Partially commutative inverse monoids

Volker Diekert, Markus Lohrey, Alexander Miller

Universität Stuttgart, FMI, Germany
{diekert,lohrey,miller}@informatik.uni-stuttgart.de

Abstract

Free partially commutative inverse monoids are investigated. Analogously to free partially commutative monoids (trace monoids), free partially commutative inverse monoids are the quotients of free inverse monoids modulo a partially defined commutation relation on the generators.

A quasi linear time algorithm for the word problem is presented, more precisely, we give an $O(n \log(n))$ algorithm for a RAM. NP-completeness of the submonoid membership problem (also known as the generalized word problem) and the membership problem for rational sets is shown. Moreover, free partially commutative inverse monoids modulo a finite idempotent presentation are studied. It turns out that the word problem is decidable if and only if the complement of the partial commutation relation is transitive.

1 Introduction

Many real systems have a determinstic behavior and they allow an undo-operation. This implies that the system is codeterministic, too. If we model such a system by some labelled transition system, then we meet two properties. The system is deterministic, thus in every state there is at most one outgoing edge for each label. Codeterminism means that for each state and label there is at most one incoming edge with this label. In this setting every label defines a partially defined injective mapping from states to states. It follows that the resulting transformations form an *inverse monoid*. Moreover, it it is well-known that every inverse monoid arises this way: It is the transformation monoid of some (possibly infinite) deterministic and codeterministic labelled transition system. Because of this background and its close connection to automata theory inverse monoids received quite an attention in theoretical computer science and there is a well-established literature on this subject, see e.g. [24,30].

In this paper we are interested in the situation where the labels describe actions where some of them can be performed independently. This leads to a partial commutation and therefore to partially commutative inverse monoids. There is a natural notion of *free partially commutative inverse monoids* and this concept has been studied in the thesis of da Costa [34], first. Da Costa showed among others that the word problem of free partially commutative inverse monoids is decidable, but he did not give any complexity bounds. Section 3 of the present paper shows that the word problem is solvable in time $O(n \log(n))$ on a random access machine (RAM).

In order to achieve this time complexity we give a direct approach to define free partially commutative inverse monoids which is in fact our first contribution. Our construction is closer to the standard Birget-Rhodes expansion [3] and to the construction of Margolis and Meakin [22,23]. However instead of using (connected) subsets (of the Cayley graph) of a group, we consider *closed* subsets for some natural closure operation on subsets of *graph groups* [12], which are also known as *free partially commutative groups*.

In Section 4, we extend our decidability result for the word problem to the *submonoid membership problem*. The submonoid membership problem asks whether a given monoid element belongs to a given finitely generated submonoid. In fact, we consider the more general membership problem for rational subsets of a free partially commutative inverse monoid and we show its NP-completeness. NP-hardness appears already for the special case of the submonoid membership problem for a 2-generator free inverse monoid. It is quite remarkable that the submonoid membership problem remains decidable in our setting, because it is known to be undecidable for direct products of free groups [27]. So there is an undecidable problem for a direct product of free groups where the same problem is decidable for a direct product of free inverse monoids.

In the second part of the paper we consider free partially commutative inverse monoids modulo a finite idempotent presentation, which is a finite set of identities between idempotent elements. We show that the resulting quotient monoids have decidable word problems if and only if the underlying dependence structure is transitive. In the transitive case, the uniform word problem (where the idempotent presentation is part of the input) turns out to be EXPTIME-complete, whereas for a fixed idempotent presentation the word problem is solvable both in linear time on a RAM and logarithmic space on a Turing machine. These results generalize corresponding results for free inverse monoids modulo an idempotent presentation from [21,23]. Our decidability result for the case of a transitive dependence structure is unexpected in light of a result of Meakin and Sapir [26], where it was shown that there exist E-unitary inverse monoids over a finitely generated abelian group, where the word problem is undecidable. The proof of this result in [26] is quite involved and relies on a sophisticated encoding of computations of Minsky machines. A slight variation of our undecidability proof for non-transitive dependence structures gives as a byproduct a simpler proof for the result of Meakin and Sapir. Moreover, we exhibit such a situation for every free abelian group of rank at least two. This is tight since free abelian groups of rank one are virtually free and there positive results are known [9].

Finally, in Appendix B we exhibit a connection between the reachability problem for Petri nets and our formalism.

A preliminary version of this paper appeared as a conference abstract in [8].

2 Preliminaries

In the following Σ denotes a finite alphabet and we let $\Sigma^{-1} = \{a^{-1} \mid a \in \Sigma\}$ be a disjoint copy of Σ. We define $\Gamma = \Sigma \cup \Sigma^{-1}$. The set Γ is equipped with an involution ${}^{-1} : \Gamma \to \Gamma$ by $(a^{-1})^{-1} = a$ for all $a \in \Sigma$. We extend this involution to an involution ${}^{-1} : \Gamma^* \to \Gamma^*$ on words over Γ by setting $(a_1 \cdots a_n)^{-1} = a_n^{-1} \cdots a_1^{-1}$ for $a_i \in \Gamma,\ 1 \le i \le n,\ n \ge 0$. The *free group* generated by Σ is denoted by $F(\Sigma)$; it can be defined as the quotient monoid $\Gamma^*/\{aa^{-1} = 1 \mid a \in \Gamma\}$. If the elements of Σ are listed explicitly, we omit the surrounding braces, e.g., we write $F(a, b)$ instead of $F(\{a, b\})$.

Let M be a finitely generated monoid and let Γ be a finite generating set for M, i.e., there exists a surjective homomorphism $h : \Gamma^* \to M$. The *word problem* for M is the computational problem that asks for two given words $u, v \in \Gamma^*$, whether $h(u) = h(v)$. The *submonoid membership problem* for M asks whether for given words $u, v_1, \ldots, v_n$ the element $h(u)$ belongs to $\{h(v_1), \ldots, h(v_n)\}^* \subseteq M$, which is the submonoid generated by $h(v_1), \ldots, h(v_n)$.

The *Cayley graph* of a group G with respect to a generating set Γ is the concrete undirected graph $\mathcal{C}(G, \Gamma) = (G, \{\{u, v\} \mid u^{-1}v \in \Gamma\})$. Note that the undirected edge $\{u, v\}$ can be viewed as a pair of directed edges (u, v) and (v, u), where (u, v) is labelled with $u^{-1}v \in \Gamma$ and (v, u) is labelled with $v^{-1}u \in \Gamma$.

2.1 Free partially commutative inverse monoids

An *inverse monoid* is a monoid M such that for every $x \in M$ there is a *unique* $x^{-1} \in M$ with

$$xx^{-1}x = x \quad \text{and} \quad x^{-1}xx^{-1} = x^{-1}. \tag{1}$$

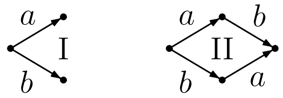
It is well-known that uniqueness of the inverse x^{-1} follows, if we require additionally to (1) that for all $x, y \in M$ we have:

$$xx^{-1}yy^{-1} = yy^{-1}xx^{-1}. \tag{2}$$

Exactly the elements xx^{-1} are the idempotents in M.

Every inverse monoid M can be viewed as a monoid of partially defined injections over a set Q. The inverse of a partial injection $a : \mathrm{dom}(a) \hookrightarrow Q$ is the partial injection a^{-1} with domain $\mathrm{dom}(a^{-1}) = \{\, a(q) \mid q \in \mathrm{dom}(a) \,\}$.

We are interested in inverse monoids with an independence relation. Consider the following situations:



In situation I the transitions a and b commute: The result of ab is the same as ba; it is the undefined mapping, which corresponds to a zero in the monoid of partial injections. It is clear however that a and b should not be called independent, because a can *disable* b (and vice versa). Hence, whether or not b is possible depends on a. The situation II is different: Again a and b commute, but this time a and b act truly independently. We can view Q as the set of global states of the asynchronous product of two independent components. Both components have two states. The first component can perform an action a and the second one a b.

This leads to the following definition. Let Q be a set (of states) and $a : \mathrm{dom}(a) \hookrightarrow Q$ and $b : \mathrm{dom}(b) \hookrightarrow Q$ be partially defined injections where $\mathrm{dom}(a) \cup \mathrm{dom}(b) \subseteq Q$. Then a and b are called *independent*, if the following three conditions are satisfied for all $q \in Q$:

(i) if $q \in \mathrm{dom}(a)$, then: $a(q) \in \mathrm{dom}(b) \iff q \in \mathrm{dom}(b)$,
(ii) if $q \in \mathrm{dom}(b)$, then: $b(q) \in \mathrm{dom}(a) \iff q \in \mathrm{dom}(a)$,
(iii) if $q \in \mathrm{dom}(a) \cap \mathrm{dom}(b)$, then: $ab(q) = ba(q)$.

These conditions look technical, but a brief reflection shows that they are indeed natural translations of an intuitive meaning of independence. Note that independence implies that a and b commute, because $\mathrm{dom}(ab) = \mathrm{dom}(ba) = \mathrm{dom}(a) \cap \mathrm{dom}(b)$. A simple calculation shows that independence of a and b implies the independence of a^{-1} and b, too.

Given an inverse monoid M it is clear that every mapping $\varphi : \Sigma \to M$ lifts uniquely to a homomorphism $\varphi : \Gamma^* \to M$ such that $\varphi(u^{-1}) = \varphi(u)^{-1}$ for all $u \in \Gamma^*$. Next, we lift the notion of independence.

An *independence relation* over Γ is an irreflexive and symmetric relation $I_\Gamma \subseteq$

$\Gamma \times \Gamma$ such that $(a,b) \in I_\Gamma$ implies $(a^{-1},b) \in I_\Gamma$ for all $a,b \in \Gamma$. Note that I_Γ is specified by $I_\Sigma = I_\Gamma \cap \Sigma \times \Sigma$. The pair (Σ, I_Σ) yields an undirected graph with $I_\Sigma \subseteq \binom{\Sigma}{2}$. In the following we simply write I, if the reference to Σ or Γ is clear.

For words $u, v \in \Gamma^*$ we write $(u,v) \in I$ if $u = a_1 \cdots a_m, v = b_1 \cdots b_n$, and $(a_i, b_j) \in I$ for all $1 \le i \le m$ and $1 \le j \le n$.

An *inverse monoid over* (Σ, I) is an inverse monoid M together with a mapping $\varphi : \Sigma \to M$ such that $\varphi(a)\varphi(b) = \varphi(b)\varphi(a)$ and $\varphi(a)^{-1}\varphi(b) = \varphi(b)\varphi(a)^{-1}$ for all $(a,b) \in I = I_\Sigma$. Thus, we can define the *free inverse monoid over* (Σ, I) by

$$\mathrm{FIM}(\Sigma, I) = \mathrm{FIM}(\Sigma)/\{ab = ba, a^{-1}b = ba^{-1} \mid (a,b) \in I\}.$$

Here, $\mathrm{FIM}(\Sigma)$ denotes the free inverse monoid over Σ, which is defined as the quotient monoid of $\Gamma^* = (\Sigma \cup \Sigma^{-1})^*$ modulo the equations in (1) and (2) for all $x, y \in \Gamma^*$. The monoid $\mathrm{FIM}(\Sigma, I)$ is also called a *free partially commutative inverse monoid*.

Da Costa has studied $\mathrm{FIM}(\Sigma, I)$ in his Ph.D. thesis from a more general viewpoint of graph products [34]. As a consequence he showed that $\mathrm{FIM}(\Sigma, I)$ has a decidable word problem. In his construction he used the general approach via Schützenberger graphs and Stephen's iterative procedure [33]. The decidability of the word problem follows because da Costa can show that the procedure terminates. However, no complexity bounds are given in [34].

Another starting point for defining free partially commutative inverse monoids is the Birget-Rhodes expansion [3]. One would start with the free partially commutative group $G(\Sigma, I)$ (defined below) and consider as elements of an inverse monoid the pairs (A, g) where A is a finite and connected subset of the Cayley graph of $G(\Sigma, I)$ with $1, g \in A$. Although this construction yields for $I = \emptyset$ indeed $\mathrm{FIM}(\Sigma)$ by a result of Munn [29], it fails for $I \neq \emptyset$, simply because independent generators do not commute. Thus we have to do something else. Fortunately it is enough to modify the construction of Birget and Rhodes slightly in order to achieve a simple and convenient description of the elements in $\mathrm{FIM}(\Sigma, I)$. Our approach is based on the notion of coherently prefix-closed subsets which we make precise in Section 2.2.

2.2 Trace monoids and graph groups

Recall that $\Gamma = \Sigma \cup \Sigma^{-1}$ and $I \subseteq \Gamma \times \Gamma$ is an irreflexive and symmetric relation such that $(a,b) \in I$ implies $(a^{-1},b) \in I$. Let $M(\Gamma, I) = \Gamma^*/\{ab = ba \mid (a,b) \in I\}$ be the *free partially commutative monoid* (or *trace monoid*) over (Γ, I). Due to $(a,b) \in I \Rightarrow (a^{-1},b) \in I$, the involution ${}^{-1} : \Gamma^* \to \Gamma^*$ is well-defined on $M(\Gamma, I)$. The relation $D = (\Gamma \times \Gamma) \setminus I$ is called the *dependence relation*.

