# Inverse monoids: decidability and complexity of algebraic questions

Markus Lohrey , Nicole Ondrusch

*Universität Stuttgart, FMI, Germany*
{lohrey,ondrusch}@informatik.uni-stuttgart.de

**Abstract**

This paper investigates the word problem for inverse monoids generated by a set $\Gamma$ subject to relations of the form $e = f$, where $e$ and $f$ are both idempotents in the free inverse monoid generated by $\Gamma$. It is shown that for every fixed monoid of this form the word problem can be solved both in linear time on a RAM as well as in deterministic logarithmic space, which solves an open problem of Margolis and Meakin. For the uniform word problem, where the presentation is part of the input, EXPTIME-completeness is shown. For the Cayley-graphs of these monoids, it is shown that the first-order theory with regular path predicates is decidable. Regular path predicates allow to state that there is a path from a node $x$ to a node $y$ that is labeled with a word from some regular language. As a corollary, the decidability of the generalized word problem is deduced.

## 1 Introduction

The decidability and complexity of algebraic questions in various kinds of structures is a classical topic at the borderline of computer science and mathematics. The most basic algorithmic question concerning algebraic structures is the word problem, which asks whether two given expressions denote the same element of the underlying structure. Markov [29] and Post [38] proved independently that the word problem for finitely presented monoids is undecidable in general. This result can be seen as one of the first undecidability results dealing with algebraic structures. Later, Novikov [35] and Boone [3] extended the result of Markov and Post to finitely presented groups.

In this paper, we are interested in a class of monoids that lies somewhere between groups and general monoids: inverse monoids [37]. In the same way as groups can be represented by sets of permutations, inverse monoids can be represented by sets of partial injections [37]. Algorithmic questions for inverse

*Preprint submitted to Elsevier Science* *15 December 2006*

monoids received increasing attention in the past and inverse monoid theory found several applications in combinatorial group theory, see e.g. [1,8,9,12,27,30,41,43,44] and the survey [28]. In [27], Margolis and Meakin presented a large class of finitely presented inverse monoids with decidable word problems. An inverse monoid from that class is of the form $\mathrm{FIM}(\Gamma)/P$, where $\mathrm{FIM}(\Gamma)$ is the free inverse monoid generated by the set $\Gamma$ and $P$ is a presentation consisting of a finite number of identities between idempotents of $\mathrm{FIM}(\Gamma)$; we call such a presentation idempotent. In fact, in [27] it is shown that even the uniform word problem for idempotent presentations is decidable. In this problem, also the presentation is part of the input. An alternative proof for the decidability of the uniform word problem was given in [43].

The decidability proof of Margolis and Meakin uses Rabin's seminal tree Theorem [39], concerning the decidability of the monadic second-order theory of the complete binary tree. From the view point of complexity, the use of Rabin's tree Theorem is somewhat unsatisfactory, because it leads to a nonelementary algorithm for the word problem. Therefore, in [27] the question for a more efficient approach was asked. A partial answer was obtained in [1], where it was shown that for an idempotent presentation with only one identity the word problem can be solved in polynomial time. In Section 6 we present a full solution to the question of Margolis and Meakin: by using tree automata techniques we show that for every fixed idempotent presentation $P$ the word problem for $\mathrm{FIM}(\Gamma)/P$ can be solved both in linear time on a RAM as well as in deterministic logarithmic space. For the uniform word problem for idempotent presentations we prove completeness for EXPTIME (deterministic exponential time). Similarly to the method of Margolis and Meakin, we use results from logic for the EXPTIME upper bound. But instead of translating the uniform word problem into monadic second-order logic over the complete binary tree, we exploit a translation into the modal $\mu$-calculus, which is a popular logic for the verification of reactive systems. Then, we can use a result from [19,49] stating that the model-checking problem of the modal $\mu$-calculus over context-free graphs [33] is EXPTIME-complete.

In Section 7 we will investigate Cayley-graphs of inverse monoids of the form $\mathrm{FIM}(\Gamma)/P$. The Cayley-graph of a finitely generated monoid $\mathcal{M}$ w.r.t. a finite generating set $\Gamma$ is a $\Gamma$-labeled directed graph with node set $\mathcal{M}$ and an $a$-labeled edge from a node $x$ to a node $y$ if $y = xa$ in $\mathcal{M}$. Cayley-graphs of groups are a fundamental tool in combinatorial group theory [26] and serve as a link to other fields like topology, graph theory, and automata theory, see, e.g., [32,33]. Here we consider Cayley-graphs of monoids from a logical point of view, see [5,20,21] for previous results in this direction. In [5] it was shown that the monadic second-order theory of the Cayley-graph of the free inverse monoid generated by only one element is undecidable. In Section 7 we present a still quite powerful fragment of monadic second-order logic, which remains decidable for Cayley-graphs of inverse monoids of the form $\mathrm{FIM}(\Gamma)/P$ (for $P$

an idempotent presentation). More precisely, we consider an expansion $G_{\mathrm{reg}}$ of the Cayley-graph $G$ of a monoid $\mathcal{M}$ that contains for every regular language $L$ over the generators of $\mathcal{M}$ a binary predicate $\mathrm{reach}_L$. Two nodes $u$ and $v$ of $G$ are related by $\mathrm{reach}_L$ if there exists a path from $u$ to $v$ in the Cayley-graph $G$, which is labeled with a word from the language $L$. It is not hard to translate first-order formulas over this expansion $G_{\mathrm{reg}}$ into monadic second-order formulas over the (plain) Cayley-graph $G$. Our main result of Section 7 states that $G_{\mathrm{reg}}$ has a decidable first-order theory, whenever the underlying monoid is of the form $\mathrm{FIM}(\Gamma)/P$ for an idempotent presentation $P$ (Theorem 15). An immediate corollary of this result is that the generalized word problem of $\mathrm{FIM}(\Gamma)/P$ is decidable. The generalized word problem asks whether for given elements $w, w_1, \ldots, w_n \in \mathrm{FIM}(\Gamma)/P$, $w$ belongs to the submonoid of $\mathrm{FIM}(\Gamma)/P$ generated by $w_1, \ldots, w_n$. Our decidability result for Cayley-graphs should be also compared with the undecidability result for the existential theory of the free inverse monoid $\mathrm{FIM}(\{a,b\})$ [41], which consists of all true statements over $\mathrm{FIM}(\{a,b\})$ of the form $\exists x_1 \cdots \exists x_m : \varphi$, where $\varphi$ is a boolean combination of word equations (with constant).

A short version of this paper appeared in [25].

## 2 Preliminaries

Let $\Gamma$ be a finite alphabet. The *empty word* over $\Gamma$ is denoted by $\varepsilon$. Let $s = a_1 \cdots a_n \in \Gamma^*$ be a word over $\Gamma$, where $n \geq 0$ and $a_1, \ldots, a_n \in \Gamma$ for $1 \leq i \leq n$. The *length* of $s$ is $|s| = n$. Furthermore for $a \in \Gamma$ we define $|s|_a = |\{i \mid a_i = a\}|$. For $1 \leq i \leq n$ let $s[i] = a_i$ and for $1 \leq i \leq j \leq n$ let $s[i,j] = a_i a_{i+1} \cdots a_j$. If $i > j$ we set $s[i,j] = \varepsilon$. We denote with $\Gamma^{-1} = \{a^{-1} \mid a \in \Gamma\}$ a disjoint copy of $\Gamma$. For $a^{-1} \in \Gamma^{-1}$ we define $(a^{-1})^{-1} = a$; thus, ${}^{-1}$ becomes an involution on the alphabet $\Gamma \cup \Gamma^{-1}$. We extend this involution to words from $(\Gamma \cup \Gamma^{-1})^*$ by setting $(a_1 \cdots a_n)^{-1} = a_n^{-1} \cdots a_1^{-1}$, where $a_i \in \Gamma \cup \Gamma^{-1}$. The set of all regular languages over an alphabet $\Gamma$ will be denoted by $\mathrm{REG}(\Gamma)$.

We assume that the reader has some basic background in complexity theory [36]. We will make use of alternating Turing-machines, see [7] for more details. Roughly speaking, an *alternating Turing-machine* $T = (Q, \Sigma, \delta, q_0, q_f)$ (where $Q$ is the state set, $\Sigma$ is the tape alphabet, $\delta$ is the transition relation, $q_0$ is the initial state, and $q_f$ is the unique accepting state) is a nondeterministic Turing-machine, where the set of non-final states $Q \setminus \{q_f\}$ is partitioned into two sets: $Q_\exists$ (existential states) and $Q_\forall$ (universal states). We assume that $T$ cannot make transitions out of the accepting state $q_f$. A configuration $C$ with current state $q$ is accepting, if

- $q = q_f$, or

- $q \in Q_\exists$ and there exists a successor configuration of $C$ that is accepting, or
- $q \in Q_\forall$ and every successor configuration of $C$ is accepting.

An input word $w$ is accepted by $T$ if the corresponding initial configuration is accepting. It is known that EXPTIME (deterministic exponential time) equals APSPACE (the class of all problems that can be accepted by an alternating Turing-machine in polynomial space) [7].

## 3 Relational Structures and Logic

See [15] for more details on the subject of this section. A signature is a countable set $\mathcal{S}$ of relational symbols, where each relational symbol $R \in \mathcal{S}$ has an associated arity $n_R$. A (relational) structure over the signature $\mathcal{S}$ is a tuple $\mathcal{A} = (A, (R^{\mathcal{A}})_{R \in \mathcal{S}})$, where $A$ is a set (the universe of $\mathcal{A}$) and $R^{\mathcal{A}}$ is a relation of arity $n_R$ over the set $A$, which interprets the relational symbol $R$. We will assume that every signature contains the equality symbol $=$ and that $=^{\mathcal{A}}$ is the identity relation on the set $A$. As usual, a constant $c \in A$ can be encoded by the unary relation $\{c\}$. Usually, we denote the relation $R^{\mathcal{A}}$ also with $R$. For $B \subseteq A$ we define the restriction $\mathcal{A}{\restriction}B = (B, (R^{\mathcal{A}} \cap B^{n_R})_{R \in \mathcal{S}})$; it is again a structure over the signature $\mathcal{S}$.

Next, let us introduce *monadic second-order logic (MSO-logic).* Let $\mathbb{V}_1$ (resp. $\mathbb{V}_2$) be a countably infinite set of *first-order variables* (resp. *second-order variables*) which range over elements (resp. subsets) of the universe $A$. First-order variables (resp. second-order variables) are denoted $x$, $y$, $z$, $x'$, etc. (resp. $X$, $Y$, $Z$, $X'$, etc.). *MSO-formulas* over the signature $\mathcal{S}$ are constructed from the atomic formulas $R(x_1, \ldots, x_{n_R})$ and $x \in X$ (where $R \in \mathcal{S}$, $x_1, \ldots, x_{n_R}, x \in \mathbb{V}_1$, and $X \in \mathbb{V}_2$) using the boolean connectives $\neg, \wedge$, and $\vee$, and quantifications over variables from $\mathbb{V}_1$ and $\mathbb{V}_2$. The notion of a free occurrence of a variable is defined as usual. A formula without free occurrences of variables is called an *MSO-sentence.* If $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ is an MSO-formula such that at most the first-order variables among $x_1, \ldots, x_n$ and the second-order variables among $X_1, \ldots, X_m$ occur freely in $\varphi$, and $a_1, \ldots, a_n \in A$, $A_1, \ldots, A_m \subseteq A$, then $\mathcal{A} \models \varphi(a_1, \ldots, a_n, A_1, \ldots, A_m)$ means that $\varphi$ evaluates to true in $\mathcal{A}$ if the free variable $x_i$ (resp. $X_j$) evaluates to $a_i$ (resp. $A_j$). The *MSO-theory* of $\mathcal{A}$, denoted by $\mathrm{MSOTh}(\mathcal{A})$, is the set of all MSO-sentences $\varphi$ such that $\mathcal{A} \models \varphi$. For an MSO-formula $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ and a variable $Y \in \mathbb{V}_2 \setminus \{X_1, \ldots, X_m\}$ we need the relativation $\varphi{\restriction}_Y(x_1, \ldots, x_n, X_1, \ldots, X_m, Y)$. It is inductively defined by restricting every quantifier in $\varphi$ to the set $Y$. Then for all $B \subseteq A$ and all $a_1, \ldots, a_n \in B$, $A_1, \ldots, A_m \subseteq B$ we have $\mathcal{A}{\restriction}_B \models \varphi(a_1, \ldots, a_n, A_1, \ldots, A_m)$ if and only if $\mathcal{A} \models \varphi{\restriction}_Y(a_1, \ldots, a_n, A_1, \ldots, A_m, B)$.

