The idempotent problem for an inverse monoid

N. D. Gilbert and R. Noonan Heale *
School of Mathematical and Computer Sciences
and the Maxwell Institute for the Mathematical Sciences,
Heriot-Watt University, Edinburgh EH14 4AS, U.K.
N.D.Gilbert@hw.ac.uk, rfn1@hw.ac.uk

May 7, 2010

Abstract

We generalize the *word problem* for groups, the formal language of all words in the generators that represent the identity, to inverse monoids. In particular, we introduce the *idempotent problem*, the formal language of all words representing idempotents, and investigate how the properties of an inverse monoid aer related to the formal language properties of its idempotent problem. In particular, we show that if an inverse monoid is either E-unitary or has a finite set of idempotents, then its idempotent problem is regular if and only if the inverse monoid is finite. We also give examples of inverse monoids with context-free idempotent problems, including all Bruck-Reilly extensions of finite groups.

1 Introduction

Significant structural results on finitely generated groups have emerged from the study of the *word problem*, interpreted as the formal language of all words in the generators that represent the identity element of the group. The first such result is the theorem of Anisimov [1] classifying groups with a regular word problem: a finitely generated group has a regular word problem if and only if it is finite. Moving up the Chomsky hierarchy to the class of context-free languages leads us to the Muller-Schupp Theorem [12] (supplemented by a deep result of Dunwoody [3] on the accessibility of finitely presented groups): a finitely generated group has a context-free word problem if and only if it has a free subgroup of finite index. We mention two more recent contributions amongst a number of interesting results in this area. Holt, Rees, Röver, and Thomas [6] study the class of coCF–groups, for which the complement of the word problem is a context-free language. They establish various closure properties of this class, and show that for a coCF–group, the word problem is decidable in cubic time. Elder, Kambites, and Ostheimer [4] study groups whose word problems are accepted by counter automata and show that a group has word problem accepted by a blind n–counter automaton if and only if it is virtually free abelian of rank n. This result can be seen as an abelian analogue of the Muller-Schupp theorem.

*The second author acknowledges the support of a PhD studentship from the Maxwell Institute for the Mathematical Sciences, held whilst some of this work was carried out.

Steps have been taken to generalise some of these interactions between formal language theory and group theory to corresponding interactions with semigroup theory. Kambites [8] associates to each finitely generated semigroup or monoid a formal language called the *loop problem*. For a monoid M generated by X this is the language of closed paths at $1 \in M$ in the automaton obtained from the Cayley graph $\mathrm{Cay}(M, X)$ by adding, for each edge labelled x, an inverse edge labelled by a formal inverse $\overline{x}$. The loop problem of a semigroup S is defined to be the loop problem of the monoid S^1 obtained by adjoining an identity to S. Kambites' detailed study includes an analogue of Anisimov's Theorem – a monoid (or semigroup) has a regular loop problem if and only if it is finite – and subtle results on the loop problems of completely simple semigroups and right cancellative monoids.

For inverse semigroups and inverse monoids, there is a natural alternative approach. The chosen analogue for the word problem is the language of all words that represent idempotents. We call this language the *idempotent problem*. In this paper we choose to work primarily with inverse monoids, but most of our results have natural parallels for inverse semigroups, since the definition of the idempotent problem affords no special status to an identity element. We first look for an analogue to Anisimov's Theorem. A finite inverse monoid will certainly have a regular idempotent problem, and we show that an inverse monoid M with regular idempotent problem will have finite maximum group image. Our main results show that E–unitary (or more generally, strongly E–reflexive) inverse monoids with regular idempotent problem are finite.

We may, of course, choose to afford special status to some idempotent $e \in E(M)$. The *idempotent problem at* e is the language of all words that represent e. If this language is regular, then Green's $\mathcal{R}$–class R_e containing e is finite. We show that if the set $E(M)$ is finite, then for all $e \in E(M)$, the idempotent problem at e is regular; it then follows that M is finite.

For inverse semigroups and monoids, the Schützenberger graphs introduced by Stephen [14] are a powerful tool for studying presentation-theoretic questions such as the word problem in its original guise as a decision problem: given two words on the generators, is there an algorithm to decide if they represent the same element? For an idempotent e of an inverse monoid M, the Schützenberger graph of e is an inverse automaton that accepts the language of all words that represent elements lying above e in the natural partial order on M. We show that this language is regular if and only if Green's $\mathcal{R}$–class R_e containing e is finite.

We conclude the paper by looking at some examples of inverse monoids whose idempotent problems are context-free. The bicyclic monoid has a context-free idempotent problem, as do the polycyclic monoids. Our main results in this direction are that a Bruck-Reilly extension of a finite group and a graph inverse monoid of a finite graph both have context-free idempotent problems.

2 Languages and automata

An *alphabet* X is a set of symbols which we call *letters*. A *word over* X is a finite string of letters from X. We refer to the string with no symbols as the *empty word* and denote it by ϵ. The set of all nonempty, finite words is denoted X^+ and by including the empty word, we obtain $X^* = X^+ \cup \{\epsilon\}$. Under the operation of concatenation, X^+ is the free semigroup over X and X^* is the free monoid over X. A *language* L over X is a subset of X^*.