A *clique covering* of the dependence relation D is a tuple $(\Gamma_i)_{1 \leq i \leq k}$ such that

$$\Gamma = \bigcup_{i=1}^{k} \Gamma_i,$$
$$D = \bigcup_{i=1}^{k} \Gamma_i \times \Gamma_i.$$

Note that we have $a \in \Gamma_i$ if and only if $a^{-1} \in \Gamma_i$. Let $\pi_i : M(\Gamma, I) \to \Gamma_i^*$ the *projection homomorphism* which deletes all letters from $\Gamma \setminus \Gamma_i$. The morphism $\pi : M(\Gamma, I) \to \prod_{i=1}^{k} \Gamma_i^*$ defined by $\pi(u) = (\pi_1(u), \ldots, \pi_k(u))$ is injective. This property is sometimes called *projection lemma* in the literature.

There is a rich theory on trace monoids [11], but we need only a few more basic results. The projection lemma can be used to show that traces (i.e., elements of $M(\Gamma, I)$) have a unique description as *dependence graphs*, which are node-labelled acyclic graphs. Let $u = a_1 \cdots a_n \in \Gamma^*$ be a word. The vertex set of the dependence graph of u is $\{1, \ldots, n\}$ and vertex i is labelled with $a_i \in \Gamma$. There is an arc from vertex i to j if and only if $i < j$ and $(a_i, a_j) \in D$. Now, two words define the same trace in $M(\Gamma, I)$ if and only if their dependence graphs are isomorphic. A dependence graph is acyclic, so it induces a labelled partial order, which can be uniquely represented by its Hasse diagram. This means whenever the partial order contains arcs from i to j and j to k for three vertices i, j, k, then we do not draw the arc from i to k.

For $u, v \in M(\Gamma, I)$ we write $u \leq v$ if u is a prefix of v, i.e., $v = uw$ in $M(\Gamma, I)$ for some trace w. A trace f is a *factor* of u if we can write $u = pfq$ in $M(\Gamma, I)$. Let $\max(u) = \{a \in \Gamma \mid u = ta \text{ for some trace } t\}$; it is the set of labels of the maximal nodes in the dependence graph for u.

A basic tool for working with traces is Levi's Lemma, see e.g. [11, p. 10]:

Lemma 1 (Levi Lemma) *Let $x, y, z, t \in M(\Gamma, I)$, then the following assertions are equivalent.*

$$xy = zt,$$
$$\exists p, r, s, q \in M(\Gamma, I) : x = pr,\ y = sq,\ z = ps,\ t = rq,\ \text{ and }\ (r, s) \in I.$$

Another important fact about traces is the following: Consider the prefix order $\leq$ and assume $u \leq w$ and $v \leq w$ for some $u, v, w \in M(\Gamma, I)$. Then the supremum $u \sqcup v \in M(\Gamma, I)$ exists. We can define $u \sqcup v$ by restricting the dependence graph of w to the domain of u and v, where u and v are viewed as downward-closed subsets of the dependence graph of w. If $(\Gamma_i)_{1\leq i\leq k}$ is a clique covering of the dependence relation, then for every i, either $\pi_i(u) \leq \pi_i(v)$ and $\pi_i(u \sqcup v) = \pi_i(v)$ or $\pi_i(v) \leq \pi_i(u)$ and $\pi_i(u \sqcup v) = \pi_i(u)$.

A trace p is called *prime* if $|\max(p)| = 1$. For $t \in M(\Gamma, I)$ let $\mathbb{P}(t) = \{p \leq t \mid p \text{ is prime}\}$. Note that $t = \sqcup \mathbb{P}(t)$ (the supremum of the traces in $\mathbb{P}(t)$). Let $A \subseteq M(\Gamma, I)$. We define $\mathbb{P}(A) = \bigcup_{t\in A} \mathbb{P}(t)$. The set A is called *prefix-closed*, if $u \leq v \in A$ implies $u \in A$. It is called *coherently-closed* if for every $C \subseteq A$ such that $\sqcup C$ exists, $\sqcup C \in A$. One can show that A is coherently-closed if and only if for all $u, v \in A$ such that $u \sqcup v$ exists, we have $u \sqcup v \in A$. In the following we say that A is *closed*, if it is both prefix-closed and coherently-closed.

Note however that closed sets do not form a topology, since the union of two closed sets is not closed, in general. Indeed, let $(a, b) \in I$. Then $\{1, a\}$ and $\{1, b\}$ are closed, but the union $\{1, a, b\}$ is not, since $a \sqcup b = ab$ is missing. The notations prime and coherence are standard in domain theory and the connection to trace theory is exposed in [11, Sec. 11.3].

Clearly, for every $A \subseteq M(\Gamma, I)$ there is a smallest closed set

$$\overline{A} = \{ \sqcup C \ \mid C \text{ is a set of prefixes of } A \text{ such that } \sqcup C \text{ exists} \}$$

with $A \subseteq \overline{A} = \overline{\overline{A}}$. We also have

$$\overline{A} = \{\sqcup C \mid C \subseteq \mathbb{P}(A), \sqcup C \text{ exists}\},$$

since the set on the right side of the equation is prefix-closed, coherently-closed, and it contains A. In particular, $\mathbb{P}(A) = \mathbb{P}(B)$ implies $\overline{A} = \overline{B}$. The converse holds, too:

Lemma 2 *For $A, B \subseteq M(\Gamma, I)$ we have $\overline{A} = \overline{B}$ if and only if $\mathbb{P}(A) = \mathbb{P}(B)$.*

Proof. As we have mentioned above $\mathbb{P}(A) = \mathbb{P}(B)$ implies $\overline{A} = \overline{B}$. The other direction follows because every trace is the supremum of its primes. $\square$

A *trace rewriting system* R over $M(\Gamma, I)$ is just a finite subset of $M(\Gamma, I) \times M(\Gamma, I)$, for details see [7]. We can define the one-step rewrite relation $\to_R \subseteq M(\Gamma, I) \times M(\Gamma, I)$ by: $x \to_R y$ if and only if there are $u, v \in M(\Gamma, I)$ and $(\ell, r) \in R$ such that $x = u\ell v$ and $y = urv$. The notion of a *confluent* and *terminating* trace rewriting system is defined as for other types of rewriting systems. A trace u is an *irreducible normal form* of t if $t \xrightarrow{*}_R u$ and there does not exist a trace v with $u \to_R v$.

The *free partially commutative group* (or *graph group* [12]) over (Σ, I), briefly $G(\Sigma, I)$, is the quotient of the free group $F(\Sigma)$ modulo the defining relations

$ab = ba$ for all $(a, b) \in I$. Clearly,

$$G(\Sigma, I) = M(\Gamma, I)/\{aa^{-1} = 1 \mid a \in \Gamma\}.$$

We can define a confluent and terminating trace rewriting system.

$$R = \{aa^{-1} \to 1 \mid a \in \Gamma\}.$$

Thus, as long as a trace contains a factor aa^{-1} we replace this factor by the empty trace $1 \in M(\Gamma, I)$. Given a trace $u \in M(\Gamma, I)$ by any representing word $u \in \Gamma^*$ we can compute its irreducible normal form $\widehat{u} \in M(\Gamma, I)$ w.r.t. R in linear time, see [7] or [36] for a similar method. We also say that the trace $\widehat{u}$ is *reduced*. Thus, a reduced trace is a trace without any factor of the form aa^{-1} for $a \in \Gamma$. We have $u = v$ in $G(\Sigma, I)$ if and only if $\widehat{u} = \widehat{v}$. This allows us to solve the word problem in $G(\Sigma, I)$ in linear time, too.

The other idea, to use a projection lemma in order solve the word problem of $G(\Sigma, I)$, works for trace monoids, but fails for graph groups: Indeed, let (a, b), $(b, c) \in D$, but $(a, c) \in I$. Then $abcb^{-1}a^{-1}bc^{-1}b^{-1}$ is reduced, but it is in the kernel of the projections to the free groups $F(a, b)$ and $F(b, c)$. (In fact, the free product of $\mathbb{Z}$ and the direct product $\mathbb{Z} \times \mathbb{Z}$ is no subgroup of any finitely generated direct product of free groups, [13].)

In the following, whenever $u \in \Gamma^*$ (or $u \in M(\Gamma, I)$ or $u \in G(\Sigma, I)$), then $\widehat{u} \in M(\Gamma, I)$ denotes the unique reduced trace such that $u = \widehat{u}$ in $G(\Sigma, I)$. The set $\widehat{M}(\Gamma, I) = \{\widehat{u} \mid u \in M(\Gamma, I)\}$ is in canonical one-to-one correspondence with $G(\Sigma, I)$, hence we may identify $\widehat{u}$ with the corresponding group element.

Here comes a crucial definition: A subset $A \subseteq G(\Sigma, I)$ is called *closed*, if the set of reduced traces $\widehat{A} = \{\widehat{g} \in \widehat{M}(\Gamma, I) \mid g \in A\}$ is closed. Clearly, for every $A \subseteq G(\Sigma, I)$, there is a smallest closed subset $\overline{A} \subseteq G(\Sigma, I)$ such that $A \subseteq \overline{A}$. We have $\overline{A} = \overline{\overline{A}}$ and we can identify $\overline{A}$ with $\widehat{\overline{A}} \subseteq \widehat{M}(\Gamma, I)$. Note that $\widehat{M}(\Gamma, I)$ is closed. Recall that I is irreflexive, hence $\{1, a, a^{-1}\}$ is closed since $(a, a^{-1}) \in D$.

We now give a geometric interpretation of closed sets. Let $g, h \in G(\Sigma, I)$. A *geodesic* between g and h is a shortest path between g and h in the Cayley graph of $G(\Sigma, I)$. The labelling of such a path is unique as a reduced trace $\widehat{u} \in \widehat{M}(\Gamma, I)$ such that $g\widehat{u} = h$ in $G(\Sigma, I)$. We have $\widehat{u} = \widehat{g^{-1}h}$ and we say that $f \in G(\Sigma, I)$ is *on a geodesic* from g to h if $\widehat{f} \leq \widehat{u}$.

Proposition 3 *A subset $A \subseteq G(\Sigma, I)$ is closed if and only if both $1 \in A$ and whenever f is on a geodesic from g to h with $g, h \in A$, then $gf \in A$, too.*

Proof. First, let $1 \in A$ and assume that whenever f is on a geodesic from

g to h with $g, h \in A$, then $gf \in A$, too. We have to show that $\widehat{A} \subseteq M(\Gamma, I)$ is prefix-closed and coherently-closed, where $\widehat{A} = \{\widehat{g} \in \widehat{M}(\Gamma, I) \mid g \in A\}$. If $u \leq \widehat{g} \in \widehat{A}$, then u is on a geodesic from 1 to g, hence $u \in A$. Let $\widehat{g}, \widehat{h} \in \widehat{A}$ such that $\widehat{g} \sqcup \widehat{h}$ exists. Then, by Levi Lemma 1 we have $\widehat{g} = pu$, $\widehat{h} = pv$, and $\widehat{g} \sqcup \widehat{h} = puv$ with $(u, v) \in I$. This implies that $(u^{-1}, v) \in I$ and vu^{-1} is reduced. Hence $vu^{-1} = u^{-1}v$ is a geodesic from g to h and $f = gv = puv = \widehat{g} \sqcup \widehat{h} \in A$. Together this shows that $\widehat{A}$ is closed.

For the converse assume that $\widehat{A}$ is closed. This implies $1 \in A$. Let $\widehat{g}, \widehat{h} \in \widehat{A}$ and let f be on a geodesic from g to h. We have to show $gf \in A$. We can write $\widehat{g}\widehat{w} = \widehat{h}$ in $M(\Gamma, I)$ with $f \leq \widehat{w}$. As shown in [10, Lemma 23] this implies that there are $p, x, y \in \widehat{M}(\Sigma, I)$ with $\widehat{g} = px^{-1}$, $\widehat{w} = xy$, and $\widehat{h} = py$. Moreover, $xy = \widehat{w} = fz$ for some $z \in \widehat{M}(\Sigma, I)$. Again, by lemma 1 $x = ru$, $y = vs$, $f = rv$, and $z = us$ with $(u, v) \in I$. Now, $pu^{-1} \leq \widehat{g} \in \widehat{A}$, hence $pu^{-1} \in \widehat{A}$, and $pv \leq pvs = py = \widehat{h} \in \widehat{A}$, hence $pv \in \widehat{A}$. Since $\widehat{A}$ is closed and $(u^{-1}, v) \in I$, we have $pu^{-1}v \in \widehat{A}$. Finally, in $G(\Sigma, I)$ we obtain $gf = px^{-1}rv = pu^{-1}r^{-1}rv = pu^{-1}v \in A$. $\square$

Corollary 4 *Let $A \subseteq G(\Sigma, I)$ be closed and $g \in A$. Then $g^{-1}A$ is closed.*

Proof. Since $g \in A$, we have $1 \in g^{-1}A$. The property

"f is on a geodesic from h_1 to h_2 with $h_1, h_2 \in A$ implies $h_1 f \in A$"

is invariant by translation. Thus, A satisfies this property if and only if $g^{-1}A$ satisfies this property. $\square$

