The Omphalos Context-Free Grammar Learning Competition

Bradford Starkie[1], François Coste[2], and Menno van Zaanen[3]

[1] Telstra Research Laboratories**,
770 Blackburn Rd Clayton, Melbourne, Victoria, Australia 3127,
Brad.Starkie@telstra.com.au,
http://www.cs.newcastle.edu.au/~bstarkie
[2] IRISA,
Campus de Beaulieu, 35042 Rennes, France,
francois.coste@irisa.fr,
http://www.irisa.fr/symbiose/people/coste
[3] Tilburg University,
Postbus 90153, 5000LE, Tilburg, the Netherlands,
mvzaanen@uvt.nl,
http://ilk.uvt.nl/~mvzaanen

Abstract. This paper describes the Omphalos Context-Free Grammar Learning Competition held as part of the International Colloquium on Grammatical Inference 2004. The competition was created in an effort to promote the development of new and better grammatical inference algorithms for context-free languages, to provide a forum for the comparison of different grammatical inference algorithms and to gain insight into the current state-of-the-art of context-free grammatical inference algorithms. This paper discusses design issues and decisions made when creating the competition. It also includes a new measure of the complexity of inferring context-free grammars, used to rank the competition problems.

1 Introduction

Omphalos is a context-free language learning competition held in conjunction with the International Colloquium on Grammatical Inference 2004.[4] The aims of the competition are:

- to promote the development of new and better grammatical inference (GI) algorithms,
- to provide a forum in which the performance of different grammatical inference algorithms can be compared on a given task, and
- to provide an indicative measure of the complexity of grammatical inference problems that can be solved with the current state-of-the-art techniques.

** Opinions expressed in this paper do not necessarily represent the views of Telstra Corporation Limited (ACN 051 775 556, ABN 33 051 775 556).

[4] The competition data will continue to be available after the completion of the competition and can be accessed via http://www.irisa.fr/Omphalos/.

2 The Competition Task

The competition task was to infer a model of a context-free language from unstructured examples. During the development of the competition, two main design issues needed to be resolved as follows:

Method of evaluation A method of determining the winner of the competition needed to be decided upon. For instance entries could be judged by measuring the difference between the inferred language and the target language, or, alternatively, entries could be judged by measuring the difference between the derivation trees assigned to sentences by the inferred grammar and the derivation trees assigned to sentences by the target grammar.

Complexity of tasks As one of the goals of the competition was to determine the state-of-the-art in grammatical inference, the competition tasks should be selected so as to be neither too simple, nor too difficult to solve. That is the complexity of the learning task should be quantifiable.

Both of these issues will be discussed in the following subsections below.

2.1 Method of Evaluation

The evaluation approach selected to be used by the competition should be automatic, objective and easy. Van Zaanen et al. [1] describe several approaches to the evaluation of grammatical inference systems. These include the following:

Rebuilding known grammars. Using a pre-defined grammar, unstructured data is generated. Based on this data, the GI system tries to induce the original grammar. The problem here is that for most languages, there is more than one grammar that can be used to describe that language. Although all regular grammars can be transformed into a canonical form, no such canonical form exists for context-free grammars. Therefore there is no automated way to determine that the language described by a grammar submitted by a competitor and the target grammar are the same.

Comparison of labelled test data with treebank. Plain data is extracted from a set of structured sentences. A GI system must now find the original structure using the unstructured data. A distance measure is then used to rank the similarity between the derivation trees assigned to test sentences by the inferred grammar and derivation trees assigned to test sentences by the target grammar. Once again because for most languages there is more than one grammar describing that language a grammar submitted by a competitor may describe the target language exactly, but would rank poorly according to the distance measure. In addition the approach is problematic when ambiguity occurs in either the target or inferred grammar as multiple solutions are then possible.

Classification of unseen examples. The GI systems receive unstructured (positive or positive and negative) training data. This training data is generated according to an underlying grammar. Next, test data is provided and the system designed by competitors should assign language membership information to this test data. In other words, the system must say for each sentence if it is contained in the language described by the underlying grammar. The main disadvantage of this evaluation approach is that since this task is a classification task, no real grammatical information has to be learned as such.