**Remark 1** *We will use the well-known fact that the reflexive and transitive closure $E^*$ of a binary relation $E$ can be defined in MSO: if* $\mathrm{reach}(x, y)$ *is the formula*

$$\forall X : ((x \in X \ \wedge \ \forall u, v : (u \in X \wedge E(u, v) \ \Rightarrow \ v \in X)) \ \Rightarrow \ y \in X),$$

*then for every directed graph $G = (V, E)$ and all nodes $s, t \in V$ we have*

$$G \models \mathrm{reach}(s, t) \text{ if and only if } (s, t) \in E^*.$$

*Another important fact is that finiteness of a subset of a finitely-branching tree can be expressed in MSO, i.e., there is an MSO-formula* $\mathrm{fin}(X)$ *(over the signature containing a binary relation symbol $E$) such that for every (finitely-branching and undirected) tree $T = (V, E)$ and all subsets $U \subseteq V$ we have $T \models \mathrm{fin}(U)$ if and only if $U$ is finite, see also [39, Lemma 1.8]. First, let us define two auxiliary formulas, where $N(x)$ denotes the set $\{y \in V \mid (x, y) \in E\}$:*

$$\begin{aligned}
\omega\text{-}path(x, X) = {} & x \in X \ \wedge \ |N(x) \cap X| = 1 \ \wedge \\
& \forall y \in X \setminus \{x\} : |N(y) \cap X| = 2 \ \wedge \\
& \forall y \in X : \mathrm{reach}\restriction_X(x, y, X) \\
\mathit{fin}\text{-}path(x, y, X) = {} & (X = \{x\} \wedge x = y) \ \vee \ (x \neq y \ \wedge \ x, y \in X \ \wedge \\
& \quad |N(x) \cap X| = |N(y) \cap X| = 1 \ \wedge \\
& \quad \forall z \in X \setminus \{x, y\} : |N(z) \cap X| = 2 \ \wedge \\
& \quad \forall z \in X : \mathrm{reach}\restriction_X(x, z, X))
\end{aligned}$$

*Then we have $T \models \omega$-path$(u, U)$ if and only if $U$ is an $\omega$-path starting in node $u$, whereas $T \models$ fin-path$(u, v, U)$ if and only if $U$ is a finite path with end points $u$ and $v$. Now $U \subseteq V$ is finite if and only if the following holds:*

$$\begin{aligned}
& \exists r \, \exists X : \forall x : (x \in X \Leftrightarrow \exists y \in U \, \exists Y : (\mathit{fin}\text{-}path(r, y, Y) \wedge x \in Y)) \wedge \\
& \qquad \neg \exists Z : (\omega\text{-}path(r, Z) \wedge Z \subseteq X)
\end{aligned}$$

*We select first an arbitrary root $r$. Then the formula $\forall x : (x \in X \Leftrightarrow \exists y \in U \, \exists Y : (\mathit{fin}\text{-}path(r, y, Y) \wedge x \in Y))$ says that $X$ is the upward-closure of the set $U$, when $r$ is the root of the tree. Finally, we say that there does not exist an infinite path $Z$ that is contained in $X$. Since $T$ is finitely-branching, by König's lemma this is equivalent to the fact that $X$ (and hence $U$) is finite.*

A *first-order formula* over the signature $\mathcal{S}$ is an MSO-formula that does not contain any occurrences of second-order variables. In particular, first-order formulas do not contain atomic subformulas of the form $x \in X$. The *first-order theory* $\mathrm{FOTh}(\mathcal{A})$ of the structure $\mathcal{A}$ is the set of all first-order sentences $\varphi$ such that $\mathcal{A} \models \varphi$.

In Section 6 we will make use of the *modal $\mu$-calculus*, which is a popular logic for the verification of reactive systems, see [48] for more details. Formulas of this logic are interpreted over edge-labeled directed graphs. Let $\Sigma$ be a finite set of edge labels. The syntax of the modal $\mu$-calculus is given by the following grammar:

$$\varphi ::= \text{true} \mid \text{false} \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

Here $X \in \mathbb{V}_2$ is a second-order variable ranging over sets of nodes and $a \in \Sigma$. Variables from $\mathbb{V}_2$ are bounded by the $\mu$- and $\nu$-operator. We define the semantics of the modal $\mu$-calculus w.r.t. an edge-labeled graph $G = (V, (E_a)_{a\in\Sigma})$ ($E_a \subseteq V \times V$ is the set of all $a$-labeled edges) and a valuation $\sigma : \mathbb{V}_2 \to 2^V$. To each formula $\varphi$ we assign the set $\varphi^G(\sigma) \subseteq V$ of nodes where $\varphi$ evaluates to true under the valuation $\sigma$. For a valuation $\sigma$, a variable $X \in \mathbb{V}_2$, and a set $U \subseteq V$ define $\sigma[U/X]$ as the valuation with $\sigma[U/X](X) = U$ and $\sigma[U/X](Y) = \sigma(Y)$ for $X \neq Y$. Now we can define $\varphi^G(\sigma)$ inductively as follows:

- $\text{true}^G(\sigma) = V$, $\text{false}^G(\sigma) = \emptyset$
- $X^G(\sigma) = \sigma(X)$ for every $X \in \mathbb{V}_2$
- $(\varphi \vee \psi)^G(\sigma) = \varphi^G(\sigma) \cup \psi^G(\sigma)$
- $(\varphi \wedge \psi)^G(\sigma) = \varphi^G(\sigma) \cap \psi^G(\sigma)$
- $(\langle a \rangle\varphi)^G(\sigma) = \{u \in V \mid \exists v \in V : (u, v) \in E_a \ \wedge \ v \in \varphi^G(\sigma)\}$
- $([a]\varphi)^G(\sigma) = \{u \in V \mid \forall v \in V : (u, v) \in E_a \ \Rightarrow \ v \in \varphi^G(\sigma)\}$
- $(\mu X.\varphi)^G(\sigma) = \bigcap\{U \subseteq V \mid \varphi^G(\sigma[U/X]) \subseteq U\}$
- $(\nu X.\varphi)^G(\sigma) = \bigcup\{U \subseteq V \mid U \subseteq \varphi^G(\sigma[U/X])\}$

The set $(\mu X.\varphi)^G(\sigma)$ is the smallest fixpoint of the monotonic mapping $U \mapsto \varphi^G(\sigma[U/X])$, whereas $(\nu X.\varphi)^G(\sigma)$ is the largest fixpoint of this mapping. Note that in order to determine $\varphi^G(\sigma)$, only the values of the valuation $\sigma$ for free variables of $\varphi$ are important. In particular, if $\varphi$ is a sentence (i.e., a formula where all variables are bound by fixpoint operators), then the valuation $\sigma$ is not relevant and we can write $\varphi^G$ instead of $\varphi^G(\sigma)$, where $\sigma$ is an arbitrary valuation. For a sentence $\varphi$ and a node $v \in V$ we write $(G, v) \models \varphi$ if $v \in \varphi^G$. It is known that for every sentence $\varphi$ of the modal $\mu$-calculus one can construct an MSO-formula $\psi(x)$ such that for every node $v \in V$: $(G, v) \models \varphi$ if and only if $G \models \psi(v)$.

A *context-free graph* [33] is the transition graph of a pushdown automaton, i.e., nodes are the configurations of a given pushdown automaton, and edges are given by the transitions of the automaton. A more formal definition is not necessary for the purpose of this paper. We will only need the following result:

**Theorem 2 ([19,49])** *The following problem is in EXPTIME:*

*INPUT: A pushdown automaton $A$ defining a context-free graph $G(A)$, a node $v$ of $G(A)$, and a formula $\varphi$ of the modal $\mu$-calculus*

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

*Moreover, there exists already a fixed formula $\varphi$ for which this question becomes EXPTIME-complete.*

## 4 Word problems and Cayley-graphs

Let $\mathcal{M} = (M, \circ, 1)$ be a finitely generated monoid with identity 1 and let $\Sigma$ be a finite generating set for $\mathcal{M}$, i.e., $\Sigma \subseteq M$ and the canonical morphism $h : \Sigma^* \to \mathcal{M}$ is surjective. The *word problem* for $\mathcal{M}$ w.r.t. $\Sigma$ is the following problem:

INPUT: Words $u, v \in \Sigma^*$

QUESTION: $h(u) = h(v)$?

The following fact is well-known:

**Proposition 3** *Let $\mathcal{M}$ be a finitely generated monoid and let $\Sigma_1$ and $\Sigma_2$ be two finite generating sets for $\mathcal{M}$. Then the word problem for $\mathcal{M}$ w.r.t. $\Sigma_1$ is logspace reducible to the word problem for $\mathcal{M}$ w.r.t. $\Sigma_2$.*

Thus, the computational complexity of the word problem does not depend on the underlying set of generators. Since we are only interested in the complexity (resp. decidability) status of word problems, we can just speak of the word problem for a given monoid.

The *Cayley-graph* of $\mathcal{M}$ w.r.t. $\Sigma$ is the following relational structure:

$$\mathcal{C}(\mathcal{M}, \Sigma) = (M, (\{(u, v) \in M \times M \mid u \circ a = v\})_{a \in \Sigma}, 1)$$

It is a rooted (1 is the root) directed graph, where every edge has a label from $\Sigma$ and $\{(u, v) \mid u \circ a = v\}$ is the set of $a$-labeled edges. Since $\Sigma$ generates $\mathcal{M}$, every $u \in M$ is reachable from the root 1.

Cayley-graphs of groups play an important role in combinatorial group theory [26], see also the survey of Schupp [42]. Cayley-graphs of monoids received less attention, see e.g. [6,18] for some recent work. In [24,45,46], Cayley-graphs of automatic monoids are investigated.

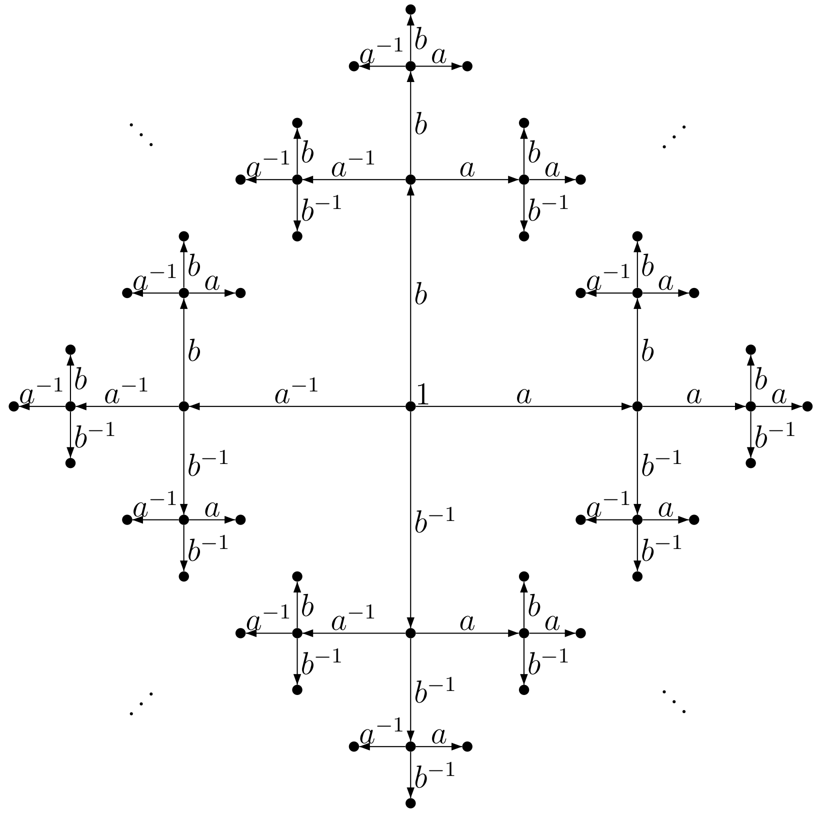



Fig. 1. The Cayley-graph $\mathcal{C}(\{a,b\})$ of the free group $\mathrm{FG}(\{a,b\})$

The *free group* $\mathrm{FG}(\Gamma)$ generated by the set $\Gamma$ is the quotient monoid

$$\mathrm{FG}(\Gamma) = (\Gamma \cup \Gamma^{-1})^*/\delta,$$

where $\delta$ is the smallest congruence on $(\Gamma \cup \Gamma^{-1})^*$ that contains all pairs $(bb^{-1}, \varepsilon)$ for $b \in \Gamma \cup \Gamma^{-1}$. Let

$$\gamma : (\Gamma \cup \Gamma^{-1})^* \to \mathrm{FG}(\Gamma)$$

denote the canonical morphism mapping a word $u \in (\Gamma \cup \Gamma^{-1})^*$ to the group element represented by $u$. It is well known that for every $u \in (\Gamma \cup \Gamma^{-1})^*$ there exists a unique word $r(u) \in (\Gamma \cup \Gamma^{-1})^*$ (the *reduced normal form of* $u$) such that $\gamma(u) = \gamma(r(u))$ and $r(u)$ does not contain a factor of the form $bb^{-1}$ for $b \in \Gamma \cup \Gamma^{-1}$. The word $r(u)$ can be calculated from $u$ in linear time [2]. It holds $\gamma(u) = \gamma(v)$ if and only if $r(u) = r(v)$.

The Cayley-graph of $\mathrm{FG}(\Gamma)$ w.r.t. the standard generating set $\Gamma \cup \Gamma^{-1}$ will be denoted by $\mathcal{C}(\Gamma)$; it is a finitely-branching tree and a context-free graph [33]. Figure 1 shows a finite portion of $\mathcal{C}(\{a,b\})$. Here, and in the following, we only draw one directed edge between two points. Thus, for every drawn $x$-labeled edge we omit the $x^{-1}$-labeled reversed edge.

The concrete shape of a Cayley-graph $\mathcal{C}(\mathcal{M}, \Sigma)$ depends on the chosen set of generators $\Sigma$. Nevertheless, and similarly to the word problem, the chosen generating set has no influence on the decidability (or complexity) of the first-order (resp. monadic second-order) theory of the Cayley-graph:

**Proposition 4 ([21])** *Let $\Sigma_1$ and $\Sigma_2$ be finite generating sets for the monoid $\mathcal{M}$. Then the first-order theory of $\mathcal{C}(\mathcal{M}, \Sigma_1)$ is logspace reducible to the first-order theory of $\mathcal{C}(\mathcal{M}, \Sigma_2)$ and the same holds for the MSO-theories.*

Thus, similarly to the word problem, we will just speak of the Cayley-graph of a monoid in statements concerning the complexity (resp. decidability) of the first-order (monadic second-order) theory of Cayley-graphs.