2.3 A realization of free partially commutative inverse monoids

We are now ready to give a concrete realization of the free inverse monoid over (Σ, I). The realization is very much in the spirit of the Birget-Rhodes expansion [3], but differs in the subtle point that we allow closed subsets of $G(\Sigma, I)$, only. Consider the set of pairs (A, g) where $A \subseteq G(\Sigma, I)$ is a finite and closed subset of the graph group $G(\Sigma, I)$ and $g \in A$. This set becomes a monoid by

$$(A, g) \cdot (B, h) = (\overline{A \cup gB}, gh).$$

An immediate calculation shows that the operation is associative and that $(\{1\}, 1)$ is a neutral element. Moreover, the idempotents are the elements of the form $(A, 1)$ and idempotents commute. By Corollary 4, if $g \in A \subseteq G(\Sigma, I)$ and A is closed, then $g^{-1}A$ is closed, too. Hence, we can define $(A, g)^{-1} = (g^{-1}A, g^{-1})$. A simple calculation shows that (1) and (2) are satisfied. Thus our monoid is an inverse monoid. We view Γ as a subset of this monoid by identifying $a \in \Gamma$ with the pair $(\{1,a\},a)$, and this yields a canonical homomorphism γ defined by

$$\gamma(u) = (\overline{\{\widehat{v} \mid v \leq u\}}, \widehat{u})$$

for $u \in \Gamma^*$. We obtain $\gamma(ab) = (\{1,a\},a) \cdot (\{1,b\},b) = (\overline{\{1,a,ab\}}, ab)$. Now, if $(a,b) \in I$, then $\overline{\{1,a,ab\}} = \{1,a,b,ab\} = \overline{\{1,b,ba\}}$, i.e., $\gamma(ab) = \gamma(ba)$. Hence, we obtain an inverse monoid over (Σ, I) since $(a,b) \in I$ implies $(a^{-1},b) \in I$. As a consequence, the homomorphism γ can be viewed as a canonical homomorphism

$$\gamma : \mathrm{FIM}(\Sigma, I) \to \{\, (\overline{A}, g) \mid g \in A \subseteq G(\Sigma, I),\ A \text{ finite}\}. \tag{3}$$

Theorem 5 *The morphism γ in (3) is an isomorphism.*

Proof. Consider a pair (A,g) with $A \subseteq G(\Sigma,I)$ finite and closed and $g \in A$. Recall that $\widehat{A} = \{\widehat{u} \in \widehat{M}(\Gamma, I) \mid u \in A\}$. Let $w \in \Gamma^*$ be an arbitrary word representing the trace $(\prod_{\widehat{u} \in \widehat{A}} \widehat{u}\,\widehat{u}^{-1})\widehat{g}$ where the product is taken in any order. Then a simple reflection shows $\gamma(w) = (A,g)$. Hence γ is surjective. It remains to show that γ is injective. To see this let $w \in \Gamma^*$ and $\gamma(w) = (A,g)$. Note that $\widehat{w} = \widehat{g}$. It suffices to show

$$w = (\prod_{u \in \widehat{A}} uu^{-1})\widehat{w} \ \text{ in } \mathrm{FIM}(\Sigma, I). \tag{4}$$

This is enough because then $\gamma(w) = \gamma(w')$ implies $w = w'$ in $\mathrm{FIM}(\Sigma, I)$ for all $w, w' \in \Gamma^*$. If $w = 1$ then $(A,g) = (\{1\},1)$ and (4) is true. Hence let $w = va$ with $a \in \Gamma$. By induction $v = (\prod_{u \in \widehat{B}} uu^{-1})\widehat{v}$ in $\mathrm{FIM}(\Sigma, I)$, where $\gamma(v) = (B,h)$ and $\widehat{v} = \widehat{h}$. We obtain $\widehat{A} = \overline{\widehat{B} \cup \{\widehat{w}\}}$ and $w = (\prod_{u \in \widehat{B}} uu^{-1})\widehat{v}a$ in $\mathrm{FIM}(\Sigma, I)$.

We distinguish whether $\widehat{v}a$ is reduced or not. If $\widehat{v}a$ is not reduced, then $\widehat{v} = \widehat{w}a^{-1} \in \widehat{B}$, i.e., $\widehat{w} \in \widehat{B}$ since $\widehat{B}$ is prefix-closed. It follows $\widehat{B} = \widehat{A}$. We obtain in $\mathrm{FIM}(\Sigma, I)$:

$$\begin{aligned} w &= (\prod_{u \in \widehat{B}} uu^{-1})\widehat{v}a = (\prod_{u \in \widehat{A}} uu^{-1})\widehat{w}a^{-1}a = (\prod_{u \in \widehat{A}} uu^{-1})\widehat{w}\,\widehat{w}^{-1}\widehat{w}\,a^{-1}a \\ &= (\prod_{u \in \widehat{A}} uu^{-1})\widehat{w}a^{-1}a\widehat{w}^{-1}\widehat{w} = (\prod_{u \in \widehat{A}} uu^{-1})\widehat{v}\,\widehat{v}^{-1}\widehat{w} = (\prod_{u \in \widehat{A}} uu^{-1})\widehat{w}. \end{aligned}$$

It remains the case where $\widehat{va} = \widehat{v}a = \widehat{w}$. We obtain in $\mathrm{FIM}(\Sigma, I)$:

$$w = (\prod_{u \in \widehat{B}} uu^{-1})\widehat{w} = (\prod_{u \in \widehat{B}} uu^{-1})\widehat{w}\widehat{w}^{-1}\widehat{w}.$$

Clearly, $\widehat{w} \in \widehat{A}$. Hence $w = (\prod_{u \in A'} uu^{-1})\widehat{w}$ in $\mathrm{FIM}(\Sigma, I)$ for some subset $A' \subseteq \widehat{A}$ such that $\overline{A'} = \widehat{A}$ (set $A' = \widehat{B} \cup \{\widehat{w}\}$). Therefore it is enough to show $\prod_{u \in A'} uu^{-1} = \prod_{u \in \widehat{A}} uu^{-1}$ in $\mathrm{FIM}(\Sigma, I)$. This is the assertion of the following claim.

Claim: Let $A \subseteq M(\Gamma, I)$. Then $\prod_{u \in A} uu^{-1} = \prod_{u \in \overline{A}} uu^{-1}$ in $\mathrm{FIM}(\Sigma, I)$.

To prove this claim, let $v \leq u \in A$. Then $u = vw = vv^{-1}vw = vv^{-1}u$ in $\mathrm{FIM}(\Sigma, I)$. Hence we may assume that A is prefix closed. Now, let $u, v \in A$ such that $w = u \sqcup v$ exists. Then, by Levi Lemma 1, $u = pr$, $v = ps$, and $w = prs$ with $(r, s) \in I$. We obtain in $\mathrm{FIM}(\Sigma, I)$: $uu^{-1}vv^{-1} = prr^{-1}p^{-1}pss^{-1}p^{-1} = prr^{-1}ss^{-1}p^{-1} = prss^{-1}r^{-1}p^{-1} = ww^{-1}$. This means that we may assume that A is coherently-closed, too. But if A is both prefix-closed and coherently-closed, then $A = \overline{A}$ by definition of $\overline{A}$. Hence the claim and the theorem follow. $\square$

For $I = \emptyset$, Theorem 5 yields Munn's theorem [29] as a special case. Note that for $I = \emptyset$, the closure $\overline{A}$ of a prefix-closed subset A of the free group $F(\Sigma)$ equals A itself. It follows that $\gamma(u) = (\{\widehat{v} \mid v \leq u\}, \widehat{u})$, where $\widehat{v} \in \Gamma^*$ is the unique irreducible word corresponding to $v \in \Gamma^*$. The set $\{\widehat{v} \mid u \leq v\}$ is also called the *Munn tree* of u.

Since we are interested in computational problems, we are concerned with the input size of elements in $\mathrm{FIM}(\Sigma, I)$. The standard input representation for word problems over monoids generated by Γ is just a word u over the alphabet Γ. If $\gamma(u) = (A, g)$, then $|A| \leq |u|^k$, where k is the number of cliques in a clique covering for the dependence relation, because $t \in \widehat{A}$ implies that $\pi_i(t)$ is a prefix of $\pi_i(u)$ for all $1 \leq i \leq k$. Hence, for a fixed (Σ, I), the size of A is bounded polynomially in the length of u, and moreover A can be calculated in polynomial time from u. Thus, if we care only for polynomial time, we can represent $(A, g) \in \mathrm{FIM}(\Sigma, I)$ by listing all the elements of A followed by g. In fact, instead of writing down all elements of the closed set A, it suffices to list the primes in $\mathbb{P}(A)$ by Lemma 2. The set $\mathbb{P}(A)$ has size at most $|u|$ whatever (Σ, I) is. But in general, the more concise representation is still the standard representation where the input is just a word, and the input size is the length of the word. This is our input measure.

3 The word problem in $\mathrm{FIM}(\Sigma, I)$

Using Munn's theorem [29], it is easy to solve the word problem for a free inverse monoid in linear time on a random access machine (RAM). Working with a Turing machine would slow down the time complexity by poly-logarithmic factors, since internally the RAM algorithm uses pointers. For free partially commutative inverse monoids the solution of the word problem is slightly more involved, even for a RAM. We are able to present an $O(n\log(n))$-algorithm on a RAM by using some combination of simple data structures.

Theorem 6 *For every independence relation $I \subseteq \Sigma \times \Sigma$, the word problem of* $\mathrm{FIM}(\Sigma, I)$ *can be solved in time $O(n\log(n))$ on a RAM.*

Proof. Let $u, v \in \Gamma^*$. As mentioned above or by [7,36], we can test in linear time whether $u = v$ in $G(\Sigma, I)$, i.e., whether $\widehat{u} = \widehat{v}$. It remains to check equality of the closures:

$$\overline{\{\widehat{u'} \mid u' \le u\}} = \overline{\{\widehat{v'} \mid v' \le v\}}.$$

Let $(\Gamma_i)_{1\le i\le k}$ be a clique covering of the dependence relation $D = (\Gamma \times \Gamma) \setminus I$. With $w \in \Gamma^*$ we associate the following data:

- the prefix-closed set of words $T_i(w) = \{\pi_i(\widehat{s}) \mid s \le w\}$,
- the word $p_i(w) = \pi_i(\widehat{w}) \in T_i$ for every $1 \le i \le k$,
- the set of primes $P(w) = \mathbb{P}(\{\widehat{s} \mid s \le w\})$,
- the linearly ordered (w.r.t. the prefix order) set $P_i(w) = \mathbb{P}(\widehat{w}) \cap \{p \mid \max(p) \in \Gamma_i\}$.

By Lemma 2, we have to check whether $P(u) = P(v)$. Before we present an efficient implementation of the data structures above, let us first show how to compute $(T_i(wa), p_i(wa), P_i(wa))_{1\le i\le k}$ and $P(wa)$ from $(T_i(w), p_i(w), P_i(w))_{1\le i\le k}$ and $P(w)$ for $a \in \Gamma$. For this, we have to distinguish the two cases $a^{-1} \notin \max(\widehat{w})$ and $a^{-1} \in \max(\widehat{w})$ and we use the following: For $a \in \Gamma$ such that a occurs in a trace t, define the prime $\delta_a(t)$ by the maximal prefix of t such that $\max(\delta_a(t)) = \{a\}$. In case that a does not occur in t let $\delta_a(t) = 1$. We obtain

$$\delta_a(ta) = (\sqcup\{\delta_b(t) \mid (a,b) \in D\})\, a.$$

Case 1. $a^{-1} \notin \max(\widehat{w})$, i.e., $\widehat{wa} = \widehat{w}a$. We have:

$$T_i(wa) = \begin{cases} T_i(w) \cup \{p_i(w)a\} & \text{if } a \in \Gamma_i \\ T_i(w) & \text{otherwise} \end{cases}$$

$$p_i(wa) = \begin{cases} p_i(w)a & \text{if } a \in \Gamma_i \\ p_i(w) & \text{otherwise} \end{cases}$$

$$P(wa) = P(w) \cup \{\delta_a(\widehat{w}a)\}$$

$$P_i(wa) = \begin{cases} P_i(w) \cup \{\delta_a(\widehat{w}a)\} & \text{if } a \in \Gamma_i \\ P_i(w) & \text{otherwise} \end{cases}$$

Note that

$$\delta_a(\widehat{w}a) = (\sqcup\{\delta_b(\widehat{w}) \mid (a,b) \in D\})\, a = (\sqcup\{\max P_i(w) \mid 1 \le i \le k, a \in \Gamma_i\})\, a.$$

Case 2. $a^{-1} \in \max(\widehat{w})$, i.e., $\widehat{w} = sa^{-1}$ and $\widehat{wa} = s$ for an irreducible trace $s \in \widehat{M}(\Gamma, I)$. Moreover, note that $\max P_i(w) = \delta_{a^{-1}}(\widehat{w})$ for all i with $a^{-1} \in \Gamma_i$ (i.e., $a \in \Gamma_i$). We have:

$$\begin{aligned}
T_i(wa) &= T_i(w) \\
p_i(wa) &= \begin{cases} v & \text{if } a^{-1} \in \Gamma_i,\ p_i(w) = va^{-1} \\ p_i(w) & \text{otherwise} \end{cases} \\
P(wa) &= P(w) \cup \mathbb{P}(s) = P(w) \\
P_i(wa) &= \begin{cases} P_i(w) \setminus \max P_i(w) & \text{if } a^{-1} \in \Gamma_i \\ P_i(w) & \text{otherwise} \end{cases}
\end{aligned}$$

For the equality $P(w) \cup \mathbb{P}(s) = P(w)$ note that $\mathbb{P}(s) \subseteq P(w)$ since the trace s is a prefix of the trace $\widehat{w}$. Let us now discuss an efficient implementation of our data structures such that the updates above can be done in time $O(\log(n))$. The prefix-closed set $T_i(w)$ can be stored as a trie [14] with at most $|\pi_i(w)|$ many nodes, i.e., a rooted tree, where every node has for every $a \in \Gamma$ at most one a-labelled outgoing edge and $T_i(w)$ equals the set of all path-labels from the root to tree nodes. We assign with every node of $T_i(w)$ a key from $\mathbb{N}$. The root gets the key 1, and with every new node of $T_i(w)$ the key is increased by one. This allows to calculate $\max U$ for a subset $U \subseteq T_i(w)$, which is linearly ordered by the prefix relation, in time $O(|U|)$ by comparing the keys for the nodes in U. The word $p_i(w) = \pi_i(\widehat{w})$ is just a distinguished node of the trie $T_i(w)$. Clearly, $a^{-1} \in \max(\widehat{w})$ if and only if $p_i(w)$ ends with a^{-1} for all $1 \le i \le k$ with $a^{-1} \in \Gamma_i$. This means that whenever $a^{-1} \in \Gamma_i$, then $p_i(w)$ is the a^{-1}-successor of its parent node. This allows to distinguish between case 1 and case 2 above in constant time. In case 1, we have to add an a-successor to the node $p_i(w)$ in case $a \in \Gamma_i$ and $p_i(w)$ does not have an a-successor yet. This new node becomes $p_i(wa)$. If $a \in \Gamma_i$ but $p_i(w)$ already has an a-successor v, then v becomes $p_i(wa)$. In case 2, the tries do not change, but if $a \in \Gamma_i$, then $p_i(wa)$ is the father node of $p_i(w)$.