Precision and Recall. Each GI system receives unstructured training data generated according to a target grammar. From this training data a grammar is inferred. Next, previously unseen sentences that are in the target language are provided to the system. The percentage of these sentences that are in the inferred grammar is measured. This measure is known as the recall. Next sentences are generated according to the inferred grammar. The number of these sentences that exist in the target language is then measured. This measure is known as the precision. Recall and precision can then be merged into a single measure known as the F-score. The F-Score is a measure of the similarity between the target and inferred languages. A problem with this technique is that once a test sentence has been used to measure recall it cannot be used to measure recall a second time. This because that test sentence can also be used as an additional training example. In addition this technique requires that all grammatical inference systems be capable of generating example sentences.

All these evaluation methods have problems when applied to a generic GI competition. It was decided however that for the Omphalos competition, the technique of "classification of unseen examples" would be used to identify the winner of the competition. This was the same evaluation method that was used for the Abbadingo [2] DFA (regular language) learning competition. For Omphalos however competitors needed to tag the test sentences with 100% accuracy compared with 99% accuracy for the Abbadingo competition. This stricter requirement was used in an effort to encourage the development of new truly context-free learning algorithms. The main benefit of this technique is that it places few restrictions upon the techniques used to classify the data. In addition if the inferred grammar describes the target language exactly it will classify the test examples exactly. The main disadvantage of this technique is that it is possible for a classifier to classify all of the test examples exactly without the classifier having an accurate model of the target language. One way to overcome this problem is to only consider the problem to be solved when the precision of the inferred grammar is greater than a threshold value. We have decided not to implement this additional constraint, since it is believed that if the test sets contain negative examples that are sufficiently close to positive examples, then classification accuracy is a suitable measure of how close the inferred grammar is to the target grammar.

2.2 Complexity of the Competition Tasks

The target grammars, training and testing sentences were created with the following objectives in mind:

Requirement 1. The learning task should be sufficiently difficult. Specifically the task should be just outside of the current state-of-the-art, but not so difficult that it is unlikely that a winner will be found.

Requirement 2. It should be provable that the training sentences are sufficient to identify the target language.

From [3] it is known that it is impossible to learn a context-free grammar from positive examples only without reference to a statistical distribution, however:

- It is possible to learn a context-free grammar if both positive and negative examples are available and,
- If sufficient additional prior knowledge is known, such as a statistical distribution, it is possible to learn a context-free grammar with positive data only.

Therefore it was decided to include problems that included learning grammars from positive examples only as well as learning grammars from positive as well as negative examples.

To determine whether or not Requirement 1 was met, a measure of the complexity of the learning task was derived. This measure was derived by creating a model of the learning task based upon a brute force search. To do this a hypothetical grammatical inference algorithm called the *BruteForceLearner* algorithm was created. This model was also used to determine if the training data was sufficient to identify the target grammar. The details of the algorithm, the proof that it can identify any context-free grammar in the limit from positive and negative data, as well as the definition of the complexity measure itself can be found in [4].

The summary of these proofs is as follows:

- For each context-free grammar G there exists a set of positive examples $O(G)$ such that when $O(G)$ is presented to the *BruteForceLearner* algorithm, the *BruteForceLearner* algorithm constructs a set of grammars X such that there exists a grammar G_2 in X with the property that $L(G) = L(G_2)$. We call this set the characteristic set, and use the notation $O(G)$ to define the characteristic set of G.
- Given G there exists a simple technique to construct $O(G)$. This technique involves generating a sentence s for each rule P in G, such that all derivations of s are derived using P. This technique was used in the Omphalos competition to construct some of the training examples.
- When presented with the training examples $O(G)$ the *BruteForceLearner* need only construct a finite number of candidate grammars. Equation 1 described below defines the number of candidate grammars that could be constructed that would be sufficient to include the target language.

$$\#\text{candidate grammars} = 2^{(((\sum_j (2|O_j|-2))+1)^3+((\sum_j (2|O_j|-2)+1)T(O)))} \quad (1)$$

Where G is any context-free grammar, O is a set of positive examples in a characteristic set of G,and $T(O)$ is the number of terminals in O.

- Given G and $O(G)$, there exists an additional set of positive and negative examples $O_2(G)$ such that when $O_2(G)$ is presented to the *BruteForceLearner* algorithm after $O(G)$, the *BruteForceLearner* algorithm identifies the target language exactly.