It is easy to see that the decidability of the first-order theory of the Cayley-graph implies the decidability of the word problem. On the other hand, there exists a finitely presented monoid for which the word problem is decidable, but the first-order theory of the Cayley-graph is undecidable, see [21]. When restricting to groups, the situation is different: The Cayley-graph of a finitely generated group has a decidable first-order theory if and only if the group has a decidable word problem [20]. Moreover, the Cayley-graph of a finitely generated group has a decidable monadic second-order theory if and only if the group is virtually free (i.e., has a free subgroup of finite index) [20,33]. We will only need the latter result for the Cayley-graph $\mathcal{C}(\Gamma)$ of the free group $\mathrm{FG}(\Gamma)$:

**Theorem 5 ([33])** *For every finite set $\Gamma$, $\mathrm{MSOTh}(\mathcal{C}(\Gamma))$ is decidable.*

**Remark 6** *It is known that already the MSO-theory of $\mathbb{Z}$ with the successor function is decidable, but not elementary decidable [31], i.e., the running time of every algorithm for deciding this theory cannot be bounded by an exponent tower of fixed height. It follows that also the complexity of $\mathrm{MSOTh}(\mathcal{C}(\Gamma))$ is not elementary for every nonempty finite alphabet $\Gamma$.*

## 5 Inverse Monoids

A monoid $\mathcal{M}$ is called an *inverse monoid* if for every $m \in \mathcal{M}$ there is a *unique* $m^{-1} \in \mathcal{M}$ such that $m = mm^{-1}m$ and $m^{-1} = m^{-1}mm^{-1}$. For detailed reference on inverse monoids see [37]; here we only recall the basic notions. The class of inverse monoids forms a variety of algebras (with respect to the operations of multiplication, inversion, and the identity element). Thus, it follows from universal algebra that *free inverse monoids* exist. The free inverse monoid generated by a set $\Gamma$ is denoted by $\mathrm{FIM}(\Gamma)$. We have

$$\mathrm{FIM}(\Gamma) \simeq (\Gamma \cup \Gamma^{-1})^*/\rho,$$

where $\rho$ is the smallest congruence on the free monoid $(\Gamma\cup\Gamma^{-1})^*$ which contains for all words $v, w \in (\Gamma\cup\Gamma^{-1})^*$ the pairs $(w, ww^{-1}w)$ and $(ww^{-1}vv^{-1}, vv^{-1}ww^{-1})$ (which are also called the Vagner equations). An element $x$ of an inverse monoid $\mathcal{M}$ is idempotent (i.e., $x^2 = x$) if and only if $x$ is of the form $mm^{-1}$ for some $m \in \mathcal{M}$. Hence, by the Vagner equations, idempotent elements in an inverse monoid commute. Let

$$\alpha : (\Gamma \cup \Gamma^{-1})^* \to \mathrm{FIM}(\Gamma)$$

denote the canonical morphism mapping a word $u \in (\Gamma\cup\Gamma^{-1})^*$ to the element of $\mathrm{FIM}(\Gamma)$ represented by $u$. Since the Vagner equations are true in the free group $\mathrm{FG}(\Gamma)$, there exists a morphism

$$\beta : \mathrm{FIM}(\Gamma) \to \mathrm{FG}(\Gamma)$$

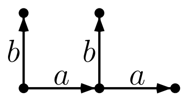
such that $\gamma = \beta \circ \alpha$, where $\gamma : (\Gamma\cup\Gamma^{-1})^* \to \mathrm{FG}(\Gamma)$ is the canonical morphism from the previous section.

The elements of the free inverse monoid $\mathrm{FIM}(\Gamma)$ can be also represented via *Munn trees*: The Munn tree $\mathrm{MT}(u)$ of $u \in (\Gamma\cup\Gamma^{-1})^*$ is a finite and connected subset of the Cayley-graph $\mathcal{C}(\Gamma)$ of the free group $\mathrm{FG}(\Gamma)$; it is defined by

$$\mathrm{MT}(u) = \{\gamma(v) \in \mathrm{FG}(\Gamma) \mid \exists w \in (\Gamma \cup \Gamma^{-1})^* : u = vw\}.$$

In other words, $\mathrm{MT}(u)$ is the set of all nodes along the unique path in $\mathcal{C}(\Gamma)$ that starts in 1 and that is labeled with the word $u$. We identify $\mathrm{MT}(u)$ with the subtree $\mathcal{C}(\Gamma)\restriction_{\mathrm{MT}(u)}$ of $\mathcal{C}(\Gamma)$.

**Example 7** *The Munn tree of $bb^{-1}abb^{-1}a$ looks as follows:*




Munn's Theorem [34] states that for all $u, v \in (\Gamma \cup \Gamma^{-1})^*$,

$$\alpha(u) = \alpha(v) \quad \Leftrightarrow \quad (r(u) = r(v) \text{ (i.e., } \gamma(u) = \gamma(v)) \;\wedge\; \mathrm{MT}(u) = \mathrm{MT}(v).$$

Thus, the element $\alpha(u) \in \mathrm{FIM}(\Gamma)$ can be uniquely represented by the pair $(\mathrm{MT}(u), r(u))$. Vice versa, for every reduced word $s \in r((\Gamma\cup\Gamma^{-1})^*)$ and every finite and connected set $U \subseteq \mathrm{FG}(\Gamma)$ with $1, \gamma(s) \in U$ we can find a word $u$ (in fact infinitely many) such that $U = \mathrm{MT}(u)$ and $r(u) = s$. If we define on the set of all pairs $(U, s) \in 2^{\mathrm{FG}(\Gamma)} \times r((\Gamma\cup\Gamma^{-1})^*)$ (with $U$ finite and connected and $1, \gamma(s) \in U$) a multiplication by

$$(U, s)(V, t) = (U \cup \gamma(s) \circ V, r(st))$$

(where $\circ$ refers to the multiplication in the free group $\mathrm{FG}(\Gamma)$), then the resulting monoid is isomorphic to $\mathrm{FIM}(\Gamma)$.

Munn's Theorem leads to a polynomial time algorithm for the word problem for $\mathrm{FIM}(\Gamma)$. For instance, the reader can easily check that the words $bb^{-1}abb^{-1}a$ and $aaa^{-1}bb^{-1}a^{-1}bb^{-1}aa$ represent the same element in $\mathrm{FIM}(\{a, b\})$ by using Munn's Theorem.

For a word $u \in (\Gamma \cup \Gamma^{-1})^*$, the element $\alpha(u) \in \mathrm{FIM}(\Gamma)$ is an idempotent element, i.e., $\alpha(uu) = \alpha(u)$, if and only if $r(u) = \varepsilon$, i.e., $\gamma(u) = 1$.

For a finite set $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ define

$$\mathrm{FIM}(\Gamma)/P = (\Gamma \cup \Gamma^{-1})^*/\tau$$

to be the inverse monoid with the set $\Gamma$ of generators and the set $P$ of relations, where $\tau$ is the smallest congruence on $(\Gamma \cup \Gamma^{-1})^*$ generated by $\rho \cup P$. Then the canonical morphism

$$\mu_P : (\Gamma \cup \Gamma^{-1})^* \to \mathrm{FIM}(\Gamma)/P$$

factors as $\mu_P = \nu_P \circ \alpha$ with

$$\nu_P : \mathrm{FIM}(\Gamma) \to \mathrm{FIM}(\Gamma)/P.$$

We say that $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ is an *idempotent presentation* if for all $(e, f) \in P$, $\alpha(e)$ and $\alpha(f)$ are both idempotents of $\mathrm{FIM}(\Gamma)$, i.e., $r(e) = r(f) = \varepsilon$ by the remark above. In this paper, we are concerned with inverse monoids of the form $\mathrm{FIM}(\Gamma)/P$ for a finite idempotent presentation $P$. In this case, since every identity $(e, f) \in P$ is true in $\mathrm{FG}(\Gamma)$ (we have $\gamma(e) = \gamma(f) = 1$), there also exists a canonical morphism $\beta_P : \mathrm{FIM}(\Gamma)/P \to \mathrm{FG}(\Gamma)$. The following commutative diagram summarizes all morphisms introduced so far.

$$\begin{tikzcd}
& (\Gamma \cup \Gamma^{-1})^* \arrow[dl, "\alpha"'] \arrow[d, "\mu_P"'] \arrow[dr, "\gamma"] & \\
\mathrm{FIM}(\Gamma) \arrow[r, "\nu_P"] \arrow[rr, bend right, "\beta"'] & \mathrm{FIM}(\Gamma)/P \arrow[r, "\beta_P"] & \mathrm{FG}(\Gamma)
\end{tikzcd}$$

For the rest of this paper, the meaning of the morphisms $\alpha, \beta, \beta_P, \gamma, \mu_P$, and $\nu_P$ will be fixed.

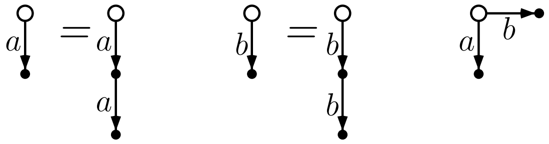
To solve the word problem for $\mathrm{FIM}(\Gamma)/P$, Margolis and Meakin [27] used a closure operation for Munn trees, which is based on work of Stephen [47]. We shortly review the ideas here. As remarked in [27], every idempotent presentation $P$ can be replaced by the idempotent presentation $P' = \{(e, ef), (f, ef) \mid (e, f) \in P\}$, i.e., $\mathrm{FIM}(\Gamma)/P = \mathrm{FIM}(\Gamma)/P'$. Since $\mathrm{MT}(e) \subseteq \mathrm{MT}(ef) \supseteq \mathrm{MT}(f)$ if $r(e) = r(f) = \varepsilon$, we can restrict in the following to idempotent presentations $P$ such that $\mathrm{MT}(e) \subseteq \mathrm{MT}(f)$ for all $(e, f) \in P$. Define a rewriting relation $\Rightarrow_P$ on subsets of $\mathrm{FG}(\Gamma)$ as follows, where $U, V \subseteq \mathrm{FG}(\Gamma)$: $U \Rightarrow_P V$ if and only if there is $(e, f) \in P$ and $u \in U$ such that

- $u \circ v \in U$ for all $v \in \mathrm{MT}(e)$ (here, $\circ$ denotes the multiplication in the free group $\mathrm{FG}(\Gamma)$) and
- $V = U \cup \{u \circ w \mid w \in \mathrm{MT}(f)\}$.

Finally, define the closure of $U \subseteq \mathrm{FG}(\Gamma)$ w.r.t. the presentation $P$ as

$$\mathrm{cl}_P(U) = \bigcup \{V \mid U \overset{*}{\Rightarrow}_P V\}.$$

**Example 8** *Assume that* $\Gamma = \{a, b\}$, $P = \{(aa^{-1}, a^2a^{-2}), (bb^{-1}, b^2b^{-2})\}$ *and* $u = aa^{-1}bb^{-1}$. *The Munn trees for the words in the presentation* $P$ *and* $u$ *look as follows; the bigger circle represents the* $1$ *of* $\mathrm{FG}(\Gamma)$*:*




*Then the closure* $\mathrm{cl}_P(\mathrm{MT}(u))$ *is* $\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\} \subseteq \mathrm{FG}(\Gamma)$.

In the next section, instead of specifying a word $w \in (\Gamma \cup \Gamma^{-1})^*$ (that represents an idempotent element of $\mathrm{FIM}(\Gamma)$, i.e., $r(w) = 1$) explicitly, we will only draw its Munn tree, where as in Example 8 the 1 of $\mathrm{FG}(\Gamma)$ is drawn as a bigger circle. In fact, one can replace $w$ by any word that labels a path from the circle back to the circle and that visits all nodes in the tree; by Munn's Theorem, the resulting word represents the same element of $\mathrm{FIM}(\Gamma)$ (and hence also of $\mathrm{FIM}(\Gamma)/P$) as the original word.

The following result of Margolis and Meakin is central for our further investigations:

**Theorem 9 ([27])** *Let* $P$ *be an idempotent presentation and let* $u, v \in (\Gamma \cup \Gamma^{-1})^*$. *Then* $\mu_P(u) = \mu_P(v)$ *if and only if* $r(u) = r(v)$ *(i.e.,* $\gamma(u) = \gamma(v)$*) and* $\mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_P(\mathrm{MT}(v))$.

The result of Munn for $\mathrm{FIM}(\Gamma)$ mentioned above is a special case of this result for $P = \emptyset$, because $\mathrm{cl}_\emptyset(\mathrm{MT}(u)) = \mathrm{MT}(u)$.

**Remark 10** *Note that* $\mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_P(\mathrm{MT}(v))$ *if and only if* $\mathrm{MT}(u) \subseteq$

$\mathrm{cl}_P(\mathrm{MT}(v))$ *and* $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$.

Margolis and Meakin used Theorem 9 in order to give a solution for the word problem for the monoid $\mathrm{FIM}(\Gamma)/P$. More precisely, they have shown that from a finite idempotent presentation $P$ one can effectively construct an MSO-formula $\mathrm{CL}_P(X,Y)$ over the signature of the Cayley-graph $\mathcal{C}(\Gamma)$ such that for all words $u \in (\Gamma \cup \Gamma^{-1})^*$ and all subsets $A \subseteq \mathrm{FG}(\Gamma)$: $\mathcal{C}(\Gamma) \models \mathrm{CL}_P(\mathrm{MT}(u), A)$ if and only if $A = \mathrm{cl}_P(\mathrm{MT}(u))$. The decidability of the word problem for $\mathrm{FIM}(\Gamma)/P$ is an immediate consequence of Theorem 5 and Theorem 9.