An *automaton* $\mathcal{A} = (Q, X, \delta, S_0, T)$ consists of a set Q of *states*, an *input alphabet* X, a *transition function* $\delta : Q \times X \to Q$ (δ may only be partially defined on its domain), an *initial state* $S_0 \in Q$, and a collection of *accept states* $T \subseteq Q$. We refer to an element $(S_i, x) \mapsto S_j$ of δ as a *transition*. We say that $\mathcal{A}$ *recognises* a word $w \in X^*$ if there is a sequence of transitions $(S_0, w_1) \mapsto S_1$, $(S_1, w_2) \mapsto S_2$, $\dots$, $(S_{n-1}, w_n) \mapsto S_n$ such that $w = w_1 w_2 \dots w_n$ and $S_n \in T$. The language associated to an automaton $\mathcal{A}$ is the set of all words accepted by $\mathcal{A}$. An automaton is *coaccessible* if from any state there is a sequence of transitions that lead to an accept state. It is *finite* if it has a finite number of states and X is finite. A language is *regular* if and only if it is accepted by a finite automaton.
It is often useful to convey the information about an automaton $\mathcal{A}$ using a directed graph with edges labelled by elements of X. With this interpretation, states correspond to vertices, transitions to labelled edges, and an accepted word w to a path labelled by w from S_0 to an accept state.
A *pushdown automaton* $\mathcal{A} = (Q, X, Y, \delta, S_0, z_0, T)$ consists of a set Q of states, an input alphabet X, a *stack alphabet* Y, a transition function $\delta : Q \times (X \cup \{\epsilon\}) \times (Y \cup \{\epsilon\}) \to Q \times Y^*$ (again, δ may only be partially defined), an initial state $S_0 \in Q$, an *initial stack symbol* $z_0 \in Y$, and a collection of accept states $T \subseteq Q$. The *stack* is a data structure which stores a string of symbols from Y; it is *last in first out*, meaning that only the topmost symbol on the stack can be read at any time. We will use the notation Q_a to mean *popping* the symbol a from the stack; the notation P_a stands for *pushing* the symbol a onto the stack. Before processing a word, the initial symbol z_0 is pushed onto the stack. When giving the transition function δ, rather than stating a transition as $(S_i, x, y) \mapsto (S_j, a)$, we will use the form: $S_i \to S_j$ and $(x,\ Q_y P_a)$. This gives the information that the automaton is in state S_i, reads input letter x, and pops the letter y from the stack; then the automaton moves to state S_j and pushes the symbols $a_1, \dots, a_n$ onto the stack (where $a = a_1 \dots a_n$ and each $a_i \in Y$). When no symbol is pushed onto the stack in a transition, we will usually write $(x,\ Q_y)$ instead of $(x,\ Q_y P_\epsilon)$. We say that $\mathcal{A}$ recognises a word $w \in X^*$ if $\mathcal{A}$ can process a sequence of transitions:

$$\begin{gathered} S_0 \to S_1 \text{ and } (w_1,\ Q_{z_0} P_{a_1}) \\ S_1 \to S_2 \text{ and } (w_2,\ Q_{b_1} P_{b_2}) \\ \vdots \\ S_{n-1} \to S_n \text{ and } (w_n,\ Q_{z_0}) \end{gathered}$$

such that $w = w_1 w_2 \dots w_n$, and $S_n \in T$. A language is *context-free* if and only if it is accepted by a some pushdown automaton. We refer the reader to [7] for more information about languages and automata. Additionally, some of our notation is borrowed from [5], which contains a graph-based description of automata accepting context-free languages.

3 Inverse monoids

We shall consider certain languages over an alphabet that is given as a set of generators for a group or an inverse monoid. Let X be a non-empty set, and let X^{-1} be a disjoint set of formal inverses for X. We set $A = X \cup X^{-1}$.
Let S be an inverse monoid. We say that a map $\alpha : A^* \to S$ is *a choice of generators* for S if the set $X\alpha$ generates S as an inverse monoid. Similarly, if G is a group, a choice of

generators for G is a map $\alpha_G : A^* \to G$ such that $X\alpha_G$ generates G as a group. We say that a choice of generators is *finite* if the set X is finite. When it is clear which map is intended, we will often write $[w]$ instead of $w\alpha$.

Let S be an inverse monoid, and let σ be its *minimum group congruence*. For $s, t \in S$, we have $s\sigma t$ if and only if there exists an idempotent $e \in S$ such that $es = et$. The quotient of S by its minimum group congruence is the *maximum group image* of S, denoted $\widehat{S}$. We denote by σ_S the map from S to $\widehat{S}$ which is induced by the minimum group congruence. If $\alpha : A^* \to S$ is a choice of generators for S, then α induces a (group) choice of generators $\widehat{\alpha}$ for $\widehat{S}$ given by $\widehat{\alpha} = \alpha\sigma$.

A choice of generators $\alpha : A^* \to S$ for an inverse monoid S determines two graphs of particular interest. The *Cayley graph* $\mathrm{Cay}(S, X)$ has vertex set S and edge set $S \times A$ with the edge (s, a) having initial vertex s and terminal vertex $s[a]$. Better adapted to the study of inverse monoids is the *Schützenberger graph* $\mathrm{Sch}(S, X)$, which is a subgraph of $\mathrm{Cay}(S, X)$. Recall that in an inverse monoid, Green's $\mathcal{R}$ relation may be defined by $s\mathcal{R}t$ if and only $ss^{-1} = tt^{-1}$. Then $\mathrm{Sch}(S, X)$ has vertex set S and edge set $\{(s, a) \in S \times A : s\mathcal{R}s[a]\}$. The vertex sets of the connected components of $\mathrm{Sch}(S, X)$ are the $\mathcal{R}$–classes in S. If $w = a_1a_2 \cdots a_m \in A^+$, there is a path $(ww^{-1}, a_1), (s_1, a_2) \cdots (s_{m-1}, a_m)$ from ww^{-1} to w in $\mathrm{Sch}(S, X)$. Stephen's approach [14] to the study of inverse monoid presentations regards each component of the Schützenberger graph as an automaton with input alphabet A: the component containing $s \in S$ has start state ss^{-1} and the single accept state s. This component forms the *Schützenberger automaton* $\mathrm{Sch}(S, X, s)$ and accepts the language

$$\{w \in A^* : ss^{-1}[w] = s\}.$$

Let $E(S)$ be the set of idempotents of S. Recall that S is *E–unitary* if $e, es \in E(S)$ together imply that $s \in E(S)$. We note the following result.

Lemma 3.1. *Let S be an inverse monoid. Then the following conditions are equivalent:*

(a) S is E-unitary;

(b) there is a morphism $\phi : S \to G$ onto a group G such that $E(S) = 1\phi^{-1}$;

(c) if $\sigma_S : S \to \widehat{S}$ is the quotient map induced by the minimum group congruence on S, then $1\sigma_S^{-1} = E(S)$.