The following technique can be used to construct $O_2(G)$;

- Given O(G) construct the set of hypothesis grammars H that is sufficiently large to ensure that it contains the target grammar.
- For each $H_i \in H$ such that $L(O(H_i)) \subset L(G)$ add one sentence, marked as a positive example, to $O_2(G)$ that is an element of G but not an element of H_i.
- For all other $H_i \in H$ where $L(O(H_i)) \neq L(G)$ add one sentence, marked as a negative example, to $O_2(G)$ that is an element of H_i but not an element of G.

Note that this is the only known technique for constructing the set $O_2(G)$. The number of possible grammars given $O(G)$ is described by Equation 1 which is not polynomial. Therefore the construction of the negative data using this technique is not computable in polynomial time using this technique.

3 Creation of the Example Problems

As a result of the proofs contained in [4] and summarized in the previous section, Equation 1 was considered to be a suitable measure of the complexity of the learning tasks of the Omphalos competition. This is because it defines a hypothesis space used by at least one algorithm that is guaranteed to identify any context-free language in the limit using positive and negative data. Equation 1 was also used to benchmark the target problems against other grammatical inference problems that were known to be solved using other algorithms. In addition the proofs contained in [4] showed that for all grammars, other than those that could generate all strings of a given alphabet, the *BruteForceLearner* algorithm required negative data to ensure that it uniquely identified any context-free language. As described in the previous section, the only known way that could be used to construct sufficient negative data to ensure that at least one known algorithm could identify the language exactly from positive data was not computable in polynomial time. Therefore if a target grammar was chosen that was small enough to enable sufficient negative training examples to be calculated, then the learning task would become too simple. Therefore no such set of negative data was calculated, and it is not known if for any of the competition problems the training examples are sufficient to uniquely identify the target language.

The following technique was used to construct the training sets for the Omphalos competition:

- For each target grammar a set of positive sentences were constructed, such that for every rule in the target grammar, a positive sentence was added to the training set that is derived using that rule.

– A set of positive examples were then randomly created from the target grammar of length up to five symbols longer than the longest sentence in the characteristic.
– A set of negative sentences was then created for each target grammar. For problems 1 to 6 these were constructed by randomly creating strings up to the maximum length using the symbols of the grammar. For problems 6.1 to 10 the negative examples were created from "surrogate" grammars such as regular approximations to the target languages.

The number of training examples was selected to be between 10 and 20 times as large as the characteristic set.

3.1 Creation of the Target Grammars

Firstly the literature was reviewed to identify pre-existing benchmark grammatical inference problems. The work in [5] and [6] identified some benchmark problems, i.e. grammars that can be used as some sort of standard to test the effectiveness of a GI system by trying to learn these grammars. The grammars were taken from [7] and [8]. Using Equation 1 the complexities of these grammars were calculated. A description of these grammars and their complexity measures are listed in Table 1.

Using the results of Table 1 the complexities of the target grammars of the competition problems were selected. The grammars were then created as follows:

1. The number of non-terminals, terminals and rules were selected to be greater than in grammars shown in Table 1.
2. A set of terminals and non-terminals were created. Rules were then created by randomly selecting terminals and non-terminals. A fixed number of rules were created to contain only terminal strings.
3. Useless rules were then identified. If a non-terminal could not generate a terminal string, a terminal rule was added to it. If a non-terminal was not reachable from the start symbol, rules were added to ensure the rule was reachable from the start symbol. For instance if the non-terminal N was unreachable from the start symbol, a rule was created with the start symbol on the left hand side of the rule, and N on the right hand side of the rule.
4. Additional rules were added to ensure that the grammar did not represent a regular language. Specifically rules containing center recursion were added.
5. A characteristic set of sentences was generated for the grammar. If the complexity of the grammar was not in the desired range, then the grammar was deleted.

Using this technique six grammars were created as listed in Table 2. Tests were undertaken to ensure that grammars 1–4 represented deterministic languages. Specifically $LR(1)$ parse tables were constructed from the grammars using bison. To ensure that grammars 4 and 5 represented non-deterministic languages, rules were added to the target grammars. It should be noted that

Table 1. Complexity of Benchmark Inference Problems from [5] and [6].

