Compressed word problems for inverse monoids

Markus Lohrey

Universität Leipzig, Institut für Informatik, Germany
lohrey@informatik.uni-leipzig.de

Abstract. The compressed word problem for a finitely generated monoid M asks whether two given compressed words over the generators of M represent the same element of M. For string compression, straight-line programs, i.e., context-free grammars that generate a single string, are used in this paper. It is shown that the compressed word problem for a free inverse monoid of finite rank at least two is complete for Π_2^p (second universal level of the polynomial time hierarchy). Moreover, it is shown that there exists a fixed finite idempotent presentation (i.e., a finite set of relations involving idempotents of a free inverse monoid), for which the corresponding quotient monoid has a PSPACE-complete compressed word problem. The ordinary uncompressed word problem for such a quotient can be solved in logspace [10]. Finally, a PSPACE-algorithm that checks whether a given element of a free inverse monoid belongs to a given rational subset is presented. This problem is also shown to be PSPACE-complete (even for a fixed finitely generated submonoid instead of a variable rational subset).

1 Introduction

The decidability and complexity of algorithmic problems in finitely generated monoids and groups is a classical topic at the borderline of computer science and mathematics. The most basic question of this kind is the *word problem*, which asks whether two words over the generators represent the same element. Markov and Post proved independently that the word problem for finitely presented monoids is undecidable in general. Later, Novikov and Boone extended the result of Markov and Post to finitely presented groups, see the the survey [15] for references.

In this paper, we are interested in *inverse monoids*. A monoid is inverse, if for each element x there exists a unique "inverse" x^{-1} such that $x = xx^{-1}x$ and $x^{-1} = x^{-1}xx^{-1}$ [3]. In the same way as groups can be represented by sets of permutations, inverse monoids can be represented by sets of partial injections [3]. Algorithmic questions for inverse monoids received increasing attention in the past and inverse monoid theory found several applications in combinatorial group theory, see e.g. [10] and the survey [15] for further references.

Since the class of inverse monoids forms a variety of algebras (with respect to the operations of multiplication, inversion, and the identity element), the free inverse monoid $\mathsf{FIM}(\Gamma)$ generated by a set Γ exists. Munn gave in [16] an explicit representation of the free inverse monoid $\mathsf{FIM}(\Gamma)$. Elements can be represented by finite subtrees of the Cayley-graph of the free group generated by Γ (so called *Munn trees*). Moreover, there are two distinguished nodes (an initial node and a final node). Multiplication of two elements of $\mathsf{FIM}(\Gamma)$ amounts of gluing the two Munn trees together, where the final node of the first Munn tree is identified with the initial node of the second Munn tree. This gives rise to a very simple algorithm for the word problem of $\mathsf{FIM}(\Gamma)$, which can moreover implemented in linear time. In [10], it was also shown (using Munn trees together with a result of Lipton and Zalcstein [5] saying that the word problem for a finitely generated free group can be solved in logspace) that the word problem for $\mathsf{FIM}(\Gamma)$ can be solved in logspace.

Although the word problem for a free inverse monoid can be solved very efficiently, there are several subtle differences between the algorithmic properties of free inverse monoids on the one hand and free monoids and free groups on the other hand. Let us give two examples:

- Solvability of equations: By the seminal results of Makanin, this problem is decidable for free monoids and free groups. On the other hand, solvability of equations in a finitely generated free inverse monoid of rank at least 2 (the rank is the minimal number of generators) is undecidable [19].
- Rational subset membership problem: Membership in a given rational subset of a free monoid or free group can be decided in polynomial time. The same problem is NP-complete for finitely generated free inverse monoids of rank at least two [2].

In this paper, we show that in a certain sense also the word problem is harder for free inverse monoids than free monoids (groups). For this we consider the *compressed word problem*, where the input words are given succinctly by so called *straight-line programs* (SLPs) [18]. An SLP is a context free grammar that generates only one word, see Section 4. Since the length of this word may grow exponentially with the size (number of productions) of the SLP, SLPs can be seen as a compact string representation. SLPs turned out to be a very flexible compressed representation of strings, which are well suited for studying algorithms for compressed strings; see [8] for references. In the compressed word problem for a finitely generated monoid M the input consists of two SLPs that generate words over the generators of M, and it is asked whether these two words represent the same element of M. Hence, the compressed word problem for a free monoid simply asks, whether two SLPs generate the same word. Plandowski proved in [17] that this problem can be solved in polynomial time; the best algorithm is due to Lifshits [4] and has a cubic running time. Based on Plandowski's result, it was shown in [7] that the compressed word problem for a free group can be solved in polynomial time. This result has algorithmic implications for the ordinary (uncompressed) word problem: In [11, 20] it was shown that the word problem for the automorphism group of a group G can be reduced in polynomial time to the *compressed* word problem for G (more general: the word problem for the endomorphism monoid of a monoid M can be reduced in polynomial time to the *compressed* word problem for M). Hence, the word problem for the automorphism group of a free group turned out to be solvable in polynomial time [20], which solved an open problem from combinatorial group theory. Generalizations of this result for larger classes of groups can be found in [11, 13].

Our first main result states that the compressed word problem for every finitely generated free inverse monoid of rank at least two is complete for Π_2^p, the second universal level of the polynomial time hierarchy (Thm. 4). The upper bound follows easily using Munn's solution for the word problem together with the above mentioned result of

Lipton and Zalcstein for free groups. The lower bound is based on a reduction from a variant of the SUBSETSUM problem together with an encoding of a SUBSETSUM instance by an SLP [7]. Hence, the compressed word problem for free inverse monoids is indeed computationally harder than the compressed word problem for free monoids (groups) (unless $\mathsf{P} = \Pi_2^p$). It is not difficult to see that the compressed word problem for a free inverse monoid of rank 1 can be solved in polynomial time (Prop. 1).

