended for use in TTS. First, the input text is converted into a sequence of part-of-speech tags. Next, a Markov-model is used for predicting the most likely sequence of phrase breaks for the the input, also taking the length of phrases into account. Read speech (news stories, weather reports, etc.) from the Spoken English Corpus were used for training. For testing, 10 separate stories were used, containing 7762 words and 1404 breaks. This material was manually labelled with two types of breaks: major and minor breaks. The former corresponds to our intonational phrase boundaries[7], while the latter may be comparable to our phonological phrase boundaries. Most of their results, however, concern the correct prediction of the *location* of prosodic breaks regardless of their *type* (i.e. major or minor), which precludes a comparison with our results.

In one case, however, Taylor and Black report results for a model that predicts both major and minor boundaries. The following measure was used to express its performance:

$$Junctures\ correct = \frac{N - D - S - I}{N} \times 100\%$$

[6] As a result, there is no need for manual labelling of the corpus, because all linguistic information is automatically generated, and the pitch accent model can be trained on detailed and reliable syntactic and semantic features. This shows how an NLG system can be put to use in generating a corpus for training inductive language models.

[7] "Major phrase boundaries delimit intonation phrases and are the only position where boundary tones can occur." (Taylor and Black 1998, Introduction)

N is the total number of junctures (word pairs) in the test set. D is the total number of *deletion errors*: a break that was marked in the reference sentence but not in the test sentence. I is the total number of *insertion errors*: a break that was marked in the test sentence but not in the reference. S is the total number of *substitution errors*: a break that occurred at the right place but was of the wrong type (i.e. major instead of minor or vice versa). The major-minor algorithm reached a junctures-correct score of 90.211%. This score, however, gives no further information about how many *major* breaks were correctly predicted, and can therefore not be directly compared with the agreement indices in Table 8.7.

Taylor & Black introduce two other performance measures called *Breaks-correct* and *Junctures-insertion* (or false insertions w.r.t. junctures), where B is the total number of junctures which are breaks:

$$Breaks\ correct = \frac{B - D - S}{B} \times 100\%$$

$$Junctures\ insertion = \frac{I}{N} \times 100\%$$

These measures are simililar to those of *precision* and *recall* as used in Information Retrieval. Taylor & Black argue that these are better performance indicators than junctures-correct.

> "The difference between breaks-correct and junctures-correct lies in whether non-breaks are included in the calculation. The junctures-correct score gives credit when both the test and reference sentences have a non-break at the same juncture, while the breaks-correct score only looks at junctures with breaks. In our data the number of non-breaks outnumbers the number of breaks by a ratio of about 4:1, and hence an algorithm which marks everything as non-breaks will score about 80% junctures-correct, but 0% breaks correct. [...] Because the breaks-correct score is not dependent on the relative distributions of breaks and non-breaks, we regard this as a better indicator of algorithm performance." (Taylor and Black 1998)

Admittedly, this argumentation make sense. In our case, where as few as 6% of the junctures in TD and 5% of those in in TD were obligatory boundaries (cf. Table 8.6), an algorithm that assigns no boundaries at all would still score about 95% junctures-correct.[8]

[8]Although in principle a similar argument can be made with respect to accentuation, it is far less appropriate, because the number of accented words is usually much closer to the number of non-accented words. Compare for instance Table 8.1, showing that about 40% of the words in TD were accented, and about 61% of those in NR.

Table 8.10: Confusion matrix expressing the agreement in boundary placement in TD between ProsR and human expert CO.

		CO: Obligatory:	Optional:	None:	Total:
ProsR:	Obligatory:	23	1	5	29
	Optional:	0	0	0	0
	None:	32	13	513	558
	Total:	55	14	518	587

As our algorithm predicts only major breaks (intonational boundaries) and no minor breaks, substitution errors (S) did not occur.[9] For example, breaks-correct for Table 8.10 was calculated with a zero substitution error:

$$Breaks\ correct = \frac{B - D - S}{B} \times 100\% = \frac{55 - 32 - 0}{55} \times 100\% = 41.82\%$$

Junctures-insertion was calculated in the following way:

$$Junctures\ insertion = \frac{I}{N} \times 100\% = \frac{5}{587} \times 100\% = 0.85\%$$

Table 8.11 presents the breaks-correct and junctures-insertion scores when ProsR is compared with the human experts in this way. Table 8.12 repeats the results reported in Taylor and Black (1998). At first sight, our results appear to be substantially worse. However, this comparison is rather misleading. Recall from the start of Section 8.4.1 that we only took intonational boundaries *within* sentences into account. That is, we did not count the trivial intonational boundary at the end of a sentence (or phrase in NR) as correctly predicted. In contrast, Taylor & Black tested whole texts:

[9] It should be noted that Taylor and Black's definition of a substitution error cannot be applied to our results. Major and minor breaks refer to boundaries of a different *type*, i.e. with a different acoustic realization and perceptual salience. Our distinction between obligatory and optional boundaries refers to the *occurrence* of a boundary, not to boundaries of a different nature. For instance, substitution of a major for a minor break is different from the substitution of an obligatory for an optional boundary. While the former is clearly an error, the latter is just one of two acceptable realizations of an optional boundary. This also explains why a junctures-correct calculation cannot be applied to the results in Table 8.7.

Table 8.11: Breaks-correct and junctures-insertion when comparing ProsR with human experts, excluding trivial sentence-final breaks.

	Breaks-correct (%):		Junctures-insertion (%):	
	TD:	NR:	TD:	NR:
CO:	41.82	30.77	0.85	0
JC:	43.75	33.33	0.17	0
JH:	48.89	20	0.17	0
MS:	100	50	1.70	0
$\overline{x}$:	58.62	33.53	0.72	0

Table 8.12: Minor and major phrase breaks for a part of the Spoken English Corpus as predicted by the HMM-based prosodic phrasing algorithm described in Taylor and Black (1998).

Model:	Minor cor.:	Major cor.:	Junct. cor.:	Junct. insertion:
1-gram	60.125	56.415	90.128	2.035
6-gram	68.703	59.502	90.211	3.923

Table 8.13: Breaks-correct and junctures-insertion when comparing ProsR with human experts, including trivial sentence-final breaks.

	Breaks-correct (%):		Junctures-insertion (%):	
	TD:	NR:	TD:	NR:
CO:	71.17	87.67	0.78	0
JC:	74.04	91.30	0.16	0
JH:	77.23	84.00	0.16	0
MS:	100	98.39	1.56	0
$\overline{x}$:	80.61	90.34	0.67	0

> "[...] our system is designed and optimised solely for use in a unconstrained text-to-speech system. To that end we test the system on the input format it can expect in real use: a continuous stream of ascii characters with no division into words, sentences or other units. The division of the text into such units is performed automatically within the initial text processing modules of the text-to-speech system." (Taylor and Black 1998)

In the reported experiments, a POS tagger was used that has an overall performance of 97% and includes a tag for final punctuation symbols (period, question mark, and exclamation mark), for which performance is probably near perfect. This means that the end of a sentence was virtually always correctly identified, and thus the reported results include many trivial sentence-final boundaries.[10]

If we include the trivial sentence-final boundaries as well (56 for TD and 60 for NR), we obtain the results in Table 8.13, which clearly looks much better than those in Table 8.11. The average breaks-correct scores compare favourably with the major correct scores in Table 8.12. It should be noted, however, that the latter also include substitution errors (minor instead of major break), which do not occur in our results. Assuming that major break insertions (instead of no break) make up about half of the 2 to 4% junctures-insertion in Table 8.12, junctures-insertion by ProsR is still lower as well.

Ostendorf and Veilleux (1994) present another stochastic approach to prosodic phrasing, using a dynamic programming algorithm for finding the most likely prosodic parse. Their test data consisted of 5 versions of an FM radio news story, containing 23 sentences and 386 words, read by 4 different announcers (2 male and 2 female). Prosodic break levels from 0 to 6 were manually

[10] As a matter of fact, Taylor and Black (1998) describe a deterministic phrasing algorithm which places a phrase break after every punctuation mark in the text (abstracting away from the major-minor distinction). This algorithm has a breaks-correct score of 54.274% and a junctures-insertion score of 0.852%, indicating that punctuation is a very reliable indicator of phrase breaks, and that punctuation nearly always implies a phrase break.

Table 8.14: Confusion matrix for predicted breaks in Ostendorf and Veilleux (1994) (using the hierarchical system, without syntax, and without minor phrase length ratio test).

		Actual:		
		Major:	Minor:	No-break:
	Major:	49	6	5
Predicted:	Minor:	5	12	6
	No-break:	7	10	286
	Total:	61	28	297

assigned by three listeners, and from this minor and major breaks were derived. Their major breaks are equal to intonational phrase boundaries (Ostendorf and Veilleux 1994, p. 28).

The best result achieved by the hierarchical system in Ostendorf and Veilleux (1994) is repeated in Table 8.14. It was calculated in the following way.

> "A predicted sentence is compared to each of the five spoken versions, and the closest spoken version is used as a reference for that sentence (The closeness of parses is measured using a Euclidian distance with 0 for no break, 1 for minor break and 2 for major break.) Then the correspondence between predicted and observed breaks is tabulated in a confusion matrix." (Ostendorf and Veilleux 1994, p. 40)

Using the performance measures proposed by Taylor and Black (1998), this gives a junctures-correct of 89.90%, major-breaks-correct of 80.33% and junctures-insertion of 2.85%. On the one hand, these figures include 5 substitution errors and 6 insertion errors due to *minor* break insertion. A version of the algorithm that assigns only major breaks is likely to perform even better. On the other hand, the figures in Table 8.14 resulted from comparing each sentence with the *best* match of the five available versions. According to Ostendorf and Veilleux (1994, p. 40), the correct prediction and false prediction rates that correspond to the average over the five separate test versions (as the $\overline{x}$ in Table 8.13) are roughly 10% lower. In addition, these results also included sentence-final major breaks[11], which means that 23 of the 49 correctly predicted major breaks were in fact trivial. Taking these reservations into account, we

[11] "Sentence breaks are deterministically assigned at periods, but these are included in the performance results reported here (as major breaks) to be consistent with results reported elsewhere." (Ostendorf and Veilleux 1994, p. 40)

Table 8.15: Confusion matrix for predicted breaks in Ostendorf and Veilleux (1994) (using simple classification trees, with syntax, and without minor phrase length ratio test).

		Actual: Major:	Minor:	No-break:
Predicted:	Major:	57	3	4
	Minor:	0	0	0
	No-break:	5	18	298
	Total:	62	21	302

can tentatively conclude that the major-break-correct score is comparable to ours in Table 8.11, but the junctures-insertion rate is almost twice as high.

Ostendorf and Veilleux also trained classification trees for prosodic phrasing. The results for the best tree are repeated in Table 8.15. This model predicted only major breaks, which makes a comparison easier. We can assume that the distinction between minor and no-break is irrelevant, which means that we collapse the second and third column of the confusion matrix. This gives a junctures-correct score of 91.97%, a major-breaks-correct score of 91.94%, and junctures-insertion score of 1.81%. Excluding the trivial sentence-final major-breaks decreases the major-breaks-correct score to 87.18%. We probably have to subtract another 10% to compensate for the effects of the comparison with the *best* version only, but a breaks-correct of about 77% is still considerably better than our results in Table 8.11.

Finally, Ostendorf and Veilleux also report results for the rule-based phrasing algorithm by Bachenko and Fitzpatrick (1990), which predicts both major and minor breaks.[12] This algorithm produced a junctures-correct score of 86.79%, a major-break-correct score of 77.78%, and a junctures-insertion score of 4.92%. These results stem from a comparison with the best matching sentence of the five versions available, and include sentence-final breaks. The value for junctures-insertion is particularly high in comparison with the other algorithms.

To sum up, comparing our intonational phrasing results with other results

[12] However, their interpretation of prosodic breaks appears to be different from that in Ostendorf and Veilleux (1994): “In the transcriptions, we both distinguished three types of prosodic event: a primary phrase boundary, a secondary phrase boundary, and the absence of a boundary. The most salient characteristic of the primary phrase boundary was a pause, while that of the secondary boundary involved a change in pitch.” Bachenko and Fitzpatrick (1990, p. 155). This suggests that the Ostendorf and Veilleux's major breaks (and our intonational boundaries) should actually be compared to Bachenko and Fitzpatrick's *secondary* boundaries. For this reason, we have not attempted a direct comparison of our results with those in Bachenko and Fitzpatrick (1990).

in the literature is not without difficulties. Both Taylor and Black (1998) and Ostendorf and Veilleux (1994) describe algorithms that predict major as well as minor breaks, and the results they report often concern an average over both types, precluding a direct comparison with our results, which concern only major breaks (intonational boundaries). In addition, different evaluation methods and measures are used. For instance, counting the trivial sentence-final major breaks can produce flattered results. Likewise, comparison with the nearest version as in Ostendorf and Veilleux (1994) is certainly defendable, but flatters the results as well. Still, abstracting away from details, our results appear to be somewhat better than those of Taylor and Black (1998) and Bachenko and Fitzpatrick (1990), comparable to those of the hierarchical model in Ostendorf and Veilleux (1994), and somewhat worse than those of the classification tree model in Ostendorf and Veilleux (1994) when breaks-correct scores are considered. In all cases, however, our insertion of erroneous boundaries (junctures-insertion) is considerably lower, indicating that ProsR hardly overgenerates. As a consequence, ProsR has the best performance of all when agreement or junctures-correct scores are considered (again, abstracting away from some details). Since there is a trade-off between breaks-correct and junctures-insertion, this suggests that the performance of ProsR can brought close to that of the classification tree model by inserting more boundaries.

8.6 Summary

This chapter reported on an experimental evaluation of the output of our SLG system CONPAS, more in particular the algorithms for pitch accent placement and intonational phrasing of its prosodic realization module ProsR. The evaluation involved a comparison at the symbolic level of accents and boundaries generated by the system with those assigned by four human experts. The material consisted of eight travel descriptions (TD, 643 words) and 60 expressions containing numerals (NR, 274 words). It was found that ProsR's accentuation had an average agreement of 93% for TD and 95% for NR, while its intonational phrasing showed an agreement of 96% for both domains. Comparison to other approaches suggests that its accentuation is better than that of most TTS systems, and comparable to other SLG systems, while its phrasing is superior to other approaches in terms of agreement, and competitive in terms of other performance measures. Finally, the fact that agreement among the human experts is still higher indicates that there is still room for improvement of accentuation and phrasing models.

Chapter 9

General Discussion

9.1 Introduction

The research objective of this study has been to design, build, and test a linguistically-based SLG system that generates speech with pitch accents and intonational boundaries that are most acceptable to experts (Section 1.3). Although the system generates speech output, our work focused on generating intonation at the phonological level, largely abstracting away from phonetic implementation. Similarly, little attention was paid to matters of tune choice. The scope of the work was further restricted by language (Dutch), genre (read speech), application domain (WFD and PTD), and a commitment to a particular theoretical framework (Chapter 2).

According to our research objective, the judgements of intonational experts were taken as a reference for system output. One reason is that acoustically-based approaches for assessing the intuitions of native speakers are faced with several methodological problems (Section 1.1.4). Therefore, we chose to use a symbolic assessement of the intuitions of speakers. This requires experts, because they are able to express their intonational intuitions directly in terms of a symbolic representation. Another reason is that experts have developed a greater awareness of the significance of intonation, and their judgements are taken to be superior to those of average native speakers. In an attempt to deal with the variability inherent to intonation, the intended output was narrowed down to intonation that is *most acceptable* according to the experts (Section 1.1.3). We noted that this goal is well suited to the research paradigm of NLG/SLG, where systems try to generate the *best* form of output (given the content and context of the message), rather than a form that is merely acceptable.

Designing and building an SLG system, generally conceived of as a combination of NLG and speech synthesis, led to a number of research questions regarding its architecture (Section 1.3):

1. What is the best way to combine NLG and speech synthesis?
2. What is the best place in the system to control the intonation?
3. What modifications to the 'standard' architecture of NLG and speech synthesis systems are required?
4. What additional information should be generated by the NLG system, and how can this be accomplished?

These questions were primarily addressed in Chapter 5, while the last was implicit in most of the other chapters. In the first part of the current chapter, we will review the answers to these questions provided by our work. In Section 9.2, we will recapitulate the results, compare them to other work, address unsolved issues, and speculate on further improvements.

Returning to our research objective, the intended SLG system is linguistically-based, because our motivation has been primarily linguistic. Our intention has been to use SLG as a testbed for linguistic theories that cover the relation between the intonation of an utterance and other aspects of its linguistic structure and context. The theoretical framework was presented in Chapter 2, and subsequently refined in the following chapters: phonological phrasing in Chapter 3, intonational phrasing in Chapter 4, and pitch accent placement in Chapter 7, while Chapter 8 reported on a final evaluation. This work has been driven by two research questions relating to the linguistic theory on which we based our efforts:

1. How adequate is it?
2. How can it be improved?

We will address these questions in the second part of this chapter (Section 9.3), in which we summarise our results, compare it to other work, and address remaining challenges. We will finish this chapter with a discussion of some of the implications of our work for applied speech synthesis systems in Section 9.4.

9.2 The architecture of an SLG system

9.2.1 Combining NLG and speech synthesis

Our first research question concerned the best way to combine NLG and speech synthesis in an SLG system. It was argued that using the text output of an NLG system as input to a TTS system does not make a good interface, because it fails to exploit the discourse, semantic, and syntactic information produced in the text generation process, for the generation of intonation (Section 5.2.3). Instead, the NLG system and the speech synthesiser should be integrated to a much higher degree, allowing detailed and reliable linguistic information to be used for generating adequate intonation, and prosody in general. This point has been made by many researchers, and forms the background for most work on SLG.

Our proposal for an SLG architecture resulted from several considerations. To begin with, it is an extension of the modular pipeline architecture that is currently regarded as the standard model for NLG systems. The model allows information to be concentrated and encapsulated in modules, facilitating design, implementation, documentation, maintenance, and modification of the system. In addition, it provides an opportunity to experiment with different versions of a particular module, or to reuse one or more modules in a new system. The same considerations argue in favour of using a pipeline architecture, in which the flow of information is strictly sequential, and no feedback is allowed. Although pipeline architectures have been criticised for being too restrictive (Inui, Tokunaga, and Tanaka 1992; Rubinoff 1992), we did not encounter any serious problems, suggesting that it is a viable choice for the purposes of SLG.

A second consideration affecting the architecture stems from a further research question: what is the best place in the system to control intonation? We argued that prosodic realization, i.e. generating a phonological description of the prosody (including the intonation), cannot be reduced to the standard NLG tasks of either microplanning or structure realization (Section 5.2.4). Nor should it be left to the speech synthesiser, which approximates the process by heuristic rules or purely stochastic models intended for a TTS environment. Instead, it should be recognised as a complex, inherently linguistic process, requiring a separate module. Therefore, we proposed a prosodic realization module that connects the syntactic realizer to the speech synthesiser.

A third consideration was related to reusability. For several languages, reusable surface realizers are available which translate syntactic specifications into well-formed expressions. These allow developers of NLG systems to ignore the low-level syntactic details of the target language, and instead to focus on issues of document structuring and microplanning. In contrast, reusable document planners are not yet available, primarily because planning is intrinsically bound to a certain domain. We conjectured that prosodic realization is fairly domain-independent as well. This opens up the possibility for a reusable 'content-to-speech' system that serves as the front-end for an SLG system, and allows developers of an SLG system to abstract away from the prosodic details of the target language as long as they make sure their planning components deliver the required input.

On the basis of these considerations, we proposed an SLG architecture that extends the standard modular pipeline NLG architecture with a separate module for prosodic realization, and that uses a domain-independent, reusable content-to-speech system; see Figure 9.1. Accordingly, our SLG systems, WFDGen and PTDGen, have two major parts:

1. a domain-specific text planner (PlanR) that generates the *content* in the form of a text specification containing a number of abstract phrase specifications

2. a domain-independent front-end called CONPAS ('CONtent to Prosodically Adequate Speech') that is responsible for the *form* of the spoken output.

The fact that CONPAS has been successfully used in building SLG systems for two different application domains shows the viability of a domain-independent, reusable content-to-speech system. As a result, we have a platform that allows us to build laboratory SLG systems for other domains with far less effort than it would take to build them from scratch.

Another of our research questions asked what modifications to the 'standard' NLG and speech synthesis architecture are required. As far as NLG is concerned, one of the most obvious modifications is that spoken output obviates the necessity of graphemic output by the surface realizer, and therefore also the need for a structure realization component that takes care of orthographic processing (punctuation) and visual lay-out.[1] Instead, the system needs a syntactic realizer that outputs syntactic structures, and transfers other useful discourse and semantic information from earlier stages of the generation process.

