ARTICLE IN PRESS
YINCO:4288

Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Complexity of universality and related problems for partially ordered NFAs

Markus Krötzsch [a,1], Tomáš Masopust [a,b,*,1], Michaël Thomazo [c]

[a] Institute of Theoretical Computer Science and Center of Advancing Electronics Dresden (cfaed), TU Dresden, Germany
[b] Institute of Mathematics, Czech Academy of Sciences, Žižkova 22, 616 62 Brno, Czechia
[c] Inria, France

ARTICLE INFO

Article history:
Received 12 September 2016
Received in revised form 20 June 2017
Available online xxxx

Keywords:
Automata
Nondeterminism
Partial order
Universality
Inclusion
Equivalence

ABSTRACT

Partially ordered NFAs (poNFAs) are NFAs where cycles occur only in the form of self-loops. A poNFA is universal if it accepts all words over its alphabet. Deciding universality is PSPACE-complete for poNFAs. We show that this remains true when restricting to fixed alphabets. This is nontrivial since standard encodings of symbols in, e.g., binary can turn self-loops into longer cycles. A lower coNP-complete complexity bound is obtained if all self-loops in the poNFA are deterministic. We find that such restricted poNFAs (rpoNFAs) characterize $\mathcal{R}$-trivial languages, and establish the complexity of deciding if the language of an NFA is $\mathcal{R}$-trivial. The limitation to fixed alphabets is essential even in the restricted case: deciding universality of rpoNFAs with unbounded alphabets is PSPACE-complete. Consequently, we obtain the complexity results for inclusion and equivalence problems. Finally, we show that the languages of rpoNFAs are definable by deterministic (one-unambiguous) regular expressions.

1. Introduction

The universality problem asks if a given automaton (or grammar) accepts (or generates) all possible words over its alphabet. In typical cases, deciding universality is more difficult than deciding the word problem. For example, universality is undecidable for context-free grammars [3] and PSPACE-complete for nondeterministic finite automata (NFAs) [29]. The study of universality (and its complement, emptiness) has a long tradition in formal languages, with many applications across computer science, e.g., in the context of formal knowledge representation and database theory [4,10,38]. Recent studies investigate the problem for specific types of automata or grammars, e.g., for prefixes or factors of regular languages [32].

In this paper, we are interested in the universality problem for *partially ordered NFAs* (poNFAs) and special cases thereof. An NFA is partially ordered if its transition relation induces a partial order on states: the only cycles allowed are self-loops on a single state. Partially ordered NFAs define a natural class of languages that has been shown to coincide with level $\frac{3}{2}$ of the Straubing–Thérien hierarchy [35] and with Alphabetical Pattern Constraint (APC) languages, a subclass of regular

* Corresponding author at: Institute of Mathematics, Czech Academy of Sciences, Žižkova 22, 616 62 Brno, Czechia.
E-mail addresses: markus.kroetzsch@tu-dresden.de (M. Krötzsch), masopust@math.cas.cz (T. Masopust), michael.thomazo@inria.fr (M. Thomazo).
1 This work was supported by the German Research Foundation (DFG) in Emmy Noether grant KR 4381/1-1 (DIAMOND).

http://dx.doi.org/10.1016/j.ic.2017.06.004
0890-5401/

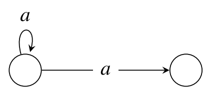

Fig. 1. Nondeterministic self-loops – the forbidden pattern of rpoNFAs.

Table 1
Complexity of deciding universality.

	Unary alphabet		Fixed alphabet		Arbitrary alphabet	
DFA	L-comp.	[21]	NL-comp.	[21]	NL-comp.	[21]
rpoNFA	NL-comp.	(Corollary 25)	coNP-comp.	(Corollary 24)	PSPACE-comp.	(Theorem 28)
poNFA	NL-comp.	(Theorem 4)	PSPACE-comp.	(Theorem 3)	PSPACE-comp.	[1]
NFA	coNP-comp.	[39]	PSPACE-comp.	[1]	PSPACE-comp.	[1]

languages effectively closed under permutation rewriting [6]. Deciding whether an automaton recognizes an APC language (and hence whether it can be recognized by a poNFA) is PSPACE-complete for NFAs and NL-complete for DFAs [6].