The set of primes $P(w)$ is stored as the set of tuples $\{(\pi_i(t))_{1 \le i \le k} \mid t \in P(w)\}$, where every projection $\pi_i(t)$ is represented by the corresponding node in the trie $T_i(w)$. For the set $P(w)$ we use a data structure which allows $O(\log(n))$ time implementations for the operations insert and find. The linearly ordered set $P_i(w)$ is stored as a list of tuples $(\pi_i(t))_{1 \le i \le k}$ for $t \in P_i(w)$. Using this representation, the necessary updates for case 2 are possible in constant time. For case 1, we have to calculate the tuple corresponding to $\delta_a(\widehat{w}a) = (\sqcup\{\max P_i(w) \mid 1 \le i \le k, a \in \Gamma_i\})\, a$. Note that

$$\pi_j(\delta_a(\widehat{w}a)) = \sqcup\{\pi_j(\max P_i(w)) \mid 1 \le i \le k, a \in \Gamma_i\}\, \pi_j(a) \tag{5}$$

for $1 \leq j \leq k$. The $\sqcup$ in (5) refers to the prefix order on words. Note that since $\max P_i(w)$ is a prefix of the trace $\widehat{w}$ for every i, the set $\{\pi_j(\max P_i(w)) \mid 1 \leq i \leq k, a \in \Gamma_i\}$ is linearly ordered by the prefix relation on Γ_j^*, i.e., the supremum exists. Moreover, this supremum can be computed in time $O(k)$, where k is the number of cliques (which is a constant) by using the keys associated with the nodes from $T_j(w)$. This concludes the description of our data structures. Now for our input words $u, v \in \Gamma^*$ we first compute $(T_i(u), p_i(u), P_i(u))_{1 \leq i \leq k}$, $P(u)$ and $(T_i(v), p_i(v), P_i(v))_{1 \leq i \leq k}$, $P(v)$ in time $O(n \log(n))$. When building up the tries $T_i(u)$ and $T_i(v)$ we have to use the same node name for a certain string over Γ_i. Then we can check $P(u) = P(v)$ in time $O(n \log(n))$ using the set data structures for $P(u)$ and $P(v)$. $\square$

For the uniform word problem, where the independence relation I is part of the input, the above algorithm still yields a polynomial time algorithm. More precisely, the running time is $O((k^2 + \log(n))n)$, where k is the number of cliques in a clique covering for the dependence relation and n is the length of the input words.

4 The submonoid membership problem and the membership problem for rational sets in $\mathrm{FIM}(\Sigma, I)$

Rational subsets in a monoid M are defined inductively: Finite subsets are rational, and if L_1 and L_2 are rational, then the following subsets are rational as well:

$$\begin{aligned} &L_1 \cup L_2 \\ L_1 \cdot L_2 &= \{\, uv \;\mid\, u \in L_1,\, v \in L_2 \,\} \\ (L_1)^* &= \text{ submonoid of } M \text{ generated by } L_1 \end{aligned}$$

For $M = \Gamma^*$, rational subsets are also called *regular languages*. By Kleene's Theorem a regular language can be specified by a non-deterministic finite automaton. Hence, if M is generated by Γ, then we may use non-deterministic finite automata over the alphabet Γ for a concise specification of rational subsets over M as well, since the homomorphic image of a rational subset is rational again. Hence, a rational subset of $\mathrm{FIM}(\Sigma, I)$ can always be represented by some non-deterministic finite automaton over Γ.

The submonoid membership problem is a special instance of the membership problem for rational sets. In this section, we show that the submonoid membership problem for the free inverse monoid $\mathrm{FIM}(a, b)$ is NP-hard, and we show that the membership problem for rational sets of $\mathrm{FIM}(\Sigma, I)$ is still in NP. So, the submonoid membership problem and the membership problem for rational sets are both NP-complete for free partially commutative inverse monoids, in general.

Theorem 7 *For every independence relation $I \subseteq \Sigma \times \Sigma$, the membership problem for rational subsets of* $\mathrm{FIM}(\Sigma, I)$ *belongs to* NP.

Proof. For given $(B, g) \in \mathrm{FIM}(\Sigma, I)$ and a finite automaton $\mathcal{A}$ over the alphabet Γ we have to determine whether $(B, g) \in \gamma(L(\mathcal{A}))$, where $\gamma : \Gamma^* \to \mathrm{FIM}(\Sigma, I)$ is the canonical morphism and $L(\mathcal{A}) \subseteq \Gamma^*$ denotes the accepted language. In the following, we view B as a closed subset of of $G(\Sigma, I)$. In a first step we guess a connected (in the Cayley graph of $G(\Sigma, I)$) subset $C \subseteq B$ with $1, g \in C$ such that its closure $\overline{C}$ equals B. For this it is enough to check that all primes of B appear in C. It remains to check in NP whether there is a path in the Cayley-graph from 1 to g, which visits exactly the nodes in C and such that this path is labelled with a word from $L(\mathcal{A})$.

Let n be the number of states of the automaton $\mathcal{A}$. Assume that $p = (v_1, \ldots, v_m)$ is a path in C such that $v_1 = 1$, $v_m = g$, $C = \{v_1, \ldots, v_m\}$ and let $q_1 \ldots, q_m$ be a corresponding path in the automaton $\mathcal{A}$, where q_1 is some initial state and q_m is some final state. Let $i_1 < \cdots < i_\ell$ be exactly those positions $j \in \{2, \ldots, m\}$ such that $v_j \notin \{v_1, \ldots, v_{j-1}\}$. Clearly, $\ell < |C|$. Set $i_0 = 1$ and $i_{\ell+1} = m + 1$. Assume that $|i_{k+1} - i_k| > |C| \cdot n$ for some $k \in \{0, \ldots, \ell\}$. Then there are positions $i_k \leq \alpha < \beta < i_{k+1}$ such that $v_\alpha = v_\beta$ and $q_\alpha = q_\beta$. It follows that $v_1, \ldots, v_\alpha, v_{\beta+1}, \ldots, v_m$ is a again a path from 1 to g, which visits all nodes of C, and $q_1, \ldots, q_\alpha, q_{\beta+1}, \ldots, q_m$ is a corresponding path in the automaton $\mathcal{A}$.

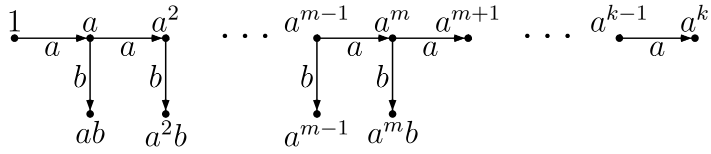
From the above consideration it follows that if there exists a path from 1 to g in C, which visits all nodes of C and such that this path is labelled with a word from $L(\mathcal{A})$, then there exists such a path of length at most $|C|^2 \cdot n$. Such a path can be guessed in polynomial time. This finishes the proof. $\square$

NP-hardness can be already shown for the submonoid membership problem of $\mathrm{FIM}(a, b)$:

Theorem 8 *The submonoid membership problem for* $\mathrm{FIM}(a, b)$ *is* NP*-hard.*

Proof. We prove the theorem by a reduction from CNF-SAT (Satisfiability of Boolean formulas in conjunctive normal form).

Let $\Psi = \{C_1, \ldots, C_m\}$ be a set of clauses over variables $x_1, \ldots, x_n$. Let $k = m + n$. For $1 \leq i \leq n$ let $P_i = \{j \mid x_i \in C_j\}$ and $N_i = \{j \mid \neg x_i \in C_j\}$. Let $t = (A, a^n) \in \mathrm{FIM}(a, b)$ be defined such that the subgraph of the Cayley graph of $F(a, b)$ induced by A looks as follows:



This means that we have

$$t = (abb^{-1})^m\, a^n\, a^{-m}.$$

The idea is that the node $a^j b$ represents the clause C_j. For every $1 \le i \le n$ define $t_{i,p} = (A_{i,p}, a) \in \mathrm{FIM}(a, b)$ and $t_{i,f} = (A_{i,f}, a) \in \mathrm{FIM}(a, b)$, where:

$$\begin{aligned} A_{i,p} &= a^{-i+1}(\{1, a, \dots, a^k\} \cup \{a^j b \mid j \in P_i\}) \subseteq F(a, b) \\ A_{i,f} &= a^{-i+1}(\{1, a, \dots, a^k\} \cup \{a^j b \mid j \in N_i\}) \subseteq F(a, b) \end{aligned}$$

Then we have $t \in \{t_{1,p}, t_{1,f}, \dots, t_{n,t}, t_{n,f}\}^*$ if and only if Ψ is satisfiable. $\square$

The NP upper bound in Theorem 7 generalizes to the uniform case, where the independence relation I is part of the input but the number of cliques in a clique covering for $D = I \setminus (\Sigma \times \Sigma)$ is bounded by a fixed constant. However, the uniform complexity becomes PSPACE-complete:

Theorem 9 *The following problem is* PSPACE*-complete:*

INPUT: An alphabet Σ, an independence relation $I \subseteq \Sigma \times \Sigma$, and words $u, u_1, \dots, u_n \in \Gamma^*$.

QUESTION: $u \in \{u_1, \dots, u_n\}^*$ in $\mathrm{FIM}(\Sigma, I)$?

In fact we prove stronger theorems. We give an PSPACE-upper bound for the membership problem in rational sets, and we show PSPACE-hardness in the case where the graph group $G(\Sigma, I)$ is free abelian. In fact, our hardness proof of Theorem 11 is very similar to the proof that the reachability problem for 1-safe Petri nets is PSPACE-hard [6]. The similarity is no surprise by Appendix B.

Theorem 10 *The following problem is in* PSPACE*:*

INPUT: An alphabet Σ, an independence relation $I \subseteq \Sigma \times \Sigma$, a word $u \in \Gamma^*$ and a rational subset $L \subseteq \mathrm{FIM}(\Sigma, I)$ (given by a nondeterministic finite state automaton over the alphabet Γ).

QUESTION: Is $u \in L$ if we read $u \in \mathrm{FIM}(\Sigma, I)$?

Proof. Let (Σ, I) be an independence alphabet, let $u \in \Gamma^*$, and let $\mathcal{A}$ be a non-deterministic finite state automaton over the alphabet Γ. Let $(B, g) \in \mathrm{FIM}(\Sigma, I)$ be the element represented by the word u. Note that $|B|$ may be of size $|u|^{|\Sigma|}$, but the set of primes $\mathbb{P} = \mathbb{P}(\{\widehat{s} \mid s \leq u\})$ has size at most $|u|$. Using the algorithm from the proof of Theorem 6, we can construct the set $\mathbb{P}$ in polynomial time. Every reduced trace from the set B can be represented by a subset of at most $|\Sigma|$ many elements from $\mathbb{P}$. In order to check whether $u \in L(\mathcal{A})$ in $\mathrm{FIM}(\Sigma, I)$, we have to guess a word $v \in L(\mathcal{A})$ such that $\widehat{v} = \widehat{u}$ and $\mathbb{P} = \mathbb{P}(\{\widehat{s} \mid s \leq v\})$. If we have guessed a prefix s of v so far, then instead of storing the whole prefix s, we only store the current state of the automaton $\mathcal{A}$ as well as the two sets of primes $\mathbb{P}(\widehat{s})$ and $\mathbb{P}(\{\widehat{t} \mid t \leq s\})$. If the latter set is no longer contained in $\mathbb{P}$, then we immediately reject. We accept if we reach a final state of $\mathcal{A}$ and at the same time $\sqcup \mathbb{P}(\widehat{s}) = \widehat{u}$ and $\mathbb{P}(\{\widehat{t} \mid t \leq s\}) = \mathbb{P}$. Since at any step of this algorithm we only have to store a polynomial amount of information, the PSPACE upper bound follows. $\square$

Theorem 11 *The following problem is* PSPACE*-hard:*

INPUT: An alphabet Σ with the independence relation $I = (\Sigma \times \Sigma) \setminus \mathrm{id}_\Sigma$, and words $u, u_1, \ldots, u_n \in \Gamma^*$

QUESTION: $u \in \{u_1, \ldots, u_n\}^*$ in $\mathrm{FIM}(\Sigma, I)$?

Proof. We make a reduction from the following PSPACE-complete problem:

INPUT: A finite alphabet Θ, two strings $u, v \in \Theta^*$ with $|u| = |v|$, and a string rewriting system R, where all rules of R have the form $ab \to cd$ for $a, b, c, d \in \Theta$.