## 6 Complexity of the word problem

The direct use of Theorem 5 leads to a nonelementary algorithm for the word problem for the monoid $\mathrm{FIM}(\Gamma)/P$, see Remark 6. Using tree automata techniques we will show:

**Theorem 11** *For every finite idempotent presentation* $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ *the word problem for* $\mathrm{FIM}(\Gamma)/P$ *can be solved in (i) linear time on a RAM and (ii) in deterministic logspace.* [1]

**Proof.** Let us fix a finite and idempotent presentation $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ and let $u, v \in (\Gamma \cup \Gamma^{-1})^*$. By Theorem 9 we have to check whether $r(u) = r(v)$ and $\mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_P(\mathrm{MT}(v))$. The first property $r(u) = r(v)$ (i.e., the word problem for the free group $\mathrm{FG}(\Gamma)$) can be checked in linear time on a RAM [2] as well as in deterministic logspace [22]. By Remark 10, the property $\mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_P(\mathrm{MT}(v))$ is equivalent to

$$\mathrm{MT}(u) \subseteq \mathrm{cl}_P(\mathrm{MT}(v)) \ \wedge \ \mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u)).$$

It suffices to show that $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$ can be checked both in linear time on a RAM and in deterministic logspace. We will first present an algorithm for this problem, which will be easily seen to be a polynomial time algorithm. In a second step, we will show that this algorithm can be implemented in linear time on a RAM as well as in deterministic logspace.

Recall that there is an MSO-formula $\mathrm{CL}_P(X,Y)$ over the signature of the Cayley-graph $\mathcal{C}(\Gamma)$ such that for all subsets $A \subseteq \mathrm{FG}(\Gamma)$: $\mathcal{C}(\Gamma) \models \mathrm{CL}_P(\mathrm{MT}(u), A)$

[1] We do not state the existence of one algorithm that runs simultaneously in linear time and logarithmic space.

if and only if $A = \mathrm{cl}_P(\mathrm{MT}(u))$. Define the MSO-formula

$$\text{in-cl}_P(X,Y) = \exists Z : \mathrm{CL}_P(X,Z) \wedge Y \subseteq Z.$$

Thus, we have to check whether $\mathcal{C}(\Gamma) \models \text{in-cl}_P(\mathrm{MT}(u), \mathrm{MT}(v))$. Here, it is important to note that since $P$ is a fixed presentation, $\text{in-cl}_P(X,Y)$ is a fixed MSO-formula over the signature of the Cayley-graph $\mathcal{C}(\Gamma)$.

Let $T_\Gamma$ be the $(2 \cdot |\Gamma|)$-ary tree

$$T_\Gamma = ((\Gamma \cup \Gamma^{-1})^*, (\mathrm{suc}_a)_{a \in \Gamma \cup \Gamma^{-1}}),$$

where $\mathrm{suc}_a = \{(w, wa) \mid w \in (\Gamma \cup \Gamma^{-1})^*\}$, and let $\mathrm{IRR}(\Gamma) = \{r(w) \mid w \in (\Gamma \cup \Gamma^{-1})^*\}$ be the set of all reduced normal forms. In a next step, we translate the fixed MSO-formula $\text{in-cl}_P(X,Y)$ into a fixed MSO-formula $\psi_P(X,Y)$ over the signature of $T_\Gamma$ such that for every $A, B \subseteq \mathrm{IRR}(\Gamma)$ we have $T_\Gamma \models \psi_P(A,B)$ if and only if $\mathcal{C}(\Gamma) \models \text{in-cl}_P(\gamma(A), \gamma(B))$. For this, one has to notice that $\mathcal{C}(\Gamma)$ is isomorphic to the structure

$$\begin{aligned}(\mathrm{IRR}(\Gamma), (\{(u, ua) \mid u \in \mathrm{IRR}(\Gamma) \setminus (\Gamma \cup \Gamma^{-1})^* a^{-1}\} \cup \\ \{(ua^{-1}, u) \mid u \in \mathrm{IRR}(\Gamma) \setminus (\Gamma \cup \Gamma^{-1})^* a\})_{a \in \Gamma \cup \Gamma^{-1}}, \varepsilon).\end{aligned}$$

Since $\mathrm{IRR}(\Gamma)$ is a regular subset of $(\Gamma \cup \Gamma^{-1})^*$ and hence MSO-definable in $T_\Gamma$, it follows that $\mathcal{C}(\Gamma)$ is MSO-definable in $T_\Gamma$, see also [27].

We now calculate the sets

$$\begin{aligned}U &= \{r(p) \mid \exists s \in (\Gamma \cup \Gamma^{-1})^* : u = ps\} \subseteq \mathrm{IRR}(\Gamma)\\ V &= \{r(p) \mid \exists s \in (\Gamma \cup \Gamma^{-1})^* : v = ps\} \subseteq \mathrm{IRR}(\Gamma),\end{aligned}$$

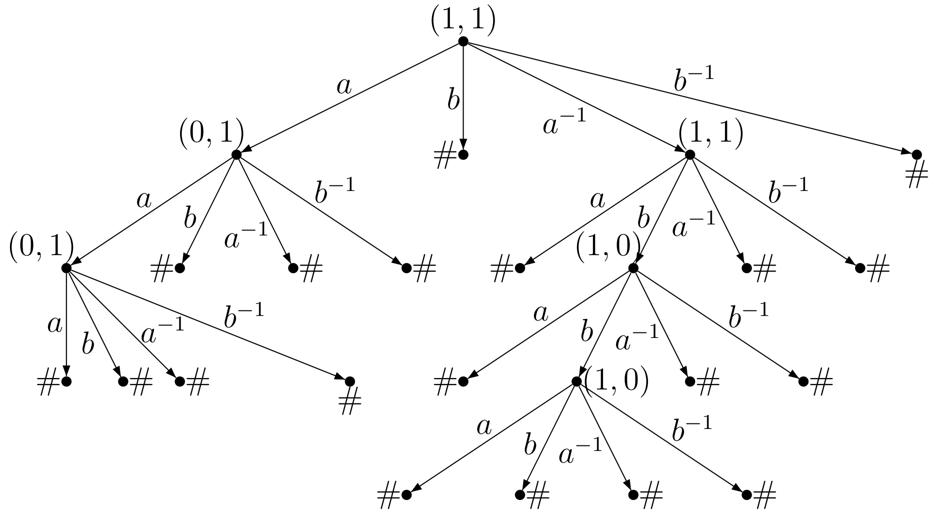
which uniquely represent $\mathrm{MT}(u)$ and $\mathrm{MT}(v)$. Thus, it remains to check whether $T_\Gamma \models \psi_P(U, V)$.

Next, we translate the fixed MSO-formula $\psi_P(X,Y)$ into a fixed (top-down) $\omega$-tree automaton $\mathcal{A}_P$, which runs on a labeled $\omega$-tree $((\Gamma \cup \Gamma^{-1})^*, (\mathrm{suc}_a)_{a \in \Gamma \cup \Gamma^{-1}}, \lambda)$, where $\lambda : (\Gamma \cup \Gamma^{-1})^* \to \{0,1\} \times \{0,1\}$ is the labeling function. The property of $\mathcal{A}_P$ is that $T_\Gamma \models \psi_P(U,V)$ if and only if $\mathcal{A}_P$ accepts the $\omega$-tree

$$T_{U,V} = ((\Gamma \cup \Gamma^{-1})^*, (\mathrm{suc}_a)_{a \in \Gamma \cup \Gamma^{-1}}, \lambda),$$

where for all $w \in (\Gamma \cup \Gamma^{-1})^*$ with $\lambda(w) = (i,j)$ we have: $i = 1$ if and only if $w \in U$ and $j = 1$ if and only if $w \in V$. Again, since $\psi_P(X,Y)$ is a fixed MSO-formula, $\mathcal{A}_P$ is a fixed $\omega$-tree automaton. The translation from $\psi_P(X,Y)$ to $\mathcal{A}_P$ is the standard translation from MSO-formulas to automata, see [39, Theorem 1.7]. It remains to check whether $\mathcal{A}_P$ accepts the $\omega$-tree $T_{U,V}$.

The final step translates $T_{U,V}$ into a finite tree $t^{\text{fin}}_{U,V}$ . Note that in $T_{U,V}$ almost all nodes are labeled with $(0,0)$ ($U$ and $V$ are finite sets of words). Let $B$ be the set of all words of the form $wa$, where $w \in (\Gamma \cup \Gamma^{-1})^*$, $a \in \Gamma \cup \Gamma^{-1}$, $\lambda(wat) = (0,0)$ for every $t \in (\Gamma \cup \Gamma^{-1})^*$, but $\lambda(w) \neq (0,0)$. We construct the tree $t^{\text{fin}}_{U,V}$ by taking $T_{U,V}$ but making every node $w \in B$ to a leaf of $t^{\text{fin}}_{U,V}$ that is labeled with the new symbol # (all proper prefixes of words from $B$ are labeled as in $T_{U,V}$). Note that $t^{\text{fin}}_{U,V}$ is a finite tree that can be constructed from $U$ and $V$ in polynomial time. Before we continue, let us give an example. Let $u = a^{-1}b^2$ and $v = a^2a^{-3}$. Then $U = \{\varepsilon, a^{-1}, a^{-1}b, a^{-1}b^2\}$ and $V = \{\varepsilon, a, a^2, a^{-1}\}$ and $t^{\text{fin}}_{U,V}$ is the following tree.




Now, from the fixed $\omega$-tree automaton $\mathcal{A}_P$ it is easy to construct a fixed tree automaton $\mathcal{A}^{\text{fin}}_P$ (working on finite trees) such that $\mathcal{A}_P$ accepts $T_{U,V}$ if and only if $\mathcal{A}^{\text{fin}}_P$ accepts $t^{\text{fin}}_{U,V}$. Basically, $\mathcal{A}^{\text{fin}}_P$ has the same states and transitions as $\mathcal{A}_P$, except that $\mathcal{A}^{\text{fin}}_P$ accepts in a #-labeled leaf in state $q$ if and only if $\mathcal{A}_P$ accepts the full $\omega$-tree with all nodes labeled $(0,0)$ when starting in state $q$. Since $\mathcal{A}_P$ is a fixed $\omega$-tree automaton, this information can be hardwired into $\mathcal{A}^{\text{fin}}_P$. Finally, whether $\mathcal{A}^{\text{fin}}_P$ accepts $t^{\text{fin}}_{U,V}$ can be checked in polynomial time.

It remains to argue that the above procedure can be implemented both in linear time on a RAM as well as in deterministic logspace. For the linear time algorithm, note that a pointer representation of the tree $t^{\text{fin}}_{U,V}$ can be constructed in linear time from the input words $u$ and $v$. The following algorithm builds a pointer representation of $\mathrm{MT}(u)$:

$k := 1$; $c := 1$;
for all $a \in \Gamma \cup \Gamma^{-1}$: $\mathrm{out}(1, a) := \mathrm{nil}$;
for $i := 1$ to $|u|$ do
    if $\mathrm{out}(c, u[i]) \neq \mathrm{nil}$ then
        $c := \mathrm{out}(c, u[i])$
    else

    $k := k + 1$;
    $\mathrm{out}(c, u[i]) := k$;
    $\mathrm{out}(k, u[i]^{-1}) := c$;
    for all $a \in (\Gamma \cup \Gamma^{-1}) \setminus \{u[i]^{-1}\}$: $\mathrm{out}(k, a) :=$ nil;
    $c := k$
  endif
endfor

The idea behind this algorithm is the following: The nodes of $\mathrm{MT}(u)$ are represented by numbers from $\{1, \ldots, \ell\}$, where $\ell$ is the final value of the variable $k$. During the run of the algorithm, $k$ stores the maximal node generated so far. The tree $\mathrm{MT}(u)$ is build by running once over the word $u$ from left to right. The current node in the partially generated Munn tree is stored in the variable $c$. In order to navigate in the tree, we store in $\mathrm{out}(j, a)$ for every node $j$ the node that can be reached from $j$ with an $a$-labeled edge; this node may be nil. The linear running time of the algorithm is obvious.

After running the above algorithm, we set the current node $c$ to the root 1 and run the same algorithm (without changing the other global variables) with the word $v$ instead of $u$. This results in a pointer representation of $\mathrm{MT}(u) \cup \mathrm{MT}(v)$. Finally, we add for every node $1 \leq i \leq k$ and every $a \in \Gamma \subseteq \Gamma^{-1}$ such that either $\mathrm{out}(i, a) =$ nil or $\mathrm{out}(i, a) < i$ (which means that the $a$-labeled edge leaving $i$ goes up in the tree) a new node $j$ and set $\mathrm{out}(i, a) := j$. The resulting pointer structure represents $t^{\mathrm{fin}}_{U,V}$.

Finally note that the tree automaton $\mathcal{A}^{\mathrm{fin}}_P$ can be evaluated in linear time on the pointer representation of the tree $t^{\mathrm{fin}}_{U,V}$. This finishes our presentation of a linear time algorithm for the word problem for $\mathrm{FIM}(\Gamma)/P$.

For the logspace algorithm we use the fact that the membership problem for the fixed tree automaton $\mathcal{A}^{\mathrm{fin}}_P$ can be solved in deterministic logspace, when the input tree is given by a pointer representation: By [23, Theorem 1], the membership problem for a fixed tree automaton can be even solved in $\mathrm{NC}^1 \subseteq \mathrm{L}$ if the input tree is represented by a well-bracketed expression string. On the other hand, as noted in [4,17], transforming the pointer representation of a tree into its expression string is possible in logspace.

Since deterministic logspace is closed under logspace reductions, it suffices to show that the pointer representation of the tree $t^{\mathrm{fin}}_{U,V}$ can be constructed in deterministic logspace from the words $u$ and $v$. This construction will be presented by a chain of logspace reductions, recall that logspace reducibility is transitive [36].

