Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Regulated variants of limited context restarting automata

Friedrich Otto [a,*], František Mráz [b,1]

[a] Fachbereich Elektrotechnik/Informatik, Universität Kassel, 34109 Kassel, Germany
[b] Charles University, Faculty of Mathematics and Physics, Department of Computer Science, Malostranské nám. 25, 118 00 Praha 1, Czech Republic

ARTICLE INFO

Article history:
Received 31 July 2016
Received in revised form 21 November 2016
Accepted 7 December 2016
Available online xxxx

Keywords:
Limited context restarting automaton
Regulated rewriting
Language class

ABSTRACT

In the literature various types of restarting automata have been studied that are based on contextual rewriting. A word w is accepted by such an automaton if, starting from the initial configuration that corresponds to input w, the tape contents is reduced to a particular word within a finite number of applications of these contextual rewritings. Here we extend the limited context restarting automata by using additional global means to structure the reductions they execute. In fact, we study regulated lc-R-automata, for which a regular control language is used to restrict the set of admissible reduction sequences, and we propose random context conditions to restrict the place at which a transition of an lc-R-automaton can be applied.

1. Introduction

Restarting automata have been introduced in the 1990s to model the linguistic technique of *analysis by reduction* [1]. Since then many different types of restarting automata have been defined and studied intensively, see for example [2]. The deterministic context-free languages, the context-free languages, the Church–Rosser languages [3], and the growing context-sensitive languages [4] have all been characterized by certain types of restarting automata. Further, it has been shown that some languages, like the marked copy language $L_{\text{copy}} = \{ w\#w \mid w \in \{a,b\}^* \}$ and the Gladkij language $L_{\text{Glad}} = \{ wcw^Rcw \mid w \in \{a,b\}^* \}$, which are not even growing context-sensitive [5], are accepted by certain types of restarting automata. In fact, a restarting automaton does not only determine a language (the language of input words that it accepts), but in addition, a restarting automaton (analysis by reduction) enables nice error localization in rejected words/sentences (see, e.g. [6]). However, only few tools have been developed that support the design of restarting automata.

Several attempts for learning restarting automata by genetic algorithms have been made [7,8], but the results are far from being applicable. Another method based on the concept of identification in the limit from positive data was proposed in [9]. This method uses positive samples of simplifications (reductions) and positive samples of so-called simple words (sentences) of the language to be learnt. It allows to learn the subclass of *strictly locally testable* restarting automata. Their definition as well as the protocol for learning them is based on the notion of strictly locally testable languages [10,11]. As it turned out the strictly locally testable restarting automata are quite expressive as they accept a proper superclass of the growing context-sensitive languages.

* Corresponding author.
E-mail addresses: otto@theory.informatik.uni-kassel.de (F. Otto), frantisek.mraz@mff.cuni.cz (F. Mráz).
1 The second author was supported by the Czech Science Foundation under the project 15-04960S.

http://dx.doi.org/10.1016/j.tcs.2016.12.010

In [12,13] an even simpler version of the restarting automaton was introduced: the so-called *clearing restarting automaton*. While in general a restarting automaton scans its tape from left to right until it detects a position at which a rewrite operation applies, the rewriting done by a clearing restarting automaton only depends on a fixed context around the subword to be rewritten. In fact, a clearing restarting automaton can only delete symbols. For these automata a simple learning algorithm exists, but not surprisingly, clearing restarting automata are quite limited in their expressive power. They accept all regular languages and even some languages that are not context-free, but they do not even accept all context-free languages. Accordingly, they were extended to the so-called Δ-*clearing restarting automata* and the Δ^*-*clearing restarting automata* that can use a marker symbol Δ in their rewrite operations. As it turned out these types of restarting automata can accept all context-free languages [14].

Finally in [15], *limited context restarting automata* were introduced, which can be seen as an extension of the clearing restarting automaton. They also apply rewrite steps just based on context information, but their rewrite rules are more general. In fact, the most general of these automata accept exactly the growing context-sensitive languages. In [15], a special version of a genetic algorithm is proposed to learn these automata from positive and negative samples, and in [16,17], the various types of limited context restarting automata and their confluent versions were put in relation to some McNaughton families of languages [18].