	Description	Example phrase	$\log_2$ compl.	Properties
1	(aa)$*$	aa, aaaa, aaaaaa	1.34×10^3	Regular
2	(ab)$*$	ab, abab, ababab	2.22×10^3	Regular
3	Operator precedence (small)	(a+(b+a))	9.37×10^3	Not regular
4	Parentheses	(), (()), ()(), ()(())	2.22×10^3	Not regular
5	English verb agreement (small)	that is a woman, i am there	5.33×10^5	Finite
6	English lzero grammar	a circle touches a square, a square is below a triangle	4.17×10^6	Finite
7	English with clauses (small)	a cat saw a mouse that saw a cat	6.59×10^5	Not regular
8	English conjugations (small)	the big old cat heard the mouse	9.13×10^5	Regular
9	Regular expressions	ab$*$(a)$*$	9.39×10^3	Not regular
10	$\{\omega = \omega^{\mathbb{R}}, \omega \in \{a,b\}\{a,b\}+\}$	aaa, ba	6.90×10^4	Not regular
11	Number of a's=number of b's	aabbaa	4.29×10^4	Not regular
12	Number of a's=2$\times$number of b's	aab, babaaa	3.01×10^5	Not regular
13	$\{\omega\omega \mid \omega \in \{a,b\}+\}$	aba, aa	9.12×10^4	Regular
14	Palindrome with end delimiter	aabb$, ab$, baab$	1.18×10^5	Not regular
15	Palindrome with center mark	aca, abcba	4.96×10^3	Not regular
16	Even length palindrome	aa, abba	9.30×10^3	Not regular
17	Shape grammar	da, bada	2.45×10^4	Not regular

these grammars are complex enough that they cannot be learned using a brute force technique in time to be entered into the Omphalos competition. Having said that, even the smallest of grammars could not be inferred using the *BruteForceLearner*. After problem 6 was solved, problems 7 to 10 were added to the competition. Problems 6.1 to 6.6 were added some time later.

Table 2. Complexities of Benchmark Inference Problems in Omphalos Competition.

	Training data	Properties	$\log_2$ compl.
1	Positive and negative	Not regular, deterministic	1.10×10^9
2	Positive only	Not regular, deterministic	7.12×10^8
3	Positive and negative	Not regular, deterministic	1.65×10^{10}
4	Positive only	Not regular, deterministic	1.13×10^{10}
5	Positive and negative	Not regular, non-deterministic	5.46×10^{10}
6	Positive only	Not regular, non-deterministic	6.55×10^{10}
6.1	Positive and negative	Not regular, deterministic	1.10×10^9
6.2	Positive only	Not regular, deterministic	7.12×10^8
6.3	Positive and negative	Not regular, deterministic	1.65×10^{10}
6.4	Positive only	Not regular, deterministic	1.13×10^{10}
6.5	Positive and negative	Not regular, non-deterministic	5.46×10^{10}
6.6	Positive only	Not regular, non-deterministic	6.55×10^{10}
7	Positive and negative	Not regular, deterministic	5.88×10^{11}
8	Positive only	Not regular, deterministic	1.63×10^{11}
9	Positive and negative	Not regular, non-deterministic	1.08×10^{12}
10	Positive only	Not regular, non-deterministic	9.92×10^{11}

The grammars listed in Table 2 represent three axes of difficulty in grammatical inference. Specifically:

1. The complexity of the underlying grammar,
2. whether or not negative data is available and,
3. how similar the negative examples in the test set are to positive examples of the language. For instance whether or not the test set includes sentences that can be generated by regular approximations to the target language but not the target language itself.

The competition adopted a linear ordering for the benchmark problems based upon these axes. Correctly labelling a test set in which the negative sentences closely resembled the positive sentences was ranked higher than correctly labelling a test set where the negative examples differed greatly from the positive examples. For instance, problem 6.1 is ranked higher than problem 1 and even problem 6. Similarly, solving a problem with a higher complexity measure was ranked higher than solving one with a lower complexity measure. For instance, problem 3 is ranked higher than problem 1. Solving a problem without using negative data was considered to be a more difficult problem than when negative data was used. For instance, problem 2 is ranked higher than problem 1. In addition it has been noted by [6] that the inference of non-deterministic languages is a more difficult task than the inference of deterministic languages. Therefore solving those problems that involved non-deterministic languages was ranked higher than solving those problems that involved deterministic languages.

3.2 Construction of training and testing sets.

Once the target grammars were constructed, characteristic sets were constructed for each grammar. Sets of positive examples were then created using the GenR-GenS software [9].