QUESTION: $u \xrightarrow{*}_R v$?

PSPACE-hardness of this problem is well-known and can be easily shown by a reduction from the word problem of LBA (linear bounded automata). Let us take an input Θ, $u, v \in \Theta^*$, together with a system R as described above. Let $n = |u| = |v| \geq 1$. For all $1 \leq i \leq n$ let $\Theta_i = \{[a, i] \mid a \in \Theta\}$ be a new copy of Θ. Let $\Sigma = \bigcup_{i=1}^n \Theta_i$ and let $I = (\Sigma \times \Sigma) \setminus \mathrm{id}_\Sigma$. Note that $G(\Sigma, I)$ is a free abelian group with rank $n\,|\Theta|$. However, $\mathrm{FIM}(\Sigma, I)$ is not commutative. Although most generators commute, we have $aa^{-1} \neq a^{-1}a$ in $\mathrm{FIM}(\Sigma, I)$.

For a word $w = a_1 a_2 \cdots a_n$ of length n with $a_i \in \Theta$ let

$$h(w) = [a_1, 1][a_2, 2] \cdots [a_n, n] \in \Theta_1 \Theta_2 \cdots \Theta_n$$

and define:

$$g(w) = (\prod_{a \in \Sigma} aa^{-1})h(w) \in \mathrm{FIM}(\Sigma, I).$$

Note that for $|w| = |w'| = n$ and $g(w) = g(w')$ we have $w = w'$.

Next, for a rule $(ab \to cd) \in R$ and $1 \le i < n$ let

$$g(i, ab \to cd) = [a,i]^{-1}[b,i+1]^{-1}[c,i][d,i+1].$$

Note that the ordering of the elements on the right hand side is irrelevant, if $a \neq c$ and $b \neq d$. However if e.g. $a = c$, then we insist that $[a,i]^{-1}$ appears before $[c,i]$.

Now assume that $w = w_1 ab w_2$ with $|w_1| + 1 = i$ and $|w| = n$ and $(ab \to cd) \in R$. Then we obtain $w' = w_1 cd w_2$ by applying the rule at position i.

Moreover, we obtain:

$$\begin{aligned} g(w) \cdot g(i, ab \to cd) &= (\prod_{a \in \Sigma} aa^{-1})h(w) \cdot g(i, ab \to cd) \\ &= (\prod_{a \in \Sigma} aa^{-1})h(w) \cdot [a,i]^{-1}[b,i+1]^{-1}[c,i][d,i+1] \\ &= (\prod_{a \in \Sigma} aa^{-1}) \cdot [a,i][a,i]^{-1}[b,i+1][b,i+1]^{-1} \cdot h(w') \qquad (6) \\ &= (\prod_{a \in \Sigma} aa^{-1}) \cdot h(w') \\ &= g(w') \end{aligned}$$

Then the following two statements are equivalent:

(1) $u \xrightarrow{*}_R v$
(2) $g(v) \in \{g(u),\ g(i,r) \mid 1 \le i < n, r \in R\}^*$ in $\mathrm{FIM}(\Sigma, I)$

The calculation (6) yields (1) $\Rightarrow$ (2): Indeed let $u \xrightarrow{*}_R v$ be any derivation from u to v. Then (6) shows that $g(v)$ belongs to $\{g(u),\ g(i,r) \mid 1 \le i < n, r \in R\}^*$.

For the other direction let

$$g(v) = g_1 \cdots g_m \text{ with } g_j \in \{g(u),\ g(i,r) \mid 1 \le i < n, r \in R\}.$$

We have to show $u \xrightarrow{*}_R v$. Let $g(v) = (V, x)$ in $\mathrm{FIM}(\Sigma, I)$ where $V = \overline{\Sigma} \subseteq G(\Sigma, I)$ and $x \in V$. We must have $m \ge 1$. Moreover, we cannot have $g_1 = [a,i]^{-1}[b,i+1]^{-1}[c,i][d,i+1]$, because then $[a,i]^{-1}$ would belong to V, which is not the case. Hence $g_1 = g(u)$. We claim that for all $1 \le k \le m$:

- for every $2 \le j \le k$, the element g_j is of the form $g(i_j, r_j)$ for some $1 \le i_j < n$ and $r_j \in R$,

- there is a word $w \in \Theta^n$ such that $g_1 g_2 \cdots g_k = g(w)$ and w results from u by applying the rules $r_2, \ldots, r_k$ in that order.

For $k = m$ this implies $u \xrightarrow{m}_R v$. For $k = 1$ there is nothing to show. Hence assume that the conditions above are true for some $k < m$. Since $g(v) = g_1 g_2 \cdots g_k g_{k+1} \cdots g_m = g(w) g_{k+1} \cdots g_m$ we cannot have $g_{k+1} = g(u)$, because then an element of Σ^2 would belong to V, which is not the case. More precisely, $[b,1][a,1]$ would belong to V where b is the first symbol of w and a is the first symbol of u.

Hence, $g_{k+1} = [a,i]^{-1}[b,i+1]^{-1}[c,i][d,i+1]$ for some $1 \le i < n$ and $(ab \to cd) \in R$. We must have $w = w_1 ab w_2$ with $|w_1|+1 = i$, since otherwise V would contain the group element $[a,i]^{-1}$ or $[b,i+1]^{-1}$. Thus, by abuse of notation we may write:

$$\begin{aligned} g(w) g_{k+1} &= (\prod_{a \in \Sigma} aa^{-1}) w_1 [a,i][b,i+1] w_2 [a,i]^{-1}[b,i+1]^{-1}[c,i][d,i+1] \\ &= \prod_{a \in \Sigma} aa^{-1} w_1 [c,i][d,i+1] w_2 \\ &= g(w') \end{aligned}$$

Hence, w' results from w by applying the rule $ab \to cd$ at position i. $\square$

5 FIM(Σ, I) modulo an idempotent presentation

Let $I \subseteq \Sigma \times \Sigma$ be an independence relation. An *idempotent presentation* over (Σ, I) is a finite set of identities $P = \{(e_i, f_i) \mid 1 \le i \le n\}$, where every e_i and f_i are idempotent elements in FIM(Σ, I). Based on a reduction to Rabin's tree theorem, Margolis and Meakin have shown that for $I = \emptyset$, the word problem for quotient monoids of the form FIM$(\Sigma)/P$ is decidable [23].

In this section we prove that the uniform word problem for monoids of the form FIM$(\Sigma, I)/P$, where P is an idempotent presentation over FIM(Σ, I), is decidable if and only if the dependence relation $D = (\Sigma \times \Sigma) \setminus I$ is transitive. Transitivity means that FIM(Σ, I) is a direct product of free inverse monoids. In the transitive case we prove EXPTIME-completeness for the uniform problem. For the upper bound, we use analogously to [23] a closure operation on subsets of $G(\Sigma, I)$. The lower bound follows directly from [21].

Assume that P is an idempotent presentation. Consider a pair $(e, f) \in P$. Then we have $e = (E, 1)$ and $f = (F, 1)$, where E and F are finite and closed subsets of the graph group $G(\Sigma, I)$ and $1 \in E \cap F$. In the following, we identify the pair $(E, 1)$ with the finite closed set E. Since e and f are idempotents of

$\mathrm{FIM}(\Sigma, I)$ and idempotents commute, we can replace the relation $e = f$ by the two relations $e = ef$ and $f = ef$ without changing the quotient monoid. This means that for every pair $(E, F) \in P$, we can assume $E \subseteq F$.

Now assume that $A, B \subseteq G(\Sigma, I)$ are finite and closed. We write $A \Rightarrow_P B$ if and only if there exist $(E, F) \in P$ (hence $E \subseteq F$) and $f \in G(\Sigma, I)$ such that $fE \subseteq A$ and $B = \overline{A \cup fF}$. Note that $A \cup fF$ is not necessarily closed, even if both, A and fF are closed. Therefore we take the closure $\overline{A \cup fF}$. The relation $\Rightarrow_P$ is strongly confluent, i.e., if $A \Rightarrow_P B$ and $A \Rightarrow_P C$, then there exists D such that $B \Rightarrow_P D$ and $C \Rightarrow_P D$. Hence, $A \overset{*}{\Leftrightarrow}_P B$ if and only if there exists C such that $A \overset{*}{\Rightarrow}_P C$ and $B \overset{*}{\Rightarrow}_P C$. Define

$$\mathrm{cl}_P(A) = \bigcup\{B \subseteq G(\Sigma, I) \mid A \overset{*}{\Rightarrow}_P B\}.$$

Clearly, $\mathrm{cl}_P(A) \subseteq G(\Sigma, I)$ is closed.

Lemma 12 *We have $(A, g) = (B, h)$ in $\mathrm{FIM}(\Sigma, I)/P$ if and only if both, $g = h$ in $G(\Sigma, I)$ and $A \overset{*}{\Leftrightarrow}_P B$.*

Proof. Straightforward, and left to the reader. □

Lemma 13 *Let $(A, g), (B, h) \in \mathrm{FIM}(\Sigma, I)$. Then $(A, g) = (B, h)$ in $\mathrm{FIM}(\Sigma, I)/P$ if and only if both, $g = h$ in $G(\Sigma, I)$ and $\mathrm{cl}_P(A) = \mathrm{cl}_P(B)$.*

Proof. By Lemma 12, it suffices to show that $\mathrm{cl}_P(A) = \mathrm{cl}_P(B)$ if and only if $A \overset{*}{\Leftrightarrow}_P B$. First assume that $A \overset{*}{\Leftrightarrow}_P B$ and $x \in \mathrm{cl}_P(A)$. Thus, $A \overset{*}{\Rightarrow}_P C$ for some C with $x \in C$. Hence, $B \overset{*}{\Leftrightarrow}_P A \overset{*}{\Rightarrow}_P C$ and there exists D with $B \overset{*}{\Rightarrow}_P D$ and $C \overset{*}{\Rightarrow}_P D$. This implies $x \in C \subseteq D$, i.e, $x \in \mathrm{cl}_P(B)$. We have shown $\mathrm{cl}_P(A) \subseteq \mathrm{cl}_P(B)$; the other inclusion follows analogously.

Now assume that $\mathrm{cl}_P(A) = \mathrm{cl}_P(B)$. Thus, there must exist C, D such that $A \overset{*}{\Rightarrow}_P C \supseteq B$ and $B \overset{*}{\Rightarrow}_P D \supseteq A$. By induction over the length of the derivation $A \overset{*}{\Rightarrow}_P C$ one can show that $D \overset{*}{\Rightarrow}_P \overline{C \cup D}$ and analogously $C \overset{*}{\Rightarrow}_P \overline{C \cup D}$. Hence, $A \overset{*}{\Leftrightarrow}_P B$. □

In the following we use the modal μ-calculus with simultaneous fixpoints. The necessary details can be found in the Appendix A.

Theorem 14 *The following problem is* EXPTIME*-complete:*

INPUT: An independence relation $I \subseteq \Sigma \times \Sigma$ such that $D = (\Sigma \times \Sigma) \setminus I$ is transitive, an idempotent presentation P over (Σ, I) and words $u, v \in \Gamma^*$.

QUESTION: $u = v$ *in* $\mathrm{FIM}(\Sigma, I)/P$*?*

Proof. Hardness for EXPTIME follows from [21, Theorem 5]. For membership in EXPTIME we will analogously to [21] reduce the uniform word problem to the model-checking problem for the modal μ-calculus (with simultaneous fixpoints) over a context-free graph. Since $D = (\Sigma \times \Sigma) \setminus I$ is transitive, the graph (Σ, D) is a disjoint union of cliques $\Sigma_1, \dots, \Sigma_n$. Thus, $G(\Sigma, I) = \prod_{i=1}^{n} F(\Sigma_i)$. In the following, we assume that $n = 2$ in order to simplify notation. Let $P = \{(e_i, f_i) \mid 1 \leq i \leq \ell\}$ be the given idempotent presentation over (Σ, I). Let $(A, g) \in \mathrm{FIM}(\Sigma, I)$ be the element of $\mathrm{FIM}(\Sigma, I)$ represented by the word $u \in \Gamma^*$. Similarly (B, h) (resp. $(C_i, 1)$, $(D_i, 1)$) is the element represented by v (resp. e_i, f_i). Here $g, h \in G(\Sigma, I) = F(\Sigma_1) \times F(\Sigma_2)$ and A, B, C_i, D_i are closed subsets of $F(\Sigma_1) \times F(\Sigma_2)$. Thus, these sets can be written as $A = A_1 \times A_2$, $B = B_1 \times B_2$, $C_i = C_{i,1} \times C_{i,2}$, and $D_i = D_{i,1} \times D_{i,2}$, where $A_j, B_j, C_{i,j}, D_{i,j}$ are finite and closed subsets of $F(\Sigma_j)$ (i.e., Munn trees) for $1 \leq j \leq 2$, $1 \leq i \leq \ell$. If $g \neq h$ in $G(\Sigma, I)$ then $u \neq v$ in $\mathrm{FIM}(\Sigma, I)/P$. Thus assume that $g = h$. It remains to check in exponential time whether $\mathrm{cl}_P(A) = \mathrm{cl}_P(B)$, i.e., whether $A \subseteq \mathrm{cl}_P(B)$ and $B \subseteq \mathrm{cl}_P(A)$. Let us consider the inclusion $B \subseteq \mathrm{cl}_P(A)$. Let $(b_1, b_2) \in B = B_1 \times B_2$. We will show that we can check in exponential time whether $(b_1, b_2) \in \mathrm{cl}_P(A_1 \times A_2)$. Then we can check the inclusion $B_1 \times B_2 \subseteq \mathrm{cl}_P(A_1 \times A_2)$ in exponential time as well. In the following, we assume that $|\Sigma_1| = |\Sigma_2|$. This is not a restriction; if say $|\Sigma_1| < |\Sigma_2|$, then $F(\Sigma_1)$ can be naturally embedded into $F(\Sigma_2)$. Hence, by renaming symbols, we can assume that $A_1, A_2, B_1, B_2, C_{i,j}, D_{i,j} \subseteq F(\Theta)$ for $1 \leq j \leq 2$, $1 \leq i \leq \ell$, where Θ is an alphabet of size $|\Sigma_1| = |\Sigma_2|$.

