Context-free word problem semigroups

Tara Brough[1][0000−0002−3576−0670], Alan J. Cain[1][0000−0002−0706−1354], and
Markus Pfeiffer[2][0000−0002−9881−4429]

[1] Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa, 2829–516 Caparica, Portugal
{t.brough,a.cain}@fct.unl.pt
[2] School of Computer Science, University of St Andrews, North Haugh, St Andrews,
Fife KY16 9SS, United Kingdom markus.pfeiffer@st-andrews.ac.uk

Abstract. This paper studies the classes of semigoups and monoids with
context-free and deterministic context-free word problem. First, some ex-
amples are exhibited to clarify the relationship between these classes and
their connection with the notions of word-hyperbolicity and automatic-
ity. Second, a study is made of whether these classes are closed under
applying certain semigroup constructions, including direct products and
free products, or under regressing from the results of such constructions
to the original semigroup(s) or monoid(s).

1   Introduction

The deep connections between formal language theory and group theory are per-
haps most clearly evidenced by the famous 1985 theorem of Muller and Schupp,
which says that a group has context-free word problem if and only if it is vir-
tually free [22,9]; indeed, virtually free groups have *deterministic* context-free
word problem. Since then, many studies have analyzed the classes of groups
with word problems in various families of formal langues. Herbst and Thomas
characterized the groups with one-counter word problem [14, Theorem 5.1]. (For
a later elementary proof of this result, see [17].) The first author of the present
paper investigated groups whose word problem is an intersection of finitely many
context-free languages [2,5]. Holt et al. studied the class of groups whose co-word
problem is context-free [18] and Holt and Röver studied the the class of groups
whose co-word problem is indexed [19].

   The word problem of a group is the language of words representing the iden-
tity over some set of generators and their inverses. Thus two words u and v are
equal in a group G if and only if uV is in the word problem, where V is obtained
from v by replacing each symbol by its inverse and reversing the word. A natural
question is how to generalize this definition to semigroups. Duncan and Gilman
[8, Definition 5.1] defined the word problem of a semigroup S with respect to a
generating set A to be

$$\mathrm{WP}(S, A) = \big\{\, u\#v^{\mathrm{rev}} : u, v \in A^+, u =_S v \,\big\}, \tag{1}$$

where v^{rev} is the reverse of v. This definition fits well with the group definition and is natural when considering word problems recognizable by automata equipped with a stack. It was used by Holt, Owens, and Thomas in their study of groups and semigroups with one-counter word problem [17], and by Hoffmann et al. in their study of semigroups with context-free word problem [15].

The main conclusions of Hoffmann et al.'s earlier study were the result that the class of semigroups with context-free word problem is closed under passing to finite Rees index subsemigroups and extensions [15, Theorem 1] and a characterization of completely simple semigroups with context-free word problem as Rees matrix semigroups over virtually free groups [15, Theorem 2].

This paper explores new directions in the study of the class of semigroups with context-free word problem, including monoids with context-free word problem, and also considers the classes of semigroups and monoids with deterministic context-free word problem. First, Section 3 exhibits some natural classes of semigroups and monoids that lie within and outside these classes; in particular Example 2 shows that having context-free and deterministic context-free word problem do not coincide for semigroups or monoids, unlike (as noted above) for groups. Section 4 discusses connections with the theories of word-hyperbolic and automatic semigroups: any semigroup or monoid with context-free word problem is word-hyperbolic, but there are non-automatic semigroups that have context-free word problem. The remainder of the core of the paper (Sections 5–8) focusses on various constructions: direct products, free products, strong semilattices of semigroups, Rees matrix semigroups and Bruck–Reilly extensions. For each construction, the questions of interest are: (1) Are the classes of semigroups and monoids with context-free or deterministic context-free word problem closed under that construction? (2) If the result of applying such a construction lies in one of these classes, must the original semigroup(s) or monoids(s) lie in that same class? Finally, Section 10 lists some open problems.

2   Preliminaries

The *word problem* for a semigroup S is defined as (1) above. Similarly, the *word problem* for a monoid M with respect to a generating set A is the language

$$\mathrm{WP}(M, A) = \left\{\, u \# v^{\mathrm{rev}} : u, v \in A^*, u =_M v \,\right\}. \tag{2}$$

Proposition 1 ([15, Proposition 8]). *Let $\mathfrak{C}$ be a class of languages closed under inverse homomorphisms and intersection with regular languages. Then*

1. *If a semigroup or monoid has word problem in $\mathfrak{C}$ with respect to some generating set, then it has word problem in $\mathfrak{C}$ with respect to any generating set.*
2. *The class of semigroups (resp. monoids) with word problem in $\mathfrak{C}$ is closed under taking finitely generated subsemigroups (resp. submonoids).*

The preceding result applies in particular when $\mathfrak{C}$ is the class of context-free or deterministic context-free languages [12,20].

If a semigroup (resp. monoid) has word problem in a class of languages $\mathfrak{C}$, it is said to be a $U(\mathfrak{C})$ semigroup (resp. monoid). We denote the classes of context-free and deterministic context-free languages by $\mathcal{CF}$ and $\mathcal{DCF}$ respectively. The 'U' notation is because (1) and (2) treat the word problem as an 'unfolded' relation rather than a 'two-tape' relation; see [4] for a systematic study.

