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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On store languages of language acceptors

Oscar H. Ibarra [a,1], Ian McQuillan [b,*,2]

[a] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
[b] Department of Computer Science, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada

ARTICLE INFO

Article history:
Received 15 March 2017
Received in revised form 29 May 2018
Accepted 30 May 2018
Available online xxxx
Communicated by P.G. Spirakis

Keywords:
Store languages
Turing machines
Storage structures
Right quotient
Automata

ABSTRACT

It is well known that the "store language" of every pushdown automaton — the set of store configurations (state and stack contents) that can appear as an intermediate step in accepting computations — is a regular language. Here many models of language acceptors with various store structures are examined, along with a study of their store languages. For each model, an attempt is made to find the simplest model that accepts their store languages. Some connections between store languages of one-way and two-way machines are demonstrated, as with connections between nondeterministic and deterministic machines. A nice application of these store language results is also presented, showing a general technique for proving families accepted by many deterministic models are closed under right quotient with regular languages, resolving some open questions (and significantly simplifying proofs for others that are known) in the literature. Lower bounds on the space complexity of Turing machines for having non-regular store languages are obtained.

1. Introduction

A store configuration of a one-way or two-way language acceptor consists of the state followed by the contents of its memory (store) structure. It does not include the input and the position of the input head. For example, for a nondeterministic pushdown automaton (NPDA), a store configuration is represented by a string qx, where q is a state and x is the contents of the pushdown stack. For multi-tape acceptors, such as for an NPDA augmented with k reversal-bounded counters (NPCM) [1], the store configuration is represented by the string $qxc_1^{j_1}\cdots c_k^{j_k}$, where j_i represents the value of counter i in unary notation, and the c_i symbols and the symbols of x are disjoint. For a machine M, let $S(M)$ be the set of store configurations that can appear as an intermediate step in accepting computations of M.

It is well-known that $S(M)$ is a regular language for any NPDA M [2,3]. Greibach used this result to provide an alternative proof [3] that regular canonical systems produce regular languages [4]. Also, it was a key component to showing that it is decidable whether the set of all infixes (subwords) of the language accepted by a reversal-bounded[3] NPDA is equal to Σ^* (i.e., is dense) [5]. Connections between store languages and the area of verification and model checking have also been recently explored [6].

* Corresponding author.
E-mail addresses: ibarra@cs.ucsb.edu (O.H. Ibarra), mcquillan@cs.usask.ca (I. McQuillan).
1 Supported, in part, by NSF Grant CCF-1117708 (Oscar H. Ibarra).
2 Supported, in part, by Natural Sciences and Engineering Research Council of Canada Grant 2016-06172 (Ian McQuillan).
3 Reversal-bounded means that there is a bound on the number of switches between non-decreasing and non-increasing the size of the pushdown.

https://doi.org/10.1016/j.tcs.2018.05.036
0304-3975/

Due to the usefulness of the store language concept, the store languages of several models of language acceptors are studied in this paper. For machine models with an undecidable emptiness problem, membership in the store language is undecidable. Hence, the investigation of store languages is particularly focused on machine models with a decidable emptiness problem. Results are given here that generalize (in often non-obvious ways) the aforementioned result concerning NPDAs to many other machine models, such as the following:

1. The following nondeterministic machine models with one-way read-only input have regular store languages: k-flip pushdown automata [7] (which are like pushdown automata but can flip the pushdown store up to k times), reversal-bounded queue automata, nondeterministic Turing machines with a reversal-bounded worktape, and stack automata [8,9]. The result for stack automata was shown recently [10] and so our result becomes an alternate proof that follows from existing results in the literature. Also, a new simple but general method is presented for translating results between two-way machines and one-way machines.
2. The store languages of finite-crossing[4] two-way nondeterministic machines with reversal-bounded counters can be accepted by one-way deterministic machines with reversal-bounded counters (DCM).
3. There is a non-finite-crossing two-way deterministic machine with one reversal-bounded counter whose store language cannot be accepted by any NPCM.
4. Some machine models (e.g., deterministic pushdown automata with reversal-bounded counters, DPCM) cannot accept their own store languages.

