The Freeness Problem for Automaton Semigroups

Daniele D'Angeli
Università degli Studi Niccolò Cusano, Roma, Italy

Emanuele Rodaro
Department of Mathematics, Politecnico di Milano, Italy

Jan Philipp Wächter
Fachrichtung Mathematik, Universität des Saarlandes, Saarbrücken, Germany

Abstract

We present a new technique to encode Post's Correspondence Problem into automaton semigroups and monoids. The encoding allows us to precisely control whether there exists a relation in the generated semigroup/monoid and thus show that the freeness problems for automaton semigroups and for automaton monoids (listed as open problems by Grigorchuk, Nekrashevych and Sushchanskĭı) are undecidable. The construction seems to be quite versatile and we obtain the undecidability of further problems: Is a given automaton semigroup (monoid) (left) cancellative? Is it equidivisible (which – together with the existence of a (proper) length function – characterizes free semigroups and monoids)? Does a given map extend into a homomorphism between given automaton semigroups? Finally, our construction can be adapted to show that it is undecidable whether a given automaton generates a free monoid whose basis is given by the states (but where we allow one state to act as the identity). In the semigroup case, we show a weaker version of this.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Problems, reductions and completeness

Keywords and phrases Automaton Monoid, Automaton Semigroup, Freeness Problem, Free Presentation

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.44

Related Version *Full Version*: https://arxiv.org/abs/2402.01372

Funding *Jan Philipp Wächter*: funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 492814705 – while visiting the Department of Mathematics at Politecnico di Milano. The listed affiliation is his current one, partly funded by ERC grant 101097307.

1 Introduction

In the 1980s, Grigorchuk solved a famous question by Milnor (see [20] for a nice introduction) by presenting the first group with intermediate growth: the number of elements that can be written as a word of length at most n over the generators grows sub-exponentially but super-polynomially. The group has even more noteworthy properties. It is amenable but not elementary amenable (e. g. [24]) and an infinite 2-group (giving a counter-example to Burnside's problem, e. g. [33, 3]). Its peculiar properties stirred interest in Grigorchuk's group and groups of similar form where it soon became important that Grigorchuk's group has a nice description using what is simply called an *automaton* in this context (e. g. [33] or [3]). The simplicity of this presentation (the automaton only uses a binary alphabet and four states – with an additional identity state) contrasts the complex nature of the group. An "automaton" here is what more precisely is called a finite-state letter-to-letter transducer (i. e. an automaton with input and output). The idea is that in such an automaton every state induces a mapping of input to output words and the closure of these functions under composition forms a semigroup. If the automaton is additionally invertible, the functions are bijections and

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 44; pp. 44:1–44:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

we may consider the generated group. This leads to the classes of *automaton semigroups* and *groups*, which contain further noteworthy examples (e. g. Gupta-Siki p-groups [22], the lamplighter group [21] and more general lamplighter-like groups [37, 38]).

Being able to finitely describe groups without classical finite presentations (consisting of generators and relations) additionally highlights the usefulness of considering (semi)groups generated by automata. Starting from Grigorchuk's group, the study of automaton groups and semigroups is nowadays a thriving research field with important connections to many neighboring areas (such as geometry, dynamical systems and symbolic dynamics; see e. g. [33, 3] for more background information). The extensive research in Mathematics and Computer Science on the semigroup (and monoid) case (e. g. [9, 26, 7, 34, 1, 15]) arises naturally from the group case for example via the *dual automaton* where states and input/output letters swap places. The connection between an automaton and its dual has been exploited algebraically and algorithmically (e. g. [18, 41, 42, 26, 27, 11]).

In this work, we look further at the algorithmic aspects of this interesting class by showing that its *freeness problem* is undecidable. This problem asks whether a given automaton generates a free semigroup (or monoid). It has been studied extensively for other classes of groups and semigroups. Since freeness is a Markov property, the problem is undecidable for classical finite group (and, thus, semigroup) presentations (see e. g. [29]). Further important results include the undecidability of the freeness problem for matrix semigroups, originally shown using a reduction from Post's Correspondence Problem [25], which has been improved and contrasted in many further publications (e. g. [31, 10, 4]). Interestingly, matrix (semi)groups and automaton (semi)groups are connected in the sense that the former can be presented as subgroups of the latter [8] (see also [40, 12, 43]) but this does not help to prove the freeness problem undecidable for automaton (semi)groups [13].

With our result, we continue this line of research but also further contribute to the study of freeness in self-similar (i. e. generated by infinite automata) and automaton structures as well as their algorithmic aspects. For the former, we refer the reader to the survey [36] and only point out that, while it is known that free groups are automaton groups [41, 42, 39], these constructions are usually deemed rather difficult. For automaton semigroups and monoids, the situation seems to be simpler: every free semigroup of (finite) rank at least two can be generated by an automaton (see [9] or Example 2.5) but the free semigroup of rank one cannot [9]. All free monoids of finite rank are automaton semigroups, though.

Regarding algorithmic questions for automaton (semi)groups, we point out that, while one may easily be misled into believing that using a finite automaton as the generating combinatorial object should be rather simple, the situation is actually quite complex and only a few natural algorithmic problems are known to be undecidable while many others notoriously remain open problems. An exception here seems to be that the word problem for automaton (semi)groups is PSpace-complete. Interestingly, this was first known for semigroups [14] and was later extended to groups [43]. Some subclasses have simpler word problems. For example, using finitary automata to present finite groups results in a coNP-complete word problem [28] and the word problem of an automaton group of polynomial activity is in polylogarithmic space [5] (see [44] for more information). On the other hand, there is an automaton group with an undecidable conjugacy problem [40] ("are two given group elements conjugate in the group?"). The construction used there also shows that the isomorphism problem for automaton groups ("are the groups generated by two given automata isomorphic?") and, thus, automaton semigroups is undecidable.[1] There are two constructions for an automaton group with undecidable order problem ("has a given group

[1] Unfortunately, this does not seem to be written down explicitly anywhere.

element finite or infinite order?") [17, 2]. The latter of the two even yields a contracting automaton. The undecidability was also first known for automaton semigroups [16] and the problem is decidable for bounded automaton groups [6] and monoids [1].

All these constructions encoding Turing machines in automaton (semi)groups make a statement about individual (semi)group elements. Since the interaction between the generating automaton and generated algebraic structure is often surprising and still not well understood, it is much more challenging to construct reductions where the entire generated (semi)group (or monoid) has a certain property (based on whether we input a positive or negative problem instance). The only known result of this kind seems to be that the finiteness problem for automaton semigroups ("Is the semigroup generated by a given automaton finite?") is undecidable [16]. The corresponding group problem is still open [19].

Our reduction from Post's Correspondence Problem [35] to the freeness problems for automaton semigroups and for monoids in this paper is a second result of this form. It solves the corresponding open problem by Grigorchuk, Nekrashevych and Sushchanskĭĭ [19, 7.2 b)] and, despite previous attempts [12, 13] and a positive result for semigroups generated by invertible and reversible automata with two states [26] as well as a negative result on testing for relations of the form $w = \mathbb{1}$ [12], the problem had remained open quite a while for groups and for semigroups. The main challenge seems to be that we need very precise control over the relations in the generated semigroup (which seems to be much more difficult than, e. g., ensuring that the semigroup is finite or infinite) while the interaction between the structure of the generating automaton and the semigroup/monoid relations is highly non-obvious.

Our construction yields further results beyond the freeness problem(s). Namely, testing whether a given automaton generates a (left) cancellative semigroup/monoid and whether the semigroup/monoid generated by a given automaton is equidivisible (a notion strongly related to freeness by Levi's lemma, see Fact 2.2) are undecidable. We also obtain that it is undecidable whether a given automaton generates a free semigroup with a given basis and whether a given map between the state sets of two given automata can be extended into an iso- or homomorphism. The latter problem is connected to the (undecidable, see above) isomorphism problem for automaton semigroups in the sense that it asks whether all relations of the first automaton semigroup also hold in the second one.

Finally, the construction seems to be flexible enough to be adapted to similar problems, which gives us hope that our results could also contribute towards showing that the freeness problem is undecidable in the group case. For example, it can be adapted to show that the free presentation problem for automaton monoids is undecidable: does a given automaton generate a free monoid whose rank is equal to the number of its states (minus an identity state)? In other words, we cannot test whether a given automaton monoid contains any relations (although this is semi-decidable as the word problem is decidable, see above).

Adapting our construction for this is necessary because the construction in the semigroup case always yields semigroup relations since we need to use a result on the closure of the class of automaton semigroups under (certain) free products [30] in order to construct some kind of "partial" powers of the generating automaton. However, no details of this construction will be required to understand our results. More generally, the presentation in this work is meant to be self-contained (although the construction may be considered to be rather technical).

2 Preliminaries

Fundamentals, Semigroups and Monoids. We write $A \uplus B$ for the disjoint union of sets and consider the set of natural numbers $\mathbb{N}$ to contain 0. We assume the reader to be familiar with fundamental notions of semigroup theory (see e. g. [23]). We write $\mathbb{1}_M$ for the neutral element of a monoid M or, if M is clear from the context, simply $\mathbb{1}$. For a monoid M, we let $M^{\mathbb{1}} = M$ and, if S is a semigroup but not a monoid, we may adjoin a neutral element $\mathbb{1} \notin S$ to S by letting $\mathbb{1}\mathbb{1} = \mathbb{1}$ and $\mathbb{1}s = s = s\mathbb{1}$ for all S and denote the resulting monoid by $S^{\mathbb{1}}$.

Words, Free Semigroups and Free Monoids. Let B be a finite, non-empty set, which we call an *alphabet*. A *word* w over B is a finite sequence $a_1 \dots a_n$ with $a_1, \dots, a_n \in B$, whose *length* is $|w| = n$. We denote the unique word of length 0 (i. e. the *empty word*) by ε. The set of all words over B is denoted by B^*. Words have the natural operation of juxtaposition (where we let $uv = a_1 \dots a_m b_1 \dots b_n$ for $u = a_1 \dots a_m$ and $v = b_1 \dots b_n$ with $a_1, \dots, a_m, b_1, \dots, b_n \in B$), which turns B^* into a monoid with the neutral element ε. This monoid B^* is *the free monoid* with *basis* B (or *over* B) and a monoid M is *free* (with basis B) if it is isomorphic to B^* (for some alphabet B). Closely related to the free monoid is *the free semigroup* B^+, which is formed by the set of all non-empty words (i. e. $B^+ = B^* \setminus \{\varepsilon\}$) and (again) juxtaposition as operation. Similarly, a semigroup S is *free* (with basis B) if it is isomorphic to B^+ (for some alphabet B). Note that B^* is (isomorphic to) $(B^+)^{\mathbb{1}}$ and that the basis of a free monoid or semigroup is unique (see e. g. [23, Proposition 7.1.3]). The *rank* of a free monoid or semigroup is the cardinality $|B|$ of its basis B. We will use common conventions from formal language theory and, e. g., write q^+ and q^* for $\{q\}^+$ and $\{q\}^*$.

