Journal of Computer and System Sciences 78 (2012) 1814–1827

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Reversible pushdown automata

Martin Kutrib *, Andreas Malcher

Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany

ARTICLE INFO

Article history:
Received 15 September 2010
Received in revised form 1 March 2011
Accepted 17 November 2011
Available online 21 December 2011

Keywords:
Reversible computations
Pushdown automata
Formal languages
Closure properties
Decidability questions

ABSTRACT

Reversible pushdown automata are deterministic pushdown automata that are also backward deterministic. Therefore, they have the property that any configuration occurring in any computation has exactly one predecessor. In this paper, the computational capacity of reversible computations in pushdown automata is investigated and turns out to lie properly in between the regular and deterministic context-free languages. Furthermore, it is shown that a deterministic context-free language cannot be accepted reversibly if more than realtime is necessary for acceptance. Closure properties as well as decidability questions for reversible pushdown automata are studied. Finally, we show that the problem to decide whether a given nondeterministic or deterministic pushdown automaton is reversible is P-complete, whereas it is undecidable whether the language accepted by a given nondeterministic pushdown automaton is reversible.

1. Introduction

Computers are information processing devices which are physical realizations of abstract computational models. It may be difficult to define exactly what information is or how information should be measured suitably. It may be even more difficult to analyze in detail how a computational device processes or transmits information while working on some input. Thus, one first step towards a better understanding of information is to study computations in which no information is lost. Another motivation to study information preserving computations is the physical observation that a loss of information results in heat dissipation [3,17]. A first study of this kind has been done in [3] for Turing machines where the notion of *reversible* Turing machines is introduced. Deterministic Turing machines are called reversible when they are also backward deterministic. One fundamental result shown in [3] is that every, possibly irreversible, Turing machine can always be simulated by a reversible Turing machine in a constructive way. This construction is significantly improved in [20] with respect to the number of tapes and tape symbols. Thus, for the powerful model of Turing machines, which describe the recursively enumerable languages, every computation can be made information preserving. At the other end of the Chomsky hierarchy there are the regular languages. Reversible variants of deterministic finite automata have been defined and investigated in [2,23]. It turns out that there are regular languages for which no reversible deterministic finite automaton exists. Thus, there are computations in which a loss of information is inevitable. Another result of [23] is that the existence of a reversible automaton can be decided for a regular language in polynomial time.

Nowadays, reversible computing has become a field of intensive study from several perspectives. In [21] one may find a recent survey which summarizes results on reversible Turing machines, reversible cellular automata, which are a massively parallel model consisting of interacting deterministic finite automata, and other reversible models such as logic gates, logic

* Corresponding author.
E-mail addresses: kutrib@informatik.uni-giessen.de (M. Kutrib), malcher@informatik.uni-giessen.de (A. Malcher).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.12.004

circuits, or logic elements with memory. Reversible deterministic finite automata are also studied in the context of algorithmic learning theory [2,14,18] and quantum computing [8,9] whereas construction problems are investigated in [5,6,19]. A recent paper which motivates the study of reversible computing from the vantage point of physics is [4]. How to compute reversibly by using a reversible programming language is presented in [26]. Reversibility has also been studied for other computational models such as, for example, flowcharts [25] or process calculi [22]. See also the references in [26].

Reversible variants of the massively parallel model of cellular automata and iterative arrays have been also studied in [15,16] with regard to the acceptance of formal languages. One main result there is the identification of data structures and constructions in terms of closure properties which can be implemented reversibly. Another interesting result is that, in contrast to regular languages, there is no algorithm which decides whether a given cellular device is reversible.

In this paper, the investigation of reversibility in computational devices is complemented by the study of *reversible pushdown automata*. These are deterministic pushdown automata that are also backward deterministic. First, it is shown that all regular languages as well as some non-regular languages are accepted by reversible deterministic pushdown automata. On the other hand, we prove that there is a deterministic context-free language which cannot be accepted in a reversible way. Thus, the computational capacity of reversible pushdown automata lies properly in between the regular and deterministic context-free languages. Moreover, every deterministic context-free language which needs more than realtime is shown not to be acceptable by reversible pushdown automata. In the second part of the paper, closure properties and decidability questions of the language class are investigated. It turns out that the closure properties of reversible pushdown automata are similar to those of deterministic pushdown automata. The main difference is the somehow interesting result that the language class accepted by reversible pushdown automata is *not* closed under union and intersection with regular languages. Finally, the questions of whether a given automaton is a reversible pushdown automaton, and whether its language is reversible are investigated. We show that the problem to decide whether a given nondeterministic or deterministic pushdown automaton is reversible is P-complete, whereas it is undecidable whether the language accepted by a given nondeterministic pushdown automaton is reversible.