NPCMs and NCMs have been extensively studied since their introductions in [1,11]. They have found applications in areas such as timed automata [12], model-checking and verification [13,14], membrane computing [15], and Diophantine equations [16].

Another interesting application is presented here showing the closure of many families of languages accepted by deterministic machines under right quotient with regular languages. Some of these resolve open problems in the literature, and others simplify existing known proofs. These include deterministic stack automata (known with a lengthy proof in [17]), deterministic k-flip pushdown automata (stated as an unresolved open problem in [18]), certain types of deterministic Turing machines, deterministic checking stack automata, and deterministic reversal-bounded queue automata. An alternate proof of the result for deterministic pushdown automata that was shown in [19] is also given. This general closure is somewhat surprising given the determinism of the machines and the nondeterministic nature of deletion occurring with quotients.

Finally, lower bounds are obtained on the space complexity of different types of Turing machines in order to have non-regular store languages.

2. Notation

An alphabet Σ is a set of symbols (usually assumed to be finite unless stated otherwise). The set of all words over Σ is denoted by Σ^*, and the set of all non-empty words is denoted by Σ^+. A *language* L over Σ is any subset of Σ^*. Given a word $w \in \Sigma^*$, the *length* of w is denoted by $|w|$. Given $a \in \Sigma$, then $|w|_a$ is the number of a's in w. The *empty word* is denoted by ϵ. The *reverse* of a word w is denoted by w^R, extended to the reverse L^R of a language L in the natural way. Given two languages L_1, L_2, the *left quotient* of L_2 by L_1, $L_1^{-1}L_2 = \{y \mid xy \in L_2, x \in L_1\}$, and the *right quotient* of L_1 by L_2 is $L_1 L_2^{-1} = \{x \mid xy \in L_1, y \in L_2\}$. A language $L \subseteq \Sigma^*$ is *letter-bounded* if there exists (not necessarily distinct) $a_1, \ldots, a_l \in \Sigma$ such that $L \subseteq a_1^* \cdots a_l^*$. A language L is *bounded* if there exists $w_1, \ldots, w_l \in \Sigma^*$ such that $L \subseteq w_1^* \cdots w_l^*$. Given two words $u, v \in \Sigma^*$, u is a *prefix* of v if $v = ux$, for some $x \in \Sigma^*$, u is a *suffix* of v if $v = xu$ for some $x \in \Sigma^*$, u is an *infix* of v if $v = xuy$, for some $x, y \in \Sigma^*$, and u is a *subsequence* of v if $v = x_0 u_1 x_1 \ldots x_{n-1} u_n x_n, x_0, \ldots, x_n, u_1, \ldots, u_n \in \Sigma^*, u = u_1 \cdots u_n$.

In this paper, introductory knowledge of automata and formal language theory is assumed (see [20] for an introduction), including finite automata (NFAs and DFAs), pushdown automata (NPDAs), Turing machines (NTMs and DTMs), and generalized sequential machines (gsms). Let $\mathcal{L}(\text{REG})$ be the family of languages accepted by NFAs.

3. Store languages of one-way machines

Many different kinds of machine models are studied in this paper, such as finite automata, pushdown automata [20], reversal-bounded multicounter machines [1], stack automata (similar to a pushdown automata with the ability to read, but not change on the inside of the pushdown) [8,9], Turing machines [20], queue automata [21], flip-pushdown automata (machines with the ability to flip the pushdown at most k times) [7], and also combinations of their stores within individual machines. The store language of each depends on the precise definition of each type of machine. It is possible to define all such models generally by varying the "store type" similar to Abstract Families of Automata [22] or storage types [23], and then the store language only needs to be defined once for all types of machines. A similar approach is followed here due to the large number of machine models considered, because it allows to make general connections between types of machines, and because store languages depend considerably on the precise definition of the machines.

[4] Finite-crossing means that the input head crosses the boundary of any two adjacent input symbols at most a fixed number of times.

Please cite this article in press as: O.H. Ibarra, I. McQuillan, On store languages of language acceptors, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.05.036