In [14], 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 $\mathsf{FIM}(\Gamma)/P$, where P is a presentation consisting of a finite number of relations $e = f$, where e and f are idempotents of $\mathsf{FIM}(\Gamma)$; we call such a presentation idempotent. An alternative proof for the decidability result of Margolis and Meakin was given in [21]. In [10] it was shown that the word problem for every inverse monoid $\mathsf{FIM}(\Gamma)/P$, where P is an idempotent presentation, can be solved in logspace. This implies that the compressed word problem for each of these inverse monoids belongs to the class PSPACE. Our second main result states that the are specific idempotent presentations P such that the compressed word problem for $\mathsf{FIM}(\Gamma)/P$ is PSPACE-complete (Thm. 5).

In the last part of the paper we consider the compressed variant of the rational subset membership problem. The class of rational subsets of a monoid M is the smallest class of subsets, which contains all finite subsets, and which is closed under union, product and Kleene star (A^* is the submonoid generated by the subset $A \subseteq M$). If M is finitely generated by Γ, then a rational subset of M can be represented by a finite automaton over the alphabet Γ. In this case, the rational subset membership problem asks, whether a given element of M (given by a finite word over Γ) belongs to a given rational subset (given by a finite automaton over Γ). Especially for groups, this problem is intensively studied, see e.g. [12]. In [2], it was shown that the rational subset membership problem for a free inverse monoid of finite rank at least two is NP-complete. Here, we consider the *compressed rational subset membership problem*, where the input consists of an SLP-compressed word over the generators and a finite automaton over the generators. We show that the compressed rational subset membership problem for a free inverse monoid of finite rank at least two is PSPACE-complete. The difficult part of the proof is to show membership in PSPACE. PSPACE-hardness holds already for the case that the rational subset is a fixed finitely generated submonoid (Thm. 6).

Proofs that are omitted in this paper can be found in the long version [9].

2 Preliminaries

Let Γ be a finite alphabet. The *empty word* over Γ is denoted by ε. Let $s = a_1 \cdots a_n \in \Gamma^*$ be a word over Γ, where $n \geq 0$ and $a_1, \ldots, a_n \in \Gamma$ for $1 \leq i \leq n$. The *length* of s is $|s| = n$. 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 $s[i,j] = \varepsilon$. For $n \in \mathbb{N}$ let $\Gamma^{\leq n} = \{w \in \Gamma^* \mid |w| \leq n\}$. We write $s \preceq t$ for $s, t \in \Gamma^*$, if s is a prefix of t. A set $A \subseteq \Gamma^*$ is *prefix-closed*, if $u \preceq v \in A$ implies $u \in A$. We denote with $\Gamma^{-1} = \{a^{-1} \mid a \in \Gamma\}$ a disjoint copy of the finite alphabet Γ. 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 $a_i \in \Gamma \cup \Gamma^{-1}$. For $a \in \Gamma \cup \Gamma^{-1}$ and $n \geq 0$ we use a^{-n} as an abbreviation for the word $(a^{-1})^n$. We use standard terminology from automata theory. A *nondeterministic finite automaton* (NFA) over an input alphabet Γ is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is the set of states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. For a *deterministic finite automaton,* $\delta : Q \times \Sigma \rightarrow_p Q$ is a partial mapping from $Q \times \Sigma$ to Q.

Complexity theory: We assume some basic background in complexity theory. Recall that Π_2^p (the second universal level of the polynomial time hierarchy) is the class of all languages L for which there exists a polynomial time predicate $P(x, y, z)$ and a polynomial $p(n)$ such that $L = \{x \in \Sigma^* \mid \forall y \in \Sigma^{\leq p(|x|)} \exists z \in \Sigma^{\leq p(|x|)} : P(x, y, z)\}$. POLYLOGSPACE denotes the class $\mathsf{NSPACE}(\log(n)^{O(1)}) = \mathsf{DSPACE}(\log(n)^{O(1)})$. A PSPACE-transducer is a deterministic Turing machine with a read-only input tape, a write-only output tape and a working tape, whose length is bounded by $n^{O(1)}$, where n is the input length. The output is written from left to right on the output tape, i.e., in each step the transducer either outputs a new symbol on the output tape, in which case the output head moves one cell to the right, or the transducer does not output a new symbol in which case the output head does not move. Moreover, we assume that the transducer terminates for every input. This implies that a PSPACE-transducer computes a mapping $f : \Sigma^* \to \Theta^*$, where $|f(w)|$ is bounded by $2^{|w|^{O(1)}}$. A POLYLOGSPACE-transducer is defined in the same way as a PSPACE-transducer, except that the length of the working tape is bounded by $\log(n)^{O(1)}$. The proof of the following lemma uses the same idea that shows that logspace reducibility is transitive.

Lemma 1. *Assume that $f : \Sigma^* \to \Theta^*$ can be computed by a* PSPACE-*transducer and that $g : \Theta^* \to \Delta^*$ can be computed by a* POLYLOGSPACE-*transducer. Then the mapping $f \circ g$ can be computed by a* PSPACE-*transducer. In particular, if the language $L \subseteq \Theta^*$ belongs to* POLYLOGSPACE, *then $f^{-1}(L)$ belongs to* PSPACE.

Free groups: It is common to identify a congruence α on a monoid M with the surjective homomorphism from M to the quotient M/α that maps an element $m \in M$ to the congruence class of m with respect to α. The *free group* $\mathsf{FG}(\Gamma)$ generated by the set Γ is the quotient monoid

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

where δ is the smallest congruence on $(\Gamma \cup \Gamma^{-1})^*$ that contains all pairs (bb^{-1}, ε) for $b \in \Gamma \cup \Gamma^{-1}$. 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 $\delta(u) = \delta(r(u))$ and $r(u)$ does not contain a factor of the form bb^{-1} for $b \in \Gamma \cup \Gamma^{-1}$. It holds $\delta(u) = \delta(v)$ if and only if $r(u) = r(v)$. Since the word $r(u)$ can be calculated from u in linear time, the word problem for $\mathsf{FG}(\Gamma)$ can be solved in linear time. Let $\mathsf{IRR}(\Gamma) = \{r(u) \mid u \in (\Gamma \cup \Gamma^{-1})^*\}$ be the set of all *irreducible* words. The epimorphism $\delta : (\Gamma \cup \Gamma^{-1})^* \to \mathsf{FG}(\Gamma)$ restricted to $\mathsf{IRR}(\Gamma)$ is a bijection.