Speech synthesisers have traditionally been used in a TTS environment, where a stage of text preprocessing and linguistic analysis precedes actual synthesis. An SLG environment, however, has no need for expanding abbreviations, tagging of word classes, or syntactic parsing, because the resulting information is available from the NLG system. Preceded by a prosodic realization module generating prosody at the symbolic level, the tasks remaining for the speech synthesiser are the application of segmental rules, phonetic implementation of prosody in terms of fundamental frequency, duration, and intensity, followed by actual synthesis of the wave form. In our system, we chose to employ the existing lexica, and therefore left grapheme-to-phoneme conversion to the speech synthesiser. However, the syntactic realizer could equally generate phonemic forms instead of, or in addition to, the graphemic forms.

In the next two sections, the NLG modules (the text planner and the syntactic realizer) will be discussed in the light of the two remaining research questions: what additional information must be generated by the NLG system to facilitate the control of intonation, and how can this be accomplished?

9.2.2 The contribution of the text planner

Recall that the task of a text planner is to determine the content of the text, and to structure this content in the form of an abstract text specification (see Section 5.2.1 for a more precise description). It was argued that the distinction between document planning and microplanning, while fully compatible with the current architecture, is not essential for SLG as far as intonational phrasing and pitch accent placement are concerned. As our interest did not concern the planning process itself, but rather the question what information a

[1] With a view to possible NLG applications, our syntactic realizer is capable of generating text as well, although structure realization is still rather limited.

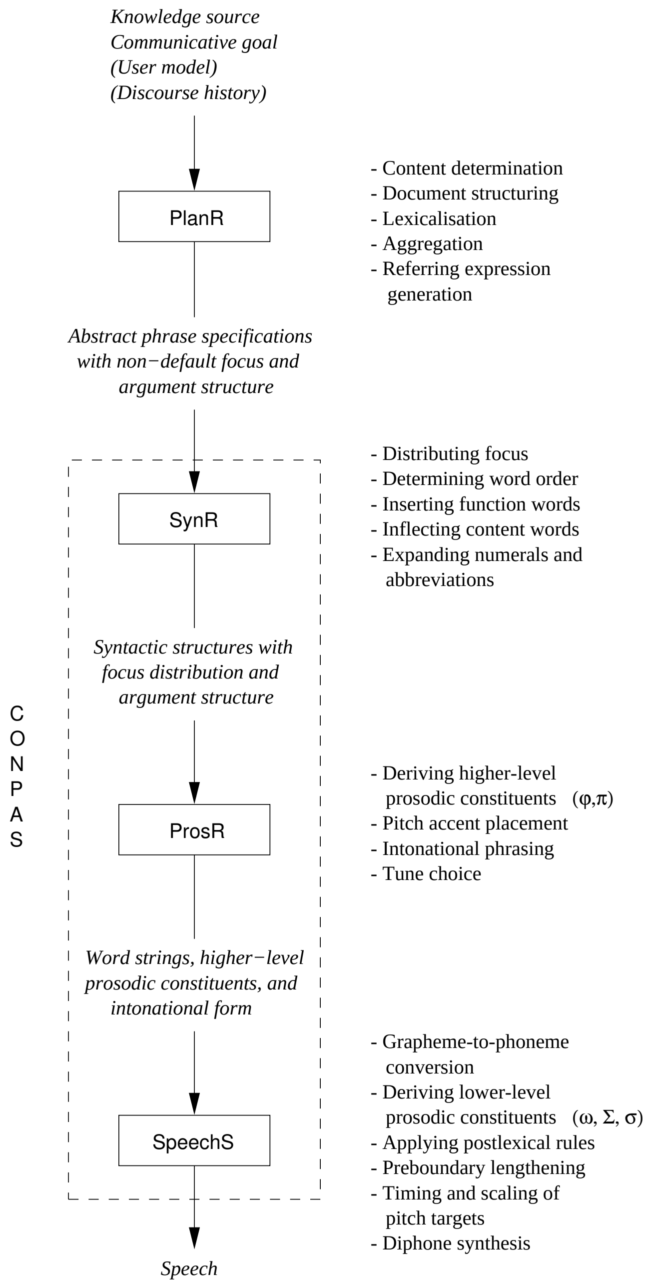


Figure 9.1: Overview of the input/output and tasks of each module in an SLG system based on CONPAS.

text planner should contribute for prosodic realization, we used relatively simple, domain-specific text planners (PTDPlanR and WFDPlanR) in which both planning processes were merged. Moreover, since our SLG systems were primarily intended as a test bed for linguistic theories, we aimed at generating maximal variation on the basis of limited input. This was accomplished using a straightforward approach to text planning, employing template-based generation techniques coded in a general-purpose programming language. This proved to be a concise, reliable and effective approach for our purposes.

The output of PlanR consists of a list of Abstract Phrase Specifications (APS's); we will consider the adequacy of this level of representation in the next section. Besides the abstract syntactic form, PlanR contributes two other types of information to an APS that are important for controlling the intonation. First, it specifies non-default focus, that is, it may explicitly focus or background elements of an APS. This affects the accentuation, as the focus distribution determines initial pitch accent placement. Rather than predicting focus distributions, which is a task for discourse theories, our work concerned their intonational *realization* (see also Section 2.5.1). Therefore, the non-default focus decisions are currently hard-wired in the APS templates used by PlanR. As the final evaluation revealed (Section 8.3.2), some accentuation errors could be traced back to in inappropriate distribution of focus; in particular, placenames were too often backgrounded by PTDPlanR. However, more advanced text planners will certainly be capable of dynamically generating a more reliable focus distribution by keeping track of the information status of entities in a discourse history, combined with pragmatic reasoning about cooperative behaviour, and other factors that may affect a speaker's focus decisions. As a matter of fact, focus has already been used in NLG work on text planning, e.g. McKeown (1985b), McCoy and Cheng (1991), Jokinen, Tanaka, and Yokoo (1998), Fraczak, Lapalme, and Zock (1998).[2] One approach at context modelling specifically for SLG is described in van Deemter and Odijk (1997).

The second type of information contributed by PlanR is the argument structure (predicates, arguments, modifiers). In our approach to accentuation, the argument structure is used by SAAR to decide whether the predicate needs a pitch accent. Like the focus distribution, the argument structure is currently explicitly encoded in the APS templates used by the text planner. It is nevertheless reasonable to assume that this information originates from the text planner, because knowing the number and type of arguments of a verb is essential during microplanning, in particular for lexicalisation and aggregation (Stede 1995). The argument structure must therefore either be listed in the lexicon or encoded in the microplanning algorithms (e.g. decision trees). In addition, we found that the argument structure is required during syntactic realization for determining VP-internal word order (Section 6.3.7.4).

Contrary to pitch accent placement, intonational phrasing appeared to require no additional information from the text planner (apart from abstract

[2] Although, rather confusingly, *focus* is defined in some NLG work as 'that which is being talked about'.

syntax). This suggests that whereas accentuation is closely related to the informational and semantic structure of an utterance, intonational phrasing is independent of it, and relies only on the syntactic structure. Some other approaches to SLG (Steedman 1991; Prevost and Steedman 1994; Hiyakumoto, Prevost, and Cassell 1997) assume that in addition to the focus distribution, the information structure of an utterance consists of a *theme* (what the participants have agreed to talk about) and a *rheme* (what the speaker has to say about the theme), and claim that theme and rheme must belong to separate intonational phrases, characterised by different tunes. However, their examples are limited to a rather specific type of contrast found in a certain type of dialogue. Actually, we have not encountered any examples of intonational phrasing or tune choice in our application domains that suggest the need to incorporate phrasal distinctions related to theme/rheme or topic/comment.[3]

In addition to abstract syntax, focus distribution and argument structure, a text planner may contribute several other types of information that can be used to control the intonation. One of these is the text specification. It is commonly assumed in NLG that text planning results in a hierarchically-structured text specification in the form of a tree whose the terminal nodes are phrase specifications and whose non-terminal nodes are textual units such as *paragraph* and *list*. In PTDGen, for instance, each description of a travel section (from A to B) forms a separate paragraph. In the case of text generation, the surface realizer maps the textual units onto appropriate punctuation symbols and mark-up codes for document presentation. For instance, a paragraph may be signalled by indentation or a blank line, whereas the items of a list may be preceded by numbers or bullets. In the case of SLG, as mentioned earlier, there is no use for either punctuation or visual lay-out. However, there is abundant evidence that the textual structure, often referred to as the global discourse structure (Grosz and Sidner 1986), is signalled by intonational and durational cues. For example, the end of discourse units is signalled by changes in pitch range, pitch register, and pitch contour (Swerts and Geluykens 1993). Likewise, paragraphs are usually closed with a pause that is longer than that between utterances within a paragraph (Lehiste 1975; Lehiste 1979), and intonational peaks are pushed rightwards at the start of a paragraph (Wichmann, House, and Rietveld 2000). Similar findings are reported in Grosz and Hirschberg (1992), Sluijter and Terken (1993), Swerts, Collier, and Terken (1994), Swerts (1994), van Donzel (1999), Noordman, Dassen, Swerts, and Terken (1999). It is obvious that the availability of a text specification in SLG provides an excellent opportunity to explore the relation between text structure and prosody.

In fact, our use of Downstep for simple coordinations (lists), and a %H boundary tone to mark the start of new paragraph can be considered as instances of this relation between text structure and prosody. However, as tune choice is beyond the scope of our research objective, we have not further explored this

[3] We did, however, insert intonational boundaries around appositions. Since an apposition may be interpreted in informational terms as a piece of backgrounded information, this may be considered a relation between informational structure and intonational phrasing.

topic. For the same reason, we did not go into the phonetic implementation of text structure. However, we think that this would require at least two things. First, the textual units (i.e. the non-terminal nodes in the text specification) have to be made available to the speech synthesiser, possibly in the form of prosodic annotations similar to those currently used for AD-boundaries and φ-boundaries. Second, the speech synthesiser needs additional rules for interpreting these textual units in terms of global intonational and durational phenomena (like gradual pitch lowering and pause insertion).

Besides signalling the hierarchical structure of the discourse, intonation can have other discourse functions as well, for instance, signalling the type of speech act. In fact, we made a crude distinction between questions and declaratives to guide the choice of the final pitch accent in an utterance (H*L or H*LH for questions versus the default H*L). Of course, the relation between speech acts and tune choice deserves a more elaborate approach. SLG work by Teich, Haen, Grote, and Bateman (1997) explored the role of intonation in information exchange dialogues, relating tune choice to dialogue acts such as requesting, confirming, and updating information. In addition, tune may serve as a cue to turn-taking in dialogue, which is clearly relevant for any SLG system embedded in a spoken dialogue system.

A cross-fertilisation between intonational and SLG research could result in significant contributions here. On the one hand, SLG systems can be used as a test bed by building text planners that supply the prosodic realizer and speech synthesiser with the type of information that is required for the prosodic realization and phonetic implementation of the discourse structure. On the other hand, some practical formalisations of discourse theory and techniques of implementation have been developed in the field of NLG. In particular, Rhetorical Structure Theory (RST) (Mann and Thompson 1988) has stimulated a lot of work and has proven to be very useful for NLG (Hovy 1988; Hovy 1991). It would be interesting to see what RST has to offer intonational research with respect to discourse prosody.

9.2.3 The contribution of the syntactic realizer

In NLG, syntactic realization refers to translating phrase specifications to grammatical expressions. In our SLG systems, it is taken to include the distribution of (default) focus, and the expansion of abbreviations and numeral expressions. In addition to words, the output must contain the syntactic structures. Syntactic realization by simple string concatenation would fall short of supplying the necessary information, indicating that SLG requires a more advanced form of syntactic realization in order to fully exploit its advantages over TTS. As a matter of fact, in the ongoing discussion about the pros and cons of NLG techniques in comparison to straightforward template-based approaches (Reiter 1995), this is an additional argument in favour of NLG.

We implemented a syntactic realization module (SynR) that meets these requirements by mapping APS's onto syntactic trees.[4] An APS is a skeletal structure consisting of words, their hierarchical relations, and their informational, semantic, and syntactic features. We argued that an APS has the desired level of abstraction to function as input to a content-to-speech system (5.3.3). On the one hand, concrete specifications like canned text or string templates are not an option, because the corresponding surface realizer would fail to provide the syntactic structure required for prosodic realization. On the other hand, highly abstract specifications like lexicalised case frames appear to hold too much information for our purposes.[5]

SynR's output consists of syntactic structures in the form of labelled-bracket structures, enhanced with features for focus and argument structure. We proposed a syntactic analysis of Dutch that relies on minimal assumptions about the phrasal structure, and is compatible with most syntactic theories (Section 2.6). Nonetheless, it proved to provide sufficient input for the deaccentuation and phrasing rules, suggesting that syntactic details such as dominance relations within the verb cluster or among extraposed constituents are irrelevant for prosodic realization.

We will not repeat the discussion on SynR from an NLG point of view here (see Section 6.5), but rather point to three more issues in relation to prosodic realization. With respect to focus, SynR implements a default distribution based on the content/function word dichotomy. In addition, it applies focus projection rules to provide SAAR with the necessary information. Besides its use for pitch accent placement, the focus distribution is also used for determining the word order in a VP: within the constraints imposed by the grammar, non-focused constituents typically precede focused ones. This is just one example of the interaction between word order and accentuation mediated by focus. Thus, instead of deriving pitch accent locations from the syntactic structure, as in earlier treatments (Chomsky and Halle 1968), both syntax and pitch accent distribution are seen as dependent on the focus distribution. Earlier, we mentioned the fact that narrow focus allows a marked ordering of the adjectives in an NP. In addition, there appears to be a relation between the accentuation and the order of the verbs in Dutch verb clusters (Swerts 1999), and certain marked syntactic structures (e.g. clefts and pseudo-clefts) force particular focus patterns and corresponding accentuations (Rochemont and Cullicover 1990; Verhagen 1986). This an area that deserves further investigation, in which SLG can be instrumental.

With respect to numerals, we proposed a new syntactic analysis of numeral phrases in Dutch. It was shown that this analysis provides the required information to derive phonological phrases and apply rhythm rules, resulting in a precise account of the accentuation of numerals (see Section 7.3.5).

---

[4] We partly ignored punctuation and other matters of orthographic realization, as these are – strictly speaking – irrelevant in SLG. Nevertheless, with some additional effort in this direction, SynR can be used as a syntactic realization module for building Dutch NLG systems intended for research, which is a valuable spin-off of the current project.

[5] Section 6.5.1 discussed some additional advantages of APS's over string templates in relation to NLG in general.

Finally, it was found that SynR's output must contain some morphological structure as well. The structure of *phrasal compounds* like *groengeel* 'greenish-yellow', of which the place and number of accents is context-dependent, must be transparent to ProsR. This was accomplished by retrieving these forms from the syntactic lexicon. In addition, the LRR and URR deaccentuation rules rely on features marking exceptional words that also originate from the lexicon.

9.3 The linguistic theory

The prosodic realization module (ProsR) is based on linguistic theory about the relation between the intonation of an utterance and other aspects of its linguistic structure and context. ProsR produces a phonological description of the prosody in terms of two types of abstract representation. The first is the prosodic constituent structure of an utterance. In this respect, we followed prosodic phonology, and its central claim that every utterance has a strictly hierarchical structure consisting of a limited number of prosodic categories (Section 2.2). It was shown that this prosodic structure is instrumental in pitch accent placement (RR applies within φ's), application of rules of phrasal phonology (see Section 5.6.1), and phonetic implementation of the durational structure (preboundary lengthening and pause insertion, see Section 5.6.2). With respect to the theory of prosodic phonology, the main contribution of the current work is a detailed investigation of phonological phrasing in Dutch, which resulted in the proposal of a new algorithm.

The second type of representation generated by ProsR is the intonational structure. Here, we followed intonational phonology in adopting a two-level intonational model: phonological description and phonetic implementation. The former consists of the autosegmental description of Dutch intonation developed by Gussenhoven, which describes the intonation of an utterance in terms of pitch accents, boundary tones, AD's, tone morphemes, and their alignment to the segmental string (Section 2.3). The latter involves a set of phonetic rules that map a phonological description onto an actual pitch contour (Section 5.6.2).

Intonational phonology introduces a convenient layer of phonological abstraction, which allowed us to abstract away from phonetic details, and to focus on generating intonational descriptions at a symbolic level. The contribution of the current work is thus not so much to intonational phonology per se, but rather to the derivation of two aspects of the intonational descriptions: the locations of AD-boundaries, and the locations of pitch accents. With respect to pitch accent placement, we extended the theory developed by Gussenhoven, and subjected it to an evaluation. As far a intonational phrasing is concerned, we evaluated several existing intonational phrasing algorithms, and proposed a new algorithm specifically for Dutch, which was also subjected to a formal evaluation.

Below, we will further discuss these theoretical contributions, thereby addressing the research questions regarding the quality of our linguistic theory about pitch accent placement and intonational phrasing. We finish with some remarks on tune choice.

9.3.1 Phonological phrasing

Our algorithm for pitch accent placement relies in part on RR, which in turn requires that the φ-boundaries are correctly determined. In Chapter 3, therefore, we investigated phonological phrasing in Dutch. It was argued that the φ mapping rules proposed in Nespor and Vogel (1986) and Hayes (1986) are inadequate for Dutch. Therefore, we proposed a new φ-algorithm for Dutch based on the notion of adjacent modification. A careful examination of φ-phrasing in a substantial number of syntactic structures revealed that it was still not entirely adequate in VP's and S's. We therefore revised the definitions of φ-phrasing and adjacent modification.

For evaluation, we used a corpus of flower descriptions. The sentences were manually parsed, and pitch accents were assigned to them by a panel of experts. The algorithm was tested by deriving φ's, applying RR (in addition to SAAR), and checking if the accent deletions that took place were indeed correct. The outcome of this evaluation was that pitch accent placement using a combination of SAAR, φ-phrasing, and RR correctly accounts for 94.11% of all accent decisions in the reference corpus. Considering the length and complexity of the sentences, this appears to be a rather good result. Further analysis revealed more than three times as many undershoot errors as there were overshoot errors. In other words, the algorithm predicted too many rather than too few accents, suggesting that either the focus-based initial accentuation is too generous, or additional rules for deaccentuation are needed. Additional accent deletion rules were in fact introduced in Chapter 7.

As far as RR is concerned, it was found that it applied correctly in 39 cases and incorrectly in 5 cases. Two of these errors could be reduced to an incorrect initial accent assignment, whereas the remaining three may be real errors, requiring further research. Moreover, 19 out of the 42 undershoot errors may in theory be blamed on RR and/or φ-phrasing. It would appear that a postmodifying PP is not always preceded by a φ-boundary, as predicted by our φ-phrasing algorithm, although the exact conditions are unclear. It was suggested that some undershoot errors in compound adjectives can be explained by variability in their accent potential (they can carry one or two accents). Still, these undershoot errors do not provide compelling reasons to modify our φ-phrasing algorithm. We concluded that the proposed φ-algorithm for Dutch is fairly accurate. The later evaluation of SLG output in PTD and NUM showed that none of the errors in pitch accent placement were due to RR, confirming once more the correctness of the our φ-phrasing algorithm.

Although descriptively adequate, the algorithm is still somewhat unsatisfactory when viewed a a set of linguistic generalisations. In particular, we would like to get rid of the special conditions for adverbs and auxiliaries. Applying the algorithm in SLG also indicated that a rule for clitic movement is required to prevent clitics from ending up in the wrong φ. Perhaps an alternative syntactic analysis would allow a generalised statement of both the φ-algorithm and adjacent modification. Alternatively, other formalisations may be explored, for instance, Optimality Theory (Prince and Smolensky 1993), which may turn out to be more suitable; see Post (2000, ch. 4), Truckenbrodt (1999), and Selkirk (2000).[6]

The φ-algorithm is based exclusively on evidence from accent deletions by RR. Potentially, there are two other types of evidence. The first is formed by segmental rules that are bound to apply within a φ. However, it appears to be difficult to find compelling evidence of this type, because such rules tend to be rather sensitive to performance factors like speech rate and speaking style. Second, prosodic phonology claims that the prosodic structure of an utterance is reflected in its the durational structure. More specifically, the prediction here is that φ-boundaries trigger more preboundary lengthening, and are more likely to be accompanied by a pause than non-φ-boundaries. As a matter of fact, this regularity has been implemented in CONPAS. There is experimental work that supports this claim for the 'non-controversial' φ-boundaries, i.e. those predicted by most φ-algorithms (Gussenhoven and Rietveld 1992a; Hofhuis, Gussenhoven, and Rietveld 1995; Cambier-Langeveld 2000). It would be interesting to obtain this type of evidence for 'controversial' boundaries, i.e. those that are unique to our algorithm.