For the first six problems additional examples were then created by randomly generating sentences of length up to five symbols more than the length of the longest sentence in the characteristic set. These sentences were then parsed using the target grammar and were labeled as being either in or out of the target language. This set of sentences was then randomized and split into testing and training sets, but in such a way as to ensure that the training set contained a characteristic set. For those problems that were to be learned from positive data only the training sets had all negative examples removed.

For problems 6.1 to 10 a more rigorous method of constructing negative data was used as follows:

- For each context-free grammar an equivalent regular grammar was constructed using the superset approximation method based on Recursive Tree Network (RTN) described in [10]. Sentences that could be generated from this regular approximation to the target language were included as negative data. These sentences were included to distinguish between competitors who had created regular approximations to the underlying context-free languages, and competitors who had identified a non-regular language.
- A context-free grammar that was larger than the target language was constructed by treating the target grammar as a string rewriting system, and normalizing the right hand sides of rules using the normalization algorithm described in [11]. That is, a context-free grammar was constructed that described a language that was a superset of the target language, but in which the right hand side of each rule could not be parsed by the right hand side of any other rule. Negative examples were then created from this approximation to the target language.
- Each target grammar in the competition included some deliberate constructs designed to trick grammatical inference algorithms. For instance most included sequences that where identical to center recursion expanded to a finite depth. An example is A^nB^n where $n < m$, m is an integer > 1. To ensure that the training and testing examples tested the ability of the inferred grammars to capture these nuances, the target grammars were copied and hand modified changing the A^nB^n where $n < m$ to become A^nB^n where $n > 1$. In addition, where center recursion existed of the form A^nB^n in the target grammar the regular approximations A^*B^* were included in the

"tricky" approximation to the target grammar. Negative examples where then created from these approximations to the target grammar and added to the test set.
– There were an equal number of positive and negative examples in the test sets.

In addition for problems 6.1 to 10:

– The longest training example was shorter than the longest test example.
– The grammar rules for problems 7 to 10 were shorter than for problems 1 to 6.6 and had more recursion. Some non-$LL(1)$ constructs were also added to the target grammars for problems 7 to 10.

4 Preliminary Results

The timetable of the competition was constructed such that the competition ended two weeks prior to the ICGI 2004 conference in which this paper appears. Due to the deadlines involved in publishing the proceedings the results of the competition cannot be contained within this paper. The following table includes some important dates on the time line of the competition.

Table 3. Competition time line.

Date	Event
February 15^{th} 2004	Competition begins
February 20^{th} 2004	Problem 1 solved by Joan Andreu Sánchez
March 22^{nd} 2004	Problems 3, 4, 5 and 6 were solved by Erik Tjong Kim Sang
April 14^{th} 2004	Problems 7, 8, 9 and 10 were added
June 7^{th} 2004	New larger testing sets were added for problems 1 to 6
October 1^{st} 2004	Competition closed
October 11^{th} 2004	Competition winner announced
October 11^{th}–13^{th} 2004	Omphalos session at ICGI-2004

4.1 Problem 1

Problem 1 was solved by Joan Andreu Sánchez from the Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València. Although Sánchez originally tried to solve the problem using the Inside-Outside algorithm, he actually solved it manually. After discovering the regularities in the positive examples he used a regular expression constructed by hand to classify the test examples as being either positive or negative. Although the target grammar was not a regular language the test sets did not include a single non-regular example.

This in addition to the speed in which the problem was solved suggests that the first task was overly simple, and the negative examples were too different from the positive examples to be an accurate test of whether or not the language had been successfully learned.

4.2 Problems 3, 4, 5, and 6

Problems 3, 4, 5, and 6 were solved by Erik Tjong Kim Sang from the CNTS - Language Technology Group at the University of Antwerp in Belgium. Tjong Kim Sang used a pattern matching system that classifies strings based on n-grams of characters that appear either only in the positive examples or only in the negative examples. With the exception of problem 1 this technique was not sufficient to solve the problem, so Tjong Kim Sang generated his own negative data, using the principle that the majority of randomly generated strings would not be contained within the language. His software behaved as follows;

1. Firstly it loaded in positive examples from the training file.
2. It then generated an equal number of unseen random strings, and added these to the training data as negative examples.
3. A n-gram classifier was then created as follows: A count was made of n-grams of length 2 to 10 that appeared uniquely in the positive examples or uniquely in the negative examples. A weighted (frequency) count of such n-grams in the test strings was then made. For each sentence in the test set. If the positive count was larger than the negative count then the string was classified as positive, otherwise it was classified as negative. If a string contained two zero counts then that sentence was classified as unknown.
4. Steps 2 and 3 were repeated thirty times.
5. Strings that were always classified as positive in all thirty tests were then assumed to be positive examples. Strings that were classified as negative one or more time were classified as negative. Other strings were classified as unknown.