Following O'Carroll [13], an inverse monoid S is said to be *strongly E–reflexive* if, given $e \in E(S)$ and $s, t \in S$, then $est \in E(S)$ implies that $ets \in E(S)$. Every E–unitary inverse monoid is strongly E–reflexive, and O'Carroll shows in [13] that S is strongly E–reflexive if and only if S is isomorphic to a semilattice (with identity) of E–unitary inverse semigroups.

4 The idempotent problem and some variations

Consider a group G with choice of generators $\alpha_G : A^* \to G$. The *word problem* of G, denoted $W(G, X)$, is the language over A consisting of all words that represent the identity of G:

$$W(G, X) = \{w \in A^* : w\alpha_G = 1_G\}.$$

Given an inverse monoid S and a choice of generators $\alpha : A^* \to S$, we can define various languages over A that are analogues to the word problem. The most obvious analogue is the *idempotent problem*:

$$\mathcal{E}(S,X) = \{w \in A^* : [w]^2 = [w]\}.$$

It is also natural to consider languages related to specific idempotents. If $e \in E(S)$, we define

$$\mathcal{E}_e(S,X) = \{w \in A^* : [w] = e\},$$

and

$$\mathcal{W}_e^{\uparrow}(S,X) = \{w \in A^* : e[w] = e\}.$$

We note that $\mathcal{W}_e^{\uparrow}$ is precisely the language accepted by the Schützenberger automaton $\mathrm{Sch}(S,X,e)$.
If S is a group, then $\mathcal{E}, \mathcal{E}_e$ and $\mathcal{W}_e^{\uparrow}$ coincide and are equal to the word problem $W(S,X)$. As for the word problem, under mild hypotheses on the languages concerned, the type of language obtained, for example whether it is regular or context-free, does not depend on the choice of generators.

Proposition 4.1. *Let $\mathcal{C}$ be a class of languages closed under inverse homomorphism. Let S be an inverse monoid and let $\alpha : A^* \to S$ be a finite choice of generators for S. Further let $e \in E(S)$. If one of the languages $\mathcal{E}(S,X)$, $\mathcal{E}_e(S,X)$, or $\mathcal{W}_e^{\uparrow}(S,X)$ is in $\mathcal{C}$, then so is the corresponding language determined by any other finite choice of generators for S.*

Proof. The proof is a standard argument, and we give it in detail for the language $\mathcal{W}_e^{\uparrow}(S,X)$. Let $B = Y \cup Y^{-1}$ and let $\beta : B^* \to S$ be a second choice of generators. There is an induced homomorphism $\phi : A^* \to B^*$ such that $\alpha = \phi\beta$. Now

$$\begin{aligned} w \in \mathcal{W}_e^{\uparrow}(S,X) &\iff e(w\alpha) = e \\ &\iff e(w\phi\beta) = e \\ &\iff w\phi \in \mathcal{W}_e^{\uparrow}(S,Y). \end{aligned}$$

Hence $\mathcal{W}_e^{\uparrow}(S,Y) = \mathcal{W}_e^{\uparrow}(S,X)\phi^{-1}$. □

5 Regular idempotent problem

Our aim in this section is to understand the implications of regularity of the languages $\mathcal{E}$, $\mathcal{E}_e$, and $\mathcal{W}_e^{\uparrow}$ introduced in section 4. We first make some elementary observations about $\mathcal{E}$.

Proposition 5.1.

(a) If S is a finite inverse semigroup then $\mathcal{E}$ is regular;

(b) $\mathcal{E} \subseteq W(\widehat{S},X)$;

(c) $\mathcal{E} = W(\widehat{S},X)$ if and only if S is E–unitary;

(d) If $\mathcal{E}$ is a regular language then the maximum group image $\widehat{S}$ of S is finite.

Proof. (a) Consider the Cayley graph $\text{Cay}(S, X)$ as a finite automaton with start state $1 \in S$ and accept states $E(S)$.
(b) This is obvious.
(c) Suppose that S is E–unitary and that $w \in W(\widehat{S}, X)$. Then in S we have $[w]\sigma = 1$ and so by Lemma 3.1, $[w] \in E(S)$ and hence $w \in \mathcal{E}(S, X)$.
(d) Suppose that $\mathcal{E}$ is regular and suppose that $\mathcal{A}$ is a finite automaton that accepts $\mathcal{E}$. By taking the coaccessible part of $\mathcal{A}$ we may assume, without loss of generality, that for each state S_i of $\mathcal{A}$ there is a word $u_i \in A^*$ such that, on reading the word u_i in $\mathcal{A}$ from the state S_i, we arrive at an accept state.
Let $w \in A^*$. Since $\mathcal{A}$ accepts the word $ww^{-1} \in \mathcal{E}$, the prefix w may be read by $\mathcal{A}$. Suppose we read w and arrive at the state S_i. Then $wu_i \in \mathcal{E}$ and hence $([w][u_i])\sigma = 1 \in \widehat{S}$. Therefore $[w]\sigma = ([u_i]\sigma)^{-1}$, and since there are only finitely many distinct words u_i, we conclude that $\widehat{S}$ is finite.□

We pose the converse to part (a) of Proposition 5.1 as a question:

Question 5.2. *If S is an inverse monoid such that $\mathcal{E}(S, X)$ is regular, then is S finite?*

The answer to this question is yes for E–unitary inverse monoids. Our argument for this relies on the following lemma which may be of independent interest.