The Cayley-graph of $\mathsf{FG}(\Gamma)$ with respect to the standard generating set $\Gamma \cup \Gamma^{-1}$ will be denoted by $\mathcal{C}(\Gamma)$. Its vertex set is $\mathsf{FG}(\Gamma)$ and there is an a-labeled edge ($a \in \Gamma \cup \Gamma^{-1}$) from $x \in \mathsf{FG}(\Gamma)$ to $y \in \mathsf{FG}(\Gamma)$ if $y = xa$ in $\mathsf{FG}(\Gamma)$. Note that $\mathsf{FG}(\Gamma)$ is a finitely-branching tree. Figure 1 shows a finite portion of $\mathcal{C}(\{a, b\})$. Here, and in the following,

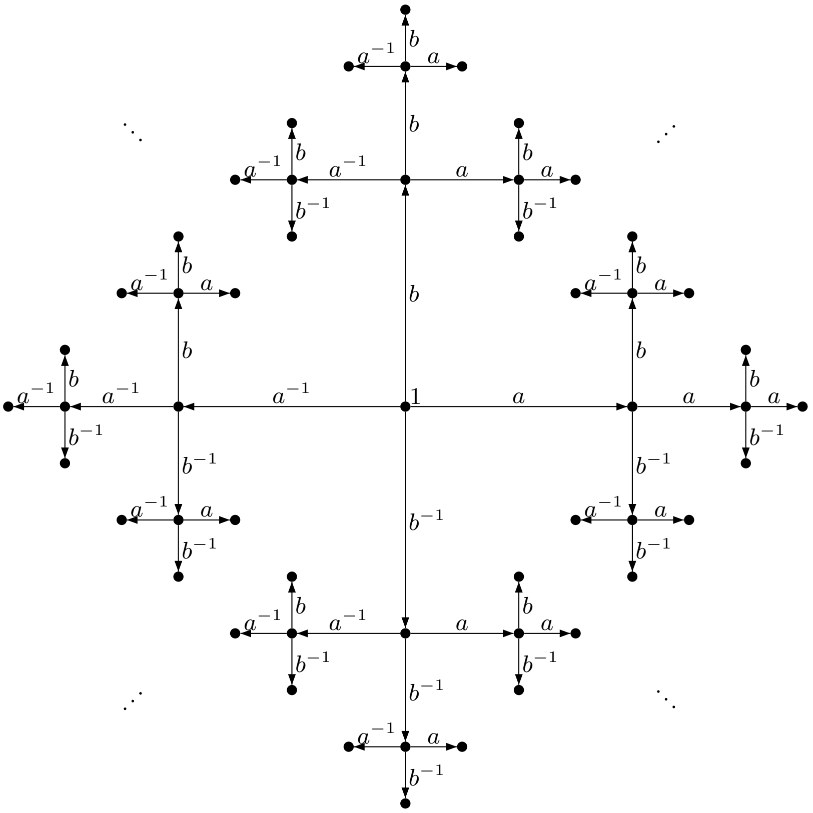


Fig. 1. The Cayley-graph $\mathcal{C}(\{a,b\})$ of the free group $\mathsf{FG}(\{a,b\})$

we only draw one directed edge between two points. Thus, for every drawn a-labeled edge we omit the a^{-1}-labeled reversed edge.

3 Inverse monoids

A monoid M is called an *inverse monoid* if for every $m \in M$ there is a *unique* $m^{-1} \in M$ such that $m = mm^{-1}m$ and $m^{-1} = m^{-1}mm^{-1}$. For detailed reference on inverse monoids see [3]; here we only recall the basic notions. Since the class of inverse monoids forms a variety of algebras (with respect to the operations of multiplication, inversion, and the identity element), the free inverse monoid $\mathsf{FIM}(\Gamma)$ generated by a set Γ exists. Vagner gave an explicit presentation of $\mathsf{FIM}(\Gamma)$: Let ρ be 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 $(ww^{-1}vv^{-1}, vv^{-1}ww^{-1})$; these identities are also called Vagner equations. Then $\mathsf{FIM}(\Gamma) \simeq (\Gamma \cup \Gamma^{-1})^*/\rho$. An element x of an inverse monoid M is idempotent (i.e., $x^2 = x$) if and only if x is of the form mm^{-1} for some $m \in M$.

Hence, Vagner's presentation of $\mathsf{FIM}(\Gamma)$ implies that idempotent elements in an inverse monoid commute. Since the Vagner equations are true in the free group $\mathsf{FG}(\Gamma)$, there exists a congruence γ on $\mathsf{FIM}(\Gamma)$ such that $\mathsf{FG}(\Gamma) = \mathsf{FIM}(\Gamma)/\gamma$. When viewing congruences as homomorphisms, we have $\delta = \rho \circ \gamma$, where δ is the congruence on $(\Gamma \cup \Gamma^{-1})^*$ from (1). Elements of $\mathsf{FIM}(\Gamma)$ can be also represented via *Munn trees*: The Munn tree $\mathsf{MT}(u)$ of $u \in (\Gamma \cup \Gamma^{-1})^*$ is a finite and prefix-closed subset of $\mathsf{IRR}(\Gamma)$; it is defined by

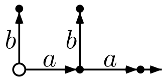
$$\mathsf{MT}(u) = \{r(v) \mid v \preceq u\}.$$

By identifying an irreducible word $v \in \mathsf{IRR}(\Gamma)$ with the group element $\delta(v)$, $\mathsf{MT}(u)$ becomes the set of all nodes along the unique path in $C(\Gamma)$ that starts in 1 and that is labeled with the word u. The subgraph of the Cayley-graph $C(\Gamma)$, which is induced by $\mathsf{MT}(u)$ is connected. Hence it is a finite tree and we can identify $\mathsf{MT}(u)$ with this tree. The following result is known as Munn's Theorem:

Theorem 1 ([16]). *For all $u, v \in (\Gamma \cup \Gamma^{-1})^*$, we have: $\rho(u) = \rho(v)$ if and only if ($r(u) = r(v)$ and $\mathsf{MT}(u) = \mathsf{MT}(v)$).*

Thus, $\rho(u) \in \mathsf{FIM}(\Gamma)$ can be uniquely represented by the pair $(\mathsf{MT}(u), r(u))$. In fact, if we define on the set of all pairs $(U, v) \in 2^{\mathsf{IRR}(\Gamma)} \times \mathsf{IRR}(\Gamma)$ (with $v \in U$ and U finite and prefix-closed) a multiplication by $(U, v)(V, w) = (r(U \cup vV), r(vw))$, then the resulting monoid is isomorphic to $\mathsf{FIM}(\Gamma)$. Quite often, we represent an element $\rho(u) \in \mathsf{FIM}(\Gamma)$ by a diagram for its Munn tree, where in addition the node ε is represented by a bigger circle and the node $r(u)$ is marked by an outgoing arrow. If $r(u) = \varepsilon$, then we omit this arrow. By Thm. 1 such a diagram uniquely specifies an element of $\mathsf{FIM}(\Gamma)$.