2. Preliminaries and definitions

Let Σ^* denote the set of all words over the finite alphabet Σ. The empty word is denoted by λ, and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. The set of words of length at most $n \geqslant 0$ is denoted by $\Sigma^{\leqslant n}$. For convenience, we use Σ_λ for $\Sigma \cup \{\lambda\}$. The reversal of a word w is denoted by w^R and for the length of w we write $|w|$. The number of occurrences of a symbol $a \in \Sigma$ in $w \in \Sigma^*$ is written as $|w|_a$. Set inclusion is denoted by $\subseteq$, and strict set inclusion by $\subset$. A *deterministic pushdown automaton* (DPDA) is a system $\mathcal{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, \bot, F\rangle$, where Q is a finite set of states, Σ is the finite input alphabet, Γ is a finite pushdown alphabet, $q_0 \in Q$ is the initial state, $\bot \in \Gamma$ is a distinguished pushdown symbol, called the bottom-of-pushdown symbol, which initially appears on the pushdown store, $F \subseteq Q$ is the set of accepting states, and δ is a mapping from $Q \times \Sigma_\lambda \times \Gamma$ to $Q \times \Gamma^*$ called the transition function. There must never be a choice of using an input symbol or of using λ input. So, it is required that for all q in Q and Z in Γ: if $\delta(q, \lambda, Z)$ is defined, then $\delta(q, a, Z)$ is undefined for all a in Σ.

A *configuration* of a pushdown automaton is a quadruple (v, q, w, γ), where q is the current state, v is the already read and w the unread part of the input, and γ the current content of the pushdown store, the leftmost symbol of γ being the top symbol. On input w the initial configuration is defined to be $(\lambda, q_0, w, \bot)$. For $q \in Q$, $a \in \Sigma_\lambda$, $v, w \in \Sigma^*$, $\gamma \in \Gamma^*$, and $Z \in \Gamma$, let $(v, q, aw, Z\gamma)$ be a configuration. Then its *successor configuration* is $(va, p, w, \beta\gamma)$, where $\delta(q, a, Z) = (p, \beta)$. We write $(v, q, aw, Z\gamma) \vdash (va, p, w, \beta\gamma)$ in this case. The reflexive transitive closure of $\vdash$ is denoted by $\vdash^*$. To simplify matters, we require that in any configuration the bottom-of-pushdown symbol appears exactly once at the bottom of the pushdown store, that is, it can neither appear at some other position in the pushdown store nor be deleted. Formally, we require that if $\delta(q, a, Z) = (p, \beta)$ then either $Z \neq \bot$ and β does not contain $\bot$, or $Z = \bot$ and $\beta = \beta'\bot$, where β' does not contain $\bot$. The *language accepted* by $\mathcal{M}$ with accepting states is

$$L(\mathcal{M}) = \left\{ w \in \Sigma^* \mid (\lambda, q_0, w, \bot) \vdash^* (w, q, \lambda, \gamma), \text{ for some } q \in F \text{ and } \gamma \in \Gamma^* \right\}.$$

In general, the family of all languages that are accepted by some device X is denoted by $\mathscr{L}(X)$.

Now we turn to reversible pushdown automata. Reversibility is meant with respect to the possibility of stepping the computation back and forth. To this end, the pushdown automata have to be also backward deterministic. That is, any configuration occurring in any computation must have at most one predecessor which, in addition, is computable by a DPDA. For reverse computation steps the head of the input tape is always moved to the *left*. Therefore, the automaton rereads the input symbol which has been read in a preceding forward step. So, for reversible pushdown automata there must exist a reverse transition function.

A *reverse transition function* $\delta_R : Q \times \Sigma_\lambda \times \Gamma \to Q \times \Gamma^*$ maps a configuration to its *predecessor configuration*. For $q \in Q$, $a \in \Sigma_\lambda$, $v, w \in \Sigma^*$, $\gamma \in \Gamma^*$, and $Z \in \Gamma$, let $(va, q, w, Z\gamma)$ be a configuration. Then its *predecessor configuration* is $(v, p, aw, \beta\gamma)$, where $\delta_R(q, a, Z) = (p, \beta)$. We write $(va, q, w, Z\gamma) \dashv (v, p, aw, \beta\gamma)$ in this case. Automaton $\mathcal{M}$ is said to be *reversible* (REV-PDA), if there exists a reverse transition function δ_R such that $c_{i+1} \dashv c_i$, $0 \leqslant i \leqslant n-1$, for any sequence $c_0 \vdash c_1 \vdash \cdots \vdash c_n$ of configurations passed through by $\mathcal{M}$ and beginning with an initial configuration c_0 (cf. Fig. 1).

To clarify our notion we continue with an example.