Lemma 5.3. *Let S be a finitely generated E–unitary inverse monoid whose maximum group image $\widehat{S}$ is finite. Then S is finite.*

Proof. Given a group G and a choice of generators $\alpha : A^* \to G$, Margolis and Meakin [9] construct the graph expansion $\mathcal{M}(G, X)$ from the Cayley graph $\text{Cay}(G, X)$ as follows. The elements of $\mathcal{M}(G, X)$ are pairs (P, g) where P is a finite connected subgraph of $\text{Cay}(G, X)$ containing the vertex 1_G and where $g \in G$ is also a vertex of P. Under the multiplication $(P, g)(Q, h) = (P \cup gQ, gh)$, the set $\mathcal{M}(G, X)$ becomes an E–unitary inverse monoid. Furthermore Margolis and Meakin prove [9, Theorem 2.2] that $\mathcal{M}(G, X)$ is an initial object in the category of E–unitary, X–generated inverse semigroups with maximum group image G. Moreover, if G is finite, then clearly so is $\mathcal{M}(G, X)$.
Now consider the E–unitary inverse monoid S, generated as an inverse monoid by the finite set X. Then S is an image of the graph expansion $\mathcal{M}(\widehat{S}, X)$ which is a finite inverse monoid, and so S is finite. □

Now part (a) of Proposition 5.1, and part (d) together with Lemma 5.3, combine to prove our main result on the language $\mathcal{E}$.

Theorem 5.4. *If S is an E–unitary inverse monoid with a finite choice of generators $\alpha : A^* \to S$, then the idempotent problem $\mathcal{E}(S, X)$ is regular if and only if S is finite.*

We are able to extend Theorem 5.4 to the class of strongly E-reflexive inverse monoids.

Theorem 5.5. *If S is a strongly E–reflexive inverse monoid with a finite choice of generators $\alpha : A^* \to S$, then the idempotent problem $\mathcal{E}(S, X)$ is regular if and only if S is finite.*

Proof. If S is finite, then $\mathcal{E}(S,X)$ is regular by part (a) of Proposition 5.1.
Now suppose that S is strongly E-reflexive. Then by [13, Theorem 1] S is isomorphic to a semilattice of E-unitary semigroups $\bigcup_{e\in\Lambda} S_e$ over a semilattice with identity Λ. Thus S comes equipped with structure maps $\psi_{e,f} : S_e \to S_f$ for each pair e, f of elements of Λ with $e \geqslant f$. These maps have the property that for all $e \in \Lambda$, $\psi_{e,e}$ is the identity, for each triple e, f, g of elements of Λ with $e \geqslant f \geqslant g$ we have $\psi_{e,f}\psi_{f,g} = \psi_{e,g}$, and if $s, t \in S$ with $s \in S_e$ and $t \in S_f$ then $st = (s\psi_{e,ef})(t\psi_{e,ef}) \in S_{ef}$. Since S is finitely generated, the semilattice Λ is finite. We wish to show that S_e is finite.
Let $\alpha : A^* \to S$ be the choice of generators. If $x \in X$ and $[x] \in S_e$ then we write $|x| = e$. Then for each $e \in \Lambda$ we let X_e be the following subset of X:

$$X_e = \{x \in X : |x| \geqslant e\}.$$

Let $A_e = X_e \cup X_e^{-1}$. Then if $x \in A_e$, the map $x \mapsto x\alpha\psi_{|x|,e}$ induces a choice of generators $A_e^* \to S_e$.
Now suppose that $\mathcal{E}(S,X)$ is regular. For each $e \in \Lambda$ we have $\mathcal{E}(S_e, X_e) = \mathcal{E}(S,X) \cap A_e^*$. Clearly A_e^* is a regular language and thus $\mathcal{E}(S_e, X_e)$ is an intersection of regular languages and so is regular. Since S_e is E–unitary, Theorem 5.4 implies that S_e is finite. Therefore S is a finite semilattice of finite inverse monoids and we conclude that S is finite. □

5.1 Regularity of the related languages

Considering the Schützenberger graph $\mathrm{Sch}(S,X)$ and its components as Schützenberger automata following [14], the languages $\mathcal{W}_e^{\uparrow}$ are the natural objects of study.

Proposition 5.6. *Let S be an inverse semigroup with finite choice of generators $\alpha : A^* \to S$ and let $e \in E(S)$. Then $\mathcal{W}_e^{\uparrow}(S,X)$ is regular if and only if the $\mathcal{R}$–class R_e is finite.*

Proof. If the $\mathcal{R}$–class R_e is finite, the Schützenberger automaton $\mathrm{Sch}(S,X,e)$ is a finite automaton that accepts the language $\mathcal{W}_e^{\uparrow}$.
Conversely, suppose that $\mathcal{W}_e^{\uparrow}$ is regular, and let $\mathcal{A}$ be a finite automaton whose language is $\mathcal{W}_e^{\uparrow}$. Arguing as in the proof of part (d) of Proposition 5.1, we may assume that $\mathcal{A}$ is coaccessible, and for each state S_i of $\mathcal{A}$ we choose a word $u_i \in A^*$ such that upon reading u_i from S_i we reach a success state. Now let $w \in A^*$ such that $[w] \in R_e$. Since $[ww^{-1}] = e$, the word w is a prefix of a word accepted by some computation in $\mathcal{A}$, and we let S_p be the state reached by this computation after reading w. Then wu_p is accepted by $\mathcal{A}$ and therefore $[wu_p] \geqslant e$ in S. Since $[w] \in R_e$, we have $[w][w]^{-1} = e$, which implies that $e[w] = [w][w]^{-1}[w] = [w]$. Then

$$[w][u_p]e[w] = [wu_p]e[w] = e[w] = [w]$$

and

$$[u_p]e[w][u_p]e = [u_p][wu_p]e = [u_p]e.$$

Therefore $[w]^{-1} = [u_p]e$. Since there are only finitely many elements $[u_p]$, there are only finitely many values of $[w]$ and so R_e is finite. □

A variation on the proof just given, using an automaton accepting the language $\mathcal{E}_e$, proves:

Proposition 5.7. *If the language $\mathcal{E}_e$ is regular then the $\mathcal{R}$–class R_e is finite. Therefore, we have*

$$\mathcal{E}_e \textit{ regular} \quad \Longrightarrow \quad R_e \textit{ finite} \quad \Longleftrightarrow \quad \mathcal{W}_e^{\uparrow} \textit{ regular.}$$

The implication

$$\mathcal{E}_e \text{ regular} \quad \Longrightarrow \quad \mathcal{W}_e^{\uparrow} \text{ regular}$$

can be seen more directly, because it an example of a standard construction for regular languages. We have

$$w \in \mathcal{W}_e^{\uparrow} \quad \Longleftrightarrow \quad \text{there exists } u \in \mathcal{E}_e \text{ such that } uw \in \mathcal{E}_e.$$

It follows easily that if $\mathcal{E}_e$ is regular, then so is $\mathcal{W}_e^{\uparrow}$.
Theorems 5.4 and 5.5 give one sort of generalisation of the Anisimov Theorem to inverse semigroups, using on a condition on the idempotents that is satisfied by any group. Another direction for such a generalisation is to assume that the set of idempotents is finite.