Properties of Free Semigroups and Monoids. We will need some properties of free semigroups and monoids. A (general) semigroup S is *left cancellative* if $st = st'$ implies $t = t'$ for all $s, t, t' \in S$. Symmetrically, it is *right cancellative* if $st = s't$ implies $s = s'$ for all $s, s', t \in S$ and, finally, it is *cancellative* if it is both left and right cancellative. It is easy to see that B^* and, thus, B^+ are cancellative (see, e. g. [23, Proposition 7.1.1]).

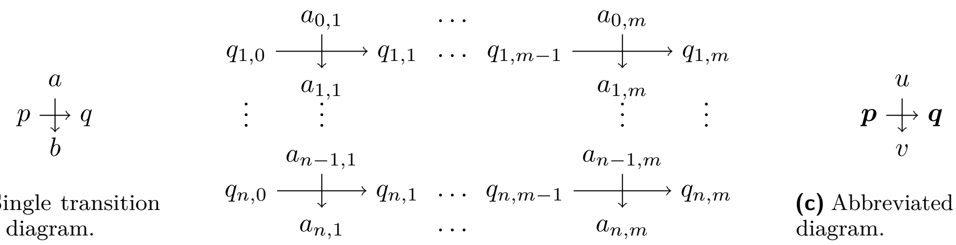
▶ **Fact 2.1.** *Free semigroups and free monoids are cancellative.*

A *length function* of a semigroup S is a homomorphism $S \to \mathbb{N}_{>0}$ where $\mathbb{N}_{>0}$ is the additive semigroup of strictly positive natural numbers. A monoid M has a *proper length function* if there is a monoid homomorphism $M \to \mathbb{N}$ (where $\mathbb{N}$ is the additive monoid of the natural numbers including 0) such that $\mathbb{1}$ is the only pre-image of 0 (i. e. only $\mathbb{1}$ has length 0, all other elements have strictly positive length). A semigroup S that is not a monoid has a length function if and only if $S^{\mathbb{1}}$ has a proper one and free semigroups and monoids do have (proper) length functions (mapping a word to its length).

A semigroup (or monoid) S is *equidivisable* if, for all $s_1, s_2, s_1', s_2' \in S$ with $s_1 s_2 = s_1' s_2'$, there is some $x \in S^{\mathbb{1}}$ with $s_1 = s_1' x$ and $x s_2 = s_2'$ or with $s_1 x = s_1'$ and $s_2 = x s_2'$. It is not difficult to see that free semigroups and monoids are equidivisible (see e. g. [23, Proposition 7.1.2]). Together with having a (proper) length function, this turns out to characterize free semigroups and monoids (see e. g. [23, Proposition 7.1.8]).

▶ **Fact 2.2** (Levi's Lemma)**.** *A semigroup (monoid) S is free if and only if it is equidivisible and has a (proper) length function.*

Free Products of Semigroups. A *semigroup presentation* is a pair $\langle Q \mid \mathcal{R} \rangle_{\mathscr{S}}$ of a set of *generators* Q and a (possibly infinite) set of *relations* $\mathcal{R} \subseteq Q^+ \times Q^+$. We will only consider presentations where Q is finite and non-empty. If we denote by $\mathcal{C}$ the smallest congruence $\mathcal{C} \subseteq Q^+ \times Q^+$ with $\mathcal{R} \subseteq \mathcal{C}$, the semigroup *presented* by such a presentation is $S = Q^+/\mathcal{C}$ formed by the congruence classes $[\cdot]$ of $\mathcal{C}$ with the (well-defined!) operation $[u] \cdot [v] = [uv]$. Every semigroup generated by a finite, non-empty set Q is presented by some semigroup presentation of this form.



(a) Single transition cross diagram.

(b) Multiple crosses combined in one diagram.

(c) Abbreviated cross diagram.

Figure 1 Combined and abbreviated cross diagrams.

The free product of the semigroups $S = \langle Q \mid \mathcal{S} \rangle_{\mathscr{S}}$ and $T = \langle P \mid \mathcal{R} \rangle_{\mathscr{S}}$ is the semigroup $S \star T = \langle Q \uplus P \mid \mathcal{S} \cup \mathcal{R} \rangle_{\mathscr{S}}$. For example, we have $\{p, q\}^+ = p^+ \star q^+$.

▶ Remark. Of course, there is also the free product of monoids (and monoid presentations). However, we will only consider free products of semigroups (in particular: $\{p, q\}^* \not\simeq p^* \star q^*$).

Automata. In the current context, an *automaton* is a triple $\mathcal{T} = (Q, \Sigma, \delta)$ consisting of a non-empty, finite set of *states* Q, an *alphabet* Σ and a set $\delta \subseteq Q \times \Sigma \times \Sigma \times Q$ of *transitions*.

▶ Remark. What we simply call an automaton here would rather be called a finite-state, letter-to-letter transducer in more general automaton-theoretic terms. However, simply using the term "automaton" is standard terminology in the area. We also do not use initial or final states as they do not interact nicely with the self-similar nature of the semigroups and monoids generated by automata we are about to define.

For transitions, we will use the graphical notation $p \xrightarrow{a/b} q$ to denote $(p, a, b, q) \in Q \times \Sigma \times \Sigma \times Q$. Such a transition *starts* in p, *ends* in q, its *input* is a and its *output* is b. This reflects the common way of depicting automata (see e. g. Figure 2). When dealing with an automaton $\mathcal{T} = (Q, \Sigma, \delta)$, we are actually dealing with two alphabets (Q and Σ). In order to avoid confusion, we call the elements of Q *states* and the elements of Q^* *state sequences*, while reserving the terms *letters* and *words* for the elements of Σ and Σ^*, respectively.

Another somewhat graphical tool that we will make heavy use of are *cross diagrams*. Here, a cross diagram as given in Figure 1a indicates the existence of a transition $p \xrightarrow{a/b} q$ in the automaton. Cross diagrams can be stacked together in order to create larger ones. For example, the diagram in Figure 1b indicates the existence of the transition $q_{i,j-1} \xrightarrow{a_{i-1,j}/a_{i,j}} q_{i,j}$ for all $0 < i \leq n$ and $0 < j \leq m$. When combining cross diagrams, we will sometimes omit unnecessary states and letters. Additionally, we will also abbreviate them: for example, if we let $\boldsymbol{p} = q_{n,0} \dots q_{1,0}$, $u = a_{0,1} \dots a_{0,m}$, $v = a_{n,1} \dots a_{n,m}$ and $\boldsymbol{q} = q_{n,m} \dots q_{1,m}$, the cross diagram in Figure 1c is an abbreviation of the cross diagram in Figure 1b. It is important here to note the order we write the state sequences in: in our example, $q_{1,0}$ is the first state in the top left of the cross diagram but it is the rightmost state in the sequence $\boldsymbol{p}$. This order will later be more natural as we will define a left action based on cross diagrams.

An automaton $\mathcal{T} = (Q, \Sigma, \delta)$ is called *complete and deterministic* if, for every $p \in Q$ and every $a \in \Sigma$, there is exactly one $q \in Q$ and exactly one $b \in \Sigma$ such that the cross diagram in Figure 1a holds (i. e. in every state p and for every letter $a \in \Sigma$, there is exactly one transition starting in p with input a). We call such an automaton a *complete $\mathscr{S}$-automaton* (as they naturally generate semigroups).

An automaton $\mathcal{S} = (P, \Sigma, \sigma)$ is a *subautomaton* of another automaton $\mathcal{T} = (Q, \Gamma, \delta)$ if $P \subseteq Q$, $\Sigma \subseteq \Gamma$ and $\sigma \subseteq \delta$. In this case, any cross diagram of $\mathcal{S}$ is also valid for $\mathcal{T}$.

Automaton Semigroups and Monoids. Let $\mathcal{T} = (Q, \Sigma, \delta)$ be a complete $\mathscr{S}$-automaton. By induction, there is exactly one $v \in \Sigma^+$ and exactly one $\boldsymbol{q} \in Q^+$ for every $\boldsymbol{p} \in Q^+$ and $u \in \Sigma^+$ such that the cross diagram in Figure 1c holds. This allows us to define a left action of Q^+ on Σ^+ by letting $\boldsymbol{p} \circ u = v$ and to define a right action of Σ^+ on Q^+, called the *dual action*, by letting $\boldsymbol{p} \cdot u = \boldsymbol{q}$. The reader may verify that this indeed defines well-defined actions by the way cross diagrams work. We may extend these into an action of Q^* on Σ^* and an action of Σ^* on Q^* by letting $\varepsilon \circ u = u$ for all $u \in \Sigma^*$, $\boldsymbol{p} \circ \varepsilon = \varepsilon$ for all $\boldsymbol{p} \in Q^*$, $\varepsilon \cdot u = \varepsilon$ again for all $u \in \Sigma^*$ and, finally, $\boldsymbol{p} \cdot \varepsilon = \boldsymbol{p}$ for (again) all $\boldsymbol{p} \in Q^*$.

By the way cross diagrams work, there is an interaction between the two actions: for all $\boldsymbol{p}, \boldsymbol{q} \in Q^*$ and all $u, v \in \Sigma^*$, we have $\boldsymbol{p} \circ uv = (\boldsymbol{p} \circ u)[(\boldsymbol{p} \cdot u) \circ v]$ and $\boldsymbol{qp} \cdot u = [\boldsymbol{q} \cdot (\boldsymbol{p} \circ u)](\boldsymbol{p} \cdot u)$.

The action $\boldsymbol{p} \circ u$ allows us to define the congruence $=_\mathcal{T} \subseteq Q^* \times Q^*$ by $\boldsymbol{p} =_\mathcal{T} \boldsymbol{q} \iff \forall u \in \Sigma^* : \boldsymbol{p} \circ u = \boldsymbol{q} \circ u$. We denote the class of $\boldsymbol{p} \in Q^*$ with respect to $=_\mathcal{T}$ by $[\boldsymbol{p}]_\mathcal{T}$. The set $\mathscr{M}(\mathcal{T}) = Q^* / =_\mathcal{T}$ of these classes forms a monoid, which is called the *monoid generated* by $\mathcal{T}$. In other words, it is the faithful quotient of Q^* with respect to the action $\boldsymbol{q} \circ u$. Note that ε acts like the identity on all $u \in \Sigma^*$ and the class of ε, thus, forms the neutral element of $\mathscr{M}(\mathcal{T})$. A monoid arising in this way is called a *complete automaton monoid*.