Example 1. The diagram for $\rho(bb^{-1}abb^{-1}a) \in \mathsf{FIM}(\{a, b\})$ looks as follows:



Thm. 1 leads to a polynomial time algorithm for the word problem for $\mathsf{FIM}(\Gamma)$. For instance, the reader can easily check that $bb^{-1}abb^{-1}a = aaa^{-1}bb^{-1}a^{-1}bb^{-1}aa$ in $\mathsf{FIM}(\{a, b\})$ by using Munn's Theorem. In fact, every word that labels a path from ε to aa (the node with the outgoing arrow) and that visits all nodes of the above diagram represents the same element of $\mathsf{FIM}(\{a, b\})$ as $bb^{-1}abb^{-1}a$. Munn's theorem also implies that an element $\rho(u) \in \mathsf{FIM}(\Gamma)$ (where $u \in (\Gamma \cup \Gamma^{-1})^*$) is idempotent (i.e., $\rho(uu) = \rho(u)$) if and only if $r(u) = \varepsilon$.

For a finite set $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ define $\mathsf{FIM}(\Gamma)/P = (\Gamma \cup \Gamma^{-1})^*/\tau_P$ to be the inverse monoid with the set Γ of generators and the set P of relations, where τ_P is the smallest congruence on $(\Gamma \cup \Gamma^{-1})^*$ generated by $\rho \cup P$. Viewed as a morphism, this congruence factors as $\tau_P = \rho \circ \nu_P$ with $\mathsf{FIM}(\Gamma)/\nu_P = \mathsf{FIM}(\Gamma)/P$. We say that $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ is an *idempotent presentation* if for all $(e, f) \in P$, $\rho(e)$ and $\rho(f)$ are both idempotents of $\mathsf{FIM}(\Gamma)$, i.e., $r(e) = r(f) = \varepsilon$ by the remark above. In this paper, we are concerned with inverse monoids of the form $\mathsf{FIM}(\Gamma)/P$ for a finite idempotent presentation P. In this case, since every identity $(e, f) \in P$ is true in $\mathsf{FG}(\Gamma)$ (we have $\delta(e) = \delta(f) = 1$), there also exists a congruence γ_P on $\mathsf{FIM}(\Gamma)/P$ with $(\mathsf{FIM}(\Gamma)/P)/\gamma_P = \mathsf{FG}(\Gamma)$. The following commutative diagram summarizes all morphisms introduced so far.

$$\begin{tikzcd}
 & (\Gamma \cup \Gamma^{-1})^* \arrow[ld, "\rho"'] \arrow[d, "\tau_P"] \arrow[rd, "\delta"] & \\
\mathsf{FIM}(\Gamma) \arrow[r, "\nu_P"] \arrow[rr, bend right, "\gamma"'] & \mathsf{FIM}(\Gamma)/P \arrow[r, "\gamma_P"] & \mathsf{FG}(\Gamma)
\end{tikzcd}$$

In the sequel, the meaning of the congruences $\rho, \delta, \gamma_P, \gamma, \tau_P$, and ν_P will be fixed.

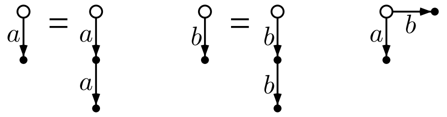
To solve the word problem for $\mathsf{FIM}(\Gamma)/P$, Margolis and Meakin [14] used a closure operation for Munn trees, which is based on work of Stephen [22]. We shortly review the ideas here. As remarked in [14], every idempotent presentation P can be replaced by the idempotent presentation $P' = \{(e, ef), (f, ef) \mid (e, f) \in P\}$, i.e., $\mathsf{FIM}(\Gamma)/P = \mathsf{FIM}(\Gamma)/P'$. Since $\mathsf{MT}(e) \subseteq \mathsf{MT}(ef) \supseteq \mathsf{MT}(f)$ if $r(e) = r(f) = \varepsilon$, we can restrict in the following to idempotent presentations P such that $\mathsf{MT}(e) \subseteq \mathsf{MT}(f)$ for all $(e, f) \in P$. Define a rewriting relation $\Rightarrow_P$ on prefix-closed subsets of $\mathsf{IRR}(\Gamma)$ as follows, where $U, V \subseteq \mathsf{IRR}(\Gamma)$: $U \Rightarrow_P V$ if and only if

$$\exists (e, f) \in P \; \exists u \in U \big(r(u\, \mathsf{MT}(e)) \subseteq U \text{ and } V = U \cup r(u\, \mathsf{MT}(f)) \big).$$

Finally, define the closure of $U \subseteq \mathsf{IRR}(\Gamma)$ with respect to the presentation P as

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

Example 2. Assume that $\Gamma = \{a, b\}$, $P = \{(aa^{-1}, a^2a^{-2}), (bb^{-1}, b^2b^{-2})\}$ and $u = aa^{-1}bb^{-1}$. The graphical representations for these elements look as follows:



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

Margolis and Meakin proved the following result:

Theorem 2 ([14]). *Let P be an idempotent presentation and let $u, v \in (\Gamma \cup \Gamma^{-1})^*$. Then $\tau_P(u) = \tau_P(v)$ if and only if $(r(u) = r(v)$ and $\mathsf{cl}_P(\mathsf{MT}(u)) = \mathsf{cl}_P(\mathsf{MT}(v)))$.*