Theorem 5.8. *If S is an inverse monoid with a finite choice of generators $\alpha : A^* \to S$ and with $E(S)$ finite, then the idempotent problem $\mathcal{E}(S, X)$ is regular if and only if S is finite.*

Proof. If S is finite, then $\mathcal{E}(S, X)$ is regular by part (a) of Proposition 5.1.
Now suppose that $E(S)$ is finite. For each $e \in E(S)$, let L_e be Green's $\mathcal{L}$–class containing e. We construct a finite automaton $\mathcal{F}_e$ that recognises the language

$$F_e = \{w \in A^* : [w] \in L_e\}$$

as follows. The state set of $\mathcal{F}_e$ is the set of idempotents $E(S)$ with start state $1 \in E(S)$. For $a \in A$ the transition on reading a at state f is $\delta(f, a) = [a]^{-1} f [a]$. The unique accept state is $e \in E(S)$. On reading a word $w \in A^*$ from the start state, we reach the state $[w]^{-1}[w]$ and which is the state e if and only if $[w] \in L_e$.
Recall that $\mathcal{E}_e = \{w \in A^* : [w] = e\}$. Then $\mathcal{E}_e = F_e \cap \mathcal{E}(S, X)$ and so, if $E(S)$ is finite and $\mathcal{E}(S, X)$ is regular, $\mathcal{E}_e$ is an intersection of regular languages and so is regular. By Proposition 5.7 the $\mathcal{R}$–class R_e is finite, and so S is the union of finitely many finite $\mathcal{R}$–classes and is finite. □

5.2 Examples

Example 5.9. Let G be a group and let $S = G^0$ be obtained by adjoining a zero to G. If Y generates G as a group, we form the generating set $X = Y \cup \{z\}$ with $z \notin Y$ and $z \mapsto 0$. Then

$$\mathcal{E} = W(G, Y) \cup A^* z A^* \cup A^* z^{-1} A^*.$$

Clearly if $W(G, Y)$ is regular, then $\mathcal{E}$ is as well. Further, as $W(G, Y)$ is the intersection of $\mathcal{E}$ with the set of words that contain neither z nor z^{-1}, if $\mathcal{E}$ is regular, then so is $W(G, Y)$. We conclude that $\mathcal{E}$ is regular if and only if $W(G, Y)$ is regular, if and only if G is finite. Of course $\widehat{S}$ is trivial, and so finiteness (or even triviality) of $\widehat{S}$ does not imply regularity of $\mathcal{E}$. We note further that

$$\mathcal{E}_1 = \mathcal{W}_1^{\uparrow} = W(G, Y),$$

$$\mathcal{E}_0 = A^* z A^* \cup A^* z^{-1} A^*, \quad \mathcal{W}_0^{\uparrow} = A^*.$$

Example 5.10. If $S = FIM(X)$, the free inverse monoid on the set X, then S is E–unitary and $\widehat{S}$ is a free group: hence the idempotent problem coincides with the word problem of the free group $\widehat{S}$ and so is context-free (and not regular). However, the $\mathcal{R}$–classes in S are all finite, and so each language $\mathcal{W}_e^{\uparrow}$ is regular.

We shall explore further examples of inverse semigroups with context-free idempotent problems in the next section.

6 Examples of context-free idempotent problems

We are motivated by Muller and Schupp's famous theorem [12], which characterises a finitely generated group with context-free word problems as being a free–by–finite group. One of their insights is that the structure of such a group is closely related to the structure of the pushdown automaton recognising its word problem: for any word w in the generators, the stack keeps tracks of which element of the free subgroup corresponds to $[w]$ and the graph part remembers which coset $[w]$ is in. In this section, we wish to explore similar structural relationships between inverse monoids and pushdown automata that recognise their respective idempotent problems. The intent of these examples is to shed light on the following question.

Question 6.1. *Can one characterize the inverse semigroups whose idempotent problems are context-free languages?*

Example 6.2. We return to $S = FIM(X)$. As mentioned in Example 5.10, S is E–unitary and thus $\mathcal{E}(S, A) = W(\widehat{S}, A)$. The group $\widehat{S}$ is a free group, and so $\mathcal{E}(S, A)$ is context-free. We give the classic example of a pushdown automaton recognising $\mathcal{E}(S, A)$, since we refer to it in other examples. Let $\mathcal{A}(FIM(X))$ consists of one state S_0 (necessarily the initial as well as the terminal state), use $A = X \cup X^{-1}$ for the input alphabet, and $A \cup \{z_0\}$ for the stack. To process an input symbol x, we examine the stack: if the top stack symbol is not x^{-1}, we push x onto the stack; if the top stack symbol is x^{-1}, we pop x^{-1} from the stack.

Example 6.3. Consider the bicyclic monoid B, which is presented as an inverse monoid by $\langle x : xx^{-1} \rangle$. Then B is E-unitary and therefore $\mathcal{E}(B, \{x\}) = W(\widehat{B}, \{x\})$. Since $\widehat{B}$ is the free group on one generator, $\mathcal{E}(B, \{x\})$ is a context-free language.