Similarly, the *semigroup generated* by $\mathcal{T}$ is the semigroup $\mathscr{S}(\mathcal{T}) = Q^+ / =_\mathcal{T}$ and any such semigroup is a *complete automaton semigroup*. Note that monoid and semigroup generated by a complete $\mathscr{S}$-automaton coincide if there is a non-empty state sequence acting trivially.

▶ Remark 2.3. We only consider complete $\mathscr{S}$-automata in this work but will make this explicit by talking about complete $\mathscr{S}$-automata and complete automaton semigroups and monoids. In the literature, these objects are often simply called "automaton semigroups" (the term "automaton monoid" is less common). This is a convention that we could also follow here but choose not to since the concepts generalize naturally also to non-complete automata, yielding (partial) automaton semigroups and monoids. It is not known whether the two classes coincide (see [15] for more details).

▶ Remark 2.4. There is a subtle difference between an automaton monoid and an automaton semigroup which happens to be a monoid. In the latter, the neutral element not necessarily acts as the identity map. In fact, it is not known whether the two classes coincide (see [9, Proposition 3.1] for the analogue for groups).

Free Semigroups (Monoids) as Automaton Semigroups (Monoids). As examples of complete automaton semigroups and monoids, we will next look at how to generate free semigroups and monoids. The free monoid of rank one is generated by an automaton known as the *adding machine* (see e. g. [33] or [3]), which turns it into both a complete automaton monoid and a complete automaton semigroup. The free semigroup of rank one, on the other hand, is neither [9, Proposition 4.3] (see also [7, Theorem 15], [15, Theorem 19] and [43, Theorem 1.2.1.4]).

However, free semigroups of higher rank (and their monoid counter-parts) are indeed complete automaton semigroups [9, Proposition 4.1]:

▶ **Example 2.5.** Let R be a finite set with $|R| \geq 2$. Consider the automaton $\mathcal{R} = (R, R, \rho)$ with $\rho = \{a \xrightarrow{b/a} b \mid a, b \in R\}$ (see Figure 2 for the binary case). One easily verifies that $\mathcal{R}$ is a complete $\mathscr{S}$-automaton and we claim that it generates R^+. For this, it suffices to show that, for every $\boldsymbol{p}, \boldsymbol{q} \in R^+$ with $\boldsymbol{p} \neq \boldsymbol{q}$, there is some $u \in R^*$ with $\boldsymbol{p} \circ u \neq \boldsymbol{q} \circ u$. We may assume $|\boldsymbol{p}| \geq |\boldsymbol{q}|$ and there needs to be some $a \in R$ with $\boldsymbol{p} \neq \boldsymbol{q}a^{|\boldsymbol{p}|-|\boldsymbol{q}|}$ (we just need to take a different to the last letter of $\boldsymbol{p}$ if the lengths differ). Now, observe that, for all $n \geq 1$ and

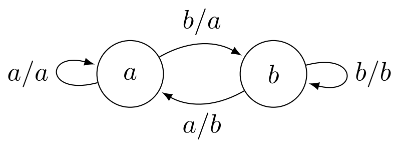


Figure 2 A complete $\mathscr{S}$-automaton generating $\{a,b\}^+$.

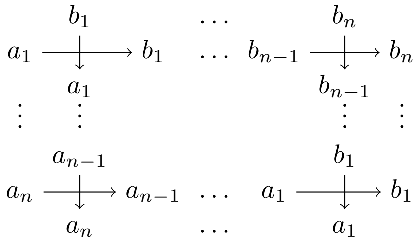


Figure 3 Cross diagram of $\mathcal{R}$.

all $a_1, \dots, a_n, b_1, \dots, b_n \in R$, we have the cross diagram in Figure 3 by the construction of $\mathcal{R}$. This shows, in particular, $\boldsymbol{p} \circ a^{|\boldsymbol{p}|} = \boldsymbol{p}$ and $\boldsymbol{p} \cdot a^{|\boldsymbol{p}|} = a^{|\boldsymbol{p}|}$. By a similar cross diagram, we obtain $\boldsymbol{p} \neq_{\mathcal{R}} \boldsymbol{q}$ (since $\boldsymbol{q} \circ a^{|\boldsymbol{p}|} = (\boldsymbol{q} \circ a^{|\boldsymbol{q}|})(a^{|\boldsymbol{q}|} \circ a^{|\boldsymbol{p}|-|\boldsymbol{q}|}) = \boldsymbol{q}a^{|\boldsymbol{p}|-|\boldsymbol{q}|} \neq \boldsymbol{p} = \boldsymbol{p} \circ a^{|\boldsymbol{p}|}$).

There is no state sequence which acts like the identity and this means that $\mathscr{M}(\mathcal{R})$ is $\mathscr{S}(\mathcal{R})^{\mathbb{1}} \simeq R^*$, which shows that R^* is a complete automaton monoid.

The construction presented in Example 2.5 is clearly computable and we obtain:

▶ **Fact 2.6.** *For every finite set R with $|R| \geq 2$, one can compute an $\mathscr{S}$-automaton $\mathcal{R} = (R, R, \rho)$ with $\mathscr{S}(\mathcal{R}) \simeq R^+$ and $\mathscr{M}(\mathcal{R}) \simeq R^*$.*

Automaton Operations. The *union* of two automata $\mathcal{T}_1 = (Q_1, \Sigma_1, \delta_1)$ and $\mathcal{T}_2 = (Q_2, \Sigma_2, \delta_2)$ is the automaton $\mathcal{T}_1 \cup \mathcal{T}_2 = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \cup \delta_2)$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are both complete $\mathscr{S}$-automaton with non-intersecting state sets ($Q_1 \cap Q_2 = \emptyset$) but a common alphabet $\Sigma_1 = \Sigma_2$, their union $\mathcal{T}_1 \cup \mathcal{T}_2$ is also a complete $\mathscr{S}$-automaton (which allows us, for example, to consider the semigroup $\mathscr{S}(\mathcal{T}_1 \cup \mathcal{T}_2)$). Similarly, the union of two complete $\mathscr{S}$-automata with the same state set but disjoint alphabets is again a complete $\mathscr{S}$-automaton. This operation basically adds the transitions of $\mathcal{T}_2$ to the existing transitions of $\mathcal{T}_1$.

The *composition* of two automata $\mathcal{T}_2 = (Q_2, \Sigma, \delta_2)$ and $\mathcal{T}_1 = (Q_1, \Sigma, \delta_1)$ over a common alphabet Σ is the automaton $\mathcal{T}_2 \circ \mathcal{T}_1 = (Q_2Q_1, \Sigma, \delta_2 \circ \delta_1)$ with

$$\delta_2 \circ \delta_1 = \left\{ p_2p_1 \xrightarrow{a/c} q_2q_1 \,\middle|\, \exists b \in \Sigma : p_1 \xrightarrow{a/b} q_1 \in \delta_1 \text{ and } p_2 \xrightarrow{b/c} q_2 \in \delta_2 \right\}$$

(where $Q_2Q_1 = \{q_2q_1 \mid q_1 \in Q_1, q_2 \in Q_2\}$ is the cartesian product of Q_2 and Q_1). If $\mathcal{T}_2$ and $\mathcal{T}_1$ are complete $\mathscr{S}$-automata, also their composition is.

The *k-th power* $\mathcal{T}^k$ of an automaton $\mathcal{T}$ is the k-fold composition of $\mathcal{T}$ with itself. It is computable and, if $\mathcal{T}$ (and, thus, $\mathcal{T}^k$) is a complete $\mathscr{S}$-automaton, the actions of some $\boldsymbol{p} \in Q^*$ of length $|\boldsymbol{p}| = k$ seen as a state of $\mathcal{T}^k$ or seen as a state sequence over $\mathcal{T}$ coincide. Thus (and by an analogue for the dual action), the notations $\boldsymbol{p} \circ u$ and $\boldsymbol{p} \cdot u$ remain unambiguous and we have $\mathscr{S}(\mathcal{T}) = \mathscr{S}(\mathcal{T} \cup \mathcal{T}^k)$ for all $k \geq 1$, which is usually used to ensure that any fixed state sequence $\boldsymbol{p} \in Q^+$ may be assumed to be congruent to a single state under $=_{\mathcal{T}}$ (i. e. equal in the semigroup or monoid).

Finally, the *dual* of an automaton $\mathcal{T} = (Q, \Sigma, \delta)$ is the automaton $\partial\mathcal{T} = (\Sigma, Q, \partial\delta)$ with $\partial\delta = \left\{ a \xrightarrow{p/q} b \,\middle|\, p \xrightarrow{a/b} q \in \delta \right\}$ (i. e. we swap the roles of the states Q and the letters Σ). Clearly, the dual of a complete $\mathscr{S}$-automaton is again a complete $\mathscr{S}$-automaton.

The dual automaton can make it sometimes more accessible to understand how a letter is transformed by a state sequence: we just have to follow a path in the graphical representation of the dual automaton. For example, from Figure 5, it is obvious that the only way for $\boldsymbol{p} \circ \alpha = \boldsymbol{q} \circ \beta$ to hold is for both of them to be equal to f.

Adding Free Generators. For our results, we will need to add new free generators to existing automaton semigroups S computationally (in the sense that we do not change the behavior of existing state sequences but add a new state q such that the new automaton generates the (semigroup) free product $S \star q^+$). More precisely, we will use the following statement, which follows from the construction used for [30, Theorem 13].

▶ **Proposition 2.7.** *On input of a complete $\mathscr{S}$-automaton $\mathcal{S} = (P, \Sigma, \sigma)$, one can compute a complete $\mathscr{S}$-automaton $\mathcal{T} = (Q, \Gamma, \delta)$ with $Q = P \uplus \{q\}$ such that the identity on Q extends into a well-defined isomorphism $\mathscr{S}(\mathcal{T}) \to \mathscr{S}(\mathcal{S}) \star q^+$ (for the free product of semigroups).*

3 The Freeness Problem for Semigroups

We reduce Post's Correspondence Problem[2] `PCP`

Constant:	an alphabet Λ
Input:	homomorphisms $\varphi, \psi : I = \{1, \dots, n\} \to \Lambda^+$
Question:	$\exists \boldsymbol{i} \in I^+ : \varphi(\boldsymbol{i}) = \psi(\boldsymbol{i})$?

to (the complement of) the freeness problem for automaton semigroups. For this, we fix an instance φ, ψ, I for `PCP`[3] over an alphabet Λ and describe how to map it to a complete $\mathscr{S}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$ in such a way that $\mathcal{T}$ can be computed and the `PCP` instance has a solution if and only if $\mathscr{S}(\mathcal{T})$ is **not** a free semigroup.

Starting from the free semigroup, we will construct $\mathcal{T}$ (in steps) such that the semigroup has a relation $\#_1 \boldsymbol{i} \#_1 =_{\mathcal{T}} \#_1 \boldsymbol{i} \#_2$ for $\boldsymbol{i} \in I^+$ if and only if $\boldsymbol{i}$ belongs to a `PCP` solution (if there is no solution, $\mathscr{S}(\mathcal{T})$ is free). Throughout this process, the reader may find it convenient to refer to Table 1 for the various symbols we are going to use.