Restricting to partially ordered deterministic finite automata (poDFAs), we can capture further classes of interest: two-way poDFAs characterize languages whose syntactic monoid belongs to the variety **DA** [35], introduced by Schützenberger [34]; poDFAs characterize $\mathcal{R}$-trivial languages [8]; and confluent poDFAs characterize level 1 of the Straubing–Thérien hierarchy, also known as $\mathcal{J}$-trivial languages or piecewise testable languages [37]. Other relevant classes of partially ordered automata include partially ordered Büchi automata [24] and two-way poDFAs with look-around [25].

The first result on the complexity of universality for poNFAs is readily obtained. It is well known that universality of regular expressions is PSPACE-complete [1, Lemma 10.2], and it is easy to verify that the regular expressions used in the proof can be expressed in poNFAs:

Corollary 1 *(Lemma 10.2 [1]). The universality problem for poNFAs is* PSPACE*-complete.*

A closer look at the proof reveals that the underlying encoding requires an alphabet of size linear in the input: PSPACE-hardness is not established for alphabets of bounded size. Usually, one could simply encode alphabet symbols σ by sequences $\sigma_1 \cdots \sigma_n$ of symbols from a smaller alphabet, say $\{0,1\}$. However, doing this requires self-loops $q \xrightarrow{\sigma} q$ to be replaced by nontrivial cycles $q \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} q$, which are not permitted in poNFAs.

We settle this open problem by showing that PSPACE-hardness is retained even for binary alphabets. This negative result leads us to ask if there is a natural subclass of poNFAs for which universality does become simpler. We consider *restricted* poNFAs (rpoNFAs), which require self-loops to be deterministic in the sense that the automaton contains no transition as in Fig. 1, which we call *nondeterministic self-loops* in the rest of the paper. Large parts of the former hardness proof hinge on transitions of this form, which, speaking intuitively, allow the automaton to navigate to an arbitrary position in the input (using the loop) and, thereafter, continue checking an arbitrary pattern. Indeed, we find that the universality becomes coNP-complete for rpoNFAs with a fixed alphabet.

However, this reduction of complexity is not preserved for unrestricted alphabets. We use a novel construction of rpoNFAs that characterize certain exponentially long words to show that universality is PSPACE-complete even for rpoNFAs if the alphabet may grow polynomially. Our complexity results are summarized in Table 1.

As a by-product, we show that rpoNFAs provide another characterization of $\mathcal{R}$-trivial languages introduced and studied by Brzozowski and Fich [8], and we establish the complexity of detecting $\mathcal{R}$-triviality and k-$\mathcal{R}$-triviality for rpoNFAs.

From the practical point of view, the problems of inclusion and equivalence of two languages, which are closely related to universality, are of interest, e.g., in optimization. Indeed, universality can be expressed either as the inclusion $\Sigma^* \subseteq L$ or as the equivalence $\Sigma^* = L$. Although equivalence can be seen as two inclusions, the complexity of inclusion does not play the role of a lower bound. For instance, for two deterministic context-free languages inclusion is undecidable [14], whereas equivalence is decidable [36]. However, the complexity of universality gives a lower bound on the complexity of both inclusion and equivalence, and we show that, for the partially ordered NFAs studied in this paper, the complexities of inclusion and equivalence coincide with the complexity of universality.

This paper is a full version of the work [23] presented at the 41st International Symposium on Mathematical Foundations of Computer Science.

2. Preliminaries and definitions

We assume that the reader is familiar with automata theory [1]. The cardinality of a set A is denoted by $|A|$ and the power set of A by 2^A. An *alphabet* Σ is a finite nonempty set. A *word* over Σ is any element of the free monoid Σ^*, the *empty word* is denoted by ε. A *language* over Σ is a subset of Σ^*. For a language L over Σ, let $\overline{L} = \Sigma^* \setminus L$ denote its complement.

A *subword* of w is a word u such that $w = w_1 u w_2$, for some words w_1, w_2; u is a *prefix* of w if $w_1 = \varepsilon$ and it is a *suffix* of w if $w_2 = \varepsilon$.

Please cite this article in press as: M. Krötzsch et al., Complexity of universality and related problems for partially ordered NFAs, Inf. Comput. (2017), http://dx.doi.org/10.1016/j.ic.2017.06.004