Example 6.4. For $n > 1$, the polycyclic monoid P_n [11] is the inverse monoid with zero presented by
$$\langle x_1, \dots, x_n : x_i x_i^{-1} = 1\,, x_i x_j^{-1} = 0\,(i \neq j) \rangle.$$
Let $X_n = \{x_1, \dots, x_n\}$ and $A = X_n \cup X_n^{-1}$. The idempotent problem of P_n is closely related to the idempotent problem of the free inverse monoid $FIM(X_n)$:
$$\mathcal{E}(P_n, X_n) = \mathcal{E}(FIM(X_n)) \cup \left(\bigcup_{i \neq j} A^* x_i x_j^{-1} A^* \right) .$$
Since context-free languages are closed under union, it follows that $\mathcal{E}(P_n, X_n)$ is context-free. We can construct a pushdown automaton $\mathcal{A}(P_n)$ recognising $\mathcal{E}(P_n, X_n)$ by adding a state Z that accepts words representing zero to the automaton $\mathcal{A}(FIM(X_n))$ from Example

6.2. The transitions are modified as follows: if $\mathcal{A}(P_n)$ is in a state other than Z, the top stack symbol is x_i, and the next input symbol is x_j^{-1} with $i \neq j$, then the automaton moves to the Z. For all other cases, the transition is the same as for $\mathcal{A}(FIM(X_n))$. If $\mathcal{A}(P_n)$ is in the state Z, then the automaton finishes reading the letters of the input word, then pops all symbols from the stack, and accepts the word.

Example 6.5. Margolis and Meakin [10] study the class $\mathfrak{M}$ of inverse monoids given by presentations of the form $\langle X : e_i = f_i \, (i \in I)\rangle$ where e_i, f_i are idempotents in $FIM(X)$. Free monoids, free groups, and the bicyclic monoid are all examples of inverse monoids in this class. Amongst the connections between the class $\mathfrak{M}$ and formal language theory established in [10], it is shown (using our current notation) that for a finitely presented inverse monoid M in the class $\mathfrak{M}$:

- [10, Theorem 4.4] Each language $\mathcal{W}_e^{\uparrow}$ is context-free. (In fact it is proved that $\mathcal{W}_s^{\uparrow}$ is deterministic context-free for any $s \in M$.)
- [10, Corollary 4.6] The idempotent problem at $1 \in M$ – our $\mathcal{E}_1$ – is deterministic context-free.

6.1 Bruck-Reilly extensions of groups

Bruck-Reilly extensions of groups generalise the bicyclic monoid, and in this section we shall show that all Bruck-Reilly extensions of finite groups have context-free idempotent problems.

Let G be a group and θ be an endomorphism of G. The Bruck-Reilly extension $\mathrm{BR}(G, \theta)$ of G with respect to θ is the set $\mathbb{N}^0 \times G \times \mathbb{N}^0$ with the binary operation:

$$(m, a, n)(p, b, q) = (m - n + s, (a\theta^{s-n})(b\theta^{s-p}), q - p + s),$$

where $s = \max(n, p)$. The idempotents of $\mathrm{BR}(G, \theta)$ are the elements of the form $(n, 1_G, n)$ where $n \in \mathbb{N}^0$.

Let G be a finite group, and consider a Bruck-Reilly extension $B = \mathrm{BR}(G, \theta)$. Let X be a finite set, $A = X \cup X^{-1}$, and suppose that $\alpha_G : A^* \to G$ is a (group) choice of generators for G. We introduce additional generators t and t^{-1}, set $\overline{A} = A \cup \{t, t^{-1}\}$, and then extend α_G to an inverse monoid choice of generators, $\alpha_B : \overline{A}^* \to B$ for B by setting $a\alpha_B = (0, [a]_G, 0)$ for $a \in A$ and $t\alpha_B = (0, 1_G, 1)$.

Theorem 6.6. *Let G be a finite group, θ an endomorphism on G, and $\alpha_G : A^* \to G$ a finite choice of generators as described above. Let B be the Bruck-Reilly extension $\mathrm{BR}(G, \theta)$. Then the idempotent problem $\mathcal{E}(B, X \cup \{t\})$ is context-free.*

Proof. We note the effects of right multiplication by $[t]_B = (0, 1_G, 1)$, $[t^{-1}]_B = (1, 1_G, 0)$ and $(0, g, 0)$ with $g \in G$ on an arbitrary element $(m, a, n) \in B$ in Table 1. We further note that, since the map θ is in the finite semigroup of endomorphisms of G, θ has finite index and period. We denote these by I and K respectively. Thus for all $g \in G$, I and K satisfy the equation $g\theta^I = g\theta^{I+K}$.

We construct a pushdown automaton $\mathcal{A}(B)$ that recognizes $\mathcal{E}(B, X \cup \{t\})$. The automaton $\mathcal{A}(B)$ has state set $\{S_{(g,n)} : g \in G,\ n \in \{0, 1, 2, \ldots, I + K - 1\}\}$; the state $S_{(1_G,0)}$ is both the

Product		Observation
$(m,a,n)(0,1_G,1) \quad = \quad (m,a,n+1)$		Right multiplication by $[t]_B$ increases the third entry by 1.
$(m,a,n)(1,1_G,0) \quad =$	$(m+1,a\theta,0) \quad n=0$	If the third entry of the left factor is zero, right multiplication by $[t^{-1}]_B$ increases the first entry by 1.
	$(m,a,n-1) \quad \text{if } n>0$	If the third entry of the left factor is positive, right multiplication by $[t^{-1}]_B$ decreases the third entry by 1.
$(m,a,n)(0,g,0) \quad = \quad (m,a(g\theta^n),n)$		In order to right multiply by $(0,g,0)$ we need to know the third entry of the left factor.

Table 1: The effects of right multiplication by specific elements in Bruck-Reilly Extensions.