The result of Munn for $\mathsf{FIM}(\Gamma)$ (Thm. 1) is a special case of this result for $P = \emptyset$. Note also that $\mathsf{cl}_P(\mathsf{MT}(u)) = \mathsf{cl}_P(\mathsf{MT}(v))$ if and only if $\mathsf{MT}(u) \subseteq \mathsf{cl}_P(\mathsf{MT}(v))$ and $\mathsf{MT}(v) \subseteq \mathsf{cl}_P(\mathsf{MT}(u))$. Margolis and Meakin used Thm. 2 in connection with Rabin's tree theorem in order to give a solution for the word problem for the monoid $\mathsf{FIM}(\Gamma)/P$. Using tree automata techniques, a logspace algorithm for the word problem for $\mathsf{FIM}(\Gamma)/P$ was given in [10]. For this result, it is important that the idempotent presentation P is not part of the input. The uniform version of the word problem, where P is part of the input, is $\mathsf{EXPTIME}$-complete [10].

4 Straight-line programs

We are using straight-line programs as a succinct representation of strings with reoccurring subpatterns [18]. A *straight-line program (SLP) over a finite alphabet Γ* is a context free grammar $\mathbb{A} = (V, \Gamma, S, P)$, where V is the set of *nonterminals*, Γ is the set of *terminals*, $S \in V$ is the *initial nonterminal*, and $P \subseteq V \times (V \cup \Gamma)^*$ is the set of *productions* such that (i) for every $X \in V$ there is exactly one $\alpha \in (V \cup \Gamma)^*$ with $(X, \alpha) \in P$ and (ii) there is no cycle in the relation $\{(X, Y) \in V \times V \mid \exists \alpha \in (V \cup \Gamma)^* Y (V \cup \Gamma)^* : (X, \alpha) \in P\}$. These conditions ensure that the language generated by the straight-line program $\mathbb{A}$ contains exactly one word $\mathsf{val}(\mathbb{A})$.

Remark 1. The following problems can be solved in polynomial time:

(a) Given an SLP $\mathbb{A}$, calculate $|\mathsf{val}(\mathbb{A})|$ in binary representation.
(b) Given an SLP $\mathbb{A}$ and two binary coded numbers $1 \leq i \leq j \leq |\mathsf{val}(\mathbb{A})|$, compute an SLP $\mathbb{B}$ with $\mathsf{val}(\mathbb{B}) = \mathsf{val}(\mathbb{A})[i, j]$.

Also notice that $\mathsf{val}(\mathbb{A})$ can be computed from $\mathbb{A}$ by a PSPACE-transducer.

Plandowski [17] presented a polynomial time algorithm for testing whether $\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B})$ for two given SLPs $\mathbb{A}$ and $\mathbb{B}$. A cubic algorithm was presented by Lifshits [4].

Let M be a finitely generated monoid and let Γ be a finite generating set for M. The *compressed word problem* for M is the following computational problem:

INPUT: SLPs $\mathbb{A}$ and $\mathbb{B}$ over the alphabet Γ.
QUESTION: Does $\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B})$ hold in M?

The above mentioned result of Plandowski [17] means that the compressed word problem for a finitely generated free monoid can be solved in polynomial time. The following result was shown in [7].

Theorem 3 ([7]). *For every finite alphabet Γ, the compressed word problem for $\mathsf{FG}(\Gamma)$ can be solved in polynomial time (and is P-complete if $|\Gamma| \geq 2$).*

5 Compressed word problem for $\mathsf{FIM}(\Gamma)$

Recall that the word problem for $\mathsf{FIM}(\Gamma)$ can be solved in logspace [10]. In the compressed setting we have:

Theorem 4. *For every finite alphabet Γ with $|\Gamma| \geq 2$, the compressed word problem for $\mathsf{FIM}(\Gamma)$ is Π_2^p-complete.*

Proof. For the Π_2^p upper bound, let $\mathbb{A}$ and $\mathbb{B}$ be SLPs over some alphabet $\Gamma \cup \Gamma^{-1}$ and let $m = |\mathsf{val}(\mathbb{A})|$ and $n = |\mathsf{val}(\mathbb{B})|$. These numbers can be computed in polynomial time by Remark 1. By Thm. 1, we have $\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B})$ in $\mathsf{FIM}(\Gamma)$ if and only if:

$$\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B}) \text{ in } \mathsf{FG}(\Gamma) \tag{2}$$

$$\forall i \in \{0, \ldots, m\}\, \exists j \in \{0, \ldots, n\} : \mathsf{val}(\mathbb{A})[1, i] = \mathsf{val}(\mathbb{B})[1, j] \text{ in } \mathsf{FG}(\Gamma) \tag{3}$$

$$\forall i \in \{0, \ldots, n\}\, \exists j \in \{0, \ldots, m\} : \mathsf{val}(\mathbb{B})[1, i] = \mathsf{val}(\mathbb{A})[1, j] \text{ in } \mathsf{FG}(\Gamma) \tag{4}$$

Thm. 3 implies that (2) can be checked in polynomial time, whereas (3) and (4) are Π_2^p-properties.

It suffices to prove the lower bound for $\Gamma = \{a, b\}$. We make a logspace reduction from the following Π_2^p-complete problem [1], where $\overline{u} \cdot \overline{v} = u_1 v_1 + \cdots + u_n v_n$ denotes the scalar product of two integer vectors $\overline{u} = (u_1, \ldots, u_n)$, $\overline{v} = (v_1, \ldots, v_n)$:

INPUT: vectors $\overline{u} = (u_1, \ldots, u_m) \in \mathbb{N}^m$, $\overline{v} = (v_1, \ldots, v_n) \in \mathbb{N}^n$, and $t \in \mathbb{N}$ (all coded binary)
QUESTION: Does $\forall \overline{x} \in \{0,1\}^m \exists \overline{y} \in \{0,1\}^n : \overline{u} \cdot \overline{x} + \overline{v} \cdot \overline{y} = t$ hold?