Let G be the edge-labelled graph that results from the Cayley graph $\mathcal{C}(\Theta)$ by labelling $1 \in F(\Theta)$ with the (only) node label 1. Since $\mathcal{C}(\Theta)$ is a context-free graph (see the Appendix A), also G is context-free. We decide $(b_1, b_2) \in \mathrm{cl}_P(A_1 \times A_2)$ by constructing a formula φ of the modal μ-calculus with simultaneous fixpoints such that $(b_1, b_2) \in \mathrm{cl}_P(A_1 \times A_2)$ if and only if $(G, 1) \models \varphi$. Then the EXPTIME upper bound follows from Theorem 20 in the Appendix A.

First, for $j \in \{1, 2\}$ let

$$\psi_j(X_1, X_2) = \bigvee_{x \in A_j} \langle x^{-1} \rangle 1 \;\vee\; \bigvee_{i=1}^{\ell} \bigvee_{y \in D_{i,j}} \langle y^{-1} \rangle \left(\bigwedge_{z \in C_{i,j}} \langle z \rangle X_j \;\wedge\; \mathrm{EF} \bigwedge_{z \in C_{i,3-j}} \langle z \rangle X_{3-j} \right).$$

The formula $\psi_j(X_1, X_2)$ holds in a node p, if either $p \in A_j$ (which is expressed by $\bigvee_{x \in A_j} \langle x^{-1} \rangle 1$) or there is $1 \leq i \leq \ell$ and nodes $q, q' \in F(\Theta)$ such that $p \in qD_j$, $qC_j \subseteq X_j$ and $q'C_{3-j} \subseteq X_{3-j}$. Here it is important that the graph G is strongly connected. Hence, with the EF-operator (EF for "exists finally", see (A.1) in Appendix A) in the above formula, we can reach every node q' of

G. Then for the sentence

$$\varphi_j = [\mu(X_1, X_2).(\psi_1(X_1, X_2), \psi_2(X_1, X_2))]_j$$

we have $(G, p) \models \varphi_j$ if and only if the node p belongs to the j-th component of $\mathrm{cl}_P(A_1 \times A_2)$. Finally, we can take for φ the sentence $\langle b_1 \rangle \varphi_1 \wedge \langle b_2 \rangle \varphi_2$. $\square$

For a fixed idempotent presentation P we can again generalize a corresponding result from [21]:

Theorem 15 *Let $I \subseteq \Sigma \times \Sigma$ be an independence relation such that $D = (\Sigma \times \Sigma) \setminus I$ is transitive, and let P be an idempotent presentation over (Σ, I). Then the word problem for $\mathrm{FIM}(\Sigma, I)/P$ can be solved both in linear time on a RAM and logarithmic space on a Turing machine.*

Proof. Let us fix a finite idempotent presentation P over (Σ, I) and let $u, v \in \Gamma^*$. Both our logspace and linear time algorithm are extensions of the corresponding algorithms for the case $I = \emptyset$ from [21]. Let us first sketch the logspace algorithm.

Let $\Sigma = \bigcup_{i=1}^{n} \Sigma_i$, where the Σ_i are the cliques of (Σ, D). As in the proof of Theorem 14, it is no essential restriction to assume that $n = 2$ and $|\Sigma_1| = |\Sigma_2|$. Let $(A_1 \times A_2, (g_1, g_2))$ and $(B_1 \times B_2, (h_1, h_2))$ be the elements of $\mathrm{FIM}(\Sigma, I)$ represented by the words u and v, respectively. Hence, $A_j, B_j \subseteq F(\Sigma_j)$ and $g_j, h_j \in F(\Sigma_j)$. The equality $(g_1, g_2) = (h_1, h_2)$ can be verified in logspace, since the word problem of a free group belongs to deterministic logspace [19]. The sets A_1, A_2, B_1, and B_2 can be calculated by a logspace transducer from the words u and v. For this, we have to use the fact that the irreducible normal form of a word can be calculated by a logspace transducer, see [21]. We will present a logspace algorithm for checking the inclusion $B_1 \times B_2 \subseteq \mathrm{cl}_P(A_1 \times A_2)$.

As in the proof of Theorem 14, let Θ be an alphabet of size $|\Sigma_1| = |\Sigma_2|$. By renaming symbols, we may assume that $A_1, B_1, A_2, B_2 \subseteq F(\Theta)$. The modal μ-calculus formulas from the proof of Theorem 14 can be used in order to construct a fixed MSO-formula $\mathrm{CL}_P(X_1, X_2, Y_1, Y_2)$ over the signature of the Cayley graph $\mathcal{C}(\Theta)$ such that for all subsets $C_1, C_2, D_1, D_2 \subseteq F(\Theta)$: $\mathcal{C}(\Theta) \models \mathrm{CL}_P(C_1, C_2, D_1, D_2)$ if and only if C_1, C_2, D_1, D_2 are finite and closed subsets of the Cayley graph $\mathcal{C}(\Theta)$ containing the 1 and $D_1 \times D_2 = \mathrm{cl}_P(C_1 \times C_2)$. Define the MSO-formula

$$\text{in-cl}_P(X_1, X_2, Y_2, Y_2) = \exists Z_1, Z_2 : \mathrm{CL}_P(X_1, X_2, Z_1, Z_2) \wedge Y_1 \subseteq Z_1 \wedge Y_2 \subseteq Z_2.$$

Thus, we have to check whether

$$\mathcal{C}(\Theta) \models \text{in-cl}_P(A_1, A_2, B_1, B_2). \tag{7}$$

Here, it is important to note that since P is a fixed presentation, in-$\mathrm{cl}_P(X_1, X_2, Y_1, Y_2)$ is a fixed MSO-formula over the signature of the Cayley graph $\mathcal{C}(\Theta)$. Hence, as in [21], by using Rabin's tree theorem [31], we can reduce (7) to the membership problem for a fixed deterministic bottom-up tree automaton. The only difference to [21] is that tree nodes are not labelled with symbols from $\{0,1\} \times \{0,1\}$ but with symbols from $\{0,1\}^4$. Such a 4-tuple encodes the information, whether the tree node belongs to A_1, A_2, B_1, and B_2, respectively. Finally, we can use the fact that the membership problem for a fixed tree automaton can be solved in deterministic logspace, when the input tree is given by a pointer representation: By [20, Theorem 1], the membership problem for a fixed tree automaton can be even solved in $\mathsf{NC}^1 \subseteq \mathsf{L}$ if the input tree is represented by a well-bracketed expression string. On the other hand, as noted in [5,15], transforming the pointer representation of a tree into its expression string is possible in logspace.

This concludes the description of the logspace algorithm. A linear time algorithm follows the same line of arguments. We only have to use the following facts:

- The word problem of a finitely generated free group can be solved in linear time on a RAM [4].
- Tries for the sets $A_1, A_2, B_1, B_2 \subseteq F(\Theta)$ can be calculated in linear time.
- The membership problem for a fixed deterministic bottom-up tree automaton can be solved in linear time. $\square$

Theorem 14 and 15 give an interesting contrast to [26], which shows that the variety of E-unitary inverse monoids over an abelian cover has an undecidable word problem. The difference is that in our setting pairs (A, g) are defined over *closed* sets, whereas [26] has not this restriction. On the other hand, if $I \subseteq \Sigma \times \Sigma$ is not transitive, then we can construct a finite idempotent presentation P over (Σ, I) such that the word problem for $\mathrm{FIM}(\Sigma, I)/P$ is undecidable.

Theorem 16 *Let $I \subseteq \Sigma \times \Sigma$ be an independence relation such that $D = (\Sigma \times \Sigma) \setminus I$ is* not *transitive. Then there exists effectively a finite idempotent presentation P over (Σ, I) such that the word problem for* $\mathrm{FIM}(\Sigma, I)/P$ *is undecidable.*

Proof. Calculations of deterministic Turing machines can be coded by some labelling of the grid $\mathbb{N} \times \mathbb{N}$ in the following way. Depending on the Turing machine we fix a finite alphabet Θ with three special symbols $\$, \cent, B \in \Theta$. The role of ¢ is the left and right end marker of the tape and the role of \$ corresponds to accepting states. The B is used for a *blank*. Configurations of the Turing machine are just words $w \in ¢(\Theta \setminus \{¢\})^*¢B^*$ and we identify a configuration $w = ¢u¢ \in ¢(\Theta \setminus \{¢\})^*¢$ with any word $w' \in ¢uB^*¢B^*$. (Blanks before or after the right marker ¢ do not change the behavior of the Turing machine.) The initial configuration is just some word $u(0) = ¢u_1 \cdots u_n¢$ where $u_i \in \Theta \setminus \{¢\}$ for $1 \le i \le n$. We define a labelling $\lambda : \mathbb{N} \times \mathbb{N} \to \Theta$ as follows. Initially the labelling is $\lambda(0,t) = ¢$ for all t, $\lambda(i,0) = u_i$ for $1 \le i \le n$, $\lambda(n+1,0) = ¢$, $\lambda(i,0) = B$ for $n+1 < i$. Inductively, assume $\lambda(i,t)$ has been defined for all i and some $t > 1$ such that $\lambda(0,t)\cdots\lambda(n_t,t) = ¢u(t)¢$ is the configuration of the Turing machine at time t and $\lambda(i,t) = B$ for all $i > n_t$. We can normalize the deterministic Turing machine in such a way that there is a partially defined function $\delta : \Theta \times \Theta \times \Theta \to \Theta$ which determines the i-th position of the configuration at time $t+1$ depending only on the positions $i-1$, i, and $i+1$ of the configuration at time t. Moreover, we may assume that $\delta(¢, B, B) = ¢$ and $\delta(a, ¢, B) = B$ for all $a \in \Theta$. These rules serve to push the right marker ¢ to the right.

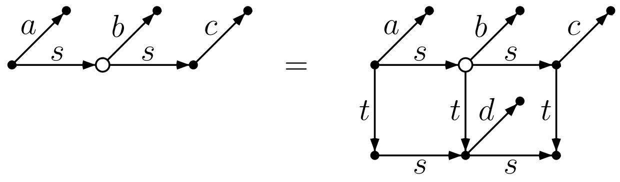
Hence, the triple $(\lambda(i-1,t), \lambda(i,t), \lambda(i+1,t))$ defines the label $\lambda(i,t+1)$ uniquely according to δ for all $i > 0$.

The question whether or not the initial configuration leads to some accepting state becomes equivalent to the question whether \$ appears at some label. This is therefore undecidable, in general.

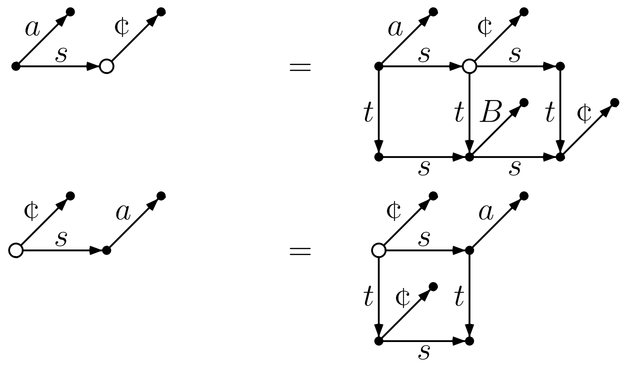
Now, we make a reduction from this undecidable problem to the word problem for $\mathrm{FIM}(\Sigma, I)/P$ for $\Sigma = \{r,s,t\}$, $I = \{(s,t),(t,s)\}$, and some fixed idempotent presentation P. This suffices to show the undecidability for every non-transitive dependence alphabet. The idempotent presentation P can be derived from Θ and δ as follows:

First, we include equations in P which allow to mimic the behaviour of the Turing machine. The elements s and t generate a two-dimensional grid where each node s^it^j corresponds to a certain string position i and time step j of the configuration of the Turing machine, and r can be used to put some information (i.e., a letter from Θ) at that node. We present the equations from P by drawing the closed subsets of the Cayley graph $\mathcal{C}(G(\Sigma, I))$ corresponding to the idempotent elements of $\mathrm{FIM}(\Sigma, I)$, where the bigger circle represents the 1. To keep notation simple, we label some edges with letters from Θ. This is a shorthand for paths labelled with a suitable encoding; for all letters except \$ we use distinct elements of the form rs^ir, $1 \le i < |\Theta|$. The letter \$ has to be encoded as a set of paths which includes the path of every other letter, thus we use $\bigcup_{i<|\Theta|} rs^ir$. Again, for simplicity we still draw a single edge labelled with \$ for this set of paths.