Transitions with input symbol ϵ or x

$S_{(1_G,n)} \to S_{(1_G,n)}$	$S_{(f,n)} \to S_{(g,n)}$ where $g = f([x]\theta^n)$
$(\epsilon,\ Q_{z_0})$	$(x,\ Q_aP_a)$ for $a = z_0, t, u, t^{-1}$,

Transitions with input symbol t

$S_{(f,0)} \to S_{(f,1)}$	$S_{(f,n)} \to S_{(f,n+1)}$ where $n \neq 0$, $n \neq I+K-1$	$S_{(f,I+K-1)} \to S_{(f,I)}$
$(t,\ Q_{z_0}P_{z_0t})$ $(t,\ Q_{t^{-1}})$	$(t,\ Q_aP_aP_a)$ for $a = t, u$	$(t,\ Q_aP_{au})$ for $a = t, u$

Transitions with input symbol t^{-1}

$S_{(f,0)} \to S_{(f\theta,0)}$	$S_{(f,n)} \to S_{(f,n-1)}$ where $n \neq 0$	$S_{(f,I)} \to S_{(f,I+K-1)}$
$(t^{-1},\ Q_aP_{at^{-1}})$ where $a = z_0, t^{-1}$	$(t^{-1},\ Q_t)$ $(t^{-1},\ Q_u)$ if $I+1 \leqslant n$	$(t^{-1},\ Q_u)$

Table 2: Transitions for the automaton recognising $\mathcal{E}(B\ ,\ X \cup \{t\})$. In the table, $n \in \{0,1,2,\dots,I+K-1\}$ unless otherwise specified, $f, g \in G$, and $x \in X \cup X^{-1}$.

start and accept state. The input alphabet is $\overline{A}$ and the stack alphabet consists of t, u, t^{-1}, and the initial symbol z_0. The transition function is given in Table 2.
The pushdown automaton $\mathcal{A}(B)$ uses the stack to monitor how many more ts there are than t^{-1}s in a word, or vice versa. The stack will either be empty, contain only t^{-1}s, contain only ts, or contain ts followed by us. Both t and u correspond to the input symbol t; the stack symbol t^{-1} corresponds to the input symbol t^{-1}.
The graph part of the automaton has $I+K-1$ copies of the vertex set G, which we refer to as $G_0, G_1, \dots, G_{I+K-1}$. The state $S_{(f,n)}$ corresponds to the element $f \in G$ in the copy G_n. The edges within each G_n are generated using $X \cup X^{-1}$ in a manner quite similar to edges in a Cayley graph. Namely there is an edge from $S_{(f,n)}$ to $S_{(g,n)}$ if there is some $x \in X \cup X^{-1}$ such that $f = g([x]_G\theta^n)$. In G_0, there are also edges related to the input symbol t^{-1}: here, there is an edge from $S_{(f,n)}$ to $S_{(g,n)}$ if $g = f\theta$. The input symbols t and t^{-1} produce edges between corresponding elements in consecutive copies of the G_n: upon

reading the input t we travel from G_n to G_{n+1} (assuming $n < I + K - 1$); upon reading the input t^{-1} we usually travel from G_n to G_{n-1} (assuming $n > 0$). The exception occurs when we are at a state in G_I, we read x^{-1}, and u is on the top of the stack. We describe this case in the end of the next paragraph.

If $w \in \overline{A}^*$ is such that $[w]_B = (m, a, n)$, where $n \leqslant I + K - 1$, upon computing w the automaton $\mathcal{A}(B)$ is in the state $S_{(a,n)}$. However we also need to be able to compute words corresponding to elements with $n > I + K - 1$. To do so, we make use of the periodicity of θ. If we are at a state in G_{I+K-1}, read input t, and see either t or u on the stack, we move to the corresponding state in G_I and add a u rather than a t to the stack. Similarly, when we are at a state in G_I, read input t^{-1}, and see a u on the stack, we move to G_{I+K-1}. If instead we see a t on the stack, we move to G_{I-1}. We accept a word w if after computing it we are in the state $S_{(1_G,0)}$ and the stack is empty. □

6.2 Graph inverse monoids

Graph inverse monoids were introduced by Ash and Hall in [2]. The polycyclic monoids are examples of graph inverse monoids, arising from bouquets of circles. In this section we show that all graph inverse monoids have context-free idempotent problems.

Let Γ be a finite directed graph with vertex set $V(\Gamma)$ and edge set $E(\Gamma)$. If $e \in E(\Gamma)$, then $\iota(e)$ stands for the *initial vertex* of e and $\tau(e)$ for the *terminal vertex*. A *path* p in Γ is either a single vertex v of Γ – the *empty path* at v, or is a sequence of consecutive edges $(e_1, e_2, \ldots, e_n)$ where $\tau(e_i) = \iota(e_{i+1})$ for $1 \leqslant i \leqslant n - 1$. We extend the maps ι and τ to paths, so that $\iota(p)$ and $\tau(p)$ are the initial and terminal vertices of a path p. The *graph inverse monoid* GIM_Γ is the set

$$\{(p, q) : p \text{ and } q \text{ are paths in } \Gamma \text{ such that } \tau(p) = \tau(q)\} \cup \{0, 1\}$$

of pairs of coterminal paths in Γ, together with 0 and 1, with the binary operation:

$$(p,q)(r,s) = \begin{cases} (pr_2, s) & \text{if there exists } r_2 \text{ such that } qr_2 = r \\ (p, sq_2) & \text{if there exists } q_2 \text{ such that } q = rq_2 \\ 0 & \text{otherwise.} \end{cases}$$

We note that if $(p, q)(r, s) \neq 0$ then one of q, r is a prefix of the other, and in particular that $\iota(q) = \iota(r)$. Further, an element (p, q) has inverse (q, p) and is idempotent if and only if $p = q$.

We obtain a choice of generators for $\mathrm{GIM}(\Gamma)$ from Γ as follows. We set $X = V(\Gamma) \cup E(\Gamma)$ and map $X \to \mathrm{GIM}(\Gamma)$ by $v \mapsto (v, v)$ for all $v \in V(\Gamma)$ and $e \mapsto (e, \tau(e))$. Setting $A = X \cup X^{-1}$ and extending the map $X \to \mathrm{GIM}(\Gamma)$ in the obvious way to $\alpha : A^* \to \mathrm{GIM}(\Gamma)$ gives a finite choice of generators.