Let $s = u_1 + \cdots + u_m + v_1 + \cdots + v_n$, $s_u = u_1 + \cdots + u_m$, and $s_v = v_1 + \cdots + v_n$. W.l.o.g. we can assume $t < s$. Using the construction from [7] (proof of Theorem 5.2) we can construct in logspace an SLP $\mathbb{A}_1$ such that $\mathsf{val}(\mathbb{A}_1) = \prod_{\overline{x} \in \{0,1\}^m} a^{\overline{u} \cdot \overline{x}} A_1 a^{s_u - \overline{u} \cdot \overline{x}}$. Here the product is taken over all tuples from $\{0,1\}^m$ in lexicographic order. By replacing A_1 by $A_2 a^{s_v}$ (which can be easily generated by a small SLP), we obtain an SLP $\mathbb{A}_2$ with $\mathsf{val}(\mathbb{A}_2) = \prod_{\overline{x} \in \{0,1\}^m} a^{\overline{u} \cdot \overline{x}} A_2 a^{s - \overline{u} \cdot \overline{x}}$. Similarly, we obtain an SLP $\mathbb{A}_3$ with $\mathsf{val}(\mathbb{A}_3) = \prod_{\overline{y} \in \{0,1\}^n} a^{\overline{v} \cdot \overline{y}} (bb^{-1} a^{-s_v}) a^{s_v - \overline{v} \cdot \overline{y}}$. Finally, be replacing A_2 in $\mathbb{A}_2$ by the start nonterminal of $\mathbb{A}_3$ we obtain an SLP $\mathbb{A}$ with

$$\mathsf{val}(\mathbb{A}) = \prod_{\overline{x} \in \{0,1\}^m} \left[a^{\overline{u} \cdot \overline{x}} \prod_{\overline{y} \in \{0,1\}^n} \left(a^{\overline{v} \cdot \overline{y}} bb^{-1} a^{-s_v} a^{s_v - \overline{v} \cdot \overline{y}} \right) a^{s - \overline{u} \cdot \overline{x}} \right].$$

Moreover, it is easy to construct a second SLP $\mathbb{B}$ such that

$$\mathsf{val}(\mathbb{B}) = \mathsf{val}(\mathbb{A}) a^{-s \cdot 2^m} \left(a^t bb^{-1} a^{s-t} \right)^{2^m}.$$

We claim that $\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B})$ in $\mathsf{FIM}(\{a, b\})$ if and only if

$$\forall \overline{x} \in \mathbb{N}^m \exists \overline{y} \in \mathbb{N}^n : \overline{u} \cdot \overline{x} + \overline{v} \cdot \overline{y} = t. \tag{5}$$

We have $r(\mathsf{val}(\mathbb{A})) = r(\mathsf{val}(\mathbb{B})) = a^{s \cdot 2^m}$. Thus, $\mathsf{val}(\mathbb{A}) = \mathsf{val}(\mathbb{B})$ holds in $\mathsf{FIM}(\{a, b\})$ if and only if $\mathsf{MT}(\mathsf{val}(\mathbb{A})) = \mathsf{MT}(\mathsf{val}(\mathbb{B}))$. Since $\mathsf{val}(\mathbb{A})$ is a prefix of $\mathsf{val}(\mathbb{B})$, we obtain $\mathsf{MT}(\mathsf{val}(\mathbb{A})) \subseteq \mathsf{MT}(\mathsf{val}(\mathbb{B}))$. Moreover, for the prefix $\mathsf{val}(\mathbb{A}) a^{-s \cdot 2^m}$ of $\mathsf{val}(\mathbb{B})$ we have $r(\mathsf{val}(\mathbb{A}) a^{-s \cdot 2^m}) = \varepsilon$ and $\mathsf{MT}(\mathsf{val}(\mathbb{A}) a^{-s \cdot 2^m}) = \mathsf{MT}(\mathsf{val}(\mathbb{A}))$. This and the fact that $\mathsf{MT}(\mathsf{val}(\mathbb{A})) \subseteq \mathsf{MT}(\mathsf{val}(\mathbb{B}))$ implies that $\mathsf{MT}(\mathsf{val}(\mathbb{A})) = \mathsf{MT}(\mathsf{val}(\mathbb{B}))$ if and only if

$$\mathsf{MT}((a^t bb^{-1} a^{s-t})^{2^m}) \subseteq \mathsf{MT}(\mathsf{val}(\mathbb{A})). \tag{6}$$

We show that (6) is equivalent to (5). We have

$$\mathsf{MT}((a^t bb^{-1} a^{s-t})^{2^m}) = \{a^i \mid 0 \leq i \leq s \cdot 2^m\} \cup \{a^{t + k \cdot s} b \mid 0 \leq k < 2^m\}.$$

Since $r(\mathsf{val}(\mathbb{A})) = a^{s \cdot 2^m}$, we have $a^i \in \mathsf{MT}(\mathsf{val}(\mathbb{A}))$ for all $0 \leq i \leq s \cdot 2^m$. Hence, (6) is equivalent to $a^{t + k \cdot s} b \in \mathsf{MT}(\mathsf{val}(\mathbb{A}))$ for every $0 \leq k < 2^m$, i.e. (for a bit vector $\overline{u} = (u_1, \ldots, u_n) \in \{0,1\}^n$ let $n(\overline{u}) = \sum_{i=1}^n u_i 2^{i-1}$ be the number represented by $\overline{u}$)

$$\forall \overline{x} \in \{0,1\}^m : a^{n(\overline{x}) \cdot s + t} b \in \mathsf{MT}(\mathsf{val}(\mathbb{A})). \tag{7}$$

Now, $\mathsf{MT}(\mathsf{val}(\mathbb{A})) \cap a^* b = \{a^{n(\overline{x}) \cdot s + \overline{u} \cdot \overline{x} + \overline{v} \cdot \overline{y}} b \mid \overline{x} \in \{0,1\}^m, \overline{y} \in \{0,1\}^n\}$. Hence, (7) if and only if $\forall \overline{x} \in \{0,1\}^m \exists \overline{y} \in \{0,1\}^n : \overline{u} \cdot \overline{x} + \overline{v} \cdot \overline{y} = t$. □

For a free inverse monoid of rank one, the compressed word problem is simpler:

Proposition 1. *The compressed word problem for* $\mathsf{FIM}(\{a\})$ *can be solved in polynomial time.*