The techniques used by Sánchez and Tjong Kim Sang suggested that more effort was required to generate negative data, to ensure that the testing sets were accurate indications of whether or not the competitor had successfully constructed a context-free grammar that was close to the exact solution. In particular the testing sets needed to include negative sentences that were in regular approximations of the target language, but not in the target language itself. As result, additional problems were added to the competition on April 14^{th}. In addition, on June 7^{th} additional test sets for problems 1 to 6 were added to the competition. Because the correct classification of these test sets was a more difficult task than the correct classification of the earlier test sets, these test sets became problems 6.1 to 6.6.

5 Conclusions

In conclusion, at the time of the writing of this paper the competition is yet to achieve the goal of encouraging the development of new grammatical inference algorithms that can infer truly context-free languages. We believe there are two reasons for this; Firstly, generic machine learning classifiers have been used to solve the "easy" problems, so GI researchers do not attempt to re-solve these. Secondly, the Omphalos problems were designed to be just out of reach of the current state-of-the-art. Since the data-sets will stay available for some time, we expect these problems to be solved in the near future. The goal of providing an indicative measure of the complexity of grammatical inference problems that can be solved using current state of the art techniques has however been partially achieved. An equation (Equation 1) has been developed that defines the size of a set of context-free grammars that can be constructed from a set of training sentences, such that the target language is guaranteed to be contained in this set of context-free grammars.

References

1. Menno van Zaanen, Andrew Roberts, and Eric Atwell. A multilingual parallel parsed corpus as gold standard for grammatical inference evaluation. In Lambros Kranias, Nicoletta Calzolari, Gregor Thurmair, Yorick Wilks, Eduard Hovy, Gudrun Magnusdottir, Anna Samiotou, and Khalid Choukri, editors, *Proceedings of the Workshop: The Amazing Utility of Parallel and Comparable Corpora; Lisbon, Portugal*, pages 58–61, May 2004.
2. Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo One DFA learning competition and a new evidence-driven state merging algorithm. In V. Honavar and G. Slutzki, editors, *Proceedings of the Fourth International Conference on Grammar Inference*, volume 1433 of *Lecture Notes in AI*, pages 1–12, Berlin Heidelberg, Germany, 1994. Springer-Verlag.
3. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
4. B. Starkie. A complexity measure for the grammatical inference of context-free languages. Technical Report RLR8561, Telstra Research Laboratories, Clayton, Australia, 2004.
5. A. Stolcke. Boogie. `ftp://ftp.icsi.berkeley.edu/pub/ai/stolcke/software/boogie.shar.Z`, 2003.
6. K. Nakamura and M. Matsumoto. Incremental learning of context-free grammars. In Pieter Adriaans, Henning Fernau, and Menno van Zaanen, editors, *Grammatical Inference: Algorithms and Applications (ICGI); Amsterdam, the Netherlands*, volume 2482 of *Lecture Notes in AI*, pages 174–184, Berlin Heidelberg, Germany, September 23–25 2002. Springer-Verlag.
7. Craig M. Cook, Azriel Rosenfeld, and Alan R. Aronson. Grammatical inference by hill climbing. *Informational Sciences*, 10:59–80, 1976.
8. J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley Publishing Company, Reading:MA, USA, 2001.
9. Alain Denise, Frédéric Sarron, and Yann Ponty. GenRGenS. URL:`http://www.lri.fr/~denise/GenRGenS/`, 2001.
10. Mark-Jan Nederhof. Practical experiments with regular approximation of context-free languages. *Computational Linguistics*, 26(1):17–44, 2000.
11. F. Otto. On deciding the confluence of a finite string-rewriting system on a given congruence class. *Journal of Computer and System Sciences*, 35:285–310, 1987.