First, note that for a given word $x \in (\Gamma \cup \Gamma^{-1})^*$ the reduced normal form $r(x)$ can be constructed in logspace: $r(x)$ will be written from left to right onto the output tape by the following procedure:

```
i := 0
while i < |x| do
    i := i + 1
    if ∀j ∈ {i + 1, ..., |x|} : x[i, j] ≠ 1 in FG(Γ) then
        write x[i] onto the output tape
    else
        let j := max{k | i < k ≤ |x|, x[i, k] = 1 in FG(Γ)}
        i := j
    endif
endwhile
```

This algorithm can be implemented in logspace, since we only have to store the two positions $i, j \in \{1, \ldots, |x|\}$. Moreover, whether $x[i, j] \neq 1$ in $\mathrm{FG}(\Gamma)$ can be decided in logspace by [22].

Thus, we can calculate in logspace (an enumeration of) the set

$$W = \{r(u[1, i]) \mid 0 \leq i \leq |u|\} \cup \{r(v[1, i]) \mid 0 \leq i \leq |v|\}.$$

Note that the set of nodes of the tree $t^{\mathrm{fin}}_{U,V}$ is the set

$$N = W \cup \{wc \mid w \in W, c \in \Gamma \cup \Gamma^{-1}\}.$$

Moreover, there is an $c$-labeled edge between $x \in N$ and $y \in N$ if and only if $y = xc$. Finally, the label $\lambda(x)$ of $x \in N$ can be defined as follows: $\lambda(x) = \#$ if $x \in N \setminus W$, otherwise $\lambda(x) = (i, j) \in \{0, 1\} \times \{0, 1\}$ with $i = 1$ if and only if $x \in \{r(u[1, i]) \mid 0 \leq i \leq |u|\}$ and $j = 1$ if and only if $x \in \{r(v[1, i]) \mid 0 \leq i \leq |v|\}$. This description of $t^{\mathrm{fin}}_{U,V}$ immediately gives rise to a logspace algorithm for calculating the pointer representation of $t^{\mathrm{fin}}_{U,V}$.

In the uniform case, where the presentation $P$ is part of the input, the complexity of the word problem increases considerably:

**Theorem 12** *There exists a fixed alphabet $\Gamma$ such that the following problem is EXPTIME-complete:*

*INPUT: Words $u, v \in (\Gamma \cup \Gamma^{-1})^*$ and a finite idempotent presentation $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$*

*QUESTION: $\mu_P(u) = \mu_P(v)$?*

*The EXPTIME upper bound even holds if the alphabet $\Gamma$ belongs to the input.*

**Proof.** For the lower bound we use the fact that EXPTIME equals APSPACE. Thus, let

$$T = (Q, \Sigma, \delta, q_0, q_f)$$

be a fixed alternating Turing machine that accepts an EXPTIME-complete language. Assume that $T$ works in space $p(n)$ for a polynomial $p$ on an input of length $n$. W.l.o.g. we may assume the following:

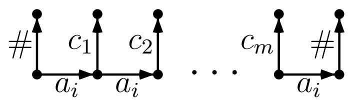
- $T$ alternates in each state, i.e., it either moves from a state of $Q_\exists$ to a state from $Q_\forall \cup \{q_f\}$ or from a state of $Q_\forall$ to a state from $Q_\exists \cup \{q_f\}$.
- $q_0 \in Q_\exists$
- For each pair $(q, a) \in (Q \setminus \{q_f\}) \times \Sigma$, the machine $T$ has precisely two choices according to the transition relation $\delta$, which we call choice 1 and choice 2.
- If $T$ terminates in the final state $q_f$, then the symbol that is currently read by the head is some distinguished symbol $\$ \in \Sigma$.

Define $\Gamma = \Sigma \cup (Q \times \Sigma) \cup \{a_1, a_2, b_1, b_2, \#\}$, where all unions are assumed to be disjoint. A configuration of $T$ is encoded as a word from $\#\Sigma^*(Q \times \Sigma)\Sigma^*\# \subseteq \Gamma^*$. Now let $w \in \Sigma^*$ be an input of length $n$ and let $m = p(n)$. Then a configuration of $T$ is a word from $\bigcup_{i=0}^{m-1} \#\Sigma^i(Q \times \Sigma)\Sigma^{m-i-1}\# \subseteq \Gamma^{m+2}$. Clearly, the symbol at position $i \in \{2, \dots, m+1\}$ at time $t+1$ in a configuration only depends on the symbols at the positions $i-1$, $i$, and $i+1$ at time $t$. Assume that $c, c_1, c_2, c_3 \in \Sigma \cup (Q \times \Sigma) \cup \{\#\}$ are such that $c_1c_2c_3 \in \{\varepsilon, \#\}\Sigma^*(Q \times \Sigma)\Sigma^*\{\varepsilon, \#\}$. We write $c_1c_2c_3 \xrightarrow{j} c$ for $j \in \{1, 2\}$ if the following holds: If three consecutive positions $i-1$, $i$, and $i+1$ of a configuration contain the symbol sequence $c_1c_2c_3$, then choice $j$ of $T$ results in the symbol $c$ at position $i$. We write $c_1c_2c_3 \xrightarrow{\exists} (d_1, d_2)$ for $c_1, c_2, c_3, d_1, d_2 \in \Sigma \cup (Q \times \Sigma) \cup \{\#\}$ if one of the following two cases holds:

- $c_1c_2c_3 \in \{\varepsilon, \#\}\Sigma^*(Q_\exists \times \Sigma)\Sigma^*\{\varepsilon, \#\}$ and $c_1c_2c_3 \xrightarrow{j} d_j$ for $j \in \{1, 2\}$
- $c_1c_2c_3 \in \{\varepsilon, \#\}\Sigma^*\{\varepsilon, \#\}$ and $d_1 = d_2 = c_2$.

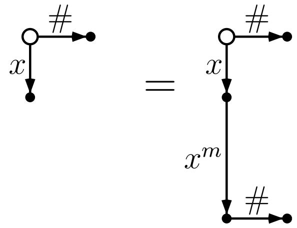
The notation $c_1c_2c_3 \xrightarrow{\forall} (d_1, d_2)$ is defined analogously, except that in the first case we require $c_1c_2c_3 \in \{\varepsilon, \#\}\Sigma^*(Q_\forall \times \Sigma)\Sigma^*\{\varepsilon, \#\}$.

Let us now briefly describe the idea for the lower bound proof. We will encode a configuration $\#c_1c_2 \cdots c_m\#$, where the current state is from $Q_\exists$ by a subgraph of the Cayley-graph $\mathcal{C}(\Gamma)$ of the following form, where $i = 1$ or $i = 2$:



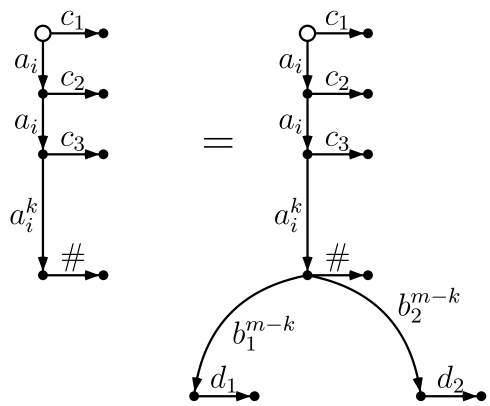

If the current state is from $Q_\forall$, then we take the same subgraph, except that $a_i$ is replaced by $b_i$. The idempotent presentation $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ is constructed in such a way from the machine $T$ that building the closure from a Munn tree that represents the initial configuration (in the above sense) corresponds to generating the whole computation tree of the Turing machine $T$ starting from the initial configuration. We will describe each pair $(e, f) \in P$ by the Munn trees $\mathrm{MT}(e)$ and $\mathrm{MT}(f)$, where $\mathrm{MT}(e) \subseteq \mathrm{MT}(f)$.

For all $x \in \{a_1, a_2, b_1, b_2\}$ put the following equation into $P$, which propagates the end-marker # along intervals of length $m+1$ (here, the $x^m$-labeled edge abbreviates a path consisting of $m$ many $x$-labeled edges):



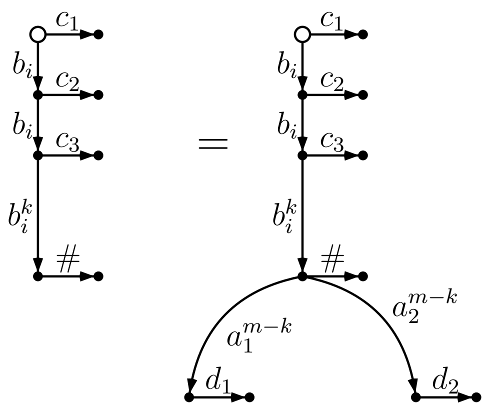

(1)

The next two equation types generate the two successor configurations of the current configuration. If $c_1c_2c_3 \xrightarrow{\exists} (d_1, d_2)$, then for every $0 \leq k \leq m-1$ and $i \in \{1, 2\}$ we include the following equation in $P$:



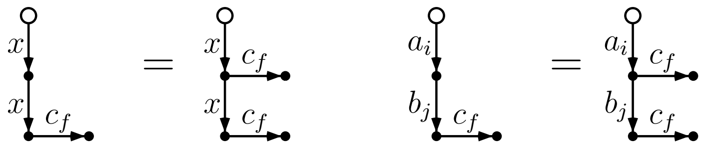

(2)

If $c_1c_2c_3 \xrightarrow{\forall} (d_1, d_2)$, then for every $0 \leq k \leq m-1$ and $i \in \{1,2\}$ we take the following equation:




(3)

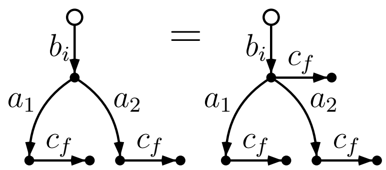
The remaining equations propagate acceptance information back to the initial Munn tree. Here the separation of the state set into existential and universal states becomes crucial. Let $c_f = (q_f, \$)$; recall that \$ is the symbol under the head of $T$ when $T$ terminates in state $q_f$. For all $x \in \{a_1, a_2, b_1, b_2\}$ and all $i, j \in \{1,2\}$ we put the following equations into $P$:




(4)

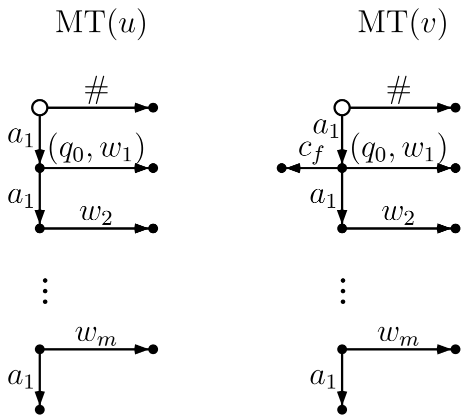
Here, the second equation expresses the fact that an existential configuration is accepting if and only if at least one successor configuration is accepting.

Finally, for $i \in \{1,2\}$ we add the following equation to $P$, which reflects the fact that a universal configuration is accepting if and only if both successor configurations are accepting.




(5)

This concludes the description of the presentation $P$. Now choose words $u, v \in (\Gamma \cup \Gamma^{-1})^*$ as follows: Assume that the input word for our alternating Turing machine $w$ is of the form $w = w_1w_2 \cdots w_n$ with $w_i \in \Sigma$. For $n+1 \leq i \leq m$ define $w_i = \Box$, where $\Box$ is the blank symbol of $T$. Then we take for $u$ and $v$ words such that $r(u) = r(v) = \varepsilon$ and such that their Munn trees look as follows:




We want to show that $\mu_P(u) = \mu_P(v)$ if and only if the machine $T$ accepts the word $w$. Since $\mathrm{MT}(u) \subseteq \mathrm{MT}(v)$, we have $a_1 c_f \in \mathrm{cl}_P(\mathrm{MT}(u))$ if and only if $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$ if and only if $\mathrm{cl}_P(\mathrm{MT}(v)) = \mathrm{cl}_P(\mathrm{MT}(u))$ (see Remark 10). Since moreover $r(u) = r(v) = \varepsilon$, it suffices by Theorem 9 to show the following equivalence:

$$T \text{ accepts the word } w \qquad \Leftrightarrow \qquad a_1 c_f \in \mathrm{cl}_P(\mathrm{MT}(u)).$$

To prove this, let us denote with $P_1$ (resp. $P_2$) the idempotent presentation consisting of the rules in (1)–(3) (resp. (4) and (5)). The rewrite relation $\Rightarrow_{P_1}$ (defined in Section 5) generates, starting from $\mathrm{MT}(u)$ (which encodes the initial configuration corresponding to the input $w$), the full computation tree $\mathrm{ct}(T)$ of the machine $T$, encoded as a subtree of the tree $\mathcal{C}(\Gamma)$. Thus, $\mathrm{cl}_{P_1}(\mathrm{MT}(u))$ encodes $\mathrm{ct}(T)$. Moreover, $\mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_{P_2}(\mathrm{cl}_{P_1}(\mathrm{MT}(u)))$. To see this latter fact, note that applications of the rules from $P_2$ do not produce new occurrences for the left hand sides from $P_1$. For this it is important that the machine $T$ terminates if it reaches state $q_f$ and hence no $c_f$-labeled edge occurs in a left hand side of $P_1$.

Now assume that $T$ accepts the word $w$. This means that there exists a subtree $S$ of $\mathrm{ct}(T)$ such that

(a) every leaf of $S$ is a configuration, where the current state is the final state $q_f$,
(b) if a non-leaf $v$ of $\mathrm{ct}(T)$ is an existential configuration, then at least one $\mathrm{ct}(T)$-successor of $v$ belongs to $S$,
(c) if a non-leaf $v$ of $\mathrm{ct}(T)$ is a universal configuration, then both $\mathrm{ct}(T)$-successors of $v$ belong to $S$, and
(d) the initial configuration is the root of $S$.