6 Compressed word problems for $\mathsf{FIM}(\Gamma)/P$

For an inverse monoid of the form $\mathsf{FIM}(\Gamma)/P$, where Γ is finite and P is a finite idempotent presentation, the word problem can be still solved in logspace [10]. In this case, the complexity of the compressed word problem reaches even PSPACE:

Theorem 5. *The following holds:*

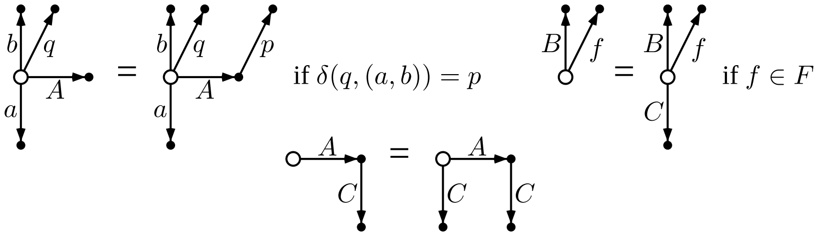
(a) For every finite idempotent presentation $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$, the compressed word problem for $\mathsf{FIM}(\Gamma)/P$ belongs to PSPACE.
(b) There exists a fixed finite idempotent presentation $P \subseteq (\Gamma \cup \Gamma^{-1})^* \times (\Gamma \cup \Gamma^{-1})^*$ such that the compressed word problem for $\mathsf{FIM}(\Gamma)/P$ is PSPACE-complete.

Proof. Let us first show (a). In [10], it was shown that the ordinary word problem for $\mathsf{FIM}(\Gamma)/P$ can be solved in logarithmic space. Since $\mathsf{val}(\mathbb{A})$ can be computed from $\mathbb{A}$ by a PSPACE-transducer (Remark 1), statement (a) follows from Lemma 1.

For the lower bound in (b), we use the following recent result from [8]: There exists a fixed regular language L over some paired alphabet $\Sigma \times \Theta$ such that the following problem is PSPACE-complete (for strings $u \in \Sigma^*, v \in \Theta^*$ with $|u| = |v| = n$ let $u \otimes v = (u[1], v[1]) \cdots (u[n], v[n]) \in (\Sigma \times \Theta)^*$):

INPUT: SLPs $\mathbb{A}$ (over Σ) and $\mathbb{B}$ (over Θ) with $|\mathsf{val}(\mathbb{A})| = |\mathsf{val}(\mathbb{B})|$
QUESTION: Does $\mathsf{val}(\mathbb{A}) \otimes \mathsf{val}(\mathbb{B}) \in L$ hold?

W.l.o.g. assume that $\Sigma \cap \Theta = \emptyset$. Let $\mathcal{A} = (Q, \Sigma \times \Theta, \delta, q_0, F)$ be a deterministic finite automaton with $L(\mathcal{A}) = L$. Let $\Gamma = \Sigma \cup \Theta \cup Q \cup \{A, B, C\}$ (all unions are assumed to be disjoint). Consider the fixed idempotent presentation over the alphabet Γ with the following relations:

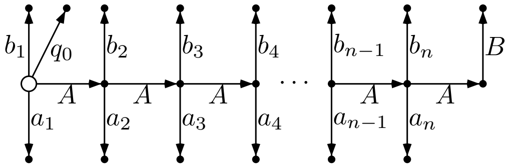


With the upper left relation, we simulate the automaton $\mathcal{A}$. The upper right relation allows to add a C-labeled edge as soon as a final state is reached; the B-labeled edge acts as a kind of end marker for the input word. Finally, the last relation allows to propagate the C-labeled edge back to the origin (node 1).

Assume that $\mathsf{val}(\mathbb{A}) = a_1 \cdots a_n$ and $\mathsf{val}(\mathbb{B}) = b_1 \cdots b_n$. Consider the string

$$w = q_0 q_0^{-1} \prod_{i=1}^{n} (a_i a_i^{-1} A) B B^{-1} \prod_{i=0}^{n-1} (A^{-1} b_{n-i} b_{n-i}^{-1}).$$

It is easy to compute from $\mathbb{A}$ and $\mathbb{B}$ in polynomial time an SLP $\mathbb{C}$ with $\mathsf{val}(\mathbb{C}) = w$. The Munn tree $\mathsf{MT}(w)$ looks as follows:



We claim that $w = CC^{-1}w$ in $\mathsf{FIM}(\Gamma)/P$ if and only if $\mathsf{val}(\mathbb{A}) \otimes \mathsf{val}(\mathbb{B}) \in L(\mathcal{A})$. Clearly, $w = CC^{-1}w = 1$ in $\mathsf{FG}(\Gamma)$. Moreover, $\mathsf{cl}_P(\mathsf{MT}(w)) = \mathsf{cl}_P(\mathsf{MT}(CC^{-1}w))$ if and only if $C \in \mathsf{cl}_P(\mathsf{MT}(w))$. Thus, it suffices to show that $C \in \mathsf{cl}_P(\mathsf{MT}(w))$ if and only if $\mathsf{val}(\mathbb{A}) \otimes \mathsf{val}(\mathbb{B}) \in L(\mathcal{A})$. First, assume that $\mathsf{val}(\mathbb{A}) \otimes \mathsf{val}(\mathbb{B}) \notin L(\mathcal{A})$. Let q_i be the state of $\mathcal{A}$ after reading $(a_1, b_1) \cdots (a_i, b_i)$ $(0 \le i \le n)$. Thus, $q_n \notin F$. This implies that $\mathsf{cl}_P(\mathsf{MT}(w)) = \mathsf{MT}(w) \cup \{A^i q_i \mid 0 \le i \le n\}$. Hence, $C \notin \mathsf{cl}_P(\mathsf{MT}(w))$. On the other hand, if $q_n \in F$, then $\mathsf{cl}_P(\mathsf{MT}(w)) = \mathsf{MT}(w) \cup \{A^i q_i, A^i C \mid 0 \le i \le n\}$ and therefore $C \in \mathsf{cl}_P(\mathsf{MT}(w))$. □

7 Rational subset membership problems

In this section we briefly outline our results on the compressed variant of the rational subset membership problem for free inverse monoids. We start with a lower bound.