The rough idea is to add an input symbol ι whose dual action turns $\boldsymbol{i}\#_1$ into $\varphi(\boldsymbol{i})$ and $\boldsymbol{i}\#_2$ into $\psi(\boldsymbol{i})$. But we also have to be careful not to introduce any unwanted relations and to keep the underlying free semigroup structure intact.

Without loss of generality, we may assume $|I| = n \geq 1$, $|\Lambda| \geq 2$ and $I \cap \Lambda = \emptyset$. In the following, we let $L = \max\{|\varphi(i)|, |\psi(i)| \mid i \in I\}$, $\hat{\Lambda} = \cup_{\ell=1}^{L} \Lambda^\ell$, $R = \Lambda \cup I$ and $\hat{R} = \hat{\Lambda} \cup I$.

Definition of $\hat{\mathcal{R}}$. First, we compute a complete $\mathscr{S}$-automaton $\hat{\mathcal{R}}$ with state set $\hat{R}$ generating the free semigroup over R:

▶ **Proposition 3.1.** *On input I, Λ and L, one can compute a complete $\mathscr{S}$-automaton $\hat{\mathcal{R}} = (\hat{R}, \Gamma, \rho)$ with state set $\hat{R} = \hat{\Lambda} \cup I$ (for $\hat{\Lambda} = \cup_{\ell=1}^{L} \Lambda^\ell$) and $\mathscr{S}(\hat{\mathcal{R}}) \simeq R^+ = (\Lambda \cup I)^+$ (where the isomorphism is given by $\hat{\lambda} \mapsto \hat{\lambda}$ for all $\hat{\lambda} \in \hat{\Lambda}$ and $i \mapsto i$ for all $i \in I$).*

Proof. Let $\mathcal{R}_1$ be an $\mathscr{S}$-automaton with state set Λ generating the free semigroup Λ^+ (see Fact 2.6) and let $\hat{\mathcal{R}}_1 = \bigcup_{\ell=1}^{L} \mathcal{R}_1^\ell$ be the union of the first L powers of $\mathcal{R}_1$. Note that the state set of $\hat{\mathcal{R}}_1$ is $\hat{\Lambda} = \cup_{\ell=1}^{L} \Lambda^\ell$ and that we still have $\mathscr{S}(\hat{\mathcal{R}}_1) \simeq \Lambda^+$ (where an isomorphism is induced by $\hat{\Lambda} \ni \hat{\lambda} \mapsto \hat{\lambda} \in \Lambda^+$). Now, we may apply Proposition 2.7 sequentially for every element of $I = \{1, \dots, n\}$, which yields the sought automaton $\hat{\mathcal{R}}$ with state set $\hat{R} = \hat{\Lambda} \cup I$ whose generated semigroup is isomorphic to $\Lambda^+ \star \bigstar_{i \in I} i^+ = \Lambda^+ \star I^+ = (\Lambda \cup I)^+$. ◀

[2] Post's statement of the problem [35] is equivalent to ours. In particular, we may assume $\varphi(i), \psi(i) \neq \varepsilon$.
[3] It is worth mentioning that we may assume I to only contain five elements [32] and Λ to be a binary alphabet (using standard encoding techniques). Note that we may only allow non-empty entries, however.

The states in $\hat{R}$ of $\hat{\mathcal{R}}$ do not form a basis of the free semigroup. To simplify working with this fact, we make the following definition(s).

▶ **Definition 3.2** (natural projection). *There is a natural projection $\pi : \hat{\Lambda}^* \to \Lambda^*$ where $\hat{\Lambda} = \bigcup_{\ell=1}^{L} \Lambda^\ell$, which interprets a letter $\hat{\lambda} \in \hat{\Lambda}$ as the corresponding word over Λ. We extend this projection into a homomorphism $\pi : \hat{R}^* \to R^*$ by setting $\pi(i) = i$ for all $i \in I$. Two elements $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2 \in \hat{R}^*$ are R-*equivalent *(written as $\hat{\boldsymbol{r}}_1 =_R \hat{\boldsymbol{r}}_2$) if $\pi(\hat{\boldsymbol{r}}_1) = \pi(\hat{\boldsymbol{r}}_2)$. Finally, $|\hat{\boldsymbol{r}}|_R$ for $\boldsymbol{r} \in \hat{R}^*$ is $|\hat{\boldsymbol{r}}|_R = |\pi(\hat{\boldsymbol{r}})|$.*

Note that we have $\hat{\boldsymbol{r}}_1 =_R \hat{\boldsymbol{r}}_2$ if and only if $\hat{\boldsymbol{r}}_1 =_{\hat{\mathcal{R}}} \hat{\boldsymbol{r}}_2$ for all $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2 \in \hat{R}^*$ as $\mathscr{S}(\hat{\mathcal{R}}) \simeq R^+$.

Definition of $\mathcal{S}$. We use the automaton $\hat{\mathcal{R}} = (\hat{R}, \Gamma, \rho)$ as a building block for our target automaton $\mathcal{T} = (Q, \Sigma, \delta)$ for the reduction. We fix some arbitrary element $\lambda_\# \in \Lambda \subseteq \hat{R}$. To compute $\mathcal{S}$ from $\hat{\mathcal{R}}$, we duplicate the state $\lambda_\#$ twice and call these copies $\#_1$ and $\#_2$. Formally, we have $\mathcal{S} = (Q, \Gamma, \sigma)$ where $Q = \hat{R} \uplus \{\#_1, \#_2\}$ for the new symbols $\#_1$ and $\#_2$ and $\sigma = \rho \cup \{\#_1 \xrightarrow{c/d} q, \#_2 \xrightarrow{c/d} q \mid \lambda_\# \xrightarrow{c/d} q \in \rho\}$. Thus, the new states $\#_1$ and $\#_2$ act in the same way as $\lambda_\#$ and we have $\mathscr{S}(\mathcal{S}) = \mathscr{S}(\hat{\mathcal{R}}) \simeq R^+$.

Definition of $\mathcal{T}$. The next step is to fix another $\lambda_R \in \Lambda \subseteq Q$ arbitrarily but different to $\lambda_\#$ and take $\mathcal{T}_1 = (Q, \Gamma \cup \{a, b\}, \delta_1) = \mathcal{S} \cup \mathcal{T}_1'$ where $\mathcal{T}_1'$ is given via its dual in Figure 4 (i. e. we add two new letters a, b to the alphabet and some additional transitions). Note that we have the transitions $\lambda_\# \xrightarrow{a/a} \lambda_R$ and the self-loops $\lambda_R^\ell \xrightarrow{a/a} \lambda_R^\ell$ for all $1 \leq \ell \leq L$ in $\mathcal{T}_1$.

The idea for this part is that we may factorize a state sequence $\boldsymbol{q} \in Q^*$ into blocks from $\hat{R}^*$ and symbols $\#_1$ and $\#_2$ and then remove the blocks one after another using the letter a. We will explain this precisely later in Fact 3.3.

Finally, we let $\mathcal{T} = (Q, \Sigma, \delta) = \mathcal{T}_1 \cup \mathcal{T}_2$ where $\mathcal{T}_2$ is given via its dual in Figure 5. Note, in particular, that we have $\varphi(i), \psi(i) \in \bigcup_{\ell=1}^{L} \Lambda^\ell = \hat{\Lambda} \subseteq \hat{R}$.

In other words, we obtain $\mathcal{T}$ from $\mathcal{T}_1$ by adding new symbols to the alphabet resulting in $\Sigma = \Gamma \cup \{a, b\} \cup \{\iota, \alpha, \alpha', f_\alpha, \beta, \beta', f_\beta, f\}$ and adding the transitions depicted in Figure 5 for all $i \in I$ and $\hat{\lambda} \in \hat{\Lambda}$. Clearly, $\mathcal{T}$ can be computed and is a complete $\mathscr{S}$-automaton.

The Role of a and b in $\mathcal{T}$. As already mentioned above, we may use the letter a to remove a block from a certain factorization of a state sequence (the proof is by induction on μ):

▶ **Fact 3.3.** *Let $\boldsymbol{p} \in Q^*$ and factorize it as $\boldsymbol{p} = (\boldsymbol{p}_s \#_{x_s}) \dots (\boldsymbol{p}_1 \#_{x_1})\, \boldsymbol{p}_0$ for $\boldsymbol{p}_0, \dots, \boldsymbol{p}_s \in \hat{R}^*$ and $x_1, \dots, x_s \in \{1, 2\}$. Then, for any $1 \leq \mu \leq s$, we have (in $\mathcal{T}$):*

$$\boldsymbol{p} \cdot a^\mu = (\boldsymbol{p}_s \#_{x_s}) \dots (\boldsymbol{p}_{\mu+1} \#_{x_{\mu+1}})\, \boldsymbol{p}_\mu \, \lambda_\# \lambda_R^{\mu-1+|\boldsymbol{p}_{\mu-1} \dots \boldsymbol{p}_0|_R}$$

Correctness. It remains to show that the `PCP` instance φ, ψ, I has a solution if and only if $\mathscr{S}(\mathcal{T})$ is **not** a free semigroup. We start with the (easier) "only if" direction and show that the additional transitions from $\mathcal{T}_1$ and $\mathcal{T}_2$ do not affect the subautomaton $\hat{\mathcal{R}}$: if two state sequences are R-equivalent, they are also equal with respect to $\mathcal{T}$.

▶ **Lemma 3.4.** *Let $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2 \in \hat{R}^*$ with $\hat{\boldsymbol{r}}_1 =_R \hat{\boldsymbol{r}}_2$. Then, we have $\hat{\boldsymbol{r}}_1 =_{\mathcal{T}} \hat{\boldsymbol{r}}_2$.*

Proof Sketch. We need to show $\hat{\boldsymbol{r}}_1 \circ u = \hat{\boldsymbol{r}}_2 \circ u$ for all $u \in \Sigma^*$ and this can be done by induction on u. Thus, write $u = cu'$ for some $c \in \Sigma = \Gamma \cup \{a, b\} \cup \{\iota, \alpha, \alpha', f_\alpha, \beta, \beta', f_\beta, f\}$ and $u' \in \Sigma^*$.