A limited context restarting automaton is *stateless*, that is, it does not use states. In fact, each of its rewrite rules is of the form $(\ell \mid x \to y \mid r)$, which is applied to a tape content (or sentential form) u by replacing a factor of the form ℓxr of u by the word ℓyr, that is, x can be rewritten into y if it is surrounded by the context (ℓ, r). This replacement can occur anywhere within the word u, that is, if u contains more than one occurrence of the factor ℓxr, then an arbitrary one of them is rewritten. Thus, we see that this rewrite process is highly nondeterministic, similar to the derivation process induced by a phrase structure grammar (see, e.g., [19]).

In order to get some additional control over the derivation process, various forms of *regulated rewriting* have been studied in formal language theory (see, e.g., [20]). Here we carry two of these control mechanisms over to limited context restarting automata. First, we study *regulated limited context restarting automata*, for which the sequence of reductions is governed by a given regular language of labels. To each rewrite rule a letter from a new alphabet is attached as a label, and it is then required that the sequence of labels that corresponds to a sequence of rule applications is an element of a given *control language*. Then we consider *random context limited context restarting automata*, where we associate a set of *permitting letters* and a set of *forbidding letters* with each rewrite rule. Now a rule can be applied to a word u only if all permitting letters of that rule occur in u, and none of the forbidding letters occurs in u.

In addition to the types of limited context restarting automata considered in [15–17], we introduce a new type (called $\widehat{\mathcal{R}}_1$) that is motivated by the ordered restarting automaton considered in [21–24]. For this particular type we also study random context conditions that are only one-sided and that allow only permitting or only forbidding letters.

Our main results show the following. For the most restricted types of limited context restarting automata (called types $\mathcal{R}_3$ and $\mathcal{R}'_3$, see Section 3 for the definitions), which are known to just accept the regular languages, it turns out that the regulated as well as the random context variants still accept only regular languages. The reason for this is the fact that a finite-state acceptor that simulates the reduction process of a given limited context restarting automaton of one of these types can additionally also check the condition expressed by a regular control language or random context conditions. On the other hand, for all other types of random context restarting automata, it turns out that the additional control on the reduction process that is provided by a regular control language or random context conditions suffices to design automata that accept a variant of the marked copy language. This means that their expressive power reaches beyond the class of growing context-sensitive languages. The reason for this is the fact that a regular control language as well as random context conditions can be used to enforce reduction sequences that process the two $\{a, b\}$-factors of a given input $w\#w'$ ($w, w' \in \{a, b\}^*$) in an alternating way, in this way making it possible to compare w and w' to each other. Finally, we show that the limited context restarting automata of type $\widehat{\mathcal{R}}_1$ characterize the regular languages just as the stateless ordered restarting automata do. However, their regulated extension as well as the extension by random context conditions also accept a variant of the marked copy language, thus increasing their expressive power beyond the growing context-sensitive languages. This is even true if we restrict the random context conditions to permitting (or forbidding) left random context conditions.

The paper is structured as follows. In Section 2, we present the basic model of the restarting automaton, describe the way in which it works, and list some fundamental results on the expressive power of some of its variants. In Section 3, we restate the definition of the limited context restarting automaton and its various types and we describe in short their expressive power. Also we introduce the limited context restarting automaton of type $\widehat{\mathcal{R}}_1$. In Section 4, we study regulated limited context restarting automata, for which a regular control language is used to restrict the set of admissible reduction sequences, and in Section 5, we study limited context restarting automata with random (two-sided) context conditions. Finally, in Section 6 we consider limited context restarting automata of type $\widehat{\mathcal{R}}_1$ with only one-sided random context conditions. The paper closes with a short summary. Throughout Sections 4 to 6 we state a number of open problems on limited context restarting automata with control that can be seen as starting points for further studies.

2. A short introduction to restarting automata

For a finite alphabet Σ, we use Σ^* (Σ^+) to denote the set of all (non-empty) words over Σ, and by λ, we denote the empty word. For $w \in \Sigma^*$, $|w|$ denotes the length of the word w, where $|\lambda| = 0$, and for a letter $a \in \Sigma$, $|w|_a$ denotes the