Theorem 6.7. *Let Γ be a finite, directed graph. Then the idempotent problem $\mathcal{E}(\mathrm{GIM}_\Gamma, X)$ is context-free.*

Proof. We outline the construction of a pushdown automaton $\mathcal{A}(\mathrm{GIM}_\Gamma)$ that accepts the language $\mathcal{E}(\mathrm{GIM}_\Gamma, X)$. The state set of $\mathcal{A}(\mathrm{GIM}_\Gamma)$ is $\{S_0, Z\} \cup \{S_v : v \in V(\Gamma)\}$; its start state is S_0 and every state is an accept state. The input alphabet is A and the stack alphabet is $E(\Gamma) \cup E(\Gamma)^{-1}$, together with the initial stack symbol z_0. The automaton is designed to act

as follows. The start state S_0 accepts the empty word and the state Z accepts all words w with $[w] = 0$. The state S_v accepts all words w such that $[w] = (p, p)$ and $\iota(p) = v$. Having read a word w with $[w] = (p, q)$, the automaton $\mathcal{A}(\mathrm{GIM}_\Gamma)$ is at state $\iota(q)$. The symbols on the stack depend upon the values of p and q. If $p = q \in V(\Gamma)$, then the stack only contains the initial stack symbol z_0. If $p \in V(\Gamma)$ and q is a non-empty path, then the stack contains z_0 and above this q^{-1}. If p is a non-empty path and $q \in V(\Gamma)$, then the stack contains z_0 and p. Finally if both p and q are both non-empty, then the stack contains z_0 and the freely reduced form of the word pq^{-1}. If the only symbol on the stack after processing w is the symbol z_0, then z_0 is popped from the stack and $\mathcal{A}(\mathrm{GIM}_\Gamma)$ accepts the word w.
The transitions that accomplish this are defined as follows. As in Example 6.4, the state Z is a sink state: if $\mathcal{A}(\mathrm{GIM}_\Gamma)$ enters state Z, the automaton finishes reading any input and pops all symbols from the stack. Suppose the automaton is in the start state S_0: upon reading $v^{\pm 1}$ with $v \in V(\Gamma)$, the state changes to S_v and we push z_0 on the stack; upon reading $e \in E(\Gamma)$, the state changes to $S_{\tau(e)}$ and we push $z_0 e$ onto the stack; upon reading e^{-1}, the state changes to $S_{\iota(e)}$ and we push $z_0 e^{-1}$ onto the stack.
Suppose $\mathcal{A}(\mathrm{GIM}_\Gamma)$ is in state S_v with $v \in V(\Gamma)$. Upon reading $u \in V(\Gamma) \cup V(\Gamma)^{-1}$, if $v = u^{\pm 1}$, then the state and stack are unchanged; if $v \neq u^{\pm 1}$, then the state changes to Z. Now suppose that $\mathcal{A}(\mathrm{GIM}_\Gamma)$ is in state S_v and reads $e \in E(\Gamma)$. If $v \neq \iota(e)$, then the state changes to Z. If $v = \iota(e)$ and the stack contains $z_0 p$ for a path p, then the state changes to $\tau(e)$ and e is pushed onto the stack. If $v = \iota(e)$ and the stack contains $z_0 pq^{-1}$ with q non-empty, then we check whether the top symbol of the stack is e^{-1}. If it is not, the state changes to Z. If it is, the state changes to $\tau(e)$ and we pop e^{-1} from the stack. Similarly, we consider when $\mathcal{A}(\mathrm{GIM}_\Gamma)$ is in state S_v and reads e^{-1} with $e \in E(\Gamma)$. If $v \neq \tau(e)$, the state changes to Z. If $v = \tau(e)$ and the stack contains $z_0 pq^{-1}$ with q non-empty, then the state changes to $\iota(e)$ and e^{-1} is pushed onto the stack. If $v = \tau(e)$ and the stack contains $z_0 p$, then the state changes to $\iota(e)$. We check whether the top symbol of the stack is e: if it is we pop it and if it is not, then we push e^{-1} onto the stack. □

References

[1] A.V. Anisimov, Group languages. *Kibernetika* 4 (1971), 18-24.

[2] C.J. Ash and T.E. Hall, Inverse semigroups on graphs. *Semigroup Forum* 11 (1975), 140-145.

[3] M.J. Dunwoody, The accessibility of finitely presented groups. *Invent. Math.* 81 (1985), 449–457.

[4] M. Elder, M. Kambites and G. Ostheimer, On groups and counter automata. *Internat. J. Algebra Comput.* 18 (2008), 1345-1364.

[5] R. Gilman, Formal languages and infinite groups, *Geometric and computational perspectives on infinite groups, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 25, 1996.

[6] D.F. Holt, S.E. Rees, C.E. Röver and R.M. Thomas, Groups with context-free co-word problem. *J. London Math. Soc. (2)* 71 (2005), 643–657

[7] J.E. Hopcroft and J.D. Ullman, *Formal languages and their relation to automata*, Addison-Wesley, 1969.

[8] M. Kambites, The loop problem for monoids and semigroups. *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 143, 271-289, 2007.

[9] S.W. Margolis and J.C. Meakin, E–unitary inverse monoids and the Cayley graph of a group presentation. *J. Pure Appl. Algebra* 58 (1989), 45-76.

[10] S.W. Margolis and J.C. Meakin, Inverse monoids, trees, and context-free languages. *Trans. Amer. Math. Soc.* 335 (1993), 259-276.

[11] M. Nivat and J-F. Perrot, Une généralization du monoïde bicyclique. *C.R. Acad. Sc.* Ser. A271 (1970), 824-827.

[12] D. Muller and P.E. Schupp, Groups, the theory of ends, pushdown automata, and second-order logic. *J. Computer and System Sciences* 26 (1983), 295-310.

[13] L. O'Carroll, Strongly E–reflexive inverse monoids. *Proc. Edinburgh Math. Soc.* 20 (1976-77), 339-354.

[14] J.B. Stephen, Presentations of inverse monoids. *J. Pure Appl. Algebra* 63 (1990), 81-112.