Theorem 6. *There exists a fixed alphabet Γ and a fixed finite subset $K \subseteq (\Gamma \cup \Gamma^{-1})^*$ such that the following problem is* PSPACE*-hard:*
INPUT: An SLP $\mathbb{A}$ over the alphabet $\Gamma \cup \Gamma^{-1}$
QUESTION: Does $\rho(\mathsf{val}(\mathbb{A})) \in \rho(K^*)$ hold?

Note that $\rho(K^*)$ is the submonoid of $\mathsf{FIM}(\Gamma)$ generated by $\rho(K)$. Let us now turn to an upper bound.

Theorem 7. *The following problem belongs to* PSPACE*:*
INPUT: An SLP $\mathbb{A}$ over an alphabet $\Gamma \cup \Gamma^{-1}$ and an NFA $\mathcal{A}$ over the alphabet $\Gamma \cup \Gamma^{-1}$.
QUESTION: Does $\rho(\mathbb{A}) \in \rho(L(\mathcal{A}))$ hold?

The proof of Thm. 7 is based on tree automata techniques. Recall that a Munn tree $\mathsf{MT}(u)$ can be viewed as an edge labeled tree. The node ε can be made the root of the tree. Such a rooted edge-labeled tree can be evaluated by a tree automaton. Usually, tree automata work on node labeled trees, but this is only a technicality. The proof of Thm. 7 is based on the following two lemmas.

Lemma 2. *There is a* PSPACE*-transducer, which computes* $\mathsf{MT}(\mathsf{val}(\mathbb{A}))$ *for a given input SLP $\mathbb{A}$.*

Lemma 3. *There is a* PSPACE*-transducer, which computes from a given nondeterministic finite automaton $\mathcal{A}$ over the alphabet $\Gamma \cup \Gamma^{-1}$ and a given SLP $\mathbb{A}$ over the alphabet $\Gamma \cup \Gamma^{-1}$ a nondeterministic tree automaton $\mathcal{B} = \mathcal{B}(\mathcal{A}, \mathbb{A})$ such that: $\rho(\mathsf{val}(\mathbb{A})) \in \rho(L(\mathcal{A}))$ if and only if* $\mathsf{MT}(\mathsf{val}(\mathbb{A}))$ *is accepted by $\mathcal{B}$.*

Proof of Thm. 7. We apply Lemma 1, where $f : (\mathbb{A}, \mathcal{A}) \mapsto (\mathsf{MT}(\mathsf{val}(\mathbb{A})), \mathcal{B}(\mathcal{A}, \mathbb{A}))$ and L is the uniform membership problem for tree automata, i.e., the set of all pairs $(T, \mathcal{B})$, where T is a tree and $\mathcal{B}$ is a tree automaton that accepts T. By [6], L belongs to LOGCFL and hence to POLYLOGSPACE. Moreover, the mapping f can be computed by a PSPACE-transducer by Lemma 2 and 3. □

References

1. P. Berman, M. Karpinski, L. L. Larmore, W. Plandowski, and W. Rytter. On the complexity of pattern matching for highly compressed two-dimensional texts. *J. Comput. Syst. Sci.*, 65(2):332–350, 2002.
2. V. Diekert, M. Lohrey, and A. Miller. Partially commutative inverse monoids. *Semigroup Forum*, 77(2):196–226, 2008.
3. M. V. Lawson. *Inverse Semigroups: The Theory of Partial Symmetries*. World Scientific, 1999.
4. Y. Lifshits. Processing compressed texts: A tractability border. In *Proc. CPM*, LNCS 4580, pages 228–240. Springer, 2007.
5. R. J. Lipton and Y. Zalcstein. Word problems solvable in logspace. *J. Assoc. Comput. Mach.*, 24(3):522–526, 1977.
6. M. Lohrey. On the parallel complexity of tree automata. In *Proc. RTA 2001*, LNCS 2051, pages 201–215. Springer, 2001.
7. M. Lohrey. Word problems and membership problems on compressed words. *SIAM J. Comput.*, 35(5):1210 – 1240, 2006.
8. M. Lohrey. Leaf languages and string compression. *Inf. Comput.*, 209(6):951–965, 2011.
9. M. Lohrey. Compressed word problems for inverse monoids. http://arxiv.org/abs/1106.1000
10. M. Lohrey and N. Ondrusch. Inverse monoids: decidability and complexity of algebraic questions. *Inf. Comput.*, 205(8):1212–1234, 2007.
11. M. Lohrey and S. Schleimer. Efficient computation in groups via compression. In *Proc. CSR 2007*, LNCS 4649, pages 249–258. Springer, 2007.
12. M. Lohrey and B. Steinberg. Tilings and submonoids of metabelian groups. *Theory Comput. Syst.*, 48(2):411–427, 2011.
13. J. Macdonald. Compressed words and automorphisms in fully residually free groups. *Internat. J. Algebra Comput.*, 20(3):343–355, 2010.
14. S. Margolis and J. Meakin. Inverse monoids, trees, and context-free languages. *Trans. Amer. Math. Soc.*, 335(1):259–276, 1993.
15. 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.
16. W. Munn. Free inverse semigroups. *Proc. London Math. Soc.*, 30:385–404, 1974.
17. W. Plandowski. Testing equivalence of morphisms on context-free languages. In *Proc. ESA'94*, LNCS 855, pages 460–470. Springer, 1994.
18. W. Plandowski and W. Rytter. Complexity of language recognition problems for compressed words. In J. Karhumäki, H. A. Maurer, G. Paun, and G. Rozenberg, editors, *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 262–272. Springer, 1999.
19. B. V. Rozenblat. Diophantine theories of free inverse semigroups. *Sib. Math. J.*, 26:860–865, 1985. English translation.
20. S. Schleimer. Polynomial-time word problems. *Comment. Math. Helv.*, 83:741–765, 2008.
21. P. V. Silva. Rational languages and inverse monoid presentations. *Internat. J. Algebra Comput.*, 2:187–207, 1992.
22. J. Stephen. Presentations of inverse monoids. *J. Pure Appl. Algebra*, 63:81–112, 1990.