Most cases for c are straight-forward (for example, for $c \in \Gamma$ – the alphabet of $\hat{\mathcal{R}}$ – we inherit this property from $\hat{\mathcal{R}}$) and we only demonstrate the case $c \in \{\alpha, \alpha', \beta, \beta'\}$. Here, we factorize $\hat{\boldsymbol{r}}_1 = \hat{\boldsymbol{s}}_1 \hat{\lambda}_1 \boldsymbol{i}_1$ with $\boldsymbol{i}_1 \in I^*$ maximal, $\hat{\lambda}_1 \in \hat{\Lambda} \cup \{\varepsilon\}$ and $\hat{\boldsymbol{s}}_1 \in \hat{R}^*$ with

Table 1 Various symbols in the order of their definition.

symbol	usage
Λ :	PCP base alphabet, $\lvert\Lambda\rvert \geq 2$
I :	PCP index set, $\lvert I\rvert \geq 1$, $I \cap \Lambda = \emptyset$
$\varphi, \psi : I \to \Lambda^+$	PCP homomorphisms
$L =$	$\max\{\lvert\varphi(i)\rvert, \lvert\psi(i)\rvert \mid i \in I\}$
$\hat{\Lambda} =$	$\bigcup_{\ell=1}^{L} \Lambda^{\ell}$
$R =$	$\Lambda \cup I$
$\hat{R} =$	$\hat{\Lambda} \cup I$: state set of $\mathcal{R}$
$\hat{\mathcal{R}} =$	$(\hat{R}, \Gamma, \rho)$: complete $\mathscr{S}$-automaton generating $R^+ = (\Lambda \cup I)^+$
ρ :	transition set of $\hat{\mathcal{R}}$
Γ :	alphabet of $\hat{\mathcal{R}}$ and $\mathcal{S}$
π :	$\hat{\Lambda}^* \to \Lambda$, $\hat{R}^* \to R^*$ natural projection with $\pi(i) = i$ for all $i \in I$
$\lvert\hat{\boldsymbol{r}}\rvert_R$:	length of $\pi(\hat{\boldsymbol{r}})$ for $\hat{\boldsymbol{r}} \in \hat{R}^*$
$\lambda_\# \in \Lambda \subseteq \hat{R}$:	arbitrarily chosen element
$\#_1, \#_2$:	copies of $\lambda_\#$
$\mathcal{S} =$	(Q, Γ, σ) : complete $\mathscr{S}$-automaton, extension of $\hat{\mathcal{R}}$ still generating R^+
$Q =$	$\hat{R} \uplus \{\#_1, \#_2\}$: state set of $\mathcal{S}$ and $\mathcal{T}$
σ :	transition set of $\mathcal{S}$
$\lambda_R \in \Lambda \subseteq Q$:	arbitrarily chosen element with $\lambda_R \neq \lambda_\#$
$a, b \notin \Gamma$:	new letters for $\mathcal{T}_1$
$\mathcal{T}_1' =$	$(Q, \{a, b\}, \delta_1')$: complete $\mathscr{S}$-automaton, additional transitions for $\mathcal{T}_1$, see Figure 4
$\mathcal{T}_1 =$	$(Q, \Gamma \uplus \{a, b\}, \delta_1) = \mathcal{S} \cup \mathcal{T}_1'$: complete $\mathscr{S}$-automaton, extension of $\mathcal{S}$ by $\mathcal{T}_1'$
δ_1 :	transition set of $\mathcal{T}_1$
$\mathcal{T} =$	$(Q, \Sigma, \delta) = \mathcal{T}_1 \cup \mathcal{T}_2$: complete $\mathscr{S}$-automaton with $e =_{\mathcal{T}} \varepsilon$, result of the reduction
$\mathcal{T}_2$:	complete $\mathscr{S}$-automaton with new transitions for $\mathcal{T}$, see Figure 5
$\Sigma =$	$\Gamma \cup \{a, b\} \cup \{\iota, \alpha, \alpha', f_\alpha, \beta, \beta', f_\beta, f\}$: alphabet of $\mathcal{T}$
$\pi_\#$:	$Q^* \to \{\#_1, \#_2\}^*$ homomorphism with $\pi_\#(\#_x) = \#_x$ but $\pi_\#(\hat{r}) = \varepsilon$ for $\hat{r} \in \hat{R}$
π' :	$Q^* \to (R \cup \{\#_1, \#_2\})^*$ homomorphism extending π with $\pi'(\#_x) = \#_x$ for $x \in \{1, 2\}$

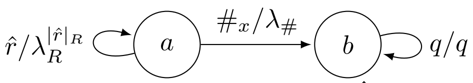


Figure 4 The dual $\partial\mathcal{T}_1'$. The transitions exist for all $\hat{r} \in \hat{R}$, $x \in \{1, 2\}$ and $q \in Q$.

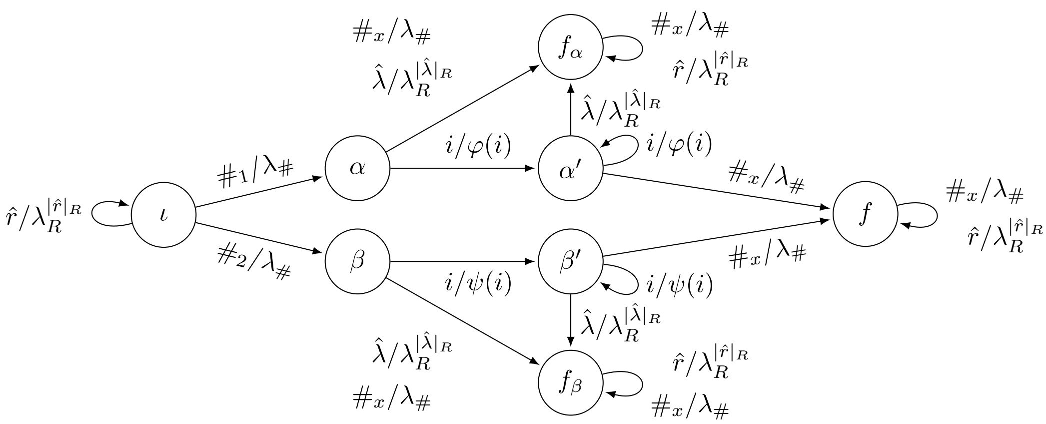
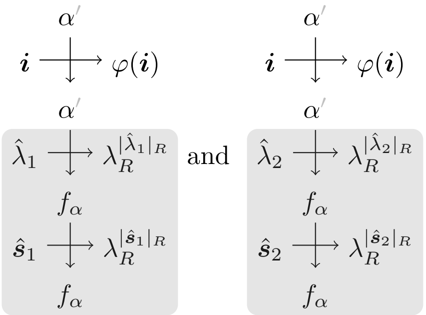
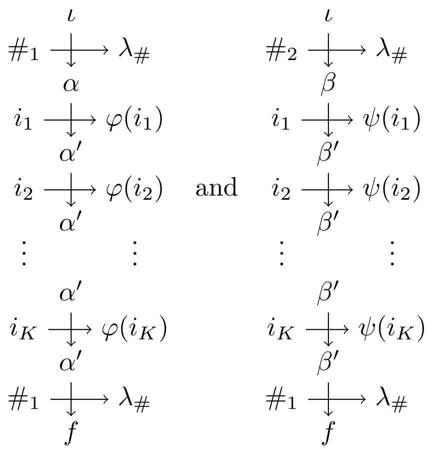


Figure 5 The dual $\partial\mathcal{T}_2$. The transitions exist for all $i \in I, \hat{r} \in \hat{R}, \hat{\lambda} \in \hat{\Lambda}$ and $x \in \{1, 2\}$.



■ **Figure 6** Cross diagrams for Lemma 3.4.



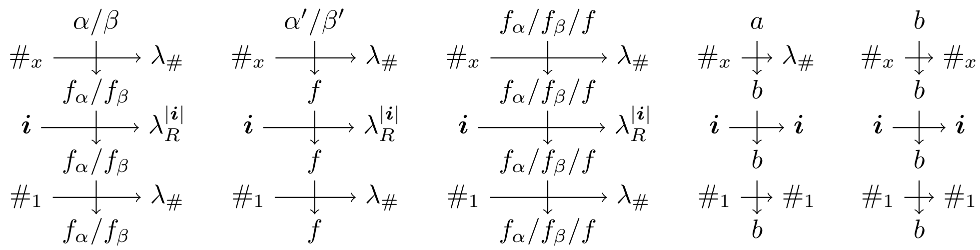
■ **Figure 7** Cross diagrams for Lemma 3.5.

$\lambda_1 = \varepsilon \implies \hat{\boldsymbol{s}}_1 = \varepsilon$. Analogously, we factorize $\hat{\boldsymbol{r}}_2 = \hat{\boldsymbol{s}}_2\hat{\lambda}_2\boldsymbol{i}_2$. Observe that, since we have $\hat{\boldsymbol{r}}_1 =_R \hat{\boldsymbol{r}}_2$, we must have $\boldsymbol{i}_1 = \boldsymbol{i}_2 = \boldsymbol{i}$, $\hat{\boldsymbol{s}}_1\hat{\lambda}_1 =_R \hat{\boldsymbol{s}}_2\hat{\lambda}_2$ and $\hat{\lambda}_1 = \varepsilon \iff \hat{\lambda}_2 = \varepsilon$. This yields the cross diagrams in Figure 6 where the shaded parts only exist if $\hat{\lambda}_1, \hat{\lambda}_2 \neq \varepsilon$ and where we have α' after applying $\boldsymbol{i}$ if $\boldsymbol{i} \neq \varepsilon$. In both diagrams, we have the same state sequence on the right hand side (because of $\hat{\boldsymbol{s}}_1\hat{\lambda}_1 =_R \hat{\boldsymbol{s}}_2\hat{\lambda}_2$) and, thus, are done. The case $c \in \{\beta, \beta'\}$ is analogous (using ψ). ◀

Finally, we show that a solution for the `PCP` instance implies a proper relation in the semigroup generated by $\mathcal{T}$ and, thus, that it is not free.

▶ **Lemma 3.5.** *If $\boldsymbol{i} \in I^+$ is a solution for the* `PCP` *instance, then we have $\#_1\boldsymbol{i}\#_1 =_{\mathcal{T}} \#_1\boldsymbol{i}\#_2$.*

Proof Sketch. We show $\#_1\boldsymbol{i}\#_1 \circ u = \#_1\boldsymbol{i}\#_2 \circ u$ for all $u \in \Sigma^*$. For $u = \varepsilon$, there is nothing to show. So, let $u = cu'$ for some $c \in \Sigma = \Gamma \cup \{a, b\} \cup \{\iota, \alpha, \alpha', f_\alpha, \beta, \beta', f_\beta, f\}$ and $u' \in \Sigma^*$. Again, we only fully demonstrate the most interesting case $c = \iota$ (the other cases may be found in Figure 8 where Figure 8e requires induction). Writing $\boldsymbol{i} = i_K \dots i_2 i_1$ for $i_1, \dots, i_K \in I$, we obtain the diagrams in Figure 7. Since $\boldsymbol{i} = i_K \dots i_2 i_1$ is a solution, we have $\varphi(i_K) \dots \varphi(i_2)\varphi(i_1) =_R \psi(i_K) \dots \psi(i_2)\psi(i_1)$. Thus, Lemma 3.4 implies $\lambda_\#\varphi(i_K) \dots \varphi(i_2)\varphi(i_1)\lambda_\# =_{\mathcal{T}} \lambda_\#\psi(i_K) \dots \psi(i_2)\psi(i_1)\lambda_\#$. ◀



(a) $c \in \{\alpha, \beta\}$ **(b)** $c \in \{\alpha', \beta'\}$ **(c)** $c \in \{f_\alpha, f_\beta, f\}$ **(d)** $c = a$ **(e)** $c = b$

■ **Figure 8** Various cases for $c \in \Sigma$. The cross diagrams hold for $x \in \{1, 2\}$.