9.3.2 Pitch accent placement

We started with an account of pitch accent placement in the form of a rule-based, linguistic grammar (Section 2.5.4). Its input consisted of the argument structure, the focus distribution, the phonological phrases, and the surface string. The initial grammar had just two rules: SAAR and RR. In Chapter 7, its shortcomings were discussed, and some modifications and extensions were proposed.

9.3.2.1 SAAR

SAAR accounts for the basic focus-accent relation: if focused, every predicate, argument, and modifier must be accented, with the exception of a predicate that, discounting unfocused constituents, is adjacent to an argument. For current purposes, it turned out to be more convenient to reformulate SAAR in two steps: (1) initial accentuation, i.e. accenting all focused constituents, followed by (2) deaccentuation, i.e. deaccenting predicates meeting the condition imposed by SAAR.

With respect to initial accentuation, the focus-accent relation is simply one-to-one for normal words. For *phrasal compounds*, however, the location and number of accents depends on the context, so the focus-accent relation may (initially) be one-to-many. Although this is hardly a problem from a theoretical

[6] However, implementing OT for SLG may not be that easy. At any rate, blindly generating candidates and applying constraints until only one is left is an inefficient process from a computational point of view.

point of view, it requires special care in the implementation. Since ProsR can only accent terminal nodes, it must have access to the internal structure of phrasal compounds. This is accomplished by having SynR deliver the internal structure of phrasal compounds, while no internal structure is generated for normal words. Similarly, some additional processing in SynR was found to be necessary for the correct accentuation of a compound verb, since its verbal part and its particle (which receives the accent, if any) may occur at separate positions in an utterance.

SAAR is formulated in semantic rather than syntactic terms. The problem with this is that a focused argument can be a phrase consisting of many words and possessing a complex internal structure. In fact, it can contain other semantic constituents itself. This raises the question which parts of a phrase have to be focused in order to count the whole phrase as focused. For instance, if only a premodifier within an NP is focused, while the nominal head is not, the NP does not constitute a focused argument as far a SAAR is concerned.

We changed the perspective somewhat by assuming that focus is assigned to words rather than to semantic constituents. A set of focus projection rules was introduced to project focus upwards onto the syntactic phrases fulfilling the semantic roles. Most of these rules fit the pattern: a phrase is focused if its syntactic head is focused. As a result, ProsR can easily identify semantic constituents that are focused when executing the deaccentuation step. In addition to the known exceptions to SAAR described in Gussenhoven (1992b, Appendix) and Faber (1987), compelling examples were found that an AD boundary between a predicate and an extraposed argument prevents deaccentuation of the former. Also, the distinction between arguments and modifiers proved to be fuzzy in some contexts. We argued that a predicate's argument structure may depend on its contextual interpretation. This is an issue that needs further exploration and formalisation before it can be implemented.

In the final evaluation reported in the previous chapter, we encountered a number of cases in which the accent on a predicate was not deleted. Although most of these could be traced back to erroneously backgrounding of arguments, a few remain unexplained. In contrast, we did not find any SAAR overshoot errors.

9.3.2.2 Additional deaccentuation rules

As already mentioned in relation to phonological phrasing, RR showed only a few overshoot errors when applied to the corpus of flower descriptions. More undershoot errors were encountered, however, suggesting that additional deaccentuation rules were required. While applying RR to the domains of PTD and NUM during the final evaluation, we did not encounter any undershoot or overshoot errors. We consider this a good result, because it means that the complex combination of syntactic analysis, φ-derivation, and RR still produces the correct predictions when applied to new utterances in a different domain.

In addition to RR, we implemented the *Lexical Rhythm Rule* (LRR) for a very limited class of quantifiers and adverbs for which the location of the pitch accent depends on the right-side context.

During the development of the SLG system, we encountered accents that should be deleted, but were beyond the scope of SAAR, RR and LRR. One type of undershoot error is the lack of an accent on an NP that functions as a premodifier of another NP (e.g. *perron* in *perron vijf* 'platform five'). To account for this, we introduced the rule *NP Premodifier Deaccentuation* (NPPD).

We also explored the accentuation of numerals in considerable depth. It was shown that their accentuation can largely be accounted for with the independently motivated φ-algorithm and RR. However, this still failed to explain the lack of an accent on 'units' such as *honderd* 'hundred' and *duizend* 'thousand'. We therefore proposed the *Unit Rhythm Rule* (URR), and showed that its use is not limited to numeral expressions, but extends to other phrases containing units, such as expressions of time and price.

The final evaluation in PTD and NUM showed that the only substantial errors caused by rhythm rules are all due to URR. While half of these could be traced back to other errors, a few errors remain unexplained, in particular, four cases in which for some reason the unit *honderd* 'hundred' is not deaccented.

Finally, there appears to be interaction between pitch accent placement and intonational phrasing. With regard to SAAR, we found that an AD-boundary between a predicate and an extraposed argument blocks the deaccentuation of the predicate. Similarly, deaccentuation of a unit by URR is prevented by an intervening AD-boundary. The problem is that this contradicts the assumption that pitch accent placement *precedes* AD-derivation, leading to a theoretical rule-order paradox for which we have no solution. Interactions of this type remain a challenge to be addressed by future research.

9.3.3 Intonational phrasing

The intonational phrase is a prosodic constituent (π) placed between the utterance (U) and the phonological phrase (φ) in the prosodic hierarchy. It is defined on the basis of three criteria: it forms the domain of a coherent intonation contour, a pause may be introduced at its end, and it serves as the application domain of certain segmental rules. We summarised Gussenhoven's arguments against this view, showing these criteria can be contradictory, and therefore that the stretch of speech covered by an intonation contour cannot consistently be identified with a particular prosodic category (Section 2.3.7). Following Gussenhoven, we assumed a constituent corresponding to a coherent intonation contour that does not define a layer in the prosodic hierarchy: the Association Domain.

In Chapter 4, several algorithms for intonational phrasing were systematically compared by means of a partly automatic method of evaluation against an intonationally annotated corpus of flower descriptions. It was found that the algorithms proposed by Nespor& Vogel ($\mathcal{N}\&\mathcal{V}$), by Capers ($\mathcal{CAS}$) and by Gussenhoven ($\mathcal{GUS}$) could not generate intonational phrases that were as good as the reference phrasing produced by the human experts. After an extensive analysis of the errors, we concluded that what is needed is a syntax-driven algorithm for intonational phrasing, which is sensitive to particular syntactic configurations, the accent distribution, and the weight of AD's.

In Section 4.8.1, we proposed an algorithm ($\mathcal{NEW}$) along these lines consisting of eight intonational phrasing rules. Each rule describes a syntactic pattern and a place to insert the boundary. There are rules for inserting boundaries in appositions, sentences, verb phrases, noun phrases and coordinations. It was shown that $\mathcal{NEW}$ correctly accounts for 97.66% of the boundary decisions in the corpus of flower descriptions. Statistical analysis showed that $\mathcal{NEW}$ performs significantly better than the algorithms proposed by Nespor& Vogel ($\mathcal{N}\&\mathcal{V}$), and by Capers ($\mathcal{CAS}$).

In addition, $\mathcal{NEW}$ was implemented in ProsR and evaluated in the domain of travel descriptions and numerals. It was found that $\mathcal{NEW}$ agrees with four human experts in 96% of the boundary decisions (ignoring optional boundaries). Differences between ProsR and the human experts showed that ProsR tends to generate too few boundaries, but inserts virtually no erroneous boundaries. In NUM, an undershoot of the rule AD-in-Coordination was observed, which could be partly explained by interaction with accentuation errors. In PTD, the majority of the errors was due to undershoot of the rule AD-in-S-2, which failed to insert a boundary before an extraposed PP. Again, many of these could be reduced to the accentuation errors.

9.3.4 Tune Choice

Although we chose to abstract away from issues of tune choice, it is an essential part of any SLG system. We therefore implemented a provisional algorithm that chooses the type of pitch accents and boundary tones, and inserts tone morphemes like downstep. It delivers tunes that are acceptable, but lack the variation, expressiveness and liveliness of those produced by skilled human readers. They can be improved by varying the type of pitch accents, and employing tone rules/morphemes like *Tone Linking*, *Half Completion*, and *Narration* (Gussenhoven 1988). Part of the work of tune choice for TTS reported in (Terken 1993) may be reinterpreted in our intonational model. However, to use various tunes in a sensible way, we need to know how they relate to semantics and pragmatics of the utterance. While this is an area that is largely unexplored, some results for English (Pierrehumbert and Hirschberg 1990) and Dutch (Gussenhoven 1984; Keijsper 1984a; Keijsper 1984b; Caspers 1988; Caspers 1998; Grabe, Gussenhoven, Haan, Marsi, and Post 1998) are available. Of course, an SLG system offers an good opportunity for testing theories about meaning and intonation.

In a preliminary attempt to enhance the aforementioned tune choice, we proposed that downstep is preferable for simple coordinations (in which each non-final conjunct has at most one accent). Also, we found that a high initial boundary tone can be used to indicate the start of a paragraph.

We found that some tunes could not be realized within the current theoretical framework and its implementation. First, it was observed that apart from the normal type of downstep, there is a variant that applies only to the 'non-primary' accents of the conjuncts of a coordination, leaving the 'primary' accents unaffected. This variant may be incorporated into the phonological description as a new tone morpheme, with an appropriate phonetic implementation. Second, we found that appositions cannot be accounted for within the current system. These are typically accompanied by temporary lowering of the register: the overall pitch level drops at the start of the apposition, but returns to the previous level when it is finished. This type of tune may also be modelled by an extra tone morpheme for pitch reduction.

9.4 Practical implications

Anyone who is familiar with speech synthesis applications will probably agree that their intonation, and their prosody in general, needs to be improved significantly in order to match human performance. Not only does good prosody contribute to the perceived naturalness of synthetic speech, there is also a growing amount of experimental evidence that it facilitates human parsing of the (possibly ambiguous) syntactic structure of utterances (Ladd 1986; Price, Ostendorf, Shattuck-Hufnagel, and Fong 1991; Beach 1991; Kelly 1992; Grabe, Warren, and Nolan 1994; Sanderman 1996; Marsi, Coppen, Gussenhoven, and Rietveld 1997) interpreting its informational structure (Birch and Clifton 1995; Brown 1983; van Donzel 1999; Fowler and Housum 1987; Nooteboom and Terken 1982; Nooteboom and Kruyt 1987), determining the discourse structure (Lehiste 1975; Lehiste 1979; Swerts and Geluykens 1993; Sluijter and Terken 1993; Swerts, Collier, and Terken 1994; Swerts 1994; van Donzel 1999; Noordman, Dassen, Swerts, and Terken 1999), and inferring the attitudinal and emotional state of the speaker (Grabe, Gussenhoven, Haan, Marsi, and Post 1998; Caspers 1998).

Speech synthesis has traditionally been applied in a TTS environment. The first step in TTS is preprocessing and linguistic analysis of the input text, which typically includes part-of-speech tagging and syntactic parsing, as the syntactic information is used for grapheme-to-phoneme conversion and prosody assignment. It is common knowledge in NLP, however, that syntactic parsing of unrestricted text is still an unsolved problem. Notorious ambiguity problems such as PP-attachment and coordination can cause a parser to consume an excessive amount of time, and output a large number of alternative analyses. In addition, there are syntactic phenomena, such as apposition and gapping, which are difficult to account for. Moreover, the input may even be ill-formed, requiring a robust form of parsing. Therefore, syntactic parsing in TTS is usually limited to shallow parsing, leaving a considerable part of the syntactic structure underspecified.

Unfortunately, what is needed for controlling the intonation appears to be primarily the type of linguistic information that can *not* be automatically extracted from a text.[7] Our intonational phrasing algorithm, for instance,

[7] In fact, this may not be a coincidence. Given the assumption that there is some similarity

requires a correct syntactic analysis of PP-attachment, coordination, and appositional elements, whereas our accentuation algorithm relies, among other things, on a detailed analysis of NP-premodifiers.

Moreover, merely POS and syntactic information is not sufficient, as predicting intonation requires semantic and discourse information as well. Assuming the correctness of SAAR, accentuation of the predicate requires identification of its arguments. Accentuation requires focus/background information, or some other type of information about the discourse status of domain entities. Although attempts to keep track of the discourse have proved useful to a certain extent (Horne, Fillipson, Johansson, Ljungqvist, and Linström 1993; Hirschberg 1992; Hirschberg 1993; Hiyakumoto, Prevost, and Cassell 1997), reliable parsing of the discourse structure is still beyond our reach.

The lack of detailed linguistic information has its impact on the naturalness of synthetic speech as well. TTS systems are made to avoid disturbing, obvious errors. The safest intonation is the one that occurs most frequently. Consequently, these systems fail to employ intonational variation, which is probably one of the main reasons that they sound unnatural and tedious in comparison to competent human speakers.

To sum up so far, the current state of NLP technology does not allow us to build TTS systems that automatically extract all the required information from unrestricted text. Consequently, applied TTS systems have to rely on POS tags and shallow syntactic analyses, and resort to statistical regularities or handcrafted heuristic rules for prosodic realization. Although these approaches take us a long way, and are certainly of great value for building practical TTS systems, they appear to offer less perspective for improvement in the long run. The same view is expressed in the following quote from Taylor and Black, a comment on their statistical model for assigning phrase breaks from POS tags in TTS:

> "While we believe more investigation into tagset composition would help in reducing errors we also believe that there is only so far that superficial text analysis techniques like this can go. From examining the errors in the test set we believe that a more sophisticated analysis will be needed to correct some of the errors. Although we have argued in the introduction against using syntactic parsers for phrase break assignment, our reasons stem from the basic inaccuracy of these parsers, not because syntactic parses themselves are unhelpful."[8] (Taylor and Black 1998)

Clearly, SLG has an advantage over TTS as far as the availability of linguistic information is concerned. The aforementioned problems with respect to

---

between a computational parser and the human language processor, both will encounter the same type of problems with regard to parsing ambiguous utterances. Adding intonation, and prosodic structure in general, may in part be a strategy to deal with ambiguity problems.

[8]It should be noted that Taylor and Black do not necessarily advocate a rule-based approach to prosody assignment, but rather acknowledge the fact that more elaborate input is required.

automatic linguistic analysis do not exist in SLG, due to the fact that the NLG module is capable of producing not only text, but also the linguistic structure and context of this text. The same point has been made by a number of other researchers, e.g. Bateman (1997), Hess (1997), McKeown and Moore (1995), Sproat (1997), and McKeown and Pan (to appear); in fact, the idea goes back as far as Young and Fallside (1979). This potential of SLG for generating better prosody holds a promise for practical application of speech synthesis. This view is supported by the fact that the experimental evaluation showed that the SLG system outperforms the tested TTS system (most clearly with respect to accent placement).[9]

Yet, our SLG system is a laboratory project which is primarily intended as a testbed for linguistic theory about intonation. In its current state, the content-to-speech component CONPAS is not ready for practical application. Although the current system has so far been applied to two quite different application domains, which we consider representative of instructional types of text, other domains will undoubtedly reveal new challenges with respect to intonation. In addition, the type of output that is generated is essentially written text that is read aloud. From the viewpoint of ergonomics or human-computer interaction, an SLG application should rather produce the type of language found in speech communication, for instance, with shorter utterances and a more liberal syntax. Finally, little attention has been paid to performance issues, and the system is too slow for on-line tasks. Nonetheless, we believe experimental systems like ours provide the foundation for the future development of applied SLG systems.

Of course, SLG is no substitute for TTS, as there will always be a need for reading aloud existing text. However, building an SLG system that generates prosody satisfactorily will at least tell us what kind of linguistic information is required for prosody assignment, and therefore, what type of information should be automatically extracted from the input text during the linguistic analysis step in a TTS system. SLG may thus serve to set the goals for linguistic analysis in TTS.

[9] Given our focus on intonation, we tend to forget the advantages of SLG with respect to text preprocessing, all the more because they look trivial. For instance, homographs with different pronunciations (e.g. the numeral *een* 'one' versus the determiner *een* 'a') are easily resolved from the available syntactic structure. Likewise, there is no need to expand numerals or abbreviations, because the NLG systems provides them in their full lexical form. Also, end-of-sentence detection, which is sometimes difficult in TTS due to ambiguity of punctuation marks, is a non-problem in SLG.

Appendix A

Phonological Phrasing in WFD

This Appendix shows the results for the evaluation of phonological phrasing in Section 3.4. The text material consisted of ten botanical descriptions of wild flowers, of which only one (with the label *aamu*) is shown here.[1] Results are given per sentence in the form of a table, with every word on a new row. A dashed line (`---`) represents a phonological phrase boundary. The first column contains the words, where capitals indicate that the word was assumed to be focused. The second column shows the pitch accents assigned by the experts, where `*` represents an obligatory accent, and `+` represents an optional accent. The third column shows the accents predicted by the accentuation algorithm. The fourth column contains comments on the application of the accentuation rules.

A.1 Aarmunt

aamu01:

```
================================================================================
Words:              Reference:     Algorithm:     Comments:
================================================================================
AARMUNT             *              *
--------------------------------------------------------------------------------
GROEIT                                            SAAR application ok.
--------------------------------------------------------------------------------
OP
VOCHTIGE            *              *
PLAATSEN            *              *
--------------------------------------------------------------------------------
LANGS
WEGEN               *              *
```

[1]The complete results can be found at
`http://lands.let.kun.nl/TSPublic/marsi/thesis`

```
--------------------------------------------------------------------------------
EN
OP
RUDERALE            *              *
TERREINEN           *              *
--------------------------------------------------------------------------------
IN
GROTE               *              *
DELEN                              *              accent not deleted!
--------------------------------------------------------------------------------
VAN
EUROPA              *              *
--------------------------------------------------------------------------------
MAAR
IS
BIJ
ONS                 *              *
ZELDZAAM            *              *
================================================================================
```

aamu02:

```
================================================================================
Words:              Reference:     Algorithm:     Comments:
================================================================================
hij
WORDT
WEL                                *              accent not deleted!
--------------------------------------------------------------------------------
IN
TUINEN              *              *
--------------------------------------------------------------------------------
GEKWEEKT            *              *
--------------------------------------------------------------------------------
EN
IN
MUNTSAUS            *              *
--------------------------------------------------------------------------------
VERWERKT            *                             SAAR application wrong!
================================================================================
```

aamu03:

```
================================================================================
Words:              Reference:     Algorithm:     Comments:
================================================================================
de
plant
--------------------------------------------------------------------------------
IS
OVERBLIJVEND        *              *
--------------------------------------------------------------------------------
DE
STENGELS            *              *
--------------------------------------------------------------------------------
WORDEN
NEGENTIG            *              *
```

```
CENTIMETER                                             RR application ok.
HOOG                   *              *
--------------------------------------------------------------------------------
EN
DE
BLADEN                 *              *
--------------------------------------------------------------------------------
ZIJN
LANCETVORMIG           *              *
--------------------------------------------------------------------------------
EN
GEZAAGD                *              *
================================================================================
```

aamu04:

```
================================================================================
Words:                 Reference:     Algorithm:     Comments:
================================================================================
hij
HEEFT                                                SAAR application ok.
--------------------------------------------------------------------------------
EEN
ON                     *              *
MISKENBARE                                           RR application ok.
GEUR                   *              *
================================================================================
```

aamu05:

```
================================================================================
Words:                 Reference:     Algorithm:     Comments:
================================================================================
IN
DE
LATE                   *              *
ZOMER                  *              *
--------------------------------------------------------------------------------
VERSCHIJNEN                                          SAAR application ok.
--------------------------------------------------------------------------------
DE
ROZE                   *              *
--------------------------------------------------------------------------------
OF
LILA                   *              *
BLOEMEN                *              *
--------------------------------------------------------------------------------
IN
AARVORMIG              *              *
SAMENGESTELDE                                        RR application ok.
SCHIJNKRANSEN          *              *
--------------------------------------------------------------------------------
AAN
DE
STENGELTOP             *              *
================================================================================
```

aamu06:

```
==============================================================================
Words:              Reference:     Algorithm:     Comments:
==============================================================================
DE
ECHTE               *              *
PEPERMUNT           *              *
------------------------------------------------------------------------------
IS
EEN
BASTAARD            *              *
------------------------------------------------------------------------------
VAN
AARMUNT             *              *
------------------------------------------------------------------------------
EN
WATERMUNT           *              *
==============================================================================
```

aamu07:

```
==============================================================================
Words:              Reference:     Algorithm:     Comments:
==============================================================================
hij
GROEIT                             *              accent not deleted!
------------------------------------------------------------------------------
IN
HET
WILD                *              *
------------------------------------------------------------------------------
OP
VOCHTIGE            *              *
PLAATSEN            *              *
------------------------------------------------------------------------------
EN
WORDT
OOK                 *              *
------------------------------------------------------------------------------
OM
DE
PEPERMUNTOLIE       *              *
------------------------------------------------------------------------------
GEKWEEKT                           *              accent not deleted!
==============================================================================
```

aamu08:

```
==============================================================================
Words:              Reference:     Algorithm:     Comments:
==============================================================================
deze
olie
------------------------------------------------------------------------------
WORDT
GEBRUIKT                                          SAAR application ok.
```

ALS SMAAK EN GEURMIDDEL	* *	* *
IN SNOEPGOED	*	*
LIKEUREN	*	*
MEDICIJNEN	*	*
EN TANDPASTA	*	*

Appendix B

Intonational Phrasing in WFD

This Appendix shows the results for the four intonational phrasing algorithms described in Chapter 4 ($\mathcal{N}\&\mathcal{V}$, $\mathcal{CAS}$, $\mathcal{GUS}$, and $\mathcal{NEW}$). The text material consisted of ten botanical descriptions of wild flowers, of which only two (*aaor* and *hera*) are shown here.[1] The reference produced by the experts is shown after `ref:` in terms of obligatory boundaries (#), optional boundaries (|), obligatory accents (*) and optional accents (+).