To this subtree $S$ there corresponds a subtree $S'$ of $\mathrm{cl}_{P_1}(\mathrm{MT}(u))$. Using the rules from $P_2$, one can add a $c_f$-labeled edge to every non-leaf of $S'$ except the root 1.

For the other direction assume that $a_1 c_f \in \mathrm{cl}_P(\mathrm{MT}(u)) = \mathrm{cl}_{P_2}(\mathrm{cl}_{P_1}(\mathrm{MT}(u)))$. This means that starting from the tree $\mathrm{cl}_{P_1}(\mathrm{MT}(u))$ (which encodes the full computation tree of the machine $T$) one can, by using only the rules (4) and (5), add a $c_f$-labeled edge to the node $a_1 \in \mathrm{FG}(\Gamma)$. By the form of the rules (4) and (5), this means that there has to exist a subtree $S$ of the computation tree $\mathrm{ct}(T)$ having properties (a)–(d) from the previous paragraph. But this implies that $T$ accepts the input word $w$. This concludes the proof for the EXPTIME lower bound.

For the upper bound let $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ be an idempotent presentation and let $u, v \in (\Gamma \cup \Gamma^{-1})^*$. Since $r(u) = r(v)$ can be checked in linear time, it suffices by Theorem 9 to verify in EXPTIME whether $\mathrm{cl}_P(\mathrm{MT}(v)) = \mathrm{cl}_P(\mathrm{MT}(u))$. By Remark 10, it is enough to show that we can check in EXPTIME, whether $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$.

Let $G$ be the edge-labeled graph that results from the Cayley-graph $\mathcal{C}(\Gamma)$ by adding a new node $v_0$ and adding a #-labeled edge from node 1 (i.e., the origin) of $\mathcal{C}(\Gamma)$ to the new node $v_0$ . Here, the edge label # is assumed to be not in $\Gamma \cup \Gamma^{-1}$ (the label set of $\mathcal{C}(\Gamma)$). We need this new edge in order to be able to recognize the 1 in $\mathcal{C}(\Gamma)$. Since $\mathcal{C}(\Gamma)$ is a context-free graph, it follows that also $G$ is context-free. We decide $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$ by constructing from $u, v$, and $P$ in polynomial time a formula $\varphi_{u,v,P}$ of the modal $\mu$-calculus such that $(G, 1) \models \varphi_{u,v,P}$ if and only if $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$. Then the EXPTIME upper bound follows from Theorem 2.

In the following, for a word $w = a_1 a_2 \cdots a_m$ ($a_i \in \Gamma \cup \Gamma^{-1}$) we use $\langle w \rangle \phi$ as an abbreviation for $\langle a_1 \rangle \langle a_2 \rangle \cdots \langle a_m \rangle \phi$. Now assume that $P = \{(e_i, f_i) \mid 1 \leq i \leq n\}$, where $\mathrm{MT}(e_i) \subseteq \mathrm{MT}(f_i)$. First, let $\varphi_{u,P}$ be the following $\mu$-sentence:

$$\mu X. \left( \bigvee_{i=0}^{|u|} \langle u[1,i]^{-1} \rangle \langle \# \rangle \mathrm{true} \ \vee \ \bigvee_{i=1}^{n} \bigvee_{j=0}^{|f_i|} \langle f_i[1,j]^{-1} \rangle ( \bigwedge_{k=0}^{|e_i|} \langle e_i[1,k] \rangle X ) \right)$$

Then $(G, x) \models \varphi_{u,P}$ if and only if the node $x$ belongs to $\mathrm{cl}_P(\mathrm{MT}(u))$. In the formula $\varphi_{u,P}$, the disjunction $\bigvee_{i=0}^{|u|} \langle u[1,i]^{-1} \rangle \langle \# \rangle \mathrm{true}$ defines all nodes from $\mathrm{MT}(u) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$. The disjunction

$$\bigvee_{i=1}^{n} \bigvee_{j=0}^{|f_i|} \langle f_i[1,j]^{-1} \rangle ( \bigwedge_{k=0}^{|e_i|} \langle e_i[1,k] \rangle X )$$

defines all nodes $x$ such that $x$ can be reached from a node $y$ via some prefix of some word $f_i$ and moreover, the whole path that starts in $y$ and that is labeled with the word $e_i$ already belongs to $X$, i.e., $\mathrm{MT}(e_i) \subseteq X$. For the correctness of the sentence $\varphi_{u,P}$, it is important to note that $\mathcal{C}(\Gamma)$ is a deterministic graph, i.e., for every $a \in \Gamma \cup \Gamma^{-1}$, every node $x$ has exactly one $a$-labeled outgoing edge. Thus, it is not relevant, whether the $[a]$- or $\langle a \rangle$-modality is used. Finally, we can take for $\varphi_{u,v,P}$ the sentence $\bigwedge_{i=0}^{|v|}\langle v[1,i]\rangle\varphi_{u,P}$.

The following result was conjectured in [49].

**Corollary 13** *There exists a fixed context-free graph, for which the model-checking problem of the modal $\mu$-calculus (restricted to formulas of nesting depth* 1*) is EXPTIME-complete.*

**Proof.** We can reuse the constructions from the previous proof. Note that the generating set $\Gamma$ from the lower bound proof is a fixed set; thus, the Cayley-graph $\mathcal{C}(\Gamma)$ is a fixed context-free graph. Hence, also the graph $G$ constructed in the upper bound proof by adding a #-labeled edge that leaves the origin 1 is a fixed context-free graph. For the input word $w$ for the Turing machine $T$ let $u$, $v$, and $P$ be the data constructed in the lower bound proof. Then $w$ is accepted by $T$ if and only if $\mathrm{MT}(v) \subseteq \mathrm{cl}_P(\mathrm{MT}(u))$ if and only if $(G, 1) \models \varphi_{u,v,P}$. This proves the corollary.

## 7 Cayley-graphs of Inverse Monoids

In [5], it was shown that the MSO-theory of the Cayley-graph of $\mathrm{FIM}(\{a\})$ is undecidable. In this section we will contrast this undecidability result with a decidability result for a still quite powerful fragment of the MSO-theory of the Cayley-graph of $\mathrm{FIM}(\Gamma)/P$ (for $P$ an idempotent presentation). For this, we extend the approach from [27] of translating the word problem for the monoid $\mathrm{FIM}(\Gamma)/P$ into a monadic second-order property of the Cayley-graph $\mathcal{C}(\Gamma)$ in order to decide more general decision problems than just the word problem. For this, we need some definitions.

Let $\mathcal{M} = (M, \circ, 1)$ be a monoid with a finite generating set $\Sigma$ and let $h : \Sigma^* \to \mathcal{M}$ be the canonical morphism. We define the following expansion $\mathcal{C}(\mathcal{M}, \Sigma)_{\mathrm{reg}}$ of the Cayley-graph $\mathcal{C}(\mathcal{M}, \Sigma)$:

$$\begin{aligned}\mathcal{C}(\mathcal{M}, \Sigma)_{\mathrm{reg}} &= (M, (\mathrm{reach}_L)_{L\in\mathrm{REG}(\Sigma)}, 1), \text{ where}\\ \mathrm{reach}_L &= \{(u, v) \in M \times M \mid \exists w \in L : u \circ h(w) = v\} \text{ for } L \subseteq \Sigma^*.\end{aligned}$$

Thus, $\mathcal{C}(\mathcal{M}, \Sigma) = (M, (\mathrm{reach}_{\{a\}})_{a\in\Sigma}, 1)$. Note that $\mathcal{C}(\mathcal{M}, \Sigma)_{\mathrm{reg}}$ is a relational structure with infinitely many binary relations, one for each regular subset of $\Sigma^*$. In a first-order formula over the structure $\mathcal{C}(\mathcal{M}, \Sigma)_{\mathrm{reg}}$, a predicate $\mathrm{reach}_L$ is represented by a finite automaton for the language $L$. Again, the decidability (resp. complexity) of the first-order theory of $\mathcal{C}(\mathcal{M}, \Sigma)_{\text{reg}}$ does not depend on the generating set $\Sigma$:

**Proposition 14** *Let $\Sigma_1$ and $\Sigma_2$ be finite generating sets for the monoid $\mathcal{M}$. Then the first-order theory of $\mathcal{C}(\mathcal{M}, \Sigma_1)_{\text{reg}}$ is reducible to the first-order theory of $\mathcal{C}(\mathcal{M}, \Sigma_2)_{\text{reg}}$.*

**Proof.** There exists a homomorphism $f : \Sigma_1^* \to \Sigma_2^*$ such that for every word $w \in \Sigma_1^*$, $f(w)$ represents the same monoid element of $\mathcal{M}$ as $w$. Then, for a given sentence $\varphi_1$ over the signature of $\mathcal{C}(\mathcal{M}, \Sigma_1)_{\text{reg}}$ we just have to replace every atomic predicate $\text{reach}_L(x, y)$ by $\text{reach}_{f(L)}(x, y)$. If $\varphi_2$ is the resulting sentence then $\mathcal{C}(\mathcal{M}, \Sigma_1)_{\text{reg}} \models \varphi_1$ if and only if $\mathcal{C}(\mathcal{M}, \Sigma_2)_{\text{reg}} \models \varphi_2$.

The main result of this section is:

**Theorem 15** *Let $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ be a finite idempotent presentation. Then the first-order theory of $\mathcal{C}(\text{FIM}(\Gamma)/P, \Gamma \cup \Gamma^{-1})_{\text{reg}}$ is decidable.*

**Remark 16** *It is easy to show that already the first-order theory of the structure $\mathcal{C}(\text{FIM}(\{a, b\}), \{a, b, a^{-1}, b^{-1}\})_{\text{reg}}$ is not elementary decidable: It is known that the first-order theory of $\mathcal{A} = (\{a, b\}^*, (\{(w, wc) \mid w \in \{a, b\}^*\})_{c \in \{a,b\}}, \preceq)$, where $\preceq$ is the prefix relation on $\{a, b\}^*$, is not elementary decidable, see e.g. [10]. It is straightforward to define $\mathcal{A}$ in $\mathcal{C}(\text{FIM}(\{a, b\}), \{a, b, a^{-1}, b^{-1}\})_{\text{reg}}$ using first-order logic.*

Before we prove Theorem 15, let us first state some consequences. Again, let $\mathcal{M}$ be a monoid with a finite generating set $\Sigma$ and let $h : \Sigma^* \to \mathcal{M}$ be the canonical morphism. Recall that a subset $L \subseteq \mathcal{M}$ is *rational* if there exists a regular language $K \subseteq \Sigma^*$ such that $L = h(K)$. Let $\text{RAT}(\mathcal{M})$ denote the set of all rational subsets of $\mathcal{M}$. The following theorem is an immediate corollary of Theorem 15; note that $x$ belongs to a rational subset $L = h(K)$ of $\mathcal{M}$ if and only if $\mathcal{C}(\mathcal{M}, \Sigma)_{\text{reg}} \models \text{reach}_K(1, x)$.

**Theorem 17** *Let $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ be a finite idempotent presentation. The following problem is decidable:*

*INPUT: A boolean combination $B$ of rational subsets from $\text{FIM}(\Gamma)/P$, where each of these rational subsets is represented by a finite automaton over the alphabet $\Gamma \cup \Gamma^{-1}$.*

*QUESTION: Is the subset of $\text{FIM}(\Gamma)/P$ defined by $B$ empty?*

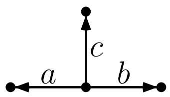
Note that for every finitely generated monoid $\mathcal{M}$ such that $\mathrm{RAT}(\mathcal{M})$ is an effective boolean algebra, the emptiness problem for boolean combinations of rational subsets of $\mathcal{M}$ is decidable. In case of $\mathcal{M} = \mathrm{FIM}(\Gamma)/P$ we cannot use this argument in order to prove Theorem 17, since by the next theorem $\mathrm{RAT}(\mathrm{FIM}(\Gamma)/P)$ is in general not a boolean algebra. This result has been obtained in collaboration with Volker Diekert and Klaus-Jörn Lange.

**Theorem 18** *If* $|\Gamma| \geq 2$, *then* $\mathrm{RAT}(\mathrm{FIM}(\Gamma))$ *is not closed under intersection and hence not under complementation.*

The proof is a corollary of the next two lemmas. Recall that $\alpha : (\Gamma \cup \Gamma^{-1})^* \to \mathrm{FIM}(\Gamma)$ denotes the canonical morphism. Let $T \subseteq \mathrm{FIM}(\Gamma)$ be the set consisting of all elements $\alpha(u) \in \mathrm{FIM}(\Gamma)$ such that the Munn tree $\mathrm{MT}(u)$ has a node of degree at least 3.

**Lemma 19** *The set* $T \subseteq \mathrm{FIM}(\Gamma)$ *is rational.*

**Proof.** We give a regular expression for a language $K \subseteq (\Gamma \cup \Gamma^{-1})^*$ with $\alpha(K) = T$ by describing the existence of a node of degree at least 3. If $\alpha(u) \in T$, then there exist $a, b, c \in \Gamma \cup \Gamma^{-1}$ such that the Munn tree $\mathrm{MT}(u)$ contains the following subgraph:




Thus, for

$$K = \bigcup_{\substack{a,b,c \in \Gamma \cup \Gamma^{-1} \\ a \neq b \neq c \neq a}} (\Gamma \cup \Gamma^{-1})^* a a^{-1} b b^{-1} c c^{-1} (\Gamma \cup \Gamma^{-1})^*$$

we have $\alpha(K) = T$.