▶ **Proposition 3.6.** *If the* PCP *instance has a solution, $\mathscr{S}(\mathcal{T})$ is not (left) cancellative and, thus, not a free semigroup.*

Converse Direction. To show that the PCP instance has a solution if the semigroup is not free, we introduce the notion of compatibility and observe that every relation is compatible. The proof relies on Fact 3.3 for removing blocks from the factorization used in Definition 3.7 and that $\mathcal{S}$ (generating R^+) survives as a subautomaton of $\mathcal{T}$. The latter allows us to use the cancellativity of R^+ to show that the individual blocks are the same in R^+.

▶ **Definition 3.7** (compatible state sequences). *Factorize $\boldsymbol{p}, \boldsymbol{q} \in Q^*$ (uniquely) as $\boldsymbol{p} = (\boldsymbol{p}_s \#_{x_s}) \dots (\boldsymbol{p}_1 \#_{x_1}) \boldsymbol{p}_0$ and $\boldsymbol{q} = (\boldsymbol{q}_t \#_{y_t}) \dots (\boldsymbol{q}_1 \#_{y_1}) \boldsymbol{q}_0$ with $\boldsymbol{p}_0, \dots, \boldsymbol{p}_s, \boldsymbol{q}_0, \dots, \boldsymbol{q}_t \in \hat{R}^*$ and $x_1, \dots, x_s, y_1, \dots, y_t \in \{1, 2\}$. They are* compatible *if $s = t$ and $\forall\, 0 \leq i \leq s = t : \boldsymbol{p}_i =_R \boldsymbol{q}_i$.*

▶ **Lemma 3.8.** *Let $\boldsymbol{p}, \boldsymbol{q} \in Q^*$ with $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$. Then, we have that $\boldsymbol{p}$ and $\boldsymbol{q}$ are compatible.*

Proof. We factorize $\boldsymbol{p}$ and $\boldsymbol{q}$ in the same way as in Definition 3.7 and show the statement by induction on $s + t$. For $s = t = 0$, we have $\boldsymbol{p}_0 = \boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q} = \boldsymbol{q}_0$. Since $\hat{\mathcal{R}}$ is a subautomaton of $\mathcal{T}$, this implies $\boldsymbol{p}_0 =_{\hat{\mathcal{R}}} \boldsymbol{q}_0$ and, equivalently, $\boldsymbol{p} = \boldsymbol{p}_0 =_R \boldsymbol{q}_0 = \boldsymbol{q}$.

For the inductive step ($s + t > 0$), we may assume $s > 0$ (due to symmetry) or, in other words, that $\boldsymbol{p}$ contains at least one $\#_1$ or $\#_2$. We have $\boldsymbol{p} \circ a = b$ (compare to Figure 4) and, thus, due to $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$, also $\boldsymbol{q} \circ a = \boldsymbol{p} \circ a = b$. This is only possible (again, compare to Figure 4) if $\boldsymbol{q}$ also contains at least one $\#_1$ or $\#_2$, i. e. if $t > 0$.

From Fact 3.3 (with $\mu = 1$), we obtain (for both $\boldsymbol{p}$ and $\boldsymbol{q}$):

$$\boldsymbol{p} \cdot a = \boldsymbol{p}' \lambda_\# \lambda_R^{|\boldsymbol{p}_0|_R}$$
$$\text{for } \boldsymbol{p}' = (\boldsymbol{p}_s \#_{x_s}) \dots (\boldsymbol{p}_2 \#_{x_2}) \boldsymbol{p}_1 \text{ and}$$
$$\boldsymbol{q} \cdot a = \boldsymbol{q}' \lambda_\# \lambda_R^{|\boldsymbol{q}_0|_R}$$
$$\text{for } \boldsymbol{q}' = (\boldsymbol{q}_t \#_{x_t}) \dots (\boldsymbol{q}_2 \#_{x_2}) \boldsymbol{q}_1$$

Now, $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$ implies $\boldsymbol{p}' \lambda_\# \lambda_R^{|\boldsymbol{p}_0|_R} = \boldsymbol{p} \cdot a =_{\mathcal{T}} \boldsymbol{q} \cdot a = \boldsymbol{q}' \lambda_\# \lambda_R^{|\boldsymbol{q}_0|_R}$ and we may apply the induction hypothesis, which yields that $\boldsymbol{p}' \lambda_\# \lambda_R^{|\boldsymbol{p}_0|_R}$ and $\boldsymbol{q}' \lambda_\# \lambda_R^{|\boldsymbol{q}_0|_R}$ are compatible. This means that we have $s = t$, $\boldsymbol{p}_\mu =_R \boldsymbol{q}_\mu$ for all $2 \leq \mu \leq s = t$ and $\boldsymbol{p}_1 \lambda_\# \lambda_R^{|\boldsymbol{p}_0|_R} =_R \boldsymbol{q}_1 \lambda_\# \lambda_R^{|\boldsymbol{q}_0|_R}$. Observe that the latter implies $\boldsymbol{p}_1 =_R \boldsymbol{q}_1$ (as we have chosen $\lambda_\#$ and λ_R as different elements of Λ). In particular, we also obtain $\boldsymbol{p}_s \lambda_\# \boldsymbol{p}_{s-1} \dots \lambda_\# \boldsymbol{p}_1 =_R \boldsymbol{q}_t \lambda_\# \boldsymbol{q}_{t-1} \dots \lambda_\# \boldsymbol{q}_1$.

Since $\mathcal{S}$ is a subautomaton of $\mathcal{T}$, $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$ implies $\boldsymbol{p} =_{\mathcal{S}} \boldsymbol{q}$. As $\#_1$ and $\#_2$ act in the same way as $\lambda_\#$ in $\mathcal{S}$ by construction, this shows $\boldsymbol{p}_s \lambda_\# \dots \boldsymbol{p}_1 \lambda_\# \boldsymbol{p}_0 =_{\mathcal{S}} \boldsymbol{q}_t \lambda_\# \dots \boldsymbol{q}_1 \lambda_\# \boldsymbol{q}_0$ and, because of $\mathscr{S}(\mathcal{S}) \simeq R^+$, also $\boldsymbol{p}_s \lambda_\# \dots \boldsymbol{p}_1 \lambda_\# \boldsymbol{p}_0 =_R \boldsymbol{q}_t \lambda_\# \dots \boldsymbol{q}_1 \lambda_\# \boldsymbol{q}_0$. Now, because R^* as a free monoid is cancellative (see Fact 2.1) and because we have $\boldsymbol{p}_s \lambda_\# \boldsymbol{p}_{s-1} \dots \lambda_\# \boldsymbol{p}_1 =_R \boldsymbol{q}_t \lambda_\# \boldsymbol{q}_{t-1} \dots \lambda_\# \boldsymbol{q}_1$ (from above), we obtain $\lambda_\# \boldsymbol{p}_0 =_R \lambda_\# \boldsymbol{q}_0$ and, finally, $\boldsymbol{p}_0 =_R \boldsymbol{q}_0$, which concludes the proof that $\boldsymbol{p}$ and $\boldsymbol{q}$ are compatible. ◀

On the other hand, not every compatible pair forms a semigroup relation. However, this is true by Lemma 3.4 if, additionally, the subsequence containing only $\#_1$ and $\#_2$ is the same in both entries. To formalize this, we introduce the following definition.

▶ **Definition 3.9** (projection on $\{\#_1, \#_2\}$). *Let $\pi_\# : Q^* \to \{\#_1, \#_2\}^*$ be the homomorphism given by $\pi_\#(\#_x) = \#_x$ for both $x \in \{1, 2\}$ and $\pi_\#(\hat{r}) = \varepsilon$ for all other $\hat{r} \in Q \setminus \{\#_1, \#_2\} = \hat{R}$.*

▶ **Lemma 3.10.** *Let $\boldsymbol{p}, \boldsymbol{q} \in Q^*$ be compatible with $\pi_\#(\boldsymbol{p}) = \pi_\#(\boldsymbol{q})$. Then, we have $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$.*

Combining the last two lemmas, we obtain that $\mathscr{S}(\mathcal{T})$ is a free semigroup if all its relations have the same projection under $\pi_\#$. Most importantly, we will later on apply the contraposition of the “only if” direction of the following lemma to obtain a relation with different images under the projection if the semigroup is not free.

▶ **Lemma 3.11.** *Let $\pi' : Q^* \to (R \cup \{\#_1, \#_2\})^*$ be the extension of the natural projection π (from Definition 3.2) with $\pi'(\#_x) = \#_x$ for $x \in \{1, 2\}$. The following are equivalent:*
1. *For all $\boldsymbol{p}, \boldsymbol{q} \in Q^+$ with $\boldsymbol{p} =_\mathcal{T} \boldsymbol{q}$, we have $\pi_\#(\boldsymbol{p}) = \pi_\#(\boldsymbol{q})$.*
2. *The map π' induces a well-defined homomorphism $\mathscr{S}(\mathcal{T}) \to (R \cup \{\#_1, \#_2\})^+$.*
3. *The map π' induces a well-defined isomorphism $\mathscr{S}(\mathcal{T}) \to (R \cup \{\#_1, \#_2\})^+$.*

In particular, $\mathscr{S}(\mathcal{T})$ is isomorphic to $(R \cup \{\#_1, \#_2\})^+$ if we have $\pi_\#(\boldsymbol{p}) = \pi_\#(\boldsymbol{q})$ for all $\boldsymbol{p}, \boldsymbol{q} \in Q^+$ with $\boldsymbol{p} =_\mathcal{T} \boldsymbol{q}$.

Now a relation whose sides have different images under $\pi_\#$ yields a `PCP` solution.

▶ **Lemma 3.12.** *If there are $\boldsymbol{p}, \boldsymbol{q} \in Q^+$ with $\boldsymbol{p} =_\mathcal{T} \boldsymbol{q}$ but $\pi_\#(\boldsymbol{p}) \neq \pi_\#(\boldsymbol{q})$, then the* `PCP` *instance has a solution.*

Proof. We factorize these $\boldsymbol{p}$ and $\boldsymbol{q}$ in the same way as in Definition 3.7 and observe that $\boldsymbol{p}$ and $\boldsymbol{q}$ are compatible by Lemma 3.8. We may assume that there is some $1 \leq \mu_0 \leq s = t$ with $\#_{x_{\mu_0}} = \#_1$ but $\#_{y_{\mu_0}} = \#_2$ (due to symmetry).