B.1 Aangebrande Orchis

aaor01

```
cas :      *        *   |             +            *
n&v :      *        *   |             +            *
gus :      *        *   #             +            *
new :      *        *                 +            *
ref :      *        *   |             +            *
wrd : aangebrande orchis ontleent de naam aan de donker kastanjebruine
```

```
cas :      *      |        * |          *    #
n&v :      *               *            *    #
gus :      *      |        *            *    #
new :      *               *            *    #
ref :      *               *            *    #
wrd : bloemknoppen aan de top van de bloeiaar
```

[1] The complete results can be found at `http://lands.let.kun.nl/TSPublic/marsi/thesis`

aaor02

```
cas :       *        *   |        *   #              *         *            *  |
n&v :       *        *   |        *   #              *         *            *
gus :       *        *            *   #              *         *  #         *  #
new :       *        *            *   #              *         *  #         *  #
ref :       *        *            *   #              *         *  #         *  #
wrd : de geopende bloemen zijn lichter met een donkerpurperen helm en een witte

cas :           *               * |         *          *   #
n&v :           *               *           *          *   #
gus :           *               * #         *          *   #
new :           *               *           *          *   #
ref :           *               * #         *          *   #
wrd : op een mannetje lijkende lip met roodpurperen vlekjes
```

aaor03

```
cas :      *            *       *  #    *         *     |     *       *     |
n&v :      *            *       *  #    *         *           *       *
gus :      *            *       *  #    *         *     #     *       *     #
new :      *            *       *  #    *         *           *       *
ref :      *            *       *  #    *         *           *       *     #
wrd : ook dit is een zeldzame soort verspreid voorkomend in korte graslanden op

cas :   *             *       |     *   | *      *    |    *     *   |  |
n&v :   *             *             *   | *      *         *     *   |
gus :   *             *       #     *   # *      *    |    *     *   #
new :   *             *       #     *   # *      *         *     *   #
ref :   *             *             *   # *      *         *     *   #
wrd : krijt en kalksteenbodems in Europa zeer zeldzaam in oost Belgie en in

cas :     *       +         *      #
n&v :     *       +         *      #
gus :     *       +         *      #
new :     *       +         *      #
ref :     *       +         *      #
wrd : Nederland helaas uitgestorven
```

aaor04

```
cas :                 *  |         *        *   |       |       *
n&v :                 *            *        *                   *
gus :                 *  #         *        *   #               *
new :                 *            *        *   #               *
ref :                 *            *        *   #               *
wrd : hij vormt een rozet van elliptische bladen met een tot twintig centimeter

cas :  *      *         *                *       *    |        *  #
n&v :  *      *         *                *       *             *  #
gus :  *  |   *         *      #         *       *             *  #
new :  *  #   *         *      #         *       *             *  #
ref :  *  #   *         *      #         *       *             *  #
wrd : hoge stevige bloeistengel en een dichte bloeiaar in de zomer
```

aaor05

```
cas :       *   |               *     #        *        *   |      *    *  #
n&v :       *   |               *     #        *        *   |      *    *  #
gus :       *                   *     #        *        *   |      *    *  #
new :       *                   *     #        *        *   #      *    *  #
ref :       *                   *     #        *        *          *    *  #
wrd : de bloemen geuren als heliotroop de afstervende bladen naar vers hooi
```

B.2 Heggerank

hera01

```
cas :     *    |           *                  *    |      *  |   *   |
n&v :     *    |           *                  *           *  |   *   |
gus :     *    #           *                  *    #      *      *   |
new :     *                *                  *    #      *      *
ref :     *    |           *                  *    #      *  |   *   |
wrd : heggerank is een borstelig behaarde klimplant met lange kantige en
```

```
cas :    *        *    |    *                   *   |         *          *   #
n&v :    *        *    |    *                   *   |         *          *   #
gus :    *        *    #    *                   *   #         *          *   #
new :    *        *    #    *                   *   #         *          *   #
ref :    *        *    #    *                   *   #         *          *   #
wrd : vertakte stengels handvormig ingesneden bladen en spiraalvormige ranken
```

hera02

```
cas :        *              *         *   |       |          *     |
n&v :        *              *         *   |                  *     |
gus :        *              *         *   #                  *
new :        *              *         *   #                  *
ref :        *              *         *   #                  *     #
wrd : de mannelijke en vrouwelijke bloemen groeien in de bladoksels aan
```

```
cas :       *          *   #
n&v :       *          *   #
gus :       *          *   #
new :       *          *   #
ref :       *          *   #
wrd : verschillende planten
```

hera03

```
cas :                 *         *          *        *                *
n&v :                 *         *          *        *                *
gus :                 *         *     #    *        *           |    *
new :                 *         *     #    *  #     *           #    *
ref :                 *         *     #    *  |     *           #    *
wrd : ze bezitten afstaande kelkbladen en vijf groenachtig witte netvormig
```

```
cas :               *     #
n&v :               *     #
gus :               *     #
new :               *     #
ref :               *     #
wrd : geaderde kroonbladen
```

hera04

```
cas :        *             |        *       *     #        *     |
n&v :        *             |        *       *     #        *     |
gus :        *                      *       *     #        *     #
new :        *                      *       *     #        *
ref :        *                      *       *     #        *
wrd : de mannelijke bloemen hebben drie meeldraden de vrouwelijke een

cas :      *             *       |     *      *    #
n&v :      *             *             *      *    #
gus :      *             *       |     *      *    #
new :      *             *             *      *    #
ref :      *             *             *      *    #
wrd : onderstandig vruchtbeginsel met drie stempels
```

hera05

```
cas :      *   |        *    * #
n&v :      *   |        *    * #
gus :      *            *    * #
new :      *            *    * #
ref :      *            *    * #
wrd : de vrucht is een rode bes
```

hera06

```
cas :         |     |     *   |    *    |       *    #                *
n&v :         |           *   |    *    |       *    #                *
gus :                     *        *            *    #                *
new :                     *        *            *    #                *
ref :                     *   |    *    |       *    #                *
wrd : de plant klimt in heggen struwelen en bosranden meestal op kalkhoudende

cas :   *   #     *        |      *   #     * |               *   |
n&v :   *   #     *               *   #     *                 *   |
gus :   *   #     *               *   #     *                 *
new :   *   #     *               *   #     *                 *
ref :   *   #     *               *   #     *                 *
wrd : bodems in grote delen van Europa bij ons vooral in de duinen en langs de

cas :          *    #
n&v :          *    #
gus :          *    #
new :          *    #
ref :          *    #
wrd : grote rivieren
```

Appendix C

Pitch Accent Placement and Intonational Phrasing in TD

This Appendix shows the results for the evaluation experiment concerning pitch accent placement and intonational phrasing described in Chapter 8. The text material consisted of eight travel descriptions generated by PTDGen, of which only two are shown here: ptd-1x-elaborate (an elaborate travel description involving one transfer), and ptd-3x-normal (a normal travel description involving three transfer).[1] Above the sentences are the pitch accents and intonational boundaries assigned by the four human experts (CO, JC, JH, and MS), the KUNTTS system (KT), and ProsR (PR), in terms of obligatory accents (*), optional accents (+), obligatory boundaries (#), and optional boundaries (|).

C.1 Ptd-1x-elaborate

ptd-1x-elaborate-1

```
CO :              *    *  #          *         *   #     *      *
JC :              *    *  |          *         *   #     *      *
JH :              *    *  |          *         *   #     *      *
KT : *            *    *             *         *                *
MS :              *    *  |          *         *   |     *      *
PR :              *    *  #          *         *   #     *      *
     U kunt van Delft Zuid naar Vlaardingen Centrum met een overstap.
```

[1]The complete results can be found at
http://lands.let.kun.nl/TSPublic /marsi/thesis

ptd-1x-elaborate-2

```
CO :          *    |               *    |        *        +
JC :          *    |               *    |        *        *
JH :          +    |               *    |        *        *
KT :                               *             *        *
MS :          *    |               *    |        *        *
PR :          *                    *             *        *
     Om te beginnen neemt u de stoptrein naar Schiedam Centrum.
```

ptd-1x-elaborate-3

```
CO :                           *        *            *    #      *
JC :  +              +         *        *    |       *    #      *
JH :                           *        *            *    #      *
KT :                 *         *        *    #       *                 *
MS :        +                  *        *            *    #      *
PR :        *                  *        *    #       *    #      *
     Die vertrekt richting Rotterdam Centraal en Dordrecht om twintig uur

CO :   *       *
JC :   *       *
JH :   *       *
KT :           *
MS :   *       *
PR :   *       *
     achten veertig.
```

ptd-1x-elaborate-4

```
CO :    +                          *    #      *          *       *
JC :    +        +                 *    #      *          *       *
JH :    +                          *    #      *          *       *
KT :                               *                 *            *
MS :    +                          *    |      *          *       *
PR :             *                             *          *       *
     Daarmee arriveert u dan in Schiedam om twintig uur vieren vijftig.
```

ptd-1x-elaborate-5

```
CO :          *              +   #    *             #          *
JC :          *              +   |    *             #          *    |
JH :          *              +   #    *             |          *
KT : *                       *                                        *
MS :          *              +   |    *             |          +
PR :          *              *   #    *                        *
     U hebt eenen twintig minuten om over te stappen op de stoptrein naar

CO :      *         *
JC :      +         *
JH :                *
KT :      *         *
MS :      *         *
PR :      *         *
     Vlaardingen Centrum.
```

ptd-1x-elaborate-6

```
 CO :  +                 +     *  #          *          *      *        *
JC :  *                       *  #   +      +          +      *  #     *
JH :                          *  #                            *  #     *
KT :                    *                   *          *      *   *
MS :       +           +     *             *          *      *        *
PR :                          *             *                 *  #     *
     Die vertrekt van perron twee richting Hoek van Holland Haven om eenen

CO :                 *
JC :                 *
JH :                 *
KT :                 *
MS :                 *
PR :                 *
     twintig uur vijftien.
```

ptd-1x-elaborate-7

```
CO :   +                            *     #     *                 *       *
JC :   *        +                   *     |     *                 *       *
JH :   +                            *     #     *                 *       *
KT :                     *          *                  *          *
MS :   +        +                   *           *                 *
PR :            *                               *                 *       *
     Daarmee arriveert u dan in Vlaardingen om eenen twintig uur drieen twintig.
```

ptd-1x-elaborate-8

```
CO :     +       *    #           *              *
JC :     *       *    |           *              *
JH :     *       *    |           *              *
KT :     *                        *              *
MS :     +       *                *              *
PR :     *       *                *              *
     De totale reistijd bedraagt vijfen dertig minuten.
```

C.2 Ptd-3x-brief

ptd-3x-brief-1

```
CO :       +          *             *       *    #        *           *
JC :       *          *    |        *       *    #        *        |  *
JH :       +          *             *       *    #        *           *
KT :                  *             *       *             *
MS :       *          *             *       *    |        *           *
PR :                  *             *       *    #        *           *
     Neem eerst de stoptrein naar Utrecht Centraal van zeventien uur achten

CO :   *
JC :   *
JH :   *
KT :   *
MS :   *
PR :   *
     dertig.
```

ptd-3x-brief-2

```
CO :        +             *           *   #      *           *
JC :        *     |       *           *   #      *           *
JH :        +             +           *   #      *           *
KT :                      *           *                      *
MS :        *             +           *   |      *           *
PR :                                  *          *           *
     Ga vervolgens van Utrecht naar Arnhem om achttien uur drie.
```

ptd-3x-brief-3

```
CO :          +                     *             *    #       *           *
JC :          *       *  |          *             *    #       *        |  *
JH :          +       *  |          *             *    #       *           *
KT :  *               *             *             *            *      *
MS :          +                     *             *    |       +           *
PR :                                *             *    #       *           *
     Stap vervolgens over op de sneltrein naar Nijmegen van achttien uur eenen

CO :    *
JC :    *
JH :    *
KT :    *
MS :    *
PR :    *
     vijftig.
```

ptd-3x-brief-4

```
CO :        +                       *    #       *           *
JC :        *             +         *    #       *           *
JH :                      +         *    #       *           *
KT :                      *         *                        *
MS :                      +         *    |       *           *
PR :                      *         *    #       *           *
     Ga tenslotte naar Nijmegen Heyendaal om negentien uur tien.
```

Appendix D

Pitch Accent Placement and Intonational Phrasing in NR

This Appendix shows the results for the evaluation experiment concerning pitch accent placement and intonational phrasing described in Chapter 8. The text material consisted of three sets of phrases containing numerals (generated by our SLG system on the basis of manually supplied APS's), of which only one (*num-1*) is shown here.[1] Above the sentences are the pitch accents and intonational boundaries assigned by the four human experts (CO, JC, JH, and MS), the KUNTTS system (KT), and ProsR (PR), in terms of obligatory accents (*), optional accents (+), obligatory boundaries (#), and optional boundaries (|).

D.1 Num-1

num-1-1

```
CO :  *      *
JC :  *      *
JH :  *      +
KT :  *      *
MS :  *      *
PR :  *      *
     Vijf honderd
```

[1]The complete results can be found at
`http://lands.let.kun.nl/TSPublic /marsi/thesis`

num-1-2

```
CO :         *   |  *      *
JC :  *          |  *      *
JH :  *          |  *      *
KT :  *             *      *
MS :  *          |  *      *
PR :  *             *      *
     Twee honderd zesen negentig
```

num-1-3

```
CO :    *      *      *       *
JC :    *      *          |   *
JH :    +      *              *
KT :    *                     *
MS :           *              *
PR :           *              *
     Honderd vieren dertig duizend
```

num-1-4

```
CO :  *             *  |   *
JC :  *          |  *      *
JH :  *             *      *
KT :  *             *      *
MS :  *          |  *      *
PR :  *             *      *
     Twee honderd negen duizend
```

num-1-5

```
CO :    *         *
JC :    *         *
JH :    *         *
KT :    *         *
MS :    *         *
PR :              *
     Honderd en zeven
```

num-1-6

```
CO :  *          | *
JC :  *            *
JH :  *            *
KT :  *            *
MS :  *            *
PR :  *            *
     Twee honderd zes
```

num-1-7

```
CO :    *       *
JC :    *       *
JH :    *       *
KT :            *
MS :    *       *
PR :            *
     Honderd duizend
```

num-1-8

```
CO :   *  #   *
JC :   *  |   *
JH :   *      *
KT :   *      *
MS :   *      *
PR :   *      *
     Negen honderd
```

num-1-9

```
CO :  *         |  *       *
JC :  *         |  *       *
JH :  *         |  *       *
KT :  *                    *
MS :  *         |  *       *
PR :  *            *       *
     Zes honderd tweeen tachtig
```

num-1-10

```
CO :  *          |   *
JC :  *              *
JH :  *              *
KT :  *              *
MS :  *              *
PR :  *              *
     Drie honderd duizend
```

num-1-11

```
CO :    *        *
JC :    *        *
JH :    *        *
KT :    *        *
MS :    *        *
PR :             *
     Honderd en twee
```

num-1-12

```
CO :   *          |   *
JC :   *              *
JH :   *              *
KT :   *              *
MS :   *              *
PR :   *              *
     Tweeen dertig honderd
```

num-1-13

```
CO :  *             *              *   # *      *       *      *
JC :  *          |  *              *   # *          |   *      *
JH :  *             *              *   # *          |   *      *
KT :  *                                  *              *      *
MS :  *          |  *              *   # *          |   *      *
PR :  *             *              *   # *              *      *
     Twee honderd vieren dertig duizend vijf honderd zevenen zestig
```

num-1-14

```
CO :   *          |   *       *
JC :   *      +   |   *       *
JH :   *      +   |   *       *
KT :   *              *       *
MS :   *          |   *       *
PR :   *              *       *
     Zeven duizend negenen vijftig
```

num-1-15

```
CO :   *              *   |  *      *
JC :   *              +   |  *      *
JH :   *              +   #  *      *
KT :                         *      *
MS :   *              +   |  *      *
PR :   *                     *      *
     Tweeen dertig honderd zesen negentig
```

num-1-16

```
CO :   *               *   |  *       *
JC :   *               +   |  *       *
JH :   *                   |  *       *
KT :   *       *                      *
MS :   *               +   |  *       *
PR :   *                      *       *
     Vieren vijftig honderd achten veertig
```

num-1-17

```
CO :   *              *
JC :   *              *
JH :   *              *
KT :   *              *
MS :   *              *
PR :   *              *
     Zeven honderd duizend
```

num-1-18

```
CO :   *               *
JC :   *               *
JH :   *           |   *
KT :   *               *
MS :   *               *
PR :   *               *
     Zesen zeventig duizend
```

num-1-19

```
CO :   *      *   |  *              *
JC :   *          |  *              *
JH :   *          |  *          |   *
KT :   *             *              *
MS :   *          |  *              *
PR :   *             *              *
     Zeven honderd vijfen dertig duizend
```

num-1-20

```
CO :  *      *   |  *      *
JC :  *      +   |  *      *
JH :  *          #  *      *
KT :  *      *             *
MS :  *      +   |  *      *
PR :  *             *      *
     Twee miljoen eenen twintig
```

num-1-21

```
CO :         *       *
JC :   *             *
JH :   *             *
KT :         *
MS :   *             *
PR :   *             *
     Eenen dertig honderd
```

num-1-22

```
CO :  *          |  *      *
JC :  *          |  *      *
JH :  *          #  *      *
KT :  *      *             *
MS :  *          |  *      *
PR :  *             *      *
     Twee duizend eenen twintig
```

num-1-23

```
CO :   *      *
JC :   *      *
JH :   *      *
KT :   *      *
MS :   *      *
PR :   *      *
     Tweeen dertig
```

num-1-24

```
CO :  *          |  *       *
JC :  *          |  *       *
JH :  *          |  *       *
KT :  *                     *
MS :  *          |  *       *
PR :  *             *       *
     Acht honderd drieen twintig
```

Appendix E

Samples of SLG output

This appendix contains examples of the output of the spoken language generators for the public transport domain (PTD) and the wild flower domain (WFD). The examples show the knowledge source (only for PTD), the abstract phrase specifications generated by the text planner (PlanR), the surface string generated by the syntactic realizer (SynR), and the prosodic constituent structure and intonational form generated by the prosodic realizer (ProsR). Due to lack of space, the syntactic structures, and phonetic form (phonemes and pitch contour) are not shown.

The intonational form consists of three aligned lines. The top line contains pitch accents and boundary tones. The middle line contains words, AD's, and tone morphemes (Downstep). The bottom line specifies the rules that are responsible for deleting the pitch accent (cf. Table 7.1), inserting the AD-boundary (cf. Table 4.14), or choosing the pitch accent or boundary tone (cf. Table 7.2).