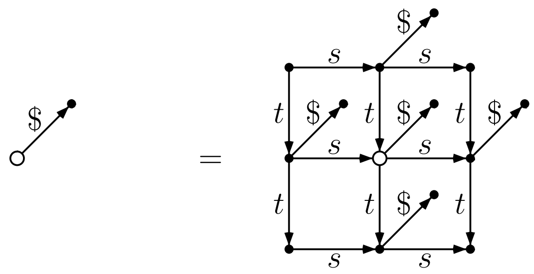
For $\delta(a,b,c) = d$, we use the following equations:



Note that we don't include the blanks after the right marker ¢ in our (s, t)-grid. Therefore, we use the following extra equations combining $\delta(a, ¢, B) = B$ and $\delta(¢, B, B) = ¢$, and for the labelling of the left vertical border.



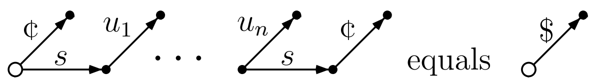
In addition, we need another equation which can fill the entire plane with \$ if there is a \$ attached to some node:



Now, for an input string $u = ¢u_1 \cdots u_n¢$ where $u_i \in \Theta \setminus \{¢\}$ for $1 \leq i \leq n$ we can ask whether we have

$$¢¢^{-1}su_1u_1^{-1} \cdots su_nu_n^{-1}s¢¢^{-1}s^{-n-1} = \$\$^{-1}.$$

This means we ask whether



in $\mathrm{FIM}(\Sigma, I)/P$.

Assume first $¢¢^{-1}su_1u_1^{-1} \cdots su_nu_n^{-1}s¢¢^{-1}s^{-n-1} = \$\$^{-1}$ in $\mathrm{FIM}(\Sigma, I)/P$. Then we can apply the rules from P from left to right until at some point we produce some \$. This defines a labelling of the grid where at some point \$

appears. For the other direction recall that we may identify a configuration $¢u¢ \in ¢(\Theta \setminus \{¢\})^*¢$ with any word in $¢uB^*¢B^*$. Therefore, using the rules from P in any order defines a marking of the grid, which corresponds to the computation of the Turing machine on the initial configuration. If the initial configuration leads to an accepting state, we produce a \$; and hence $¢¢^{-1}su_1u_1^{-1}\cdots su_nu_n^{-1}s¢¢^{-1}s^{-n-1} = \$\$^{-1}$ in $\mathrm{FIM}(\Sigma, I)/P$. $\square$

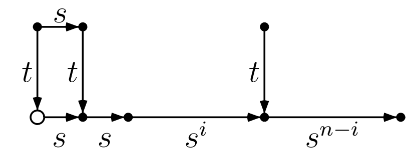
Using the Birget-Rhodes expansion [3] we can define for every group G an inverse monoid $\mathrm{IM}(G)$ as follows. The elements of $\mathrm{IM}(G)$ are the pairs (A, g) where A is a finite connected subset of $\mathcal{C}(G)$ and $1, g \in A$. Multiplication and inverses are defined by

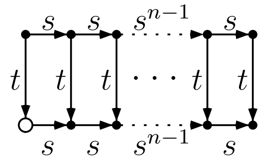
$$(A, g) \cdot (B, h) = (A \cup gB, gh) \quad \text{and} \quad (A, g)^{-1} = (g^{-1}A, g^{-1}).$$

Note that here we do not ask that subsets are closed. This makes it possible to encode letters by holes in the grid instead of adding unique new letters. Now, we can present a simpler proof an undecidability result of Meakin and Sapir [26, Theorem 4]. In fact we can state a more precise result because it applies to every abelian group G of rank at least 2, which is the best we can expect by [9]. Moreover the E-unitary monoid over G having an undecidable word problem is simply some $\mathrm{IM}(G)/P$:

Theorem 17 *Let G be a group which contains $\mathbb{Z}\times\mathbb{Z}$ as a subgroup. Then there exists an idempotent presentation P such that the word problem for $\mathrm{IM}(G)/P$ is undecidable.*

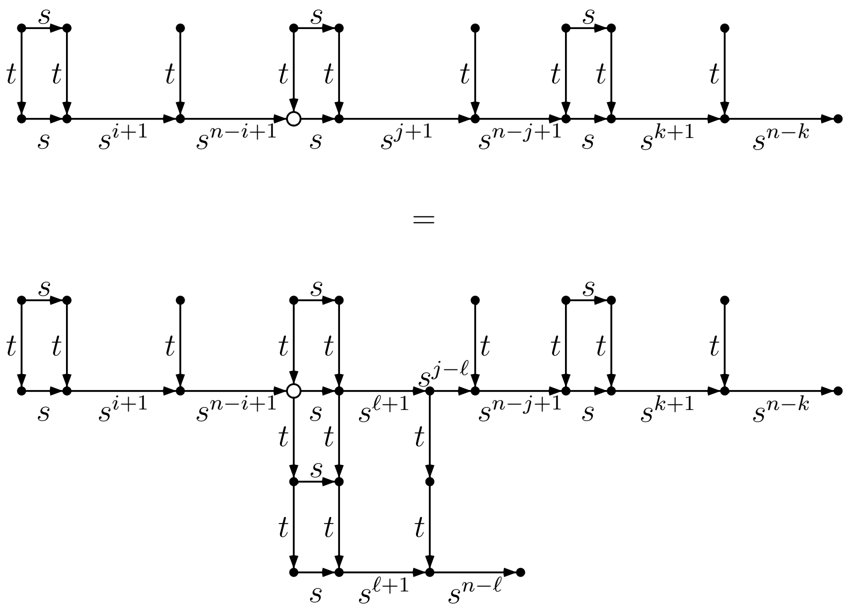
Proof. The proof is rather similar to the one of Theorem 16, so we will only sketch the differences here. Again, we reduce the acceptance problem for deterministic Turing machines to the word problem for $\mathrm{IM}(G)/P$. We also have the same types of equations in P which reproduce the labelling of the grid $\mathbb{N}\times\mathbb{N}$ corresponding to the Turing machine calculation, but we use a different encoding of the labelling. Denote the generators of the free abelian subgroup of G by s and t. Then the connected subsets of $\mathcal{C}(G)$, which correspond to idempotents in $\mathrm{IM}(G)$, may have holes in the grid generated by s and t. This suffices to encode a labelled two-dimensional grid. Let $n = |\Theta|$; for $1 \le i < n$ we encode the i-th letter of $\Theta \setminus \{\$\}$ like shown on the left in the image below, and the encoding of \$ is shown on the right.





The positions of the configuration of the Turing machine have a distance of

s^{n+3} in this representation and a new time step starts every t^2, i.e., the node (i,j) of the two-dimensional grid corresponds to the group element $s^{(n+3)i}t^{2j}$. Now, for $\delta(a,b,c) = d$, where a, b, c, and d are the i-th, j-th, k-th, and ℓ-th symbols of Θ, respectively, we get the following equation (the picture is for the case $j > \ell$):



The other equations can be translated analogously and the remaining arguments are exactly the same as in the proof of Theorem 16. □

Corollary 18 *If the dependence relation D is not transitive then there is some finite idempotent presentation P such that the submonoid membership problem is undecidable in* $\mathrm{FIM}(\Sigma, I)/P$.

Proof. By [21, Remark 7.9] we know that the submonoid membership problem for $\mathrm{FIM}(\Sigma, I)/P$ can be decidable only if the word problem for $\mathrm{FIM}(\Sigma, I)/P$ is decidable, too. Hence we can apply Theorem 16. □

However, even for a transitive dependence relation, the submonoid membership problem for $\mathrm{FIM}(\Sigma, I)$ modulo an idempotent presentation may become undecidable:

Proposition 19 *Let the graph group $G(\Sigma, I)$ contain a direct product of two free groups of rank 2, then there is some finite idempotent presentation P such that the submonoid membership problem is undecidable in* $\mathrm{FIM}(\Sigma, I)/P$.

Proof. Let $\Sigma = \{a, b, c, d\}$, $I = \{a, b\} \times \{c, d\} \cup \{c, d\} \times \{a, b\}$, and let the idempotent presentation P contain all identities $\alpha\alpha^{-1} = 1$ for $\alpha \in \Gamma$. Then $\mathrm{FIM}(\Sigma, I)/P$ is a direct product of two free groups of rank 2. By [27], this group has an undecidable submonoid membership problem. $\square$

The only remaining case is a dependence relation which consists of one non-trivial clique together with additional isolated nodes. The corresponding free partially commutative group is of the form $F \times \mathbb{Z}^k$, where F is a free group of rank at least one and $k \geq 1$. For the group $F \times \mathbb{Z}^k$ the submonoid membership problem is decidable [16], but it remains open, whether the submonoid membership problem for $\mathrm{FIM}(\Sigma, I)/P$ is decidable for every idempotent presentation P.

A Modal μ-calculus with simultaneous fixpoints over context-free graphs

In Section 5 we used an extension of *modal μ-calculus* by *simultaneous fixpoints*. In this appendix we give the basic facts. For more details see [1]. Formulas of this logic are interpreted over edge-labelled and node-labelled directed graphs. Let Σ be a finite set of edge labels, let Ξ be a finite set of node labels, and let Ω be a set of variables, which will range over sets of nodes. The syntax of the modal μ-calculus is given by the following grammar, where $X \in \Omega$, $a \in \Sigma$, and $p \in \Xi$:

$$\varphi ::= p \mid \neg p \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

Modal μ-calculus with simultaneous fixpoints extends modal μ-calculus by allowing instead of $\rho X.\varphi$ ($\rho \in \{\mu, \nu\}$) formulas of the following form, where $X_1, \dots, X_n \in \Omega$ and $1 \leq i \leq n$:

$$[\rho(X_1, \dots, X_n).(\varphi_1, \dots, \varphi_n)]_i$$

We define the semantics of formulas of the modal μ-calculus with simultaneous fixpoints w.r.t. an edge and node-labelled graph $G = (V, (E_a)_{a\in\Sigma}, (U_p)_{p\in\Xi})$ ($E_a \subseteq V \times V$ is the set of a-labelled edges, $U_p \subseteq V$ is the set of p-labelled nodes) and a valuation $\sigma : \Omega \to 2^V$. To each formula φ we assign the set $\varphi^G(\sigma) \subseteq V$ of nodes where φ evaluates to true under the valuation σ. For a valuation σ, variables $X_1, \dots, X_n \in \Omega$, and sets $U_1, \dots, U_n \subseteq V$, we define $\sigma[U_1/X_1, \dots, U_n/X_n]$ as the valuation with

$$\sigma[U_1/X_1, \dots, U_n/X_n](X_i) = U_i \quad (1 \leq i \leq n) \text{ and}$$
$$\sigma[U_1/X_1, \dots, U_n/X_n](Y) = \sigma(Y) \quad \text{for } Y \notin \{X_1, \dots, X_n\}.$$

Now we can define $\varphi^G(\sigma)$ inductively as follows:

- $p^G(\sigma) = U_p$ for $p \in \Xi$
- $(\neg p)^G(\sigma) = V \setminus U_p$ for $p \in \Xi$
- $X^G(\sigma) = \sigma(X)$ for every $X \in \Omega$
- $(\varphi \vee \psi)^G(\sigma) = \varphi^G(\sigma) \cup \psi^G(\sigma)$
- $(\varphi \wedge \psi)^G(\sigma) = \varphi^G(\sigma) \cap \psi^G(\sigma)$
- $(\langle a \rangle \varphi)^G(\sigma) = \{u \in V \mid \exists v \in V : (u, v) \in E_a \ \wedge \ v \in \varphi^G(\sigma)\}$
- $([a]\varphi)^G(\sigma) = \{u \in V \mid \forall v \in V : (u, v) \in E_a \ \Rightarrow \ v \in \varphi^G(\sigma)\}$

For a formula $[\mu(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)]_i$ we first define a monotone mapping on the lattice $(2^V)^n$ by

$$(U_1, \ldots, U_n) \mapsto (\varphi_1^G(\sigma[U_1/X_1, \ldots, U_n/X_n]), \ldots, \varphi_n^G(\sigma[U_1/X_1, \ldots, U_n/X_n])).$$

By the fixpoint theorem of Knaster and Tarski this mapping has a smallest fixpoint $(U_1', \ldots, U_n')$. We define

$$[\mu(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)]_i^G(\sigma) = U_i'.$$

The set $[\nu(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)]_i^G(\sigma)$ is defined in the same way, we only have to take the largest fixpoint. Note that in order to determine $\varphi^G(\sigma)$, only the values of the valuation σ for free variables of φ are important. In particular, if φ is a sentence (i.e., a formula where all variables are bound by fixpoint operators), then the valuation σ is not relevant and we can write φ^G instead of $\varphi^G(\sigma)$, where σ is an arbitrary valuation. For a sentence φ and a node $v \in V$ we write $(G, v) \models \varphi$ if $v \in \varphi^G$. To make formulas more readable, we introduce a few abbreviations. For a word $w = a_1 \cdots a_n \in \Sigma^*$ $(a_1, \ldots, a_n \in \Sigma)$ we write $\langle w \rangle \varphi := \langle a_1 \rangle \cdots \langle a_n \rangle \varphi$. Moreover, let

$$\mathrm{EF}\varphi := \mu Y.(\varphi \vee \bigvee_{a \in \Sigma} \langle a \rangle Y) \tag{A.1}$$

express that there exists a reachable state where φ holds.