Let now $L \subseteq \mathrm{FIM}(\Gamma)$ be the rational language

$$L = \alpha(\{a^n a^{-m} b \mid m, n \geq 1\}).$$

We will show that the intersection $L \cap T$ is not rational, which implies Theorem 18.

**Lemma 20** *Let* $T$ *and* $L$ *be as defined above. Then* $T \cap L$ *is not rational.*

**Proof.** The Munn tree $\mathrm{MT}(a^n a^{-m} b)$ $(m, n \geq 1)$ contains a node of degree 3 if and only if $n > m$. Thus, we obtain

$$T \cap L = \{\alpha(a^n a^{-m} b) \mid n > m \geq 1\}.$$

Suppose $T \cap L$ is rational. Then there exists a regular language $R \subseteq (\Gamma \cup \Gamma^{-1})^*$ such that $\alpha(R) = \{\alpha(a^n a^{-m} b) \mid n > m \geq 1\}$. Let $A$ be a finite automaton with $s$ many states, recognizing $R$ and let $n \geq s$. Then we have

$$\alpha(a^{n+1} a^{-n} b) \in T \cap L = \alpha(R).$$

This means that there exist $u, v_1, \ldots, v_n, w \in (\Gamma \cup \Gamma^{-1})^*$ such that

$$\begin{aligned}
&uv_1 \cdots v_n w \in R, \\
&\qquad \gamma(u) = \gamma(a^{n+1}), \\
&\qquad \gamma(v_i) = \gamma(a^{-1}) \text{ for } 1 \leq i \leq n, \\
&\qquad \gamma(w) = \gamma(b).
\end{aligned}$$

For $0 \leq i \leq n$ let $q_i$ be the state of $A$ after reading $uv_1 \cdots v_i$. Since $n \geq s$, there exist $i < j$ such that $q_i = q_j$. As a consequence we have for all $k \geq 0$:

$$uv_1 \cdots v_i (v_{i+1} \cdots v_j)^k v_{j+1} \cdots v_n w \in R$$

But for $k$ large enough (in fact $k \geq 2$) we obtain for some $\ell \geq 0$:

$$\gamma(uv_1 \cdots v_i (v_{i+1} \cdots v_j)^k v_{j+1} \cdots v_n w) = \gamma(a^{-\ell} b)$$

This shows $\alpha(uv_1 \cdots v_i (v_{i+1} \cdots v_j)^k v_{j+1} \cdots v_n w) \notin T \cap L$, which contradicts $\alpha(R) = T \cap L$.

**Remark 21** *The set $T$ above is a concrete example of a rational set such that* $\mathrm{FIM}(\Gamma) \setminus T$ *is not rational. To see this, just consider elements of the form* $\alpha(a^n a^{-n} b) \in \mathrm{FIM}(\Gamma) \setminus T$ *for $n$ large enough.*

The *generalized word problem* for the monoid $\mathcal{M}$ is the following computational problem:

INPUT: Words $u, u_1, \ldots, u_n \in \Sigma^*$

QUESTION: Does $h(u)$ belong to the submonoid of $\mathcal{M}$ that is generated by $h(u_1), \ldots, h(u_n)$?

**Remark 22** *In the group case the decidability of the word problem follows from the decidability of the generalized word problem. This simple fact generalizes to every monoid* $\mathrm{FIM}(\Gamma)/P$*, where* $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$

*is an idempotent presentation (whereas for arbitrary monoids, it may fail): We claim that for $u, v \in (\Gamma \cup \Gamma^{-1})^*$ we have $\mu_P(u) = \mu_P(v)$ if and only if $\mu_P(u) \in \mu_P(v^*)$ and $\mu_P(v) \in \mu_P(u^*)$. The "only if" direction is obvious. Now assume that $\mu_P(u) = \mu_P(v^n)$ and $\mu_P(v) = \mu_P(u^m)$ for some $n, m \geq 0$. If $m = 0$, then $\mu_P(v) = \mu_P(u) = 1$. Thus, assume that $m > 0$. By applying the morphism $\beta_P : \mathrm{FIM}(\Gamma)/P \to \mathrm{FG}(\Gamma)$ we get $\gamma(u) = \gamma(v)^n$ and $\gamma(v) = \gamma(u)^m$, i.e., $\gamma(u) = \gamma(u)^{m \cdot n}$. Since every free group is torsion-free, it follows $m \cdot n = 1$ (i.e., $m = n = 1$) or $\gamma(u) = \gamma(v) = 1$. In the first case, we are finished. Thus, assume that $\gamma(u) = \gamma(v) = 1$. It follows that $\alpha(u)$ is an idempotent element in $\mathrm{FIM}(\Gamma)$, i.e., $\alpha(u) = \alpha(u)^m$ (recall that $m > 0$). By applying the morphism $\nu_P : \mathrm{FIM}(\Gamma) \to \mathrm{FIM}(\Gamma)/P$ we get $\mu_P(u) = \mu_P(u)^m = \mu_P(v)$.*

Since finite subsets as well as finitely generated submonoids of a monoid are both rational, we obtain the following corollary from Theorem 17.

**Corollary 23** *Let $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ be a finite idempotent presentation. Then the generalized word problem for $\mathrm{FIM}(\Gamma)/P$ is decidable.*

### *7.1 Proof of Theorem 15*

In this section, we will prove Theorem 15. First, we need a preliminary result about arbitrary edge-labeled graphs:

**Proposition 24** *Let $\Sigma$ be a finite alphabet and let $L \in \Sigma^*$ be a regular language. There exists an MSO-formula $\mathrm{Reach}_L(x, y, X)$ over the signature consisting of binary relation symbols $E_a$, $a \in \Sigma$, such that for every directed edge-labeled graph $G = (V, (E_a)_{a \in \Sigma})$, all nodes $s, t \in V$, and every finite set of nodes $U \subseteq V$ we have: $G \models \mathrm{Reach}_L(s, t, U)$ if and only if there exist a path $(p_0, \ldots, p_m)$ ($p_i \in V$) and $a_1, \ldots, a_m \in \Sigma$ with $p_0 = s$, $p_m = t$, $(p_{i-1}, p_i) \in E_{a_i}$ for $i \in \{1, \ldots, m\}$, $a_1 \cdots a_m \in L$, and $U = \{p_1, \ldots, p_m\}$.*

Thus, $G \models \mathrm{Reach}_L(s, t, U)$ if and only if there is a path in $G$ with initial vertex $s \in U$ and terminal vertex $t \in U$ visiting precisely the vertices from $U$ and reading the labels of the path as a word from $\Sigma^*$ we obtain a word in $L$. In the short version [25] of this paper, we sketched a proof of Proposition 24 using MSO-transductions, see [11]. Here we present an alternative proof, which uses the idea from the classical proof of Kleene's Theorem (see e.g. [16]) stating that recognizable languages are rational.

*Proof of Proposition 24.* Let $L \in \Sigma^*$ be a regular language given by a finite nondeterministic automaton $A = (Q, \Sigma, \delta, I, F)$. We assume that $Q = \{1, \ldots, n\}$.

We will define a formula

$$\mathrm{Reach}[i,j](x,y,X)$$

such that for every directed edge-labeled graph $G = (V, (E_a)_{a\in\Sigma})$, all nodes $s,t \in V$, and every finite set of nodes $U \subseteq V$ we have: $G \models \mathrm{Reach}[i,j](s,t,U)$ if and only if there exist paths $(p_0,\dots,p_m)$ ($p_i \in V$) and $(q_0,\dots,q_m)$ ($q_i \in Q$) such that

- $p_0 = s$, $\{p_0,\dots,p_m\} = U$, $p_m = t$,
- for every $\ell \in \{1,\dots,m\}$ there exists $a \in \Sigma$ such that $(p_{\ell-1},p_\ell) \in E_a$ and $(q_{\ell-1},a,q_\ell) \in \delta$,
- $q_0 = i$, and $q_m = j$.

Thus, we get the following formula for our lemma:

$$\mathrm{Reach}_L(x,y,X) = \bigvee_{i\in I, f\in F} \mathrm{Reach}[i,f](x,y,X)$$

In a first part let us define by induction on $k \geq 0$ a formula

$$\mathrm{reach}[i,j,k](x,y,X),$$

where we relax the condition on $X$, but we add the constraint to restrict the automaton $A$ to the set of states $\{1,\dots,k\}$. More precisely, the semantics of $\mathrm{reach}[i,j,k](x,y,X)$ is such that for every directed edge-labeled graph $G = (V, (E_a)_{a\in\Sigma})$, all nodes $s,t \in V$, and every finite set of nodes $U \subseteq V$ we have: $G \models \mathrm{reach}[i,j,k](s,t,U)$ if and only if there exist paths $(p_0,\dots,p_m)$ ($p_i \in V$) and $(q_0,\dots,q_m)$ ($q_i \in Q$) such that

- $p_0 = s$, $\{p_0,\dots,p_m\} \subseteq U$, $p_m = t$,
- for every $\ell \in \{1,\dots,m\}$ there exists $a \in \Sigma$ such that $(p_{\ell-1},p_\ell) \in E_a$ and $(q_{\ell-1},a,q_\ell) \in \delta$,
- $q_0 = i$, $\{q_1,\dots,q_{m-1}\} \subseteq \{1,\dots k\}$, and $q_m = j$.

For $k = 0$ we define:

$$\begin{aligned}\mathrm{reach}[i,j,0](x,y,X) = {} & x,y \in X \wedge \\ & \left( (x = y \wedge i = j) \vee \bigvee_{\substack{a\in\Sigma \\ (i,a,j)\in\delta}} (x,y) \in E_a \right).\end{aligned}$$

Now let $k \geq 1$. The formula $\mathrm{reach}[k,k,k-1](x,y,X)$ is known by induction. Let $\mathrm{reach}[k,k,k-1]^*(x,y,X)$ be the reflexive and transitive closure of $\mathrm{reach}[k,k,k-1](x,y,X)$ (see Remark 1), where the set variable $X$ is treated as a fixed parameter. Then

$$\mathrm{reach}[k,k,k](x,y,X) = (x,y \in X \wedge \mathrm{reach}[k,k,k-1]^*(x,y,X)).$$

Now, analogously to the proof of Kleene's Theorem we define $\mathrm{reach}[i,j,k](x,y,X)$ for pairs $(i,j)$ with $(i,j) \neq (k,k)$ by:

$$\mathrm{reach}[i,j,k](x,y,X) = \mathrm{reach}[i,j,k-1](x,y,X) \vee \exists x' \, \exists y' \left\{ \begin{array}{l} \mathrm{reach}[i,k,k-1](x,x',X) \wedge \\ \mathrm{reach}[k,k,k](x',y',X) \wedge \\ \mathrm{reach}[k,j,k-1](y',y,X) \end{array} \right\}$$

We let $\mathrm{reach}[i,j](x,y,X) = \mathrm{reach}[i,j,n](x,y,X)$. Clearly, $\mathrm{reach}[i,j](x,y,X) \wedge \mathrm{reach}[j,k](y,z,X)$ implies $\mathrm{reach}[i,k](x,z,X)$.

Having $\mathrm{reach}[i,j](x,y,X)$ available, we can define $\mathrm{Reach}[i,j](x,y,X)$ as the following formula:

$$\exists X_1 \cdots \exists X_n \left\{ \begin{array}{l} x \in X_i \;\wedge\; \bigwedge\limits_{k \neq \ell} X_k \cap X_\ell = \emptyset \;\wedge\; X = X_1 \cup \cdots \cup X_n \;\wedge \\ \bigwedge\limits_{k,\ell} \forall u \in X_k \forall v \in X_\ell \left\{ \begin{array}{l} \mathrm{reach}[i,k](x,u,X) \wedge \\ \mathrm{reach}[k,j](u,y,X) \wedge \\ (\mathrm{reach}[k,\ell](u,v,X) \vee \\ \mathrm{reach}[\ell,k](v,u,X)) \end{array} \right\} \end{array} \right\} \tag{6}$$

In order to prove correctness, assume first that there is a path $(p_0, \ldots, p_m)$ in $G$ with $p_0 = x$ and $p_m = y$ visiting precisely the nodes from $X$ and there is a corresponding path $(q_0, \ldots, q_m)$ in the automaton with $q_0 = i$, $q_m = j$, and $(q_{\ell-1}, a, q_\ell) \in \delta$, $(p_{\ell-1}, p_\ell) \in E_a$ for some $a \in \Sigma$ $(1 \leq \ell \leq m)$. In order to show (6) we set

$$X_k = \{p_\ell \mid 0 \leq \ell \leq m, q_\ell = k, \forall r < \ell : p_r \neq p_\ell\}.$$

Thus, $X_k$ is the set of all nodes $p_\ell$ such that the automaton $A$ is in state $k$, when $p_\ell$ is visited for the first time. This defines a partition $X = X_1 \cup \cdots \cup X_n$. Note that some of the $X_k$ may be empty. Obviously we have $x \in X_i$, but it is possible that $y \in X_\ell$ with $\ell \neq j$, because we consider only the first appearance of $y$ on the path $(p_0, \ldots, p_m)$. Nevertheless, we have $\mathrm{reach}[k,j](u,y,X)$ for all $u$ and $k$ with $u \in X_k$. Now let $u \in X_k$ and $v \in X_\ell$ be on the path $(p_0, \ldots, p_m)$. Then we have $\mathrm{reach}[i,k](x,u,X)$ and we have $\mathrm{reach}[k,\ell](u,v,X)$ or $\mathrm{reach}[\ell,k](v,u,X)$, depending whether the first appearance of $u$ is before $v$ on the path or vice versa. Thus (6) holds.