E.1 Public Transport Domain

E.1.1 Example 1: brief travel description

Knowledge source

```
Station            Tijd    Spoor  Soort       Richting/Eindbestemming

Nijmegen       V   09:26   3b     Sneltrein   Arnhem, Zwolle
Arnhem         A   09:41
Arnhem         V   09:49   7b     Sneltrein   Ede-Wageningen, Rotterdam CS
Utrecht CS     A   10:30
Utrecht CS     V   10:32   7a/b   Intercity   Amsterdam CS, Haarlem
Amsterdam CS   A   11:04

Reistijd: 1:38, met 2 maal overstappen
```

Abstract phrase specification

1.
```
(V:gaan<imp>[](P:om(TIME:<heavy>(0)(9)(2)(6)))[P:van
(Nijmegen<prop>)][P:naar(Arnhem<prop>)]
(V:beginnen<om,te,top,light>))
```
2.
```
(V:gaan<imp>[](P:om(TIME:<heavy>(0)(9)(4)(9)))[P:naar
($(N-Utrecht)(N-Centraal)$<prop>)](ADV:vervolgens<top,back>))
```
3.
```
(V:nemen<imp>[][intercity<def>(P:naar($(N-Amsterdam)
(N-Centraal)$<prop>))(P:van(TIME:<heavy>(1)(0)(3)(2)))]
(ADV:tenslotte<top,back>))
```

Surface string

1. ga om te beginnen van Nijmegen naar Arnhem om negen uur
go for to begin from Nijmegen to Arnhem at nine hour
zesentwintig
six-and-twenty
'To begin with, go from Nijmegen to Arnhem at 9:26.'

2. ga vervolgens naar Utrecht Centraal om negen uur negenenveertig
go next to Utrecht Central at nine hour nine-and-forty
'Next, go to Utrecht Central at 9:49.'

3. neem tenslotte de intercity naar Amsterdam Centraal van tien
take finally the intercity to Amsterdam central of ten
uur tweeendertig
hour two-and-thirty
'Finally, take the 10:32 intercity to Amsterdam Central.'

Prosodic structure

1. $[_\pi$ $[_\varphi$ ga] $[_\varphi$ om te beginnen] $[_\varphi$ van Nijmegen] $[_\varphi$ naar Arnhem] $[_\varphi$ om negen uur] $[_\varphi$ zesen twintig]]

2. $[_\pi$ $[_\varphi$ ga] $[_\varphi$ vervolgens] $[_\varphi$ naar Utrecht Centraal] $[_\varphi$ om negen uur] $[_\varphi$ negenen veertig]]

3. $[_\pi$ $[_\varphi$ neem] $[_\varphi$ tenslotte] $[_\varphi$ de intercity] $[_\varphi$ naar Amsterdam Centraal] $[_\varphi$ van tien uur] $[_\varphi$ tweeen dertig]]

Intonational form

1.

%H	H*L			H*L		H*L		L*H	
[$_{AD}$	ga	om	te	beginnen	van	Nijmegen	naar	Arnhem	]
PMIBT	DA			DA		DA		ADFA	AD-in-S-2

%L		H*L		H*L	H*L	L%
![$_{AD}$	om	negen	uur	zesen	twintig	]
DIBT		DA	URR	DA	DA	UFBT

2.

%H				H*L	L*H	
$[_{AD}$	ga	vervolgens	naar	Utrecht	Centraal	]
PMIBT	SAAR			DA	ADFA	AD-in-S-2

%L		H*L		H*L	H*L	L%
$![_{AD}$	om	negen	uur	negenen	veertig	]
DIBT		DA	URR	DA	DA	UFBT

3.

%H				H*L		H*L	
$[_{AD}$	neem	tenslotte	de	intercity	naar	Amsterdam	
PMIBT	SAAR			DA		DA	

L*H	
Centraal	]
ADFA	AD-in-NP-3

%L		H*L		H*L	H*L	L%
$![_{AD}$	van	tien	uur	tweeen	dertig	]
DIBT		DA	URR	DA	DA	UFBT

E.1.2 Example 2: elaborate travel description

Knowledge source

```
Van Hilversum Sportpark naar Rotterdam Alexander
15 mei 2000, maandag

Station                   Tijd     Spoor   Soort       Richting/Eindbestemming

Hilversum Sportpark  V    17:24            Stoptrein   Utrecht CS
Utrecht CS           A    17:40
Utrecht CS           V    17:49    7a/b    Intercity   Rotterdam CS
Rotterdam Alexander  A    18:16

Reistijd: 0:52, met 1 maal overstappen
```

Abstract phrase specification

1. (V:kunnen[<listen>](P:van($(N-Hilversum)(N-Sportpark)$<prop>))(P:naar($(N-Rotterdam)(N-Alexander)$<prop>))(P:met(overstap<nonref>(NUM:(1)))))
2. (V:nemen[<listen>][stoptrein<def>(P:naar($(N-Utrecht)(N-Centraal)$<prop>))](ADV:allereerst<back>))
3. (V:vertrekken[<spat>](P:om(TIME:<heavy>(1)(7)(2)(4))))
4. (V:arriveren[<listen>][P:in(Utrecht<back,prop>)](P:om(TIME:<heavy>(1)(7)(4)(0)))(P:met(<spat,back,top>))(ADV:dan<back>))

5.
```
(V:gaan[<listen>](ADV:later<top>(N:minuut<plur>(NUM:(9))))
[P:van(Utrecht<back,prop>)][P:naar($(N-Rotterdam)
(N-Alexander)$<prop>)](P:met(intercity<def>
(P:(N:richting<nonref,back>(P:
(N:$(N-Rotterdam)(N-Centraal)$<prop>)))))))
```

6.
```
(V:vertrekken[<spat>](P:van(NUM:(7)(LET:a)(LET:b)
(perron<nonref>)))(P:om(TIME:<heavy>(1)(7)(4)(9))))
```

7.
```
(V:aankomen[<spat>][P:in(Alexander<back,prop>)]
(P:om(TIME:(1)(8)(1)(6))))
```

8.
```
(V:bedragen[reistijd<def>(A:totaal)][minuut<plur>(NUM:(5)(2))])
```

Surface string

1. u kunt van Hilversum Sportpark naar Rotterdam Alexander
you can from Hilversum Sportpark to Rotterdam Alexander
met twee overstappen
with two transfers
'To go from Hilversum Sportpark to Rotterdam Alexander takes two transfers'

2. u neemt allereerst de stoptrein naar Utrecht Centraal
you take first the slow-train to Utrecht Central
'First, take the slow train to Utrecht Central.'

3. die vertrekt om zeventien uur vieren twintig
which leaves at seventeen hour four-and twenty
'Which departs 17:24.'

4. daarmee arriveert u dan in Utrecht om zeventien uur veertig
there-with arrive you then in Utrecht at seventeen hour forty
'Which gets you in Utrecht at 17:40.'

5. u gaat negen minuten later van Utrecht naar Rotterdam
you go nine minutes later from Central to Rotterdam
Alexander met de intercity richting Rotterdam Centraal
Alexander with the intercity towards Rotterdam Central
'Nine minutes later, take the intercity from Utrecht to Rotterdam Alexander.'

6. die vertrekt van perron zeven a b om zeventien uur
which leaves from platform seven a b at seventeen hour
negenen veertig
nine-and forty
'which leaves from platform 7a/b at 17:49.'

7. die komt om achttien uur zestien in Alexander aan
which arrives at eighteen hour sixteen in Alexander `prt`
'Which arrives in Alexander at 18:16."

8.	de	totale	reistijd	bedraagt	tweeen	vijftig	minuten
	the	total	travel-time	amounts-to	two-and	fifty	minutes
	'The total travelling time is 52 minutes.'						

Prosodic structure

1. [$_\pi$ [$_\varphi$ u kunt] [$_\varphi$ van Hilversum Sportpark] [$_\varphi$ naar Rotterdam Alexander] [$_\varphi$ met twee overstappen]]
2. [$_\pi$ [$_\varphi$ u neemt] [$_\varphi$ allereerst] [$_\varphi$ de stoptrein] [$_\varphi$ naar Utrecht Centraal]]
3. [$_\pi$ [$_\varphi$ die vertrekt] [$_\varphi$ om zeventien uur] [$_\varphi$ vierentwintig]]
4. [$_\pi$ [$_\varphi$ daarmee arriveert u] [$_\varphi$ dan in Utrecht] [$_\varphi$ om zeventien uur] [$_\varphi$ veertig]]
5. [$_\pi$ [$_\varphi$ u gaat] [$_\varphi$ negen minuten later] [$_\varphi$ van Utrecht] [$_\varphi$ naar Rotterdam Alexander] [$_\varphi$ met de intercity] [$_\varphi$ richting] [$_\varphi$ Rotterdam Centraal]]
6. [$_\pi$ [$_\varphi$ die vertrekt] [$_\varphi$ van perron zeven] [$_\varphi$ a] [$_\varphi$ b] [$_\varphi$ om zeventien uur] [$_\varphi$ negenen veertig]]
7. [$_\pi$ [$_\varphi$ Die komt] [$_\varphi$ om achttien uur] [$_\varphi$ zestien] [$_\varphi$ in Alexander] [$_\varphi$ aan]]
8. [$_\pi$ [$_\varphi$ de totale reistijd] [$_\varphi$ bedraagt] [$_\varphi$ tweeenvijftig minuten]]

Intonational form

1

%H			H*L	L*H	
[$_{\mathrm{AD}}$	u kunt	van	Hilversum	Sportpark	]
PMIBT	SAAR		DA	ADFA	AD-in-VP

%L		H*L	L*H	
[$_{\mathrm{AD}}$	naar	Rotterdam	Alexander	]
DIBT		DA	ADFA	AD-in-VP

%L		H*L	H*L	L%
[$_{\mathrm{AD}}$	met	twee	overstappen	]
DIBT		DA	DA	UFBT

2.

%H		H*L	H*L	H*L	L%
[$_{\mathrm{AD}}$	u neemt allereerst de	stoptrein naar	Utrecht	Centraal	]
PMIBT	SAAR	DA	DA	DA	UFBT

3.

%L		H*L	H*L		H*L	L%
![$_{\mathrm{AD}}$	die	vertrekt om	zeventien	uur	vierentwintig	]
DIBT		DA	DA	URR	DA	UFBT

4.
```
%L         H*L                       H*L
![AD daarmee arriveert u dan in Utrecht om zeventien uur
DIBT       DA                        DA       URR

H*L  L%
veertig ]
     UFBT
```

5.
```
%H     H*L  H*L         L*H
[AD  u gaat negen minuten later ]
PMIBT  DA   DA    RR    ADFA AD-in-VP

%L                 H*L       L*H
[AD van Utrecht naar Rotterdam Alexander ]
DIBT               DA        ADFA       AD-in-S-2

%L         H*L            H*L       H*L     L%
[AD met de intercity richting Rotterdam Centraal ]
DIBT       DA             DA        DA      UFBT
```

6.
```
%L     H*L              H*L   H*L L*H
[AD die vertrekt van perron zeven a  b   ]
DIBT   DA          NPD  DA    DA  ADFA AD-in-S-2

%L      H*L          H*L   H*L    L%
![AD om zeventien uur  negen veertig ]
DIBT    DA       URR DA    DA     UFBT
```

7.
```
%L              H*L        H*L                   H*L L%
![AD die komt om achttien uur  zestien in Alexander aan ]
DIBT            DA       URR                     DA  UFBT
```

8.
```
%L     H*L    H*L              H*L       H*L    L%
[AD de totale reistijd bedraagt tweeen vijftig minuten ]
DIBT   DA     DA       SAAR    DA     RR  DA    UFBT
```

E.2 Wild Flower Domain

E.2.1 Example 1: Knikkend Kattekruid

Abstract phrase specification

1.
```
(V:groeien[kattekruid<prop,neut>(A:knikkend)](P:in
({(tuin<plur>)(bouwland<plur>)})))
```

2.
```
(V:{(V:voorkomen[<pers,third,masc>](P:in(Europa<prop>
(A:geheel(ADV:vrijwel)))))(V:vinden<pas,perf>[]
[soort<def,back>](P:in({(Nederland<prop>)(Belgie<prop>)}))
(ADV:nooit))(CONJ:maar)})
```

3.
```
(V:{(V:zijn[stengel<def>][A:{(A:kort)(A:dik)
(A:donkerbruin)}])(V:zijn[blad<plur,def>][A:{(A:groot)
(A:donkergekleurd)(A:vlezig)}])(V:zijn[bloem<plur,def>]
[A:klokvormig])})
```

Surface string

1. knikkend kattekruid groeit in tuinen en bouwlanden
   bend cat-herb grows in gardens and building-land
2. hij komt in vrijwel geheel Europa voor maar de soort is in
   it occurs in nearly whole Europe `prt` but the species is in
   Nederland en Belgie nooit gevonden
   the-Netherlands and Belgium never found
3. de stengel is kort dik en donker bruin de bladeren zijn groot
   the stalk is short thick and dark brown the leafs are large
   donker gekleurd en vlezig en de bloemen zijn klok vormig
   dark coloured and fleshy and the flowers are bell shaped

Prosodic structure

1. $[_{\pi}$ $[_{\varphi}$ knikkend kattekruid] $[_{\varphi}$ groeit] $[_{\varphi}$ in tuinen] $[_{\varphi}$ en bouwlanden]]
2. $[_{\pi}$ $[_{\varphi}$ hij komt] $[_{\varphi}$ in vrijwel geheel Europa] $[_{\varphi}$ voor]] $[_{\pi}$ $[_{\varphi}$ maar de soort] $[_{\varphi}$ is in Nederland] $[_{\varphi}$ en Belgie] $[_{\varphi}$ nooit gevonden]]
3. $[_{\pi}$ $[_{\varphi}$ de stengel] $[_{\varphi}$ is kort] $[_{\varphi}$ dik] $[_{\varphi}$ en donker bruin] $[_{\varphi}$ de bladeren] $[_{\varphi}$ zijn groot] $[_{\varphi}$ donker gekleurd] $[_{\varphi}$ en vlezig] $[_{\varphi}$ en de bloemen] $[_{\varphi}$ zijn klok vormig]]

Intonational form

1.

%L	H*L	H*L			H*L		H*L	L%
![$_{AD}$	knikkend	kattekruid	groeit	in	tuinen	en	bouwlanden	]
DIBT	DA	DA	SAAR		DA		DA	UFBT

2.

%L				H*L		H*L	L*H	
[$_{AD}$	hij	komt	in	vrijwel	geheel	Europa	voor	]
DIBT				DA	RR	DA	ADFA	AD-in-S-1

%L						H*L		L*H	
![$_{AD}$	maar	de	soort	is	in	Nederland	en	Belgie	]
DIBT						DA		ADFA	AD-in-VP

%L	H*L	H*L	L%
[$_{AD}$	nooit	gevonden	]
DIBT	DA	DA	UFBT

3.

%L		H*L		H*L	H*L		H*L	L*H	
![$_{\text{AD}}$	de	stengel	is	kort	dik	en	donker	bruin	]
DIBT		DA		DA	DA		DA	ADFA	AD-in-S-1

%L		H*L		L*H	
[$_{\text{AD}}$	de	bladeren	zijn	groot	]
DIBT		DA		ADFA	AD-in-coor.

%L	H*L	L*H	
[$_{\text{AD}}$	donker	gekleurd	]
DIBT	DA	ADFA	AD-in-coor.

%L		L*H	
[$_{\text{AD}}$	en	vlezig	]
DIBT		ADFA	AD-in-coor.

%L			H*L		H*L	H*L	L%
[$_{\text{AD}}$	en	de	bloemen	zijn	klok	vormig	]
DIBT			DA		DA	DA	UFBT

E.2.2 Example 2: Koolmunt

Abstract phrase specification

1.
```
(V:zijn[koolmunt<prop,neut>][plant(A:eenjarig)(P:met
((stengel<plur>(A:(A:kort)(A:dik)(A:donkerbruin)))
(blad<plur>(A:(A:veerlobbig)(A:grofgetand)))))])
```

2.
```
(V:hebben[plant<def,back>][bloem<plur>(A:(A:rood)
(A:komvormig))(P:in(bloeiwijze<plur>(A:(A:dicht)
(A:langwerpig)(V:opbreken<part,pas>(ADV:vaak)
(P:in(schijnkrans<plur>(A:afzonderlijk)))))))])
```

3.
```
(V:groeien[<pers,third,masc>](P:in(Europa<prop>(ADV:Zuid)))
(P:(P:in((bos<plur>(A:(A:open)(A:licht)))(struikgewas<nonref>)
(wegberm<plur>)))(P:op(heuvel<plur>))(P:langs(spoorlijn<plur>
(A:(A:kaal)(A:droog))))))
```

Surface string

1.

koolmunt	is	een	eenjarige	plant	met	korte	dikke	donkerbruine
cabbage-mint	is	a	annual	plant	with	short	thick	dark-brown

stengels	en	veerlobbige	grof	getande	bladeren
stalks	and	lanceolate-lobate	coarsely	denticulated	leaves

2.

de	plant	heeft	rode	komvormige	bloemen	in	dichte	langwerpige
the	plant	has	red	bowl-shaped	flowers	in	dense	elongated

vaak	in	afzonderlijke	schijnkransen	opgebroken	bloeiwijzen
often	in	separate	fake-wreaths	broken-down	inflorescences

3. hij groeit in Zuid Europa in open lichte bossen struikgewas
it grows in Southern Europe in open light forests bushes
en wegbermen op heuvels en langs kale droge spoorlijnen
and roadsides at hills and along dry bare railways

Prosodic structure

1. $[_\pi$ $[_\varphi$ koolmunt] $[_\varphi$ is een een jarige plant] $[_\varphi$ met korte] $[_\varphi$ dikke] $[_\varphi$ donker bruine stengels] $[_\varphi$ en veer lobbige] $[_\varphi$ grof getande bladeren]]
2. $[_\pi$ $[_\varphi$ de plant] $[_\varphi$ heeft rode] $[_\varphi$ komvormige bloemen] $[_\varphi$ in dichte] $[_\varphi$ langwerpige] $[_\varphi$ vaak in afzonderlijke schijnkransen opgebroken bloeiwijzen]]
3. $[_\pi$ $[_\varphi$ hij groeit in Zuid Europa] $[_\varphi$ in open] $[_\varphi$ lichte bossen] $[_\varphi$ struikgewas] $[_\varphi$ en wegbermen] $[_\varphi$ op heuvels] $[_\varphi$ en langs kale] $[_\varphi$ droge spoorlijnen]]

Intonational form

1.

%L	H*L		H*L		L*H	
$[_{AD}$	koolmunt	is een	een	jarige	plant	]
DIBT	DA		DA	RR	ADFA	AD-in-NP-1

%L		L*H		%L	L*H	
$[_{AD}$	met	korte	]	$[_{AD}$	dikke	]
DIBT		ADFA	AD-in-NP-1	DIBT	ADFA	AD-in-NP-1

%L	H*L		L*H	
$[_{AD}$	donker	bruine	stengels	]
DIBT	DA	RR	ADFA	AD-in-coor.

%L		H*L	L*H		%L	H*L	
$[_{AD}$	en	veer	lobbige	$]_{AD-in-NP-1}$	$[_{AD}$	grof	getande
DIBT		DA	ADFA		DIBT	DA	RR

H*L	%L
bladeren	]
DA	UFBT

2.

%L		L*H		%L	H*L	
$[_{AD}$	de plant heeft	rode	]	$[_{AD}$	kom	vormige
DIBT		AFFA	AD-in-NP-1	DIBT	DA	RR

L*H	
bloemen	]
ADFA	AD-in-NP-2

%L		L*H		%L	L*H	
$[_{AD}$	in	dichte	]	$[_{AD}$	langwerpige	]
DIBT		ADFA	AD-in-NP-1	DIBT	ADFA	AD-in-NP-1

%L	H*L		H*L			
[AD	vaak	in	afzonderlijke	schijnkransen	op	gebroken
DIBT	DA		DA	RR	RR	

H*L	L%
bloeiwijzen	]
DA	UFBT

3.

%L		H*L		H*L	L*H		%L		L*H	
[AD	hij	groeit	in	Zuid	Europa	]	[AD	in	open	]
DIBT		DA		DA	ADFA	AD-in-S-2	DIBT		ADFA	AD-in-NP-1

%L	H*L	L*H		%L	L*H	
[AD	lichte	bossen	]	[AD	struikgewas	]
DIBT	DA	ADFA	AD-in-coor.	DIBT	ADFA	AD-in-coor.

