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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

Separability by piecewise testable languages is PTime-complete

Tomáš Masopust

Institute of Theoretical Computer Science and Center of Advancing Electronics Dresden (cfaed), TU Dresden, Germany

ARTICLE INFO

Article history:
Received 24 November 2016
Accepted 7 November 2017
Available online xxxx
Communicated by D. Perrin

Keywords:
Separability
Piecewise testable languages
Complexity

ABSTRACT

Piecewise testable languages form the first level of the Straubing–Thérien hierarchy. The membership problem for this level is decidable and testing if the language of a DFA is piecewise testable is NL-complete. So far, this question has not been addressed for NFAs in the literature. We fill in this gap and show that it is PSpace-complete. The main interest of this paper is, however, the lower-bound complexity of separability of regular languages by piecewise testable languages. Two regular languages are separable by a piecewise testable language if the piecewise testable language includes one of them and is disjoint from the other. For languages represented by NFAs, separability by piecewise testable languages is decidable in PTime. We show that it is PTime-hard and that it remains PTime-hard even if the input automata are minimal DFAs. As a result, it is unlikely that separability of regular languages by piecewise testable languages can be solved in a restricted space or effectively parallelized.

1. Introduction

A regular language over Σ is *piecewise testable* if it is a finite boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $a_i \in \Sigma$ and $n \geq 0$. If n is bounded by a constant, k, then the language is called *k-piecewise testable*. Piecewise testable languages are exactly those regular languages whose syntactic monoid is $\mathcal{J}$-trivial [35]. Simon [36] provided various characterizations of piecewise testable languages, e.g., in terms of monoids or automata. These languages are of interest in many disciplines of mathematics, such as semigroup theory [2,3,28] for their relation to Green's relations or in logic on words [10] for their relation to first-order logic FO[$<$] and the *Straubing–Thérien hierarchy* [40,43].

For an alphabet Σ, level 0 of the Straubing–Thérien hierarchy is defined as $\mathscr{L}(0) = \{\emptyset, \Sigma^*\}$. For integers $n \geq 0$, the levels $\mathscr{L}(n)$ and $\mathscr{L}(n+\frac{1}{2})$ are defined as follows:

- $\mathscr{L}(n+\frac{1}{2})$ consists of all finite unions of languages $L_0 a_1 L_1 a_2 \ldots a_k L_k$ with $k \geq 0$, $L_0, \ldots, L_k \in \mathscr{L}(n)$, and $a_1, \ldots, a_k \in \Sigma$,
- $\mathscr{L}(n+1)$ consists of all finite Boolean combinations of languages from level $\mathscr{L}(n+\frac{1}{2})$.

The levels of the hierarchy contain only *star-free* languages [27]. Piecewise testable languages form the first level of the hierarchy. The hierarchy does not collapse on any level [5], but the problem of deciding whether a language belongs to some level ℓ is largely open for $\ell > \frac{5}{2}$ [1,31]. The Straubing–Thérien hierarchy has further close relations to the *dot-depth hierarchy* [5,7,23,41] and to complexity theory [45].

E-mail address: tomas.masopust@tu-dresden.de.

https://doi.org/10.1016/j.tcs.2017.11.004

The fundamental question is how to efficiently recognize whether a given regular language is piecewise testable. Stern [39] provided a solution that was later improved by Trahtman [44] and Klíma and Polák [21]. Stern presented an algorithm deciding piecewise testability of a regular language represented by a DFA in time $O(n^5)$, where n is the number of states of the DFA. Trahtman improved Stern's algorithm to time quadratic with respect to the number of states and linear with respect to the size of the alphabet, and Klíma and Polák found an algorithm for DFAs that is quadratic with respect to the size of the alphabet and linear with respect to the number of states. Cho and Huynh [6] proved that piecewise testability for DFAs is NL-complete. Although the complexity for DFAs has been deeply investigated, the study for NFAs is missing in the literature. We fill in this gap and show that piecewise testability for NFAs is PSPACE-complete (Theorem 2).

The knowledge of the minimal k or a reasonable bound on k for which a piecewise testable language is k-piecewise testable is of interest in applications [24,17]. The complexity of finding the minimal k has been studied in the literature [17, 20,21,26]. Testing whether a piecewise testable language is k-piecewise testable is coNP-complete for $k \geq 4$ if the language is represented as a DFA [20] and PSPACE-complete if the language is represented as an NFA [26]. The complexity for DFAs and $k < 4$ has also been discussed in detail [26]. Klíma and Polák [21] showed that the upper bound on k is given by the depth of the minimal DFA. This result has recently been generalized to NFAs [25].