For the other direction, assume that (6) holds. Consider sequences $(x_1, \ldots, x_m)$ $(x_k \in X)$ and $(q(1), \ldots, q(m))$ $(q(k) \in Q)$ with maximal length $m$ such that:

(1) $x = x_1$,

(2) $x_k \neq x_\ell$ for all $1 \leq k < \ell \leq m$,
(3) $\mathrm{reach}[q(k-1), q(k)](x_{k-1}, x_k, X)$ for all $2 \leq k \leq m$,
(4) $x_k \in X_{q(k)}$ for all $1 \leq k \leq m$.

Because $\mathrm{Reach}[i,j](x,y,X)$ is satisfied, we have $x \in X_i$ and hence $q(1) = i$. Now assume that there is some node $v \in X$ with $v \notin \{x_1, \dots, x_m\}$. Since we have $X = X_1 \cup \cdots \cup X_n$, there is an $\ell$ such that $v \in X_\ell$. Since $\mathrm{reach}[i,\ell](x,v,X)$ holds there is a maximal $k \leq m$ such that $\mathrm{reach}[q(k),\ell](x_k, v, X)$. If $k = m$, then we can set $x_{m+1} = v$ and $q(m+1) = \ell$. Then the properties (1)–(4) are again true (with $m$ replaced by $m+1$), which contradicts the maximality of $m$. If $k+1 \leq m$, then $\mathrm{reach}[q(k+1),\ell](x_{k+1}, v, X)$ does not hold ($k$ is chosen maximal). Hence, $\mathrm{reach}[\ell, q(k+1)](v, x_{k+1}, X)$, because we have $\mathrm{reach}[k,\ell](u,v,X) \vee \mathrm{reach}[\ell,k](v,u,X)$ for all $u \in X_k$ and $v \in X_\ell$. But then the sequences $(x_1, \dots, x_k, v, x_{k+1}, \dots, x_m)$ and $(q(1), \dots, q(k), \ell, q(k+1), \dots, q(m))$ satisfy again the properties (1)–(4), which contradicts the maximality of $m$. So, we have $X = \{x_1, \dots, x_m\}$. Finally, we have $\mathrm{reach}[q(m), j](x_m, y, X)$, thus there exists the desired path. $\square$

With the help of Proposition 24 we can finish the proof of Theorem 15. Let $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ be a finite idempotent presentation. We want to show that the first-order theory of the structure $\mathcal{A} = \mathcal{C}(\mathrm{FIM}(\Gamma)/P, \Gamma \cup \Gamma^{-1})_{\mathrm{reg}}$ is decidable. For this, we use Theorem 9 and translate each first-order sentence $\varphi$ over $\mathcal{A}$ into an MSO-sentence $\widehat{\varphi}$ over the Cayley graph $\mathcal{C}(\Gamma)$ of the free group $\mathrm{FG}(\Gamma)$ such that for a sentence $\varphi$ over $\mathcal{A}$ we have: $\mathcal{A} \models \varphi$ if and only if $\mathcal{C}(\Gamma) \models \widehat{\varphi}$. Together with Theorem 5 this will complete the proof of Theorem 15.

To every variable $x$ (ranging over $\mathrm{FIM}(\Gamma)/P$) in $\varphi$ we associate two variables in $\widehat{\varphi}$:

- an MSO-variable $X'$ representing $\mathrm{cl}_P(\mathrm{MT}(u))$, where $u \in (\Gamma \cup \Gamma^{-1})^*$ is any word with $\mu_P(u) = x$, and
- a first-order variable $x'$, representing $\beta_P(x) \in \mathrm{FG}(\Gamma)$ (recall from the commutative diagram in Section 5 that $\beta_P : \mathrm{FIM}(\Gamma)/P \to \mathrm{FG}(\Gamma)$ is the canonical morphism).

Thus, by Theorem 9, $x = y$ if and only if $x' = y'$ and $X' = Y'$. The relationship between $x'$ and $X'$ is expressed by the MSO-formula (over the signature of $\mathcal{C}(\Gamma)$) $\mathrm{MT}(x', X') = \exists X : \Theta(x', X, X')$, where:

$$\Theta(x', X, X') \;=\; (1, x' \in X \;\wedge\; X \text{ is connected and finite} \;\wedge\; \mathrm{CL}_P(X, X'))$$

Recall that by Remark 1, finiteness and connectedness of a subset of the finitely-branching tree $\mathcal{C}(\Gamma)$ can be expressed in MSO. Here $\mathrm{CL}_P(X, X')$ is the MSO-formula constructed by Margolis and Meakin in [27], see the remark at the end of Section 5.

Now let $\varphi$ be an FO-formula over the signature of $\mathcal{A}$. We define $\widehat{\varphi}$ inductively as follows:

- for $\varphi = \mathrm{reach}_L(x, y)$ define $\widehat{\varphi} = \exists X \exists Y \exists Z : \Theta(x', X, X') \ \wedge \ \Theta(y', Y, Y') \ \wedge \ Y \backslash X \subseteq Z \subseteq Y \ \wedge \ \mathrm{Reach}_L(x', y', Z)$
- for $\varphi = \neg\psi$ define $\widehat{\varphi} = \neg\widehat{\psi}$
- for $\varphi = \psi_1 \wedge \psi_2$ define $\widehat{\varphi} = \widehat{\psi_1} \wedge \widehat{\psi_2}$
- for $\varphi = \forall x : \psi$ define $\widehat{\varphi} = \forall x' \ \forall X' : \mathrm{MT}(x', X') \Rightarrow \widehat{\psi}$

The intuition behind the first formula $\exists X \exists Y \exists Z : \Theta(x', X, X') \wedge \Theta(y', Y, Y') \wedge Y \backslash X \subseteq Z \subseteq Y \ \wedge \ \mathrm{Reach}_L(x', y', Z)$ is the following: We express that starting from the node $x' \in \mathrm{FG}(\Gamma)$ we traverse a path $p$ in $\mathcal{C}(\Gamma)$ labeled with a word from the language $L$ that ends in the node $y' \in \mathrm{FG}(G)$. Moreover, $Y$ is the union of $X$ and the nodes along the path $p$, and the closure of $X$ (resp. $Y$) is $X'$ (resp. $Y'$). Thus, $Y = \mathrm{MT}(uv)$ for some word $uv$ such that $X = \mathrm{MT}(u)$, $\gamma(u) = x'$, $\gamma(uv) = y'$, and $v \in L$. Hence, the word $u$ (resp. $uv$) represents $x \in \mathrm{FIM}(\Gamma)/P$ (resp. $y \in \mathrm{FIM}(\Gamma)/P$) and there is a path from $x$ to $y$ in the Cayley-graph of $\mathrm{FIM}(\Gamma)/P$ that is labeled with the word $v \in L$. Now it is straightforward to verify that $\mathcal{A} \models \varphi$ if and only if $\mathcal{C}(\Gamma) \models \widehat{\varphi}$. This concludes the proof of Theorem 15.

## 8 Further Research

In the extended abstract [13], some of the results of this paper are generalized to *free partially commutative inverse monoids*. These inverse monoids result from free inverse monoids by taking the quotient with respect to a partial commutation relation.

A promising research direction might be to investigate for which monoids $\mathcal{M}$ the structure $\mathcal{C}(\mathcal{M}, \Gamma)_{\mathrm{reg}}$ has a decidable first-order theory. As we have seen, the decidability of $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Gamma)_{\mathrm{reg}})$ implies the decidability of important algebraic problems for $\mathcal{M}$. Here, in particular, the group case is interesting. It is easy to see that the decidability of the MSO-theory of $\mathcal{C}(\mathcal{M}, \Gamma)$ implies the decidability of the first-order theory of $\mathcal{C}(\mathcal{M}, \Gamma)_{\mathrm{reg}}$. The class of groups for which the first-order (resp. MSO-) theory of the Cayley-graph is decidable is precisely the class of groups with a decidable word problem (resp. the class of virtually free groups). Hence, the class of groups $\mathcal{G}$ for which $\mathcal{C}(\mathcal{G}, \Gamma)_{\mathrm{reg}}$ is decidable lies somewhere between the virtually-free groups and the groups with a decidable word problem. Moreover, these inclusions are strict: By a reduction to Presburger's arithmetic it can be easily shown that for $\mathcal{G} = \mathbb{Z} \times \mathbb{Z}$ the first-order theory of $\mathcal{C}(\mathcal{G}, \Gamma)_{\mathrm{reg}}$ is decidable, but since $\mathcal{C}(\mathcal{G}, \Gamma)$ is an infinite grid, MSOTh($\mathcal{C}(\mathcal{G}, \Gamma)$) is undecidable. Furthermore, there exists a hyperbolic group $\mathcal{G}$ [14], for which the generalized word problem is undecidable [40]. Thus, the first-order theory of $\mathcal{C}(\mathcal{G}, \Gamma)_{\text{reg}}$ is undecidable. On the other hand, every hyperbolic group has a decidable word problem [14].

**Acknowledgments** We want to thank Arnaud Carayol, Didier Caucal, Volker Diekert, and Klaus-Jörn Lange for fruitful discussion on the topic of this paper.

## References

[1] J.-C. Birget, S. W. Margolis, and J. Meakin. The word problem for inverse monoids presented by one idempotent relator. *Theoretical Computer Science*, 123(2):273–289, 1994.

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

[3] W. W. Boone. The word problem. *Annals of Mathematics (2)*, 70:207–265, 1959.

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

[5] H. Calbrix. La théorie monadique du second ordre du monoïde inversif libre est indécidable (The second-order monadic theory of the free inverse monoid is undecidable). *Bulletin of the Belgian Mathematical Society*, 4:53–65, 1997.

[6] A. Carayol and D. Caucal. The Kleene equality for graphs. 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 214–225. Springer, 2006.

[7] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.

[8] C. Choffrut. Conjugacy in free inverse monoids. In K. U. Schulz, editor, *Word Equations and Related Topics*, number 572 in Lecture Notes in Computer Science, pages 6–22. Springer, 1991.

[9] C. Choffrut and F. D'Alessandro. Commutativity in free inverse monoids. *Theoretical Computer Science*, 204(1–2):35–54, 1998.

[10] K. J. Compton and C. W. Henson. A uniform method for proving lower bounds on the computational complexity of logical theories. *Annals of Pure and Applied Logic*, 48:1–79, 1990.

[11] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of graph grammars and computing by graph transformation, Volume 1 Foundations*, pages 313–400. World Scientific, 1997.

[12] T. Deis, J. Meakin, and G. Sénizergues. Equations in free inverse monoids. *International Journal of Algebra and Computation*, 2005. Accepted for publication.

[13] 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, 2006. long version in preparation.

[14] M. Gromov. Hyperbolic groups. In S. M. Gersten, editor, *Essays in Group Theory*, number 8 in MSRI Publ., pages 75–263. Springer, 1987.

[15] W. Hodges. *Model Theory.* Cambridge University Press, 1993.

[16] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages and computation.* Addison–Wesley, Reading, MA, 1979.

[17] 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.

[18] M. Kambites. The loop problem for monoids and semigroups. Technical report, arXiv.org, 2006. `http://arxiv.org/abs/math.RA/0609293`, to appear in Mathematical Proceedings of the Cambridge Philosophical Society.

[19] 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, 2000.

[20] D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the group case. *Annals of Pure and Applied Logic*, 131(1–3):263–286, 2005.

[21] D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the monoid case. *International Journal of Algebra and Computation*, 16(2):307–340, 2006.

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

[23] 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, 2001.

[24] M. Lohrey. Decidability and complexity in automatic monoids. *International Journal of Foundations of Computer Science*, 16(4):707–722, 2005.

[25] M. Lohrey and N. Ondrusch. Inverse monoids: decidability and complexity of algebraic questions. In J. Jedrzejowicz and A. Szepietowski, editors, *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS 2005), Gdansk (Poland)*, number 3618 in Lecture Notes in Computer Science, pages 664–675. Springer, 2005.

[26] R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory*. Springer, 1977.

[27] S. Margolis and J. Meakin. Inverse monoids, trees, and context-free languages. *Trans. Amer. Math. Soc.*, 335(1):259–276, 1993.

[28] 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.

[29] A. Markov. On the impossibility of certain algorithms in the theory of associative systems. *Doklady Akademii Nauk SSSR*, 55, 58:587–590, 353–356, 1947.

[30] 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.

[31] A. R. Meyer. Weak monadic second order theory of one successor is not elementary recursive. In *Proceedings of the Logic Colloquium (Boston 1972–73)*, number 453 in Lecture Notes in Mathematics, pages 132–154. Springer, 1975.

[32] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983.

[33] 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.

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

[35] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *American Mathematical Society, Translations, II. Series*, 9:1–122, 1958.

[36] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

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

[38] E. Post. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 12(1):1–11, 1947.

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

[40] E. Rips. Subgroups of small cancellation groups. *Bulletin of the London Mathematical Society*, 14:45–47, 1982.

[41] B. V. Rozenblat. Diophantine theories of free inverse semigroups. *Siberian Mathematical Journal*, 26:860–865, 1985. English translation.

[42] P. E. Schupp. Groups and graphs: Groups acting on trees, ends, and cancellation diagrams. *Mathematical Intelligencer*, 1:205–222, 1979.

[43] P. V. Silva. Rational languages and inverse monoid presentations. *International Journal of Algebra and Computation*, 2:187–207, 1992.

[44] P. V. Silva. On free inverse monoid languages. *R.A.I.R.O. — Informatique Théorique et Applications*, 30:349–378, 1996.

[45] P. V. Silva and B. Steinberg. Extensions and submonoids of automatic monoids. *Theoretical Computer Science*, 289:727–754, 2002.

[46] P. V. Silva and B. Steinberg. A geometric characterization of automatic monoids. *The Quarterly Journal of Mathematics*, 55:333–356, 2004.

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

[48] C. Stirling. *Modal and Temporal Properties of Processes*. Springer, 2001.

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