%L		L*H		%L		L*H	
[AD	en	wegbermen	]	[AD	op	heuvels	]
DIBT		ADFA	AD-in-coor.	DIBT		ADFA	AD-in-coor.

%L			L*H		%L	H*L	H*L	L%
[AD	en	langs	kale	]	[AD	droge	spoorlijnen	]
DIBT			ADFA	AD-in-NP-1	DIBT	DA	DA	UFBT

References

Abney, S. (1986). *The English Noun Phrase in its Sentential Aspect.* PhD Dissertation, MIT.

Adorni, G. and M. Zock (1993). Introduction. In G. Adorni and M. Zock (Eds.), *Trends in Natural Language Generation*, pp. 1–16. Berlin: Springer.

Altenberg, B. (1990). Automatic text segmentation in tone units. In J. Svartik (Ed.), *The London-Lund Corpus of Spoken English*, pp. 287–323. Lund: Lund University Press.

Appelo, E., M. Leermakers, and J. Rous (1993). Template-based generation of natural language expressions with controlled m-grammar. In *IPO Annual Progress Report*, Number 28, Eindhoven, The Netherlands, pp. 131–138.

Baart, J. (1987). *Focus, Syntax and Accent Placement: towards a Rule System for the Derivation of Pitch Accents in Dutch as Spoken by Humans and Machines.* PhD Dissertation, University of Leiden.

Bachenko, J. and J. Fitzpatrick (1990). Discourse-neutral prosodic phrasing in English. *Computational Linguistics 16*(3), 155–170.

Bateman, J. A. (1994). KPML: The KOMET-Penman (multilingual) developement environment. Technical report, Institut für Integrierte Publikations- und Informationssysteme (IPSI), GMD.

Bateman, J. A. (1997). Editorial. *Speech Communication 21*, 35–36.

Bateman, J. A. and E. H. Hovy (1991). Computers and text generation: Principles and uses. In C. Butler (Ed.), *Computers and Texts: An Applied Perspective*, pp. 53–74. Oxford, England: Basil Blackwell.

Beach, C. (1991). The interpretation of prosodic patterns at points of syntactic structure ambiguity: Evidence for cue trading relations. *Journal of Memory and Language 30*, 644–663.

Beckman, M. (1995). Problems of intonation. In *Proceedings of the XIIIth International Congress of Phonetic Sciences*, Stockholm, Sweden, pp. 450–457.

Beckman, M. and J. Edwards (1994). Articulatory evidence for differentiating stress categories. In P. Keatings (Ed.), *Phonological Structure and Phonetic Form*, Volume III of *Papers in Laboratory Phonology*. Cambridge: Cambridge University Press.

Bennis, H. and T. Hoekstra (1989). *Generatieve Grammatica*. Dordrecht: Foris.

Birch, S. and C. Clifton (1995). Focus, accent, and argument structure. *Language and Speech 38*(4), 365–391.

Bolinger, D. (1983). Affirmation and default. *Folia Linguistica 17*, 99–116.

Booij, G. (1983). Principles and parameters in prosodic phonology. *Linguistics 21*, 249–280.

Booij, G. (1995). *The Phonology of Dutch*. Oxford: Clarendon Press.

Bosch, P. and R. van der Sandt (Eds.) (1999). *Focus: Linguistic, Cognitive and Computational Perspectives*. Cambridge: Cambridge University Press.

Bouma, G. (1996). Complement clauses and expletives. In G. Durieux, W. Daelemans, and S. Gillis (Eds.), *CLIN VI. Papers from the Sixth CLIN Meeting*, pp. 1–17. Antwerpen.

Bouma, G. and G. van Noord (1995). A lexicalist account of the Dutch verbal complex. In *Papers from the Fouth CLIN Meeting*. Groningen.

Brown, G. (1983). Prosodic structure and the given/new distinction. In A. Cutler and D. Ladd (Eds.), *Prosody: Models and Measurements*, pp. 67–77. Berlin: Springer.

Cambier-Langeveld, T. (2000). *Temporal marking of accents and boundaries*. PhD Thesis, University of Amsterdam. The Hague: Thesus.

Caspers, J. (1988). Experiments on the meaning of two pitch accent types: the 'pointed hat' versus the accent-lending fall in Dutch. In *Proceedings of the International Conference on Spoken Language Processing 1988*, pp. 1291–1294.

Caspers, J. (1994). *Pitch Movements under Time Pressure*. PhD Dissertation, University of Leiden. Den Haag: Holland Academic Graphics.

Caspers, J. (1998). Who's next? The melodic marking of question vs. continuation in Dutch. *Language and Speech 41*, 371–394.

Charpentier, F. and M. Stella (1986). Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Proceedings of the International Conference of Acoustics, Speech and Signal Processing*, pp. 2015–2018.

Chomsky, N. and M. Halle (1968). *The Sound Pattern of English.* New York: Harper and Row.

CoGenTex, Inc (1998). *RealPro General English Grammar User Manual.* Ithaca, New York: CoGenTex, Inc.

Coppen, P.-A. (1990). GRAMTSY 4.0: Grammaticaal Transformationeel Systeem. Manual, University of Nijmegen.

Coppen, P.-A. (1991). *Specifying the Noun Phrase.* PhD Thesis, University of Nijmegen. Amsterdam: Thesis Publishers.

Croft, W. (1995). Intonation units and grammatical structure. *Linguistics 33*, 839–882.

d'Alessandro, C. and J.-S. Liénard (1995). Synthetic speech generation. In R. A. Cole (Ed.), *Survey of the State of the Art in Human Language Technology*, pp. 196–201.

Dalianis, H. (1997). Natural language aggregation and clarification using cue words. In W. Hoeppner (Ed.), *Proceedings of the 6th European Workshop on Natural Language Generation*, Duisburg, pp. 6–16.

Dalianis, H. and E. Hovy (1993). Aggregation in natural language generation. In G. Adorni and M. Zock (Eds.), *Trends in Natural Language Generation*, pp. 88–105. Berlin: Springer.

Dalianis, H. and E. Hovy (1997). On lexical aggregation and ordering. In W. Hoeppner (Ed.), *Proceedings of the 6th European Workshop on Natural Language Generation*, Duisburg, pp. 17–27.

De Smedt, K. (1990). IPF: an incremental parallel formulator. In R. Dale, C. Mellish, and M. Zock (Eds.), *Current Research in Natural Language Generation*, pp. 167–192. London: Academic Press.

De Smedt, K. (1996). Computational models of incremental grammatical encoding. In A. Dijkstra and K. De Smedt (Eds.), *Computational Psycholinguistics: AI and Connectionist Models of Human Language Processing*, pp. 279–307. Hemel Hempstead: Prentice Hall / Harvester Wheatsheaf.

De Smedt, K., H. Horacek, and M. Zock (1996). Architectures for natural language generation: Problems and perspectives. In G. Adorni and M. Zock (Eds.), *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pp. 17–46. Berlin: Springer.

Degand, L. (1993). Dutch grammar documentation. Technical report, Institut für Integrierte Publikations- und Informationssysteme (IPSI), GMD.

Degand, L. (1996). A Dutch component for a multilingual systemic text generation system. In G. Adorni and M. Zock (Eds.), *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pp. 350–367. Berlin: Springer.

Derksen, C. and J. Potjer (2000). *Manual for the AGFL system. The GEN parser generator version 1.8.* University of Nijmegen, Computing Science Institute.

Dirksen, A. (1992). Accenting and deaccenting: A declarative approach. In *Proceedings of the 14th International Conference on Computatational Linguistics*, Nantes, France, pp. 865–869.

Dirksen, A. and H. Quené (1993). Prosodic analysis: The next generation. In V. van Heuven and L. Pols (Eds.), *Analysis and Synthesis of Speech*, pp. 131–144. Berlin: Mouton de Gruyter.

Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis.* Dordrecht: Kluwer Academic Publishers.

Dutoit, T. and H. Leich (1993). MBR-PSOLA: Text-to-speech synthesis based on an MBE resynthesis of the segments database. *Speech Communication 13*, 435–440.

Dutoit, T., V. Pagel, N. Pierret, O. van der Vreken, and F. Bataille (1996). The MBROLA project: towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia.

Elffers, E. (1992). De onaccentueerbaarheid van modale partikels. *Grammaticaal Spektakel*, 59–70.

Elhadad, M. (1993). *FUF: The Universal Unifier - User Manual, version 5.2.* New York: Columbia University. Technical Report CUCS-038-91.

Elhadad, M., K. McKeown, and J. Robin (1997). Floating constraints in lexical choice. *Computational Linguistics 23*(2), 195–239.

Elhadad, M. and J. Robin (1996). An overview of SURGE: A reusable comprehensive syntactic realization component. Technical Report 96-03, Ben Gurion University, Department of computer Science.

Elhadad, M. and J. Robin (to appear). SURGE: a comprehensive plug-in syntactic realization component for text generation. *Computational Linguistics*.

Ernestus, M. (2000). *Voice assimilation and segment reduction in casual Dutch: a corpus-based study of the phonology-phonetics interface.* PhD Dissertation, Vrije Universiteit Amsterdam.

ESPRIT (1992). Standard computer-compatible transcription. ESPRIT Project 2589, Stage Report Sen.93 SAM-UCL-037.

Faber, D. (1987). The accentuation of intransitive sentences in English. *Journal of Linguistics 23*, 341–358.

Ferreira, F. (1993). Creation of prosody during sentence production. *Psychological Review 100*(2), 233–253.

Foley, W. (1994). Information structure. In R. Asher and J. Simpson (Eds.), *The Encyclopedia of Language and Linguistics*, pp. 200–209. Oxford: Pergamon Press.

Forey, P. (1992). *Wilde Bloemen van Europa.* Baarn: Thieme.

Fowler, C. and J. Housum (1987). Talkers' signalling of "old" and "new" words in speech and listeners' perception and use of the distinction. *Journal of Memory and Language 26*, 489–504.

Fraczak, L., G. Lapalme, and M. Zock (1998). Automatic generation of subway directions: salience gradation as a factor for determining message and form. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, pp. 58–67.

Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985). *Generalized Phrase Structure Grammar.* Cambridge, MA: Harvard University Press.

Gee, J. P. and F. Grosjean (1983). Performance structures: a psycholinguistic and linguistic appraisal. *Cognitive Psychology 15*, 411–458.

Grabe, E., C. Gussenhoven, J. Haan, E. Marsi, and B. Post (1998). Preaccentual pitch and speaker attitude in Dutch. *Language and Speech 41*(1), 63–85.

Grabe, E., P. Warren, and F. Nolan (1994). Resolving category ambiguities – Evidence from stress shift. *Speech Communication 15*, 101–114.

Grimshaw, J. (1990). *Argument Structure.* Cambridge, MA: MIT Press.

Griswold, R. E. and M. T. Griswold (1993). *The Icon Programming Language* (2nd ed.). Englewood Cliffs, New Jersey: Prentice-Hall.

Grosz, B. and J. Hirschberg (1992). Some intonational characteristics of discourse structure. In *2nd International Conference on Spoken Language Processing*, Banff, Canada, pp. 429–432.

Grosz, B. and C. Sidner (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics 12*(3), 175–204.

Grosz, B. J., A. K. Joshi, and S. Weinstein (1995). Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics 21*(2), 203–225.

Gulikers, L. and R. Willemse (1992). A lexicon for a text-to-speech system. In *2nd International Conference on Spoken Language Processing*, Banff, Canada, pp. 101–104.

Gussenhoven, C. (1983a). Stress shift and the nucleus. *Linguistics 21*, 303–339.

Gussenhoven, C. (1983b). Testing the reality of focus domains. *Language and Speech 26*, 61–80.

Gussenhoven, C. (1984). *On the Grammar and Semantics of Sentence Accents.* PhD Dissertation, University of Nijmegen. Dordrecht: Foris.

Gussenhoven, C. (1988). Adequacy in intonation analysis: the case of Dutch. In H. van der Hulst and N. Smith (Eds.), *Advances in Nonlinear Phonology*, pp. 95–121. Dordrecht: Foris.

Gussenhoven, C. (1990). Tonal association domains and the prosodic hierarchy in English. In S. Ramsaran (Ed.), *Studies in the Pronunciation of English*, pp. 27–37. London, New York: Routledge.

Gussenhoven, C. (1991a). The English rhythm rule as an accent deletion rule. *Phonology 8*, 1–35.

Gussenhoven, C. (1991b). Tone segments in the intonation of Dutch. In T. Shannon and J. Snapper (Eds.), *The Berkeley Conference on Dutch Linguistics 1989*, pp. 139–155. Lanham MD: University Press of America.

Gussenhoven, C. (1992a). Intonational phrasing and the prosodic hierarchy. In W. Dressler, H. Luschützky, O. Pfeiffer, and J. Rennison (Eds.), *Phonologica 1988*, pp. 89–99. Cambridge: Cambridge University Press.

Gussenhoven, C. (1992b). Sentence accents and argument structure. In I. Roca (Ed.), *Thematic Structure: its Role in Grammar*, pp. 79–106. Berlin: Foris.

Gussenhoven, C. (1993). The Dutch foot and the chanted call. *Journal of Linguistics 29*, 37–63.

Gussenhoven, C. (1999). On the limits of focus projection in English. In P. Bosch and R. van der Sandt (Eds.), *Focus: Linguistic, Cognitive and Computational Perspectives*, pp. 43–55. Cambridge: Cambridge University Press.

Gussenhoven, C. and T. Rietveld (1992a). Intonation contours, prosodic structure and preboundary lengthening. *Journal of Phonetics 20*, 283–303.

Gussenhoven, C. and T. Rietveld (1992b). A target-interpolation model for the intonation of Dutch. In *Proceedings of the International Conference on Spoken Language Processing*, Banff, Canada, pp. 1235–1238.

Haegeman, L. (1991). *Introduction to Government & Binding Theory.* Oxford: Blackwell.

Haeseryn, W., K. Romijn, G. Geerts, J. de Rooij, and M. van den Toorn (1997). *Algemene Nederlandse Spraakkunst* (2nd ed.). Groningen: Martinus Nijhoff.

Halliday, M. (1994). *An Introduction to Functional Grammar.* London: Edward Arnold.

Hayes, B. (1984). The phonology of rhythm in English. *Linguistic Inquiry 15*, 33–74.

Hayes, B. (1986). The prosodic hierarchy in meter. In P. Kiparsky and G. Youmans (Eds.), *Rhythm and Meter*, pp. 201–260. Orlando: Academic Press.

Hayes, B. (1990). Precompliled phrasal phonology. In S. Inkelas and D. Zec (Eds.), *The Phonology-Syntax Connection*, pp. 85–108. Chicago: University of Chicago Press.

Hayes, B. and A. Lahiri (1991). Bengali intonational phonology. *Natural Language and Linguistic Theory 9*, 47–96.

Heinzel, U. (1999). Prosody: a clue for the interpretation of focus particles. In W. Bergsma, M.-J. Palmen, and M. Wester (Eds.), *Proceedings of the CLS opening Academic Year 1999/2000*, pp. 31–53.

Hess, W. J. (1997). Section introduction. a brief history of applications. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg (Eds.), *Progress in Speech Synthesis*, pp. 563–564. New York: Springer-Verlag.

Hirschberg, J. (1991). Using text analysis to predict intonational boundaries. In *Eurospeech 91*, pp. 1275–1278.

Hirschberg, J. (1992). Using discourse context to guide accent decisions in synthetic speech. In G. Bailly and C. Benoit (Eds.), *Talking Machines: Models and Design*, pp. 367–376. Amsterdam: North-Holland.

Hirschberg, J. (1993). Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence 63*, 305–340.

Hirst, D. (1993). Detaching intonational phrases from syntactic structure. *Linguistic Inquiry 24*(4), 781–788.

Hiyakumoto, L., S. Prevost, and J. Cassell (1997). Semantic and discourse information for text-to-speech intonation. In *Proceedings of the ACL/EACL Workshop on Concept to Speech Generation*, Madrid, pp. 47–56.

Hofhuis, E., C. Gussenhoven, and A. Rietveld (1995). Final lengthening at prosodic boundaries in Dutch. In *Proceedings of the International Congress of Phonetic Sciences*, Stockholm, pp. 154–157.

Horne, M. (1990). Empirical evidence for a deletion formulation of the rhythm rule in English. *Linguistics 28*, 959–981.

Horne, M. (1991). Phonetic correlates of the 'given/new' parameter. In *Proceedings of the International Congress of Phonetic Sciences 91*, Aix-en-Provence, pp. 230–233.

Horne, M., M. Fillipson, C. Johansson, M. Ljungqvist, and A. Linström (1993). Improving the prosody in TTS systems: morphological and lexical-semantic methods for tracking 'new' vs. 'given' information. In *Proceedings of the ESCA Tutorial and Research Workshop on Prosody*, Lund, pp. 208–211. (Working papers 41, Lund University, Department of Linguistics and Phonetics).

Hovy, E. H. (1988). Planning coherent multisentential text. In *Proceedings of the 28th Annual Meeting of the ACL*, pp. 163–169.

Hovy, E. H. (1991). Approaches to the planning of coherent text. In W. Swartout and W. Mann (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 83–102. Boston: Kluwer Academic Publishers.

Inkelas, S. and D. Zec (1990). *The Phonology-Syntax Connection.* Chicago: University of Chicago Press.

Inkelas, S. and D. Zec (1995). Syntax-phonology interface. In J. A. Goldsmith (Ed.), *The Handbook of Phonological Theory*, pp. 535–549. Cambridge, MA: Blackwell.

Inui, K., T. Tokunaga, and H. Tanaka (1992). Text revision: A model and its implementation. In R.Dale, E.Hovy, D.Rösner, and O.Stock (Eds.), *Aspects of Aumtomated Natural Language Generation*, Number 587 in Lecture Notes in Artificial Intelligence, pp. 215–230. Berlin: Springer.

Jackendoff, R. (1977). *$\overline{X}$ Syntax: A Study of Phrase Structure.* Cambridge, MA: MIT Press.

Jackendoff, R. (1990). *Semantic Structures.* Cambridge, MA: MIT Press.

Jokinen, K., H. Tanaka, and A. Yokoo (1998). Planning dialogue contributions with new information. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, pp. 158–167.

Kager, R. and H. Quené (1987). Deriving prosodic sentence structure without exhaustive syntactic analysis. In *Proceedings of the European Conference on Speech Technology*, pp. 243–246.

Kager, R. and H. Quené (1989). An algorithm for the assignment of sentence accents for a Dutch text-to-speech system. In *Linguistics in the Netherlands.*

Keijsper, C. (1984a). Vorm en betekenis in Nederlandse toonhoogtecontouren I. *Forum der Letteren 25*, 20–37.

Keijsper, C. (1984b). Vorm en betekenis in Nederlandse toonhoogtecontouren II. *Forum der Letteren 25*, 113–126.

Kelly, M. (1992). Using sound to solve syntactic problems: the role of phonology in grammatical category assignment. *Psychological Review 99*, 349–364.

Kempen, G. and E. Hoenkamp (1987). An incremental procedural grammar for sentence production. *Cognitive Science 11*, 201–258.

Kerkhoff, J. and T. Rietveld (1994). Prosody in NIROS with FONPARS and ALFEIOS. In P. de Haan and M. Oostdijk (Eds.), *Proceedings 18*, pp. 107–119. University of Nijmegen, Department of Language and Speech.

Kerkhoff, J. and T. Rietveld (1995). The generation of prosody in the Nijmegen Rule Oriented Speech synthesis system. In *Eurospeech 1995*, Volume 3, Madrid, pp. 1381–1384.

Kerkhoff, J., T. Rietveld, and D. van Bergem (1998). Het Nijmeegse tekst-naar-spraak systeem KUNTTS. Technical report, University of Nijmegen, Department of Language and Speech.

Klatt, D. H. (1987). Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America 82*(3), 737–793.

Koster, C. (1991). A technique for parsing ambiguous languages. In H. Albas and B. Melichar (Eds.), *Attribute Grammars, Applications and Systems.* Berlin: Springer-Verlag.

Kruyt, J. (1985). *Accents from Speakers to Listeners.* PhD Dissertation, University of Leiden.

Ladd, D. (1984). Declination: a review and some hypotheses. *Phonology Yearbook 1*, 53–74.

Ladd, D. (1986). Intonational phrasing: the case for recursive prosodic structure. *Phonology Yearbook 3*, 311–340.

Ladd, D. (1987). A phonological model of intonation for use in speech synthesis. In *Proceedings of the European Conference on Speech Technology*, Edinburgh, pp. 21–24. CEP Associates.