A labelled graph is *context-free* if it is the configuration graph of a pushdown automaton, see [28] for more details on context-free graphs, a precise definition is not necessary for the purpose of this paper. In [18,35] it was shown that it is $\mathsf{EXPTIME}$-complete to check $(G, v) \models \varphi$ for a given context-free graph G (represented by a pushdown automaton), a node v of G, and a formula φ of the modal μ-calculus. This result can be easily extended to the modal μ-calculus with simultaneous fixpoints: By Bekic's Lemma [2], a formula φ with simultaneous fixpoints can be translated into an equivalent formula φ' of (ordinary) modal μ-calculus. For instance, $\mu(X, Y).(\psi(X, Y), \theta(X, Y))$ is equivalent to $\mu X.\psi(X, \mu Y.\theta(X, Y))$. This transformation may increase the size of the formula exponentially, but the number of subformulas increases only linearly. Now, for the complexity of the model-checking problem for the modal

μ-calculus over context-free graphs, only the number of subformulas is important. For instance, in the approach from [18], a modal μ-calculus formula is translated into a two-way alternating tree automaton, where the states of this automaton are basically the subformulas of the input formula. Hence, we have:

Theorem 20 ([18,35]) *The following problem is* EXPTIME*-complete:*

INPUT: A context-free graph G given by a pushdown-automaton, a node v of G, and a sentence φ of the modal μ-calculus with simultaneous fixpoints.

QUESTION: $(G, v) \models \varphi$?

Let us denote with $\mathcal{C}(\Sigma)$ the Cayley graph of the free group $F(\Sigma)$ w.r.t. the generating set Σ. We will need Theorem 20 only for the case that the context-free graph G is $\mathcal{C}(\Sigma)$, where additionally node $1 \in F(\Sigma)$ is distinguished by a node label. It is straightforward to see that this graph is indeed context-free and for a given alphabet Σ one can easily construct a pushdown automaton, which defines this graph [28].

B Petri nets and free partially commutative inverse monoids

We do not present any result which is necessary for the other parts of the paper. However we establish some link to Petri nets which allows to see partially commutative inverse monoids from some different viewpoint.

Petri nets form a well-established formalism for studying concurrent systems, see e.g. [32]. From an abstract viewpoint a Petri net system $\mathcal{N}$ is specified by a mapping $F : T \to \mathbb{N}^S \times \mathbb{N}^S$ and an initial marking $m_0 \in \mathbb{N}^S$. Elements of the set T are called *transitions* and elements of the set S are called *places*, in German *Stellen*. Vectors $m \in \mathbb{N}^S$ are called *markings*. If $m \in \mathbb{N}^S$ is a marking and t is a transition with $F(t) = (u, v)$ such that $m \geq u$ (in the component wise ordering of $\mathbb{N}^S$), then t is *enabled* and can *fire* such that the follower marking m' is defined by the vector $m - u + v$. One also writes:

$$m\,[t\rangle\, m' \quad \text{or} \quad m' = m\,[t\rangle$$

A *firing sequence* is a word $t_1 \cdots t_n \in T^*$ such that each t_i is enabled at the marking $m\,[t_1 \cdots t_{i-1}\rangle$ and then $m\,[t_1 \cdots t_i\rangle$ is defined by $(m\,[t_1 \cdots t_{i-1}\rangle)\,[t_i\rangle$.

The *reachability problem* is to decide whether a given marking m' is reachable in a given Petri net system via some firing sequence. The problem is known to be decidable by a famous result of Mayr [25,17].

Let us establish the bridge from Petri nets to partially commutative inverse monoids.

For each place p consider the free inverse monoid $\mathrm{FIM}(p)$ over one letter. Note that in this monoid we have $pp^{-1} \neq p^{-1}p$. If $F(t) = (u, v)$ and $u(p) = k, v(p) = \ell$, then we define $f(t, p) = p^{-k}p^{\ell} \in \mathrm{FIM}(p)$. The direct product $\prod_{p \in S} \mathrm{FIM}(p)$ is a free partially commutative inverse monoid over the alphabet S with $I = S \times S \setminus \mathrm{id}_S$. Hence, $\prod_{p \in S} \mathrm{FIM}(p) = \mathrm{FIM}(S, I)$. In particular, $\mathrm{FIM}(p)$ is a submonoid of $\mathrm{FIM}(S, I)$.

We extend f to a homomorphism, which is denoted again by f:

$$\begin{aligned} f : T &\to \mathrm{FIM}(S, I) \\ t &\mapsto \prod_{p \in S} f(t, p) \end{aligned}$$

Thus, every word $v = t_1 \cdots t_n \in T^*$ becomes an element $f(v) \in \prod_{p \in S} \mathrm{FIM}(p)$. Thus, we can view $f(v)$ as a pair (V, g) where $V \subseteq \mathbb{Z}^S$ is a closed subset and $g \in \mathbb{Z}^S$. We see that $m\ [t_1 \cdots t_n\rangle\ m'$ if and only if both $m + V \subseteq \mathbb{N}^S$ and $m + g = m'$. Note also that closed subsets in $\mathbb{Z}^S$ are just direct products of intervals $[k, \ell]$ with $k \leq 0 \leq \ell$.

A Petri net system is called *1-safe*, if for every place p and every reachable marking m we have $m(p) \in \{0, 1\}$. The reachability problem for 1-safe nets is known to be PSPACE-complete, see [6]. Let us see that the reachability problem for a 1-safe net $\mathcal{N}$ can be reduced to the submonoid membership problem of $\mathrm{FIM}(S, I)$. As we have seen in Theorem 9, even more general problems for $\mathrm{FIM}(S, I)$ can be solved in PSPACE. The reduction works as follows: First, we observe that $\mathbb{N}^S$ has a natural embedding in $\mathrm{FIM}(S, I)$. Thus we can view a marking as an element of $\mathrm{FIM}(S, I)$. Let m_0 be the initial marking. Then a marking m is reachable if and only if

$$(\prod_{p \in S} pp^{-1})m \in \left\{ (\prod_{p \in S} pp^{-1})m_0,\ f(t) \;\middle|\; t \in T \right\}^* .$$

The verification of this statement is a good exercise and therefore left to the reader.

How about the reachability problem for general Petri nets? Again we can give a reduction to a submonoid membership problem, but unfortunately this does not give (yet) any algorithm because we have (yet) no decidability for this type of submonoid membership problem. The obstacle is that we move to finite idempotent presentations. Consider $\mathrm{FIM}(S, I)$ modulo defining equations of the form $pp^{-1} = 1$ for all $p \in S$. We can imagine that for each $p \in S$ there is (some new type of) a transition t_p which has this effect: $f(t_p) = pp^{-1}$. Such a transition makes no sense in 1-safe nets, because usually it destroys the 1-safeness, but for general nets there is no harm in allowing such a strange type. A transition t_p is always enabled, but the effect of firing it is the identity, hence the $pp^{-1} = 1$ is satisfied. We also need an additional new place p_0 with $m(p_0) = 1$ for every marking m and $f(t, p_0) = p_0^{-1}p_0$ for every transition t. (This ensures m_0 occurs as the first item and then never again in any factorization of a marking m as defined below.)

For a moment, let $M = \mathrm{FIM}(S, I)/\{\, pp^{-1} = 1 \mid p \in S \,\}$. Note that there is a simple representation for the elements of M. The effect of the equations $pp^{-1} = 1$ is simply that the upper bounds of the intervals $[k, \ell]$ no longer matter. For a single component, we have $p^r p^{-r} = ([0, r], 0)$ and

$$\begin{aligned}&([k_1, \ell_1], g_1) \cdot ([0, r], 0) \cdot ([k_2, \ell_2], g2) = \\ &([\min\{k_1, g_1, g_1 + k_2\}, \max\{\ell_1, g_1 + r, g_1 + \ell_2\}], g_1 + g_2).\end{aligned}$$

So, by inserting a factor $p^r p^{-r}$ the upper bound can be increased by an arbitrary number, but since $k_1 \leq g_1$, nothing else will change. Thus, we can view the elements of M as pairs $(k, g) \in \mathbb{Z}^S \times \mathbb{Z}^S$. Again, the reader is invited to check that now, a marking m is reachable if and only if

$$m \in \{\, m_0, f(t) \mid t \in T \,\}^* \subseteq M.$$

Thus, a tempting generalization of the reachability problem for Petri nets would be to prove that the submonoid membership problem for $\mathrm{FIM}(S, I)/P$ is decidable, if P is a finite idempotent presentation.

References

[1] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

[2] H. Bekic. Definable operation in general algebras, and the theory of automata and flowcharts. In C. B. Jones, editor, *Programming Languages and Their Definition*, number 177 in Lecture Notes in Computer Science, pages 30–55. Springer, 1984.

[3] J.-C. Birget and J. Rhodes. Almost finite expansions of arbitrary semigroups. *Journal of Pure and Applied Algebra*, 32(3):239–287, 1984.

[4] R. V. Book. Confluent and other types of Thue systems. *Journal of the Association for Computing Machinery*, 29(1):171–182, 1982.

[5] S. R. Buss. Alogtime algorithms for tree isomorphism, comparison, and canonization. In *Kurt Gödel Colloquium 97*, pages 18–33, 1997.

[6] A. Cheng, J. Esparza, and J. Palsberg. Complexity results for 1-safe nets. *Theoretical Computer Science*, 147(1–2):117–136, 1995.

[7] V. Diekert. *Combinatorics on Traces.* Number 454 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg, 1990.

[8] V. Diekert, M. Lohrey, and A. Miller. Partially commutative inverse monoids. In R. Kralovic and P. Urzyczyn, editors, *Proceedings of the 31th International Symposium on Mathematical Foundations of Computer Science (MFCS 2006), Bratislave (Slovakia)*, number 4162 in Lecture Notes in Computer Science, pages 292–304. Springer-Verlag, 2006.

[9] V. Diekert, M. Lohrey, and N. Ondrusch. Algorithmic problems on inverse monoids over virtually-free groups. submitted for publication, 2006.

[10] V. Diekert and A. Muscholl. Solvability of equations in free partially commutative groups is decidable. *International Journal of Algebra and Computation*, 16(6):1047–1069, 2006.

[11] V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.

[12] C. Droms. Graph groups, coherence and three-manifolds. *Journal of Algebra*, 106(2):484–489, 1985.

[13] C. Droms. Subgroup of graph groups. *Journal of Algebra*, 110:519–522, 1987.

[14] E. Fredkin. Trie memory. *Communications of the Association for Computing Machinery*, 3(9):490–499, 1960.

[15] B. Jenner, P. McKenzie, and J. Torán. A note on the hardness of tree isomorphism. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, pages 101–105. IEEE Computer Society Press, 1998.

[16] M. Kambites, P. V. Silva, and B. Steinberg. On the rational subset problem for groups. *Journal of Algebra*, 309(2):622–639, 2007.

[17] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proceedings of the 14th annual ACM symposium on Theory of computing (STOC)*, pages 267–281, 1982.

[18] O. Kupferman and M. Y. Vardi. An automata-theoretic approach to reasoning about infinite-state systems. In E. A. Emerson and A. P. Sistla, editors, *Proceedings of the 12th International Conference on Computer Aided Verification (CAV 2000), Chiacago (USA)*, number 1855 in Lecture Notes in Computer Science, pages 36–52. Springer-Verlag, 2000.

[19] R. J. Lipton and Y. Zalcstein. Word problems solvable in logspace. *Journal of the Association for Computing Machinery*, 24(3):522–526, 1977.

[20] M. Lohrey. On the parallel complexity of tree automata. In A. Middeldorp, editor, *Proceedings of the 12th International Conference on Rewrite Techniques and Applications (RTA 2001), Utrecht (The Netherlands)*, number 2051 in Lecture Notes in Computer Science, pages 201–215. Springer-Verlag, 2001.

[21] M. Lohrey and N. Ondrusch. Inverse monoids: decidability and complexity of algebraic questions. to appear in *Information and Computation*, short version appeared in *Proceedings of MFCS 2005*, number 3618 in Lecture Notes in Computer Science, pages 664–675. Springer, 2005.

[22] S. Margolis and J. Meakin. E-unitary inverse monoids and the Cayley graph of a group presentation. *Journal of Pure and Applied Algebra*, 58(1):45–76, 1989.

[23] S. Margolis and J. Meakin. Inverse monoids, trees, and context-free languages. *Transactions of the American Mathematical Society*, 335(1):259–276, 1993.

[24] S. Margolis, J. Meakin, and M. Sapir. Algorithmic problems in groups, semigroups and inverse semigroups. In J. Fountain, editor, *Semigroups, Formal Languages and Groups*, pages 147–214. Kluwer, 1995.

[25] E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal on Computing*, 13:441–460, 1984.

[26] J. Meakin and M. Sapir. The word problem in the variety of inverse semigroups with abelian covers. *Journal of the London Mathematical Society, II. Series*, 53(1):79–98, 1996.

[27] K. A. Mihailova. The occurrence problem for direct products of groups. *Math. USSR Sbornik*, 70:241–251, 1966. English translation.

[28] D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37(1):51–75, 1985.

[29] W. Munn. Free inverse semigroups. *Proc. London Math. Soc.*, 30:385–404, 1974.

[30] M. Petrich. *Inverse semigroups*. Wiley, 1984.

[31] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[32] W. Reisig. *Petri Nets (an Introduction)*. Number 4 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin Heidelberg, 1985.

[33] J. Stephen. Presentations of inverse monoids. *Journal of Pure and Applied Algebra*, 63:81–112, 1990.

[34] A. A. Veloso da Costa. *Γ-Produtos de Monóides e Semigrupos*. PhD thesis, Universidade do Porto, Faculdade de Ciências, 2003.

[35] I. Walukiewicz. Pushdown Processes: Games and Model-Checking. *Information and Computation*, 164(2):234–263, 2001.

[36] C. Wrathall. The word problem for free partially commutative groups. *Journal of Symbolic Computation*, 6(1):99–104, 1988.