We may assume $\mu_0 = 1$ without loss of generality. This is because we may substitute $\boldsymbol{p}$ by $\boldsymbol{p}' = \boldsymbol{p} \cdot a^{\mu_0 - 1}$ and $\boldsymbol{q}' = \boldsymbol{q} \cdot a^{\mu_0 - 1}$ (we still have $\boldsymbol{p}' =_\mathcal{T} \boldsymbol{q}'$) by Fact 3.3 (for $\mu_0 > 1$).

With these assumptions, we apply $\boldsymbol{p}$ and $\boldsymbol{q}$ to ι and obtain (see Figure 5) the cross diagrams depicted in Figure 9 for $\tilde{\boldsymbol{p}} = \boldsymbol{p}_s \#_{x_s} \dots \boldsymbol{p}_3 \#_{x_3} \boldsymbol{p}_2$, $\tilde{\boldsymbol{q}} = \boldsymbol{q}_t \#_{y_t} \dots \boldsymbol{q}_3 \#_{y_3} \boldsymbol{q}_2$ and some $\boldsymbol{p}_1', \tilde{\boldsymbol{p}}', \boldsymbol{q}_1', \tilde{\boldsymbol{q}}' \in Q^*$, $p_2', q_2' \in Q$ and $c_1, c_2, c, d_1, d_2, d \in \Gamma$. Since we have $\boldsymbol{p} =_\mathcal{T} \boldsymbol{q}$, we must have $c = d$ and, by the construction of $\mathcal{T}$, this is only possible if $c = f = d$ (see Figure 5). This, in turn, is only possible if we have $\boldsymbol{p}_1 = \boldsymbol{i} \in I^+$ and $\boldsymbol{q}_1 = \boldsymbol{j} \in I^+$. Since $\boldsymbol{p}$ and $\boldsymbol{q}$ are compatible, we must even have $\boldsymbol{i} = \boldsymbol{p}_1 =_R \boldsymbol{q}_1 = \boldsymbol{j}$, which implies $\boldsymbol{i} = \boldsymbol{j}$. Additionally, we also obtain $\boldsymbol{p}_1' =_R \varphi(\boldsymbol{i})$, $c_1 = \alpha'$, $p_2' = \lambda_\#$, $c_2 = f$, $\boldsymbol{q}_1' =_R \psi(\boldsymbol{i})$, $d_1 = \beta'$, $q_2' = \lambda_\#$, $d_2 = f$ and $\tilde{\boldsymbol{p}}' = \lambda_R^{|\boldsymbol{p}_s|_R} \lambda_\# \dots \lambda_R^{|\boldsymbol{p}_3|_R} \lambda_\# \lambda_R^{|\boldsymbol{p}_2|_R}$ as well as $\tilde{\boldsymbol{q}}' = \lambda_R^{|\boldsymbol{q}_t|_R} \lambda_\# \dots \lambda_R^{|\boldsymbol{q}_3|_R} \lambda_\# \lambda_R^{|\boldsymbol{q}_2|_R}$ from the construction of $\mathcal{T}$. This shows

$$\begin{aligned} &\lambda_R^{|\boldsymbol{p}_s|_R} \lambda_\# \dots \lambda_R^{|\boldsymbol{p}_3|_R} \lambda_\# \lambda_R^{|\boldsymbol{p}_2|_R} \lambda_\# \varphi(\boldsymbol{i}) \lambda_\# \lambda_R^{|\boldsymbol{p}_0|_R} \\ =_\mathcal{T} &\lambda_R^{|\boldsymbol{q}_t|_R} \lambda_\# \dots \lambda_R^{|\boldsymbol{q}_3|_R} \lambda_\# \lambda_R^{|\boldsymbol{q}_2|_R} \lambda_\# \psi(\boldsymbol{i}) \lambda_\# \lambda_R^{|\boldsymbol{q}_0|_R} \end{aligned}$$

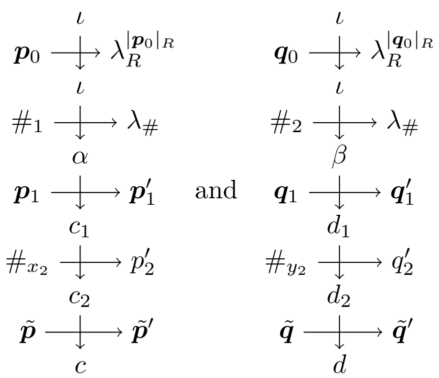


Figure 9 Cross diagrams for Lemma 3.12.

and, by Lemma 3.8, also that both sides are R-equivalent. Since $\boldsymbol{p}$ and $\boldsymbol{q}$ are compatible, we have $\lambda_R^{|\boldsymbol{p}_\mu|_R} =_R \lambda_R^{|\boldsymbol{q}_\mu|_R}$ for all $0 \leq \mu \leq s = t$. Combining this with the cancellativity of R^*, we obtain $\varphi(\boldsymbol{i}) =_R \psi(\boldsymbol{i})$ and, thus, that $\boldsymbol{i}$ is a solution for the PCP instance. ◀

We have now shown that the PCP instance has a solution if the semigroup generated by $\mathcal{T}$ is not free. A careful analysis of the proof yields more, however, which we collect in Proposition 3.14 (which follows from the lemmas and propositions above). For one part of this statement, we will first state another consequence of Lemma 3.8:

▶ **Proposition 3.13.** *Mapping $\hat{r}$ to $|\hat{r}|_R$ for every $\hat{r} \in \hat{R}$ and $\#_x$ to 1 for $x \in \{1, 2\}$ induces a well-defined proper length function of $\mathscr{M}(\mathcal{T})$ (and a well-defined length function of $\mathscr{S}(\mathcal{T})$).*

▶ **Proposition 3.14.** *The following statements are equivalent:*

1. *The* PCP *instance has a solution $\boldsymbol{i} \in I^+$.*
2. *We have $\#_1\boldsymbol{i}\#_1 =_{\mathcal{T}} \#_1\boldsymbol{i}\#_2$ for some $\boldsymbol{i} \in I^+$.*
3. *There are $\boldsymbol{p}, \boldsymbol{q} \in Q^+$ with $\boldsymbol{p} =_{\mathcal{T}} \boldsymbol{q}$ but $\pi_\#(\boldsymbol{p}) \neq \pi_\#(\boldsymbol{q})$.*
4. *$\mathscr{S}(\mathcal{T})$ is not a free semigroup.*
5. *$\mathscr{S}(\mathcal{T})$ is not isomorphic to $(R \cup \{\#_1, \#_2\})^+$.*
6. *$\mathscr{S}(\mathcal{T})$ is not (left[4]) cancellative.*
7. *$\mathscr{S}(\mathcal{T})$ is not equidivisible.*

4'. *$\mathscr{M}(\mathcal{T})$ is not a free monoid.*
5'. *$\mathscr{M}(\mathcal{T})$ is not isomorphic to $(R \cup \{\#_1, \#_2\})^*$.*
6'. *$\mathscr{M}(\mathcal{T})$ is not (left) cancellative.*
7'. *$\mathscr{M}(\mathcal{T})$ is not equidivisible.*

Main Theorem and other Consequences. Proposition 3.14 shows that we have reduced PCP to (the complements of) the freeness problem for (complete) automaton semigroups and monoids (as the construction of $\mathcal{T}$ is computable). Since PCP is undecidable [35], we obtain:

▶ **Theorem 3.15.** *The freeness problem for automaton semigroups*

Input: *a (complete) $\mathscr{S}$-automaton $\mathcal{T}$*
Question: *is $\mathscr{S}(\mathcal{T})$ a free semigroup?*

and the freeness problem for automaton monoids

Input: *a (complete) $\mathscr{S}$-automaton $\mathcal{T}$*
Question: *is $\mathscr{M}(\mathcal{T})$ a free monoid?*

are undecidable.

▶ **Theorem 3.16.** *The following problems are undecidable:*

Input: *a complete $\mathscr{S}$-automaton $\mathcal{T}$*
Question: *is $\mathscr{S}(\mathcal{T})$ (left) cancellative/equidivisible?*

Input: *a complete $\mathscr{S}$-automaton $\mathcal{T}$*
Question: *is $\mathscr{M}(\mathcal{T})$ (left) cancellative/equidivisible?*

Finally, we obtain that it is undecidable whether a given map on the generators induces a homomorphism (or an isomorphism) between two automaton semigroups (using $\mathcal{T}$ from above as $\mathcal{T}_1$ and an automaton generating $(R \cup \{\#_1, \#_2\})^+$ for $\mathcal{T}_2$). Note that the isomorphism problem for automaton groups (and, thus, also for automaton semigroups and monoids) is known to be undecidable (as it follows from [40]).

[4] Recall that we defined automaton semigroups by a left action here.

▶ **Theorem 3.17.** *The following two problems are undecidable:*

Input: *two (complete)* $\mathscr{S}$*-automata* $\mathcal{T}_1 = (Q_1, \Sigma_1, \delta_1)$ *and* $\mathcal{T}_2 = (Q_2, \Sigma_2, \delta_2)$ *and a map* $f : Q_1 \to Q_2$
Question: *does* f *extend into a homomorphism* $\mathscr{S}(\mathcal{T}_1) \to \mathscr{S}(\mathcal{T}_2)$*?*

Input: *two (complete)* $\mathscr{S}$*-automata* $\mathcal{T}_1 = (Q_1, \Sigma_1, \delta_1)$ *and* $\mathcal{T}_2 = (Q_2, \Sigma_2, \delta_2)$ *and a map* $f : Q_1 \to Q_2$
Question: *does* f *extend into an isomorphism* $\mathscr{S}(\mathcal{T}_1) \to \mathscr{S}(\mathcal{T}_2)$*?*

For our construction, we need that all $\varphi(i)$ and $\psi(i)$ are states in the automaton. This immediately yields relations of the form $u\,v =_{\hat{\mathcal{R}}} uv$ for $u, v, uv \in \hat{\Lambda}$ that still exist in the eventual automaton $\mathcal{T}$. In the monoid case, however, we may use the neutral element as a "padding symbol" and thus avoid using a power automaton. This then yields:

▶ **Theorem 3.18.** *The free presentation problem for automaton monoids is undecidable:*

Input: *a (complete)* $\mathscr{S}$*-automaton* $\mathcal{T} = (Q, \Sigma, \delta)$ *with a dedicated state* $e \in Q$ *acting as the identity map*
Question: *is* $\mathscr{M}(\mathcal{T}) \simeq (Q \setminus \{e\})^*$*?*

In the semigroup case, we only get a weaker form of this result (using $P = R \cup \{\#_1, \#_2\}$):

▶ **Theorem 3.19.** *The following problem is undecidable:*

Input: *a (complete)* $\mathscr{S}$*-automaton* $\mathcal{T} = (Q, \Sigma, \delta)$ *and a subset* $P \subseteq Q$
Question: *is* $\mathscr{S}(\mathcal{T}) \simeq P^+$*?*

4 Open Problems

Theorem 3.18 immediately raises the question whether the corresponding problem for automaton semigroups is also undecidable:

▶ **Open Problem 4.1.** *Is the following problem decidable?*

Input: *a (complete)* $\mathscr{S}$*-automaton* $\mathcal{T} = (Q, \Sigma, \delta)$
Question: *is* $\mathscr{S}(\mathcal{T}) \simeq Q^+$*?*

In Theorem 3.16, we have also shown that it is not possible to test whether a given automaton semigroup (or monoid) is equidivisible. By Levi's lemma (Fact 2.2) this is one part of a semigroup (monoid) being free while the other one is the existence of a (proper) length function. So, the following question naturally arises.