Ladd, D. (1990). Metrical representation of pitch register. In J. Kingston and M. Beckman (Eds.), *Papers in Laboratory Phonology*, pp. 35–57. Cambridge: Cambridge University Press.

Ladd, D. (1992). An introduction to intonational phonology. In G. Docherty and D. Ladd (Eds.), *Papers in Laboratory Phonology II*, pp. 321–334. Cambridge: Cambridge University Press.

Ladd, D. (1996). *Intonational Phonology*. Cambridge: Cambridge University Press.

Lavoie, B. and O. Rambow (1997). A fast and portable realizer for text generation. In *Proceedings of the Fifth Conference on Applied Natural-Language Processing*, Washington, pp. 265–268.

Lehiste, I. (1975). The phonetic structure of paragraphs. In A. Cohen and S. Nooteboom (Eds.), *Structure and Process in Speech Perception*, pp. 195–203. Heidelberg: Springer.

Lehiste, I. (1979). Perception of sentence and paragraph boundaries. In B. Lindblom and S. Öhman (Eds.), *Frontiers of Speech Research*, pp. 191–201. London: Academic Press.

Lennig, M. and J. Brassard (1984). Machine readable phonetic alphabet for English and French. *Speech Communication 3*, 165–166.

Liberman, M. and J. Pierrehumbert (1984). Intonational invariance under changes in pitch range and length. In M. Aronoff and M. T. Oehrle (Eds.), *Language Sound Structures: Studies in Phonology presented to Morris Halle*. Camdridge, MA: MIT Press.

Loots, M. (1983). Syntax and assimilation of voice in dutch. In M. van den Broecke, V. van Heuven, and W. Zonneveld (Eds.), *Sound Structures. Studies for Antonie Cohen*. Dordrecht: Foris.

Mann, W. and S. Thompson (1988). Rhetorical structure theory: towards a functional theory of text organization. *Text 8*(2), 243–281.

Marsi, E. (1995). Intonation in a spoken language generator. In H. Strik, N. Oostdijk, C. Cucchiarini, and P. Coppen (Eds.), *Proceedings 1995*, pp. 85–97. University of Nijmegen, Department of Language and Speech.

Marsi, E. (1998a). Introducing maximal variation in text planning for small domains. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, pp. 68–77.

Marsi, E. (1998b). A reusable syntactic generator for Dutch. In P.-A. Coppen, H. V. Halteren, and L. Teunissen (Eds.), *Computational Linguistics in the Netherlands 1997. Selected Papers from the Eight CLIN Meeting*. Amsterdam: Rodopi.

Marsi, E., P.-A. Coppen, C. Gussenhoven, and T. Rietveld (1997). Prosodic and intonational domains in speech synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg (Eds.), *Progress in Speech Synthesis*, pp. 477–493. New York: Springer-Verlag.

Matthiesen, C., Z. Licheng, M. Cross, I. Kobayashi, K. Teruya, and W. Canzhong (1998). The multex generator and its environment: application and developement. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, pp. 228–237.

Matthiessen, C. and J. Bateman (1991). *Text generation and Systemic Functional Grammar: Experiences from English and Japanese.* London: Francis Pinter Publishers.

McCoy, K. and J. Cheng (1991). Focus of attention: Constraining what can be said next. In C. Paris, W. Swartout, and W. Moore (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 103–124. Kluwer Academic Publishers.

McDonald, D. (1999). Natural language generation. In R. Dale, H. Moisl, and H. Somers (Eds.), *A Handbook of Natural Language Processing Techniques.* New York: Marcel Dekker Inc.

McKeown, K. (1985a). Discourse strategies for generating natural-language text. *Artificial Intelligence 27*, 1–41.

McKeown, K., S.Pan, J.Shaw, D. Jordan, and B. Allen (1997). Language generation for multimedia healthcare briefings. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pp. 227–282.

McKeown, K. R. (1985b). *Text Generation: Using Discourse Strategies and Focus Contraints to Generate Natural Language Text.* Cambridge: Cambridge University Press.

McKeown, K. R. and J. D. Moore (1995). Spoken language generation. In R. A. Cole (Ed.), *Survey of the State of the Art in Human Language Technology*, pp. 210–215.

McKeown, K. R. and S. Pan (to appear). Prosody modeling in concept-to-speech generation: Methodological issues. *Philosophical Transactions of the Royal Society Series A.*

Mel′čuk, I. (1988). *Dependency Syntax: Theory and Practice.* Albany, New York: State University of New York Press.

Menert, L. (1994). *Experiments on Voice Assimilation in Dutch : Prosodic Structures and Tempo.* PhD Dissertation, University of Utrecht. Utrecht: LEd.

Model, J. (1991). *Grammatische Analyse.* Dordrecht: ICG Publications.

Monaghan, A. (1993). The intonation of textual anomalies in text-to-speech. *Speech Communication 12*, 371–382.

Monaghan, A. and D. Ladd (1990). Symbolic output as the basis for evaluating intonation in text-to-speech systems. *Speech Communication 9*(4), 305–314.

Morais, J., L. Cary, J. Alegria, and P. Bertelson (1979). Does awareness of speech as a sequence of phones arise spontaneously? *Cognition 7*, 323–331.

Nachtegaal, D. (1997). An evaluation of Goalgetter's accentuation. Technical Report 1142, IPO, Eindhoven, The Netherlands.

Neijt, A. (1984). Fie (ϕ) vorming. In G. de Haan, M. Trommelen, and W. Zonneveld (Eds.), *Van Periferie naar Kern*, pp. 169–177. Dordrecht: Foris.

Nespor, M. and I. Vogel (1986). *Prosodic Phonology.* Dordrecht: Foris.

Noordman, L., I. Dassen, M. Swerts, and J. Terken (1999). Prosodic markers of text structure. In K. van Hoek, A. A. Kibrik, and L. Noordman (Eds.), *Discourse Studies in Cognitive Linguistics*, pp. 133–148. Amsterdam: Benjamins. Selected Papers from the Fifth International Cognitive Linguistics Conference, Amsterdam, July 1997.

Nooteboom, S. and J. Kruyt (1987). Accents, focus distribution, and the perceived distribution of given and new information: An experiment. *Journal of the Accoustical Society of America 82*(5), 1512–1524.

Nooteboom, S. and J. Terken (1982). What makes speakers omit pitch accents? An experiment. *Phonetica 39*, 317–336.

Ostendorf, M. and N. Veilleux (1994). A hierachical stochastic model for automatic prediction of prosodic boundary location. *Computational Linguistics 20*(1), 27–54.

Paiva, D. S. (1998). A survey of applied natural language generation systems. Technical report ITRI-98-03, ITRI, Brighton.

Pan, S. and J. Hirschberg (2000). Modeling local context for pitch accent prediction. In *Proceedings of ACL'2000*, Hong Kong.

Pan, S. and K. McKeown (1999). Word informativeness and automatic pitch accent modeling. In *Proceedings of EMNLP/VLC'99*, College Park, Maryland.

Paris, C. (1988). Tailoring object descriptions to the user's level of expertise. *Computational Linguistics 14*(3), 64–78.

Pierrehumbert, J. (1980). *The Phonology and Phonetics of English Intonation.* PhD Dissertation, MIT.

Pierrehumbert, J. and M. Beckman (1988). *Japanese Tone Structure.* Cambridge, MA: MIT Press.

Pierrehumbert, J. and J. Hirschberg (1990). The meaning of intonational contours in the interpretation of discourse. In P. Cohen, J. Morgan, and M. Pollack (Eds.), *Intentions in Communications*, pp. 271–311. Cambridge: MIT Press.

Pollard, C. and I. Sag (1994). *Head-Driven Phrase Structure Grammar.* Chicago: University of Chicago Press.

Post, B. (2000). *Tonal and Phrasal Structures in French Intonation.* PhD Dissertation, University of Nijmegen. Den Haag: Holland Academic Graphics.

Postal, P. (1969). On so-called "pronouns" in English. In D. Reibel and S. Schane (Eds.), *Modern Studies in English*, pp. 201–224. Englewood Cliffs, New Yersey.

Postal, P. (1971). *Cross-over Phenomena.* New York: Holt, Rinehart & Winston.

Prevost, S. and M. Steedman (1994). Specifying intonation from context for speech synthesis. *Speech Communication 15*, 139–153.

Price, P., M. Ostendorf, S. Shattuck-Hufnagel, and C. Fong (1991). The use of prosody in syntactic disambiguation. *Journal of the Accoustical Society of America 90*, 2956–2970.

Prince, A. and P. Smolensky (1993). *Optimality Theory: Constraint Interaction in Generative Grammar.* Cambridge, MA: MIT Press.

Prince, E. (1981). Towards a taxonomy of given-new information. In P. Cole (Ed.), *Radical Pragmatics*, pp. 223–255. New York: Academic Press.

Quené, H. and R. Kager (1989). Automatic accentuation and prosodic phrasing for Dutch text-to-speech conversion. In *Proceedings of Eurospeech 89*, pp. 214–217. Paris, France.

Rats, M. (1996). *Topic Management in Information Dialogues.* PhD Dissertation, University of Tilburg.

Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation (INLGW-1994)*, pp. 163–170.

Reiter, E. (1995). NLG versus templates. In *Proceedings of 7th European Workshop on Natural Language Generation*, Leiden, the Netherlands.

Reiter, E. and R. Dale (1997). Building applied natural language generation systems. *Natural Language Engineering 1*(3), 57–87.

Reiter, E. and R. Dale (2000). *Building Natural Language Generation Systems.* Cambridge: Cambridge University Press.

Rietveld, T. and C. Gussenhoven (1995). Aligning pitch targets in speech synthesis: Effects of syllable structure. *Journal of Phonetics 23*, 375–385.

Rietveld, T., J. Kerkhoff, and C. Gussenhoven (1999). Prosodic structure and vowel duration in Dutch. In *Proceedings of the International Congress of Phonetic Sciences*, pp. 463–466.

Rietveld, T. and R. van Hout (1993). *Statistical Techniques for the Study of Language and Language Behaviour.* Berlin: Mouton de Gruyter.

Rijpma, E. and F. Schuringa (1978). *Nederlandse Spraakkunst.* Groningen: Wolters-Noordhoff. (edited by Jan van Bakel).

Rochemont, M. and P. Cullicover (1990). *English Focus Constructions and the Theory of Grammar.* Cambridge: Cambridge Universtity Press.

Rooth, M. (1995). Focus. In S. Lappin (Ed.), *Handbook of Contemporary Semantic Theory*, pp. 271–297. Oxford: Blackwell.

Rubinoff, R. (1992). Integrating text planning and linguistic choice by annotating linguistic structures. In R.Dale, E.Hovy, D.Rösner, and O.Stock (Eds.), *Aspects of Automated Natural Language Generation*, pp. 45–56. Berlin: Springer.

Sanderman, A. (1996). *Prosodic phrasing: Production, Perception, Acceptability and Comprehension.* PhD Dissertation, Eindhoven University of Technology.

Sanders, E. and P. Taylor (1995). Using statistical models to predict phrase boundaries for speech synthesis. In *Proceedings of the Fourth European Conference of Speech Communication and technology (Eurospeech '95)*, Madrid, Spain.

Schmerling, S. (1976). *Aspects of English Sentence Stress.* Austin: Texas University Press.

Selkirk, E. (1984). *Phonology and Syntax: the Relation between Sound and Structure.* Cambridge, MA: MIT Press.

Selkirk, E. (1995). Sentence prosody: Intonation, stress and phrasing. In J. A. Goldsmith (Ed.), *The Handbook of Phonological Theory*, pp. 550–569. Cambridge, MA: Blackwell.

Selkirk, E. (1996). On derived domains in sentence phonology. *Phonology Yearbook 3*, 371–404.

Selkirk, E. (2000). The interaction of contraints on prosodic phrasing. In M. Horne (Ed.), *Prosody: Theory and Experiment. Studies presented to Gösta Bruce*, pp. 231–261. Dordrecht: Kluwer Academic Publishers.

Shattuck-Hufnagel, S. and A. Turk (1996). A prosody tutorial for investigators of auditory sentence processing. *Journal of Psycholinguistic Research 25*(2), 193–247.

Shaw, J. (1998). Clause aggregation using linguistic knowledge. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, pp. 138–147.

Sluijter, A. (1995). *Phonetic Corelates of Stress and Accent.* PhD Dissertation, University of Leiden.

Sluijter, A. and J. Terken (1993). Beyond sentence prosody: paragraph intonation in Dutch. *Phonetica 50*, 180–188.

Sproat, R., A. Hunt, M. Ostendorf, P. Taylor, A. Black, K. Lenzo, and M. Edginton (1998). Sable: A standard for tts markup. In *Proceedings of the International Congress of Phonetic Sciences.*

Sproat, R. W. (1997). Section introduction. the analysis of text in *text*-to-speech synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg (Eds.), *Progress in Speech Synthesis*, pp. 71–75. New York: Springer-Verlag.

Stede, M. (1995). Lexicalization in natural language generation: A survey. *Artificial Intelligence Review 8*, 309–336.

Steedman, M. (1991). Structure and intonation. *Language 67*(2), 260–296.

Swerts, M. (1994). *Prosodic Features of Discourse Units.* PhD Dissertation, Eindhoven University of Technology.

Swerts, M. (1999). Ritme als verklarende factor voor de keuze tussen groene en rode werkwoordvolgorde in het Nederlands. *Nederlandse Taalkunde 3*(4), 299–308.

Swerts, M., R. Collier, and J. Terken (1994). Prosodic predictors of discourse finality in spontaneous monologues. *Speech Communication 15*, 79–90.

Swerts, M. and R. Geluykens (1993). The prosody of information units in spontaneous monologue. *Phonetica 50*, 189–196.

't Hart, J., R. Collier, and A. Cohen (1990). *A Perceptual Study of Intonation.* Cambridge: Cambridge University Press.

Taylor, P. and A. W. Black (1998). Assigning phrase breaks from part-of-speech sequences. *Computer Language and Speech 12*(2), 97–117.

Teich, E., E. Haen, B. Grote, and J. Bateman (1997). From communicative context to speech: integrating dialogue processing, speech production, and natural language generation. *Speech Communication 21*, 73–99.

Terken, J. (1993). Synthesizing natural-sounding intonation. *Computer Speech and Language 7*, 27–48.

Teunissen, L. (1997). GENIUSS: generative implementation of Universal Semantic Syntax. In W. Hoepnner (Ed.), *6th European Workshop on Natural Language Generation*, Gerhard-Mercator-Universität-GH Duisburg.

Theune, M. (1997a). Contrastive accent in a data-to-speech system. In *Proceedings of the 35th Annual meeting of the ACL / the 8th Conference of the EACL*, Madrid, pp. 519–521.

Theune, M. (1997b). Goalgetter: Predicting contrastive accent in data-to-speech generation. In J. Landsbergen, J. Odijk, K. van Deemter, and G. V. van Zanten (Eds.), *Computational Linguistics in the Netherlands 1996; Papers from the Seventh CLIN Meeting*, pp. 177–190. Eindhoven: Eindhoven University of Technology.

Theune, M., E. Klabbers, J. Odijk, and J. de Pijper (1997). Computing prosodic properties in a data-to-speech system. In *Proceedings of the Workshop on Concept-to-Speech Generation Systems (ACL/EACL'97)*, Madrid, Spain, pp. 39–45.

Truckenbrodt, H. (1999). On the relation between syntactic phrases and phonological phrases. *Linguistic Inquiry 30*(2), 257–287.

van Bezooijen, R. (1989). Evaluation of an algorithm for the automatic assignment of sentence accents in written text. Technical Report ASSP-Report 9, Stichting Spraaktechnologie, University of Amsterdam.

van Deemter, K. (1994). What's new? A semantic perspective on sentence accent. *Journal of Semantics 11*, 1–31.

van Deemter, K. and J. Odijk (1997). Context modelling and the generation of spoken discourse. *Speech Communication 21*(1/2), 101–121.

van den Berg, R., C. Gussenhoven, and T. Rietveld (1992). Downstep in Dutch: Implications for a model. In G. Docherty and D. Ladd (Eds.), *Papers in Laboratory Phonology*, Volume II, pp. 335–359. Cambridge: Cambridge University Press.

van den Toorn, M. (1984). *Nederlandse Grammatica*. Groningen: Wolters-Noordhoff.

van der Lubbe, H. (1958). *Woordvolgorde in het Nederlands: Een Synchrone Structurele Beschouwing*. Assen: Van Gorcum.

van Donzel, M. (1999). *Prosodic Aspects of Information Structure in Discourse*. PhD Dissertation, University of Amsterdam. Den Haag: Holland Academic Graphics.

van Dreumel, S. (2000). The Amazon grammar and the last part of the middle field. In F. van Eynde, I. Schuurman, and N. Schelkens (Eds.), *Computational Linguistics in the Netherlands 1998*, pp. 93–107. Amsterdam: Rodopi.

van Eynde, F. (1997). On the notion 'minor category'. In J. Landsbergen, J. Odijk, K. van Deemter, and G. V. van Zanten (Eds.), *Computational Linguistics in the Netherlands 1996; Papers from the Seventh CLIN Meeting*, pp. 191–206. Eindhoven: Eindhoven University of Technology.

van Kuppevelt, J. (1991). *Topic en Comment. Expliciete en Impliciete Vraagstelling in Discourse*. PhD Dissertation, University of Nijmegen.

van Wijk, C. (1987a). The PSY behind PHI: a psycholinguistic model for performance structures. *Journal of Psycholinguistic Research 16*(2), 185–199.

van Wijk, C. (1987b). *Speaking, writing, and sentence form: Three psycholinguistic studies.* PhD Dissertation, University of Nijmegen.

Varga, L. (1998). Rhythmical variation in Hungarian. *Phonology 15*, 227–266.

Verhagen, A. (1986). *Linguistic Theory and the Function of Word Order in Dutch.* Dordrecht: Mouton de Gruyter/Foris.

Wichmann, A., J. House, and T. Rietveld (2000). Discourse constraints of F_0 peak timing in English. In A. Botinis (Ed.), *Intonation: Analysis, Modelling and Technology*, pp. 163–182. Dordrecht: Kluwer Academic Publishers.

Young, S. J. and F. Fallside (1979). Speech synthesis from concept: a method for speech output from information systems. *Journal of the Acoustic Society of America 66*(3), 685–695.

Zec, D. and S. Inkelas (1990). Prosodically constrained syntax. In S. Inkelas and D. Zec (Eds.), *The Phonology-Syntax Connection*, pp. 365–378. Chicago: University of Chicago Press.

Zwart, C.-W. (1993). *Dutch Syntax: A Minimalist Approach.* PhD Dissertation, University of Utrecht.

Samenvatting

Intonatie De intonatie van een uiting omvat alle aspecten van het waargenomen toonhoogtepatroon die de luisteraar gebruikt om die uiting te begrijpen, of die de spreker daartoe heeft aangebracht. In het algemeen worden drie aspecten van intonatie onderscheiden: zinsaccenten, intonationele grenzen, en zinsmelodie. Een *zinsaccent* verleent een woord nadruk, en correspondeert met een relatief snelle verandering van de toonhoogte rondom de beklemtoonde lettergreep van het woord. Een *intonationele grens* klinkt als een lichte onderbreking in het spreken (hoewel er niet noodzakelijk een pauze hoeft te zijn) die wordt waargenomen als gevolg van een verandering in de toonhoogte rondom de grens. De *zinsmelodie* betreft de specifieke vorm van het toonhoogtepatroon, die onafhankelijk van de plaats van accenten en grenzen kan variëren. Zo kan een mededeling worden omgezet in een vraag zonder de accenten of de grenzen te verplaatsen, maar enkel door de melodie te veranderen.

Net zoals Nederlandse woorden en zinnen *structureel welgevormd* moeten zijn, d.w.z. moeten gehoorzamen aan de grammatica van het Nederlands, zo moet de intonatie van een uiting minimaal voldoen aan de intonationele grammatica. Daarnaast moet de intonatie ook passen bij de vorm, betekenis en context van een uiting. Sprekers van het Nederlands kunnen beoordelen of de intonatie van een gegeven uiting acceptabel is in een bepaalde context. Vanuit het oogpunt van de generatieve taalkunde is het doel om deze oordelen te kunnen voorspellen en verklaren. Hiertoe wordt een taalkundig model opgesteld in de vorm van een formele grammatica die alle acceptabele intonatiepatronen kan genereren, maar geen onacceptabele.