The recent interest in piecewise testable languages is mainly for applications of separability of regular languages by piecewise testable languages in logic on words [30] and XML schema languages [8,17,24]. Given two languages K and L and a family of languages $\mathcal{F}$, the separability problem asks whether there exists a language S in $\mathcal{F}$ such that S includes one of the languages K and L and is disjoint from the other. Place and Zeitoun [30] used separability to obtain new decidability results of the membership problem for some levels of the Straubing–Thérien hierarchy. The separability problem for two regular languages represented by NFAs and the family of piecewise testable languages is decidable in polynomial time with respect to both the number of states and the size of the alphabet [8,29]. Separability by piecewise testable languages is of interest also outside regular languages. Although separability of context-free languages by regular languages is undecidable [18], separability by piecewise testable languages is decidable (even for some non-context-free languages) [9]. Piecewise testable languages are further investigated in natural language processing [11,32], cognitive and sub-regular complexity [33], and learning theory [12,22]. They have been extended from word languages to tree languages [4,13,15].

In this paper, we show that separability of regular languages represented by NFAs by piecewise testable languages is a PTIME-complete problem (Theorem 3) and that it remains PTIME-hard even if the input automata are minimal DFAs. As a result, the separability problem is unlikely to be solvable in logarithmic space or effectively parallelizable.

2. Preliminaries and definitions

We assume that the reader is familiar with automata theory [37]. The cardinality of a set A is denoted by $|A|$ and the power set of A by 2^A. The free monoid generated by an alphabet Σ is denoted by Σ^*. A word over Σ is any element of Σ^*; the empty word is denoted by ε. For a word $w \in \Sigma^*$, $\mathrm{alph}(w) \subseteq \Sigma$ denotes the set of all symbols occurring in w.

A *nondeterministic finite automaton* (NFA) is a quintuple $M = (Q, \Sigma, \delta, Q_0, F)$, where Q is the finite nonempty set of states, Σ is the input alphabet, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of accepting states, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function extended to the domain $2^Q \times \Sigma^*$ in the usual way. The language *accepted* by M is the set $L(M) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$. A *path* π from a state q_0 to a state q_n under a word $a_1a_2\cdots a_n$, for some $n \geq 0$, is a sequence of states and input symbols $q_0, a_1, q_1, a_2, \ldots, q_{n-1}, a_n, q_n$ such that $q_{i+1} \in \delta(q_i, a_{i+1})$, for all $i = 0, 1, \ldots, n-1$. Path π is *accepting* if $q_0 \in Q_0$ and $q_n \in F$. We write $q_0 \xrightarrow{a_1a_2\cdots a_n} q_n$ to denote that there is a path from q_0 to q_n under the word $a_1a_2\cdots a_n$. We says that M has a *cycle over an alphabet* $\Gamma \subseteq \Sigma$ if there is a state q in M and a word w over Σ such that $q \xrightarrow{w} q$ and $\mathrm{alph}(w) = \Gamma$. The NFA M is *deterministic* (DFA) if $|Q_0| = 1$ and $|\delta(q, a)| = 1$ for every $q \in Q$ and $a \in \Sigma$. Although we define DFAs as complete, we mostly depict only the most important transitions in our illustrations. The reader can easily complete such an incomplete DFA.

We say that $v = a_1a_2\cdots a_n$ is a *subsequence* of w, denoted by $v \preccurlyeq w$, if $w \in \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$. For two languages K and L, a sequence $(w_i)_{i=1}^r$ of words is a *tower between K and L* if $w_1 \in K \cup L$ and, for all $i = 1, \ldots, r-1$, $w_i \preccurlyeq w_{i+1}$, $w_i \in K$ implies $w_{i+1} \in L$, and $w_i \in L$ implies $w_{i+1} \in K$. The number of words in the sequence, r, is the *height* of the tower. In the same way, we define an infinite sequence of words as an *infinite tower between K and L*. Stern [38] defined towers between a language and its complement. Our definition naturally generalizes his definition to arbitrary two languages. Towers are sometimes called zigzags in the literature [8]. If the languages are clear from the context, we usually omit them. We do not require that the languages K and L are disjoint. However, if there is a $w \in K \cap L$, then there is a trivial infinite tower $w, w, w, \ldots$ between K and L. If we talk about a *tower between two automata*, we mean a tower between their languages.

Let K and L be languages. A language S *separates K from L* if S contains K and does not intersect L. Languages K and L are *separable by a family of languages $\mathcal{F}$* if there exists a language S in $\mathcal{F}$ that separates K from L or L from K.

3. Piecewise testability for NFAs

Given an NFA A over an alphabet Σ, the *piecewise-testability problem* asks whether the language $L(A)$ is piecewise testable. Although the containment in PSPACE follows basically from the result by Cho and Huynh [6], we prefer to provide the proof here for two reasons: (i) we would like to provide an unfamiliar reader with a method to recognize whether a

Please cite this article in press as: T. Masopust, Separability by piecewise testable languages is PTIME-complete, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.11.004