▶ **Open Problem 4.2.** *Is the following problem decidable?*

Input: *a (complete)* $\mathscr{S}$*-automaton* $\mathcal{T}$
Question: *does* $\mathscr{S}(\mathcal{T})$ *(*$\mathscr{M}(\mathcal{T})$*) admit a (proper) length function?*

We highly suspect this problem to be undecidable and it seems likely that our construction can be adapted to show this.

Of course, it also remains open whether the freeness problem for automaton groups [19, 7.2 b)] is decidable.

References

1 Laurent Bartholdi, Thibault Godin, Ines Klimann, Camille Noûs, and Matthieu Picantin. A new hierarchy for automaton semigroups. *International Journal of Foundations of Computer Science*, 31(08):1069–1089, 2020. doi:10.1142/S0129054120420046.

2 Laurent Bartholdi and Ivan Mitrofanov. The word and order problems for self-similar and automata groups. *Groups, Geometry, and Dynamics*, 14:705–728, 2020. doi:10.4171/GGD/560.

3 Laurent Bartholdi and Pedro Silva. Groups defined by automata. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, volume II, chapter 24, pages 871–911. European Mathematical Society, September 2021.

4 Paul Bell and Igor Potapov. Reachability problems in quaternion matrix and rotation semigroups. *Information and Computation*, 206(11):1353–1361, 2008. doi:10.1016/j.ic.2008.06.004.

5 Ievgen V. Bondarenko. Growth of Schreier graphs of automaton groups. *Mathematische Annalen*, 354(2):765–785, 2012. doi:10.1007/s00208-011-0757-x.

6 Ievgen V. Bondarenko, Natalia V. Bondarenko, Said N. Sidki, and Flavia R. Zapata. On the conjugacy problem for finite-state automorphisms of regular rooted trees. *Groups, Geometry, and Dynamics*, 7:232–355, 2013. doi:10.4171/GGD/184.

7 Tara Brough and Alan J. Cain. Automaton semigroups: New constructions results and examples of non-automaton semigroups. *Theoretical Computer Science*, 674:1–15, 2017. doi:10.1016/j.tcs.2017.02.003.

8 Andrew M. Brunner and Said Sidki. The generation of $\mathrm{GL}(n, \mathbb{Z})$ by finite state automata. *International Journal of Algebra and Computation*, 08(01):127–139, 1998. doi:10.1142/S0218196798000077.

9 Alan J. Cain. Automaton semigroups. *Theoretical Computer Science*, 410(47):5022–5038, 2009. doi:10.1016/j.tcs.2009.07.054.

10 Julien Cassaigne, Tero Harju, and Juhani Karhumäki. On the undecidability of freeness of matrix semigroups. *International Journal of Algebra and Computation*, 09(03n04):295–305, 1999. doi:10.1142/S0218196799000199.

11 Daniele D'Angeli, Dominik Francoeur, Emanuele Rodaro, and Jan Philipp Wächter. Infinite automaton semigroups and groups have infinite orbits. *Journal of Algebra*, 553:119–137, 2020. doi:10.1016/j.jalgebra.2020.02.014.

12 Daniele D'Angeli, Emanuele Rodaro, and Jan Philipp Wächter. Automaton semigroups and groups: On the undecidability of problems related to freeness and finiteness. *Israel Journal of Mathematics*, 237:15–52, 2020. doi:10.1007/s11856-020-1972-5.

13 Daniele D'Angeli, Emanuele Rodaro, and Jan Philipp Wächter. Erratum to "semigroups and groups: On the undecidability of problems related to freeness and finiteness". *Israel Journal of Mathematics*, 245:535–542, 2021. doi:10.1007/s11856-021-2206-1.

14 Daniele D'Angeli, Emanuele Rodaro, and Jan Philipp Wächter. On the complexity of the word problem for automaton semigroups and automaton groups. *Advances in Applied Mathematics*, 90:160–187, 2017. doi:10.1016/j.aam.2017.05.008.

15 Daniele D'Angeli, Emanuele Rodaro, and Jan Philipp Wächter. On the structure theory of partial automaton semigroups. *Semigroup Forum*, pages 51–76, 2020. doi:10.1007/s00233-020-10114-5.

16 Pierre Gillibert. The finiteness problem for automaton semigroups is undecidable. *International Journal of Algebra and Computation*, 24(01):1–9, 2014. doi:10.1142/S0218196714500015.

17 Pierre Gillibert. An automaton group with undecidable order and Engel problems. *Journal of Algebra*, 497:363–392, 2018. doi:10.1016/j.jalgebra.2017.11.049.

18 Yair Glasner and Shahar Mozes. Automata and square complexes. *Geometriae Dedicata*, 111:43–64, 2005. doi:10.1007/s10711-004-1815-2.

19 Rostislav I. Grigorchuk, Volodymyr V. Nekrashevych, and Vitaly I. Sushchanskĭi. Automata, dynamical systems, and groups. *Proceedings of the Steklov Institute of Mathematics*, 231:128–203, 2000.

20 Rostislav I. Grigorchuk and Igor Pak. Groups of intermediate growth: an introduction. *L'Enseignement Mathématique*, 54:251–272, 2008.

21 Rostislav I. Grigorchuk and Andrzej Żuk. The lamplighter group as a group generated by a 2-state automaton, and its spectrum. *Geometriae Dedicata*, 87:209–244, 2001. `doi:10.1023/A:1012061801279`.

22 Narain Gupta and Saïd Sidki. On the burnside problem for periodic groups. *Mathematische Zeitschrift*, 182(3):385–388, 1983.

23 John M. Howie. *Fundamentals of Semigroup Theory*. London Mathematical Society Monographs. Clarendon Press, 1995.

24 Kate Juschenko. *Amenability of discrete groups by examples*. American Mathematical Society, 2022.

25 David A. Klarner, Jean-Camille Birget, and Wade Satterfield. On the undecidability of the freeness of integer matrix semigroups. *International Journal of Algebra and Computation*, 1(2):223–226, 1991.

26 Ines Klimann. Automaton semigroups: The two-state case. *Theory of Computing Systems*, 58:664–680, 2016. `doi:10.1007/s00224-014-9594-0`.

27 Ines Klimann. To Infinity and Beyond. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2018.131`.

28 Maximilian Kotowsky and Jan Philipp Wächter. The word problem for finitary automaton groups. In Henning Bordihn, Nicholas Tran, and György Vaszil, editors, *Descriptional Complexity of Formal Systems*, pages 94–108, Cham, 2023. Springer Nature Switzerland.

29 Roger Lyndon and Paul Schupp. *Combinatorial Group Theory*. Classics in Mathematics. Springer, 2001. First edition 1977.

30 Tara Macalister Brough, Jan Philipp Wächter, and Janette Welker. Automaton semigroup free products revisited. *arXiv preprint*, 2023. `doi:10.48550/arXiv.2003.12810`.

31 Arnaldo Mandel and Imre Simon. On finite semigroups of matrices. *Theoretical Computer Science*, 5(2):101–111, 1977. `doi:10.1016/0304-3975(77)90001-9`.

32 Turlough Neary. Undecidability in Binary Tag Systems and the Post Correspondence Problem for Five Pairs of Words. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 649–661, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.STACS.2015.649`.

33 Volodymyr V. Nekrashevych. *Self-similar groups*, volume 117 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2005. `doi:10.1090/surv/117`.

34 Matthieu Picantin. Automatic Semigroups vs Automaton Semigroups. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 124:1–124:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2019.124`.

35 Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–269, 1946. `doi:10.1090/s0002-9904-1946-08555-9`.

36 Emanuele Rodaro and Jan Philipp Wächter. The self-similarity of free semigroups and groups. In Munehiro Iwami, editor, *Logic, Algebraic system, Language and Related Areas in Computer Science*, volume 2229 of *RIMS Kôkyûroku*, pages 11–20. Research Institute for Mathematical Sciences, Kyoto University, 2022. `doi:10.48550/arXiv.2205.10248`.

37 Pedro V. Silva and Benjamin Steinberg. On a class of automata groups generalizing lamplighter groups. *International Journal of Algebra and Computation*, 15(05n06):1213–1234, 2005. `doi:10.1142/S0218196705002761`.

38 Rachel Skipper and Benjamin Steinberg. Lamplighter groups, bireversible automata, and rational series over finite rings. *Groups, Geometry and Dynamics*, 14(2):567–589, 2020. doi:10.4171/GGD/555.

39 Benjamin Steinberg, Mariya Vorobets, and Yaroslav Vorobets. Automata over a binary alphabet generating free groups of even rank. *International Journal of Algebra and Computation*, 21(01n02):329–354, 2011. doi:10.1142/S0218196711006194.

40 Zoran Šunić and Enric Ventura. The conjugacy problem in automaton groups is not solvable. *Journal of Algebra*, 364:148–154, 2012. doi:10.1016/j.jalgebra.2012.04.014.

41 Mariya Vorobets and Yaroslav Vorobets. On a free group of transformations defined by an automaton. *Geometriae Dedicata*, 124:237–249, 2007. doi:10.1007/s10711-006-9060-5.

42 Mariya Vorobets and Yaroslav Vorobets. On a series of finite automata defining free transformation groups. *Groups, Geometry, and Dynamics*, 4:337–405, 2010. doi:10.4171/GGD/87.

43 Jan Philipp Wächter. *Automaton Structures – Decision Problems and Structure Theory*. Doctoral thesis, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2020. doi:10.18419/opus-11267.

44 Jan Philipp Wächter and Armin Weiß. *Automata and Languages – GAGTA Book 3*, chapter “The Word Problem for Automaton Groups”. DeGruyter, 2024. In preparation.