Hierbij doet zich een aantal methodologische problemen voor. Acceptabiliteitsoordelen kunnen namelijk variatie vertonen doordat ze gradueel van aard zijn (de gekozen intonatie kan in meer of mindere mate acceptabel zijn), doordat verschillende alternatieven even acceptabel kunnen zijn, of doordat proefpersonen onderling enigszins verschillen in hun oordelen. Verder is het moeilijk om perceptie-experimenten op te zetten waarbij ongeoefende luisteraars een zuiver oordeel over de intonatie kunnen vellen, aangezien ze de benodigde vaardigheid in analytische luisteren missen. Daarom is er in dit proefschrift voor gekozen om een taalkundig model te maken voor de intonatie die het *meest acceptabel* is (waarbij verschillen tussen even acceptabele varianten worden genegeerd) voor *deskundige luisteraars* (intonationele experts).

Spraakgenerering De centrale vraag binnen het onderzoeksgebied van de *natuurlijke taalgenering* is hoe we computers zo kunnen programmeren dat ze een interne representatie van informatie (bijvoorbeeld een tabel met treinreisgegevens) kunnen omzetten in een goed geschreven tekst (bijvoorbeeld een gemakkelijk te begrijpen treinreisbeschrijving). Een verwant vakgebied, dat van de *spraaksynthese*, beoogt computers tekst (of een meer abstracte fonologische representatie) om te laten zetten in goed verstaanbare en prettig klinkende spraak. *Spraakgenering* is een uitbreiding op taalgenerering zodanig dat de genenereerde tekst vervolgens zo goed mogelijk wordt uitgesproken. Een spraakgenereringssyteem is daarom te zien als een combinatie van een taalgeneringssyteem en spraaksynthese systeem.

Taal- en spraakgenerering zijn multidisciplinaire vakgebieden, waarin de redenen voor het bouwen van generingssystemen kunnen verschillen. Zo is men vanuit de psycholinguïstiek geïnteresseerd in het modelleren van menselijke taalproductie, terwijl men vanuit de taal- en spraaktechnologie streeft naar praktisch inzetbare systemen. De invalshoek in dit proefschrift is primair taalkundig: het spraakgeneringssysteem wordt gebruikt als een hulpmiddel voor het toetsen en verbeteren van taalkundige beschrijvingen van intonatie.

Bestaande taalkundige modellen voor intonatie zijn doorgaans slechts in beperkte mate getoetst. Het probleem is dat ze kennis vereisen over de syntax (de bouw) en semantiek (de betekenis), en de discourse (de context, inclusief de kennis van de spreker en luisteraar) van de uiting die als invoer dient. Veel van deze informatie is niet automatisch uit een zin of tekst af te leiden, en moet daarom met de hand worden gespecificeerd. Dit is tevens één van de redenen dat taalkundige modellen niet direct geschikt zijn voor toepassing in praktische tekst-naar-spraaksystemen. Echter, binnen een spraakgenereringssysteem kan al de benodigde informatie automatisch worden gegenereerd door de taalgenereringsmodule (mits deze voldoende geavanceerd is). Zodoende kunnen dus grote hoeveelheden betrouwbaar testmateriaal worden geproduceerd. Verder kan de daadwerkelijke implementatie van een taalkundig model verborgen aannames, mogelijke tegenspraken en andere gebreken aan het licht brengen.

Onderzoeksdoel Het centrale onderzoeksdoel in dit proefschrift is het ontwerpen, bouwen en testen van een taalkundig gemotiveerd spraakgenereringsysteem dat zinsaccenten en intonationele grenzen genereert die het meest acceptabel zijn volgens intonationele experts. Hierbij gelden een aantal restricties. Ten eerste betreft het voorgelezen spraak (in tegenstelling tot spontane spraak), en is de doeltaal het Nederlands. Ten tweede is het inherent aan de huidige taalgeneringsystemen dat ze gebonden zijn aan een bepaald domein. Dit onderzoek betreft genereringsystemen voor twee typen tekst: treinsreisbeschrijvingen, en botanische beschrijvingen van bloemen. Ten derde concentreert dit onderzoek zich op het genereren van een fonologische representatie van de intonatie, waarbij wordt geabstraheerd van de fonetische implementatie (hoewel wel degelijk hoorbare spraak wordt gesynthetiseerd). Tenslotte is er om praktische redenen voor gekozen om één bepaalde taalkundige beschrijving van de Nederlandse intonatie te evalueren.

Het onderzoek is gestuurd door enerzijds onderzoeksvragen betreffende de taalkundige theorie:

1. Hoe adequaat is de theorie?

2. Hoe kan deze verder worden verbeterd?

Anderzijds, door onderzoeksvragen betreffende de vereiste architectuur van spraakgeneringssystemen:

1. Wat is de beste manier om taalgenerering en spraaksynthese te combineren?

2. Wat is de beste plaats binnen een spraakgenereringssysteem om de intonatie te beregelen?

3. Welke aanpassingen zijn vereist aan de 'standaard' architecturen voor taalgeneringssystemen en spraaksynthese systemen?

4. Wat voor extra informatie moet worden gegenereerd door het taalgenereringssysteem, en hoe kan dit worden bereikt?

Theoretisch kader Hoofdstuk 2 beschrijft het theoretische kader van dit onderzoek. Allereerst wordt het begrip *prosodie* gedefinieerd en wordt de *prosodische fonologie* geïntroduceerd, een fonologische theorie over de hiërarchische organisatie van spraak. Volgens deze theorie heeft iedere spraakuiting een *prosodische structuur*, die samen met de tonale structuur het prosodische deel vormt van de fonologische representatie. Deze representatie wordt opgebouwd op basis van de semantiek, de syntaxis en de morfologie. Er wordt een kort overzicht gegeven van de welgevormdheidscondities voor de prosodische structuur en de argumenten die het bestaan ervan ondersteunen. Eén van de constituenten is de *fonologische frase*, die een belangrijke rol speelt in de rest van dit proefschrift.

Ten tweede introduceert dit hoofdstuk de *intonationele fonologie*, die de contrastieve intonatiepatronen van de taal definieert, meer in het bijzonder de autosegmentele intonationele fonologie van het Nederlands ontwikkeld door Gussenhoven. In navolging van het werk van Pierrehumbert, kent dit model een *toonstring* en een *segmentstring*, die onderling gesynchroniseerd zijn door middel van *associaties.* De toonstring bestaat uit toonhoogteaccenten (H*L, L*H, of H*LH), grenstonen (%H, %L, H%, or L%), en toonmorfemen zoals *downstep* (!). De segmentstring is bovendien opgedeeld in *associatie domeinen* (AD's), die in de praktijk overeenkomen met intonationele frasen. Er worden een aantal voordelen van dit model aangevoerd met betrekking tot het genereren van intonatie: (1) het biedt een precieze formalisering van accentplaatsing, intonationele frasering, en melodiekeuze; (2) het staat een concentratie op de fonologische beschrijving toe, doordat geabstraheerd wordt van de fonetische implementatie; (2) het voorkomt het genereren van intonatie die onacceptabel is op grond van fonologische onwelgevormdheid.

Ten derde wordt in navolging van Gussenhoven beargumenteerd dat de gangbare opvatting dat de intonationele frase kan worden geanalyseerd als een prosodische constituent tot tegenspraak leidt, en dat een onafhankelijk domein is vereist (het associatiedomein).

Ten vierde wordt in dit hoofdstuk een eenvoudige, regelgebaseerde grammatica voor het plaatsen van zinsaccenten beschreven, die het uitgangspunt vormt voor het verdere werk. De invoer bestaat uit de focusdistributie, de argumentstructuur, de prosodische structuur, en de opeenvolgende woorden. De eerste regel is SAAR (Sentence Accent Assignment Rule), die de focus-accent relatie modelleert, in het bijzonder voor gefocusseerde maar ongeaccentueerde predicaten. De tweede regel is de *ritmeregel*, een fonologische regel die onder bepaalde condities accenten deleert om een meer ritmische verdeling daarvan te bewerkstelligen. Er wordt beargumenteerd dat aangezien het toepassingsdomein van deze regel de fonologische frase is, het voor accentplaatsing van cruciaal belang is dat de fonologische frasering correct verloopt.

Tenslotte beschrijft dit hoofdstuk de minimale aannames over de syntactische structuur van het Nederlands die in dit onderzoek worden gehanteerd.

Fonologische frasering Hoofdstuk 3 betreft algoritmes voor fonologische frasering, die binnen het huidige kader essentieel zijn als toepassingsdomein voor de ritme regel. In het eerste deel van dit hoofdstuk worden twee algoritmes beschreven zoals voorgesteld door respectievelijk Nespor & Vogel en door Hayes, en wordt beargumenteerd dat geen van beide adequaat is voor het Nederlands. Vervolgens wordt een nieuw algoritme voorgesteld dat berust op de notie van *adjacente modificatie*. Naar aanleiding van het testen op een groot aantal verschillende syntactische structuren van het Nederlands volgt een gedeeltelijke herziening van het algoritme. Hoewel verdere generalisatie wenselijk is, voorziet het resultaat in een observationeel adequaat algoritme voor het Nederlands.

Het tweede deel van dit hoofdstuk beschrijft een evaluatie van dit algoritme door te controleren of toepassing van de ritmeregel binnen de voorspelde fonologische frasen tot de gewenste accentuering leidde. Deze evaluatie werd uitgevoerd op symbolisch niveau, met een gedeeltelijk automatische, corpusgebaseerde methode. Het corpus bestond uit 10 korte bloembeschrijvingen (935 woorden), die handmatig werden geparseerd, en door een drietal intonationele experts werden voorzien van de geprefereerde accenten. De voorspellingen van het accentueringsalgoritme, inclusief het fonologische fraseringsalgoritme en de ritmeregel, werden vervolgens vergeleken met de referentie. Voorspelling en referentie kwamen in 95% van alle gevallen overeen. Een analyse van dat deel van de fouten dat mogelijk te wijten zou zijn aan verkeerde toepassing van de ritmeregel, gaf geen directe aanleiding om het algoritme voor fonologische frasering te herzien.

Intonationele frasering Het eerste deel van Hoofdstuk 4 beschrijft een drietal algoritmes voor intonationele frasering zoals voorgesteld door respectievelijk Nespor & Vogel, Caspers, en Gussenhoven. Vervolgens wordt een evaluatie van deze algoritmes beschreven die analoog is aan de evaluatie beschreven in het vorige hoofdstuk, namelijk, op symbolisch niveau, en met behulp van van een gedeeltelijk automatische, corpusgebaseerde methode. Hiervoor werd het eerder gebruikte corpus van bloemenbeschrijvingen door de intonationele experts eveneens voorzien van de meest acceptabele intonationele grenzen. Daardoor konden de voorspellingen van de drie algoritmes worden vergeleken met deze referentie. Er wordt een gedetailleerde analyse van de fouten beschreven, die laat zien dat elk van de drie algoritmes systematische tekortkomingen heeft. Er wordt betoogd dat een beter algoritme de syntactische structuur als invoer zou moeten gebruiken, en daarbij gevoelig zou moeten zijn voor bepaalde syntactische configuraties, de accentdistributie, en het gewicht van de gevormde intonationele frasen.

In het tweede deel wordt een algoritme voorgesteld dat aan deze voorwaarden voldoet. Het bestaat uit acht regels die intonationele grenzen inserteren in bepaalde syntactische configuraties, mits aan een minimale verdeling van de accenten is voldaan. Een evaluatie van dit algoritme op het corpus van bloembeschrijvingen liet zien dat het in 97.66% van alle gevallen een correcte beslissing nam. Een statistische test toonde aan dat het algoritme significant beter was dan tenminste twee van de andere algoritmes.

Spraakgenering Hoofdstuk 5 gaat in op de onderzoeksvragen betreffende de architectuur van een spraakgeneringssysteem. De voorgestelde architectuur is een uitbreiding op de conventionele architectuur voor taalgenereringsystemen bestaande uit een pijplijn van drie modules. Voor wat betreft het combineren van taalgenering met spraaksynthese wordt beargumenteerd dat een verbinding via tekst ontoereikend is. In plaats van tekst wordt een fonologische representatie voorgesteld die bestaat uit de woorden, de intonationele vorm (accenten, grenzen, en melodie), en de hogere prosodische constituenten (waaronder de fonologische frase). Ook wordt betoogd dat prosodische realisatie (dat ondermeer het genereren van de intonationele vorm omvat) niet kan worden gereduceerd tot microplanning of oppervlakte realisatie, maar een complex, en inherent linguïstische proces is, dat een eigen module in het systeem vereist.

Vervolgens wordt een architectuur uitgewerkt bestaande uit twee hoofddelen. Ten eerste een domeinspecifieke tekstplanner die de *inhoud* van de tekst genereert in de vorm van een reeks abstracte specificaties voor syntactische frasen. Ten tweede een domeinonafhankelijk en herbruikbaar *inhoud-naar-spraaksysteem* (CONPAS genaamd) dat verantwoordelijk is voor de *vorm* van de gesproken uitvoer, gebruik makend van morfo-syntactische, fonologische, en fonetische kennis van Nederlands. Dit deel bestaat intern weer uit een drietal modules voor syntactische realisatie (SynR), prosodische realisatie (ProsR), en spraaksynthese (SpeechS). Verder wordt beargumenteerd dat de gebruikte specificaties voor syntactische frasen voorzien in het juiste niveau van abstractie.

De rest van dit hoofdstuk biedt een overzicht over de taken die binnen de verschillende modules worden uitgevoerd. Ten eerste worden de relatief eenvoudige, schemagebaseerde textplanners beschreven, die primair tot doel hebben zoveel mogelijk variatie in tekst te genereren binnen de twee toepassingsdomeinen (treinreisbeschrijvingen en bloembeschrijvingen). Ten tweede worden syntactische en prosodische realisatie kort samengevat; beide processen worden uitgebreid beschreven in de volgende twee hoofdstukken. Tenslotte worden de drie stappen in spraaksynthese beschreven: fonologie (afleiding van de lagere prosodische constituenten, en toepassing van segmentele regels), fonetische implementatie (beregeling van duur, pauze, en F_0), en difoonsynthese.

Syntactische realisatie Hoofdstuk 6 beschrijft SynR, een syntactische realisator voor het Nederlands die abstracte specificaties vertaalt naar grammaticale expressies en hun syntactische structuur, zodat een gebruiker/systeem kan abstraheren van de syntactische details van de doeltaal. Deze uitwerking laat zien dat alle syntactische informatie die nodig is voor het plaatsen van accenten en intonationele grenzen (zoals verondersteld in de eerdere hoofdstukken) inderdaad automatisch kan worden gegenereerd door een taalgeneringsssysteem.

Het systeem bestaat uit drie opeenvolgende componenten. Ten eerste een basis component, die hoofdzakelijk verantwoordelijk is voor het projecteren van de basale syntactische structuur, het toevoegen van puur syntactische informatie, en het uitschrijven van getallen, tijden, etc. Ten tweede een transformationele component die zorgt voor ondermeer de distributie van focus, verschillende vormen van congruentie, en de woordvolgorde. Ten derde een flectie component die inhoudswoorden vervoegt en functiewoorden invoegt. Het systeem voorziet in een onderscheid tussen de declaratieve/linguïstische en de procedurele/computationele aspecten, doordat alle modules zijn geïmplementeerd in speciale grammaticaformalismen (de transformationele grammatica in Gramtsy, en de andere twee als de affix grammatica's in AGFL).

Er wordt betoogd dat de voorgestelde abstracte invoer een compacte specificatie toelaat, dankzij de notationele conventies en de aanname van redelijke defaults. Verder wordt beargumenteerd dat een systeem als SynR een aantal voordelen biedt boven een eenvoudige, templaat-gebaseerde syntactische realisator, ondermeer voor wat betreft prosodische realisatie en microplanning. Tenslotte wordt SynR kort vergeleken met andere systemen, en worden een aantal mogelijke uitbreidingen besproken.

Prosodische realisatie Hoofdstuk 8 beschrijft ProsR, de module voor de prosodische realisatie van uitingen op basis van hun syntactische structuur, focusdistributie, argumentstructuur, en woordenreeks. Dit proces omvat een viertal taken: (1) het afleiden van de hogere prosodische constituenten (waaronder de fonologische frase); (2) het plaatsen van accenten; (3) het plaatsen van intonationele grenzen; (4) het kiezen van de melodie. Aangezien de eerste en derde taak reeds zijn behandeld in respectievelijk hoofdstuk 3 en 4, wordt in dit hoofdstuk hoofdzakelijk aandacht besteed aan accentplaatsing.

Op basis van de eenvoudige grammatica uit hoofdstuk 2 wordt een meer adequate methode voor accentplaatsing voorgesteld. Allereerst worden de details van de implementatie van SAAR uitgewerkt, ondermeer met betrekking tot scheidbare werkwoorden, en meer in het algemeen, samenstellingen die meer dan één accent kunnen dragen. Verder worden een aantal uitzonderingen op SAAR besproken. Er wordt betoogd dat focus projectieregels noodzakelijk zijn om de relatie tussen focus op woordniveau en op constituentniveau te definiëren. Ook wordt een aantal resterende problemen bediscussieerd, waaronder gevallen van interactie tussen SAAR en intonationele frasering.

Vervolgens worden een aantal extra regels geïntroduceerd en gemotiveerd. Ten eerste een regel voor het deaccentueren van naamwoordgroepen die fungeren als voorbepalingen bij een zelfstandig naamwoord. Ten tweede een lexicale ritmeregel voor een beperkt aantal lexicaal gespecificeerde quantificeerders en bijwoorden die ritmisch gezien een uitzondering vormen. Ten derde een speciale ritme regel voor eenheden (waaronder *honderd* en *duizend*), die in combinatie met de juiste fonologische frasen de accentuering van getallen correct voorspelt.

Het laatste deel van dit hoofdstuk presenteert een eenvoudig algoritme voor melodiekeuze, waarin ondermeer een alternatieve vorm van downstep wordt gesignaleerd.

Evaluatie Hoofdstuk 8 beschrijft een evaluatie-experiment waarbij de accenten en intonationele grenzen geproduceerd door het generingsysteem werden vergeleken, op symbolische niveau, met die voorgeschreven door een viertal intonationele experts. Tevens vond een vergelijking plaats met een representatief tekst-naar-spraaksysteem. Het materiaal bestond uit acht reisbeschrijvingen (643 woorden) en 60 expressies met getallen (274 woorden). De resultaten lieten zien dat voor de reisbeschrijvingen de accentuering van het systeem in 93% van alle gevallen overeenkwam met die van de experts, terwijl dit voor de getalsexpressies 95% bedroeg. Voor wat betreft de intonationele frasering werd in beide domeinen een overeenstemming van 95% gevonden.

Vervolgens worden deze resulaten bediscussieerd. Er wordt een hiërarchische clusteranalyse gepresenteerd die inzicht verschaft in de onderlinge verschillen tussen de systemen en de experts. Verder worden waar mogelijk systematische fouten geanalyseerd.

Tenslotte worden de resultaten vergeleken met die voor andere systemen. Hier worden ondermeer metrieken voor het berekenen van de score bediscussieerd. Er wordt geconcludeerd dat de accentuering van het ontwikkelde spraakgeneringssysteem beter is dan dat van de meeste tekst-naar-spraaksystemen, en vergelijkbaar met die van andere spraakgenereringssystemen, terwijl de frasering beter is dan die van andere systemen in termen van overeenstemming, en vergelijkbaar in termen van andere metrieken.

Algemene discussie Hoofdstuk 9 vat het proefschrift samen aan de hand van de antwoorden op de onderzoeksvragen betreffende de architectuur van spraakgeneringssystemen en de linguïstische theorie over intonatie. Ter afsluiting wordt kort aandacht besteed aan de praktische implicaties.

Curriculum Vitae

Erwin Marsi was born in Deventer on November 27 1968. He attended the Eemland College in Amersfoort, where he obtained his VWO diploma in 1986. He started his study at Nijmegen University with Philosophy, but changed after one year to Language, Speech and Computer Science. He enjoyed additional courses in Cognitive Science, Philosophy, Computer Science, and Mathematics, and graduated cum laude in 1993. From 1994 to 1998, he was employed as a PhD student (AIO) at the Department of Language and Speech, Nijmegen University. While continuing work on his thesis, he was part-time employed at Nijmegen University for teaching a course on Natural Language Generation. He recently joined the Department of Computational Linguistics and AI at Tilburg University as a postdoc researcher, and collaborates in a project about pitch accent placement and prosodic phrasing using techniques from machine learning and information retrieval.