3   Examples

We recall some less commonly-used terms from the theory of rewriting systems; see [1] for general background. A rewriting system $(A, \mathcal{R})$ is *monadic* if it is length-reducing and the right-hand side of each rewrite rule in $\mathcal{R}$ lies in $A \cup \{\varepsilon\}$. A monadic rewriting system $(A, \mathcal{R})$ is *regular* (respectively, *context-free*) if, for each $a \in A \cup \{\varepsilon\}$, the set of all left-hand sides of rewrite rules in $\mathcal{R}$ with right-hand side a is a regular (respectively, context-free) language.

Theorem 1 ([6, Theorem 3.1]). *Let $(A, \mathcal{R})$ be a confluent context-free monadic rewriting system. Then the monoid presented by $\langle A \,|\, \mathcal{R} \rangle$ is $U(\mathcal{CF})$, and a context-free grammar generating its word problem can be effectively constructed from context-free grammars describing $\mathcal{R}$.*

(The preceding result originally stated that a monoid satisfying the hypothesis was word-hyperbolic; however, the proof proceeds by constructing the word problem for the monoid. The 'effective construction' part follows easily by inspecting the construction in the proof.)

Example 1. This example shows that a $U(\mathcal{CF})$ monoid need not have a context-free cross section (that is, a language over some generating set containing a unique representative for every element).

Let $K = \{\, a^\alpha b^\alpha c^\alpha : \alpha \in \mathbb{N} \cup \{0\} \,\}$ and let $L = \{a, b, c\}^* - K$. It is well-known that K is not a context-free language but that L is a context-free language. Let $A = \{a, b, c, x, y, z\}$ and let $\mathcal{R} = \{\, (xwy, z) : w \in L \,\}$. Let M be the monoid presented by $\langle A \,|\, \mathcal{R} \rangle$. By Theorem 1, M is $U(\mathcal{CF})$. Suppose that M admits a context-free cross-section. Then M admits a context-free cross-section $J \subseteq A^*$. Let u be the unique word in J such that $u =_M z$, and let $J' = (J \setminus \{u\}) \cup \{z\}$; then J' is also a context-free cross-section of M. Let $H = J' \cap x\{a, b, c\}^* y$. Then H is context-free and comprises precisely the words xwy where $w \in K$, for if $w \in L$, then $xwy =_M z$, and the representative of z in J' is the word z itself. Hence, since the class of context-free languages is closed under right and left quotients with regular sets, $K = x\backslash H / y$ is context-free. This is a contradiction, and so M does not admit a context-free cross-section.

Example 2. This example shows that the class of $U(\mathcal{DCF})$ semigroups is properly contained in the class of $U(\mathcal{CF})$ semigroups.

Let K be the language of palindromes over $\{a, b\}$. It is well-known that K is context-free but not deterministic context-free. Let $A = \{a, b, x, y, z\}$ and let $\mathcal{R} = \{\, (xwy, z) : w \in L \,\}$. Let M be the monoid presented by $\langle A \,|\, \mathcal{R} \rangle$.

By Theorem 1, M is $U(\mathcal{CF})$. Suppose, with the aim of obtaining a contradiction, that M is $U(\mathcal{DCF})$. Then $\mathrm{WP}(M, A)$ is deterministic context-free. Let $L = (\mathrm{WP}(M, A) \cap A^*\#z)/\{\#z\} \cap \{a, b, x, y\}^*$; then L is the language of words over $\{a, b, x, y\}$ that are equal to z in M. Furthermore, L is deterministic context-free, since the class of deterministic context-free languages is closed under intersection with regular languages [20, Theorem 10.4] and right quotient by regular languages [20, Theorem 10.2].

Now, $K = x\backslash L/y$. The class of determinstic context-free languages is closed under left quotient by a singleton (since a deterministic pushdown automaton can simulate reading a fixed word before it starts reading input), and, as noted above, is closed under right quotient by regular languages. Hence K is deterministic context-free. This is a contradiction, and so M is not $U(\mathcal{DCF})$.

Example 3. An example of a monoid that is 'close' to being a free group but is not $U(\mathcal{CF})$ is the free inverse monoid of rank 1 and hence (by Proposition 1) of any finite rank. This follows from applying the pumping lemma to the intersection of the word problem and the regular language $x^*(x^{-1})^*x^*\#x^*$ (where x is the free generator); see [3, Theorem 1].

4   Relationship to word-hyperbolicity and automaticity

Hyperbolic groups have become one of the most fruitful areas of group theory since their introduction by Gromov [13]. The concept of hyperbolicity can be generalized to semigroups and monoids in more than one way, but here we consider the linguistic definition that uses Gilman's characterization of hyperbolic groups using context-free languages [11]. A *word-hyperbolic structure* for a semigroup S is a pair $(L, M(L))$, where L is a regular language over an alphabet A representing a finite generating set for S such that L maps onto S, and where

$$M(L) = \{u\#_1v\#_2w^{\mathrm{rev}} : u, v, w \in L \wedge uv =_S w\}$$

(where $\#_1$ and $\#_2$ are new symbols not in A) is context-free.

Theorem 2. *Every $U(\mathcal{CF})$ semigroup is word-hyperbolic.*

The proof is in effect the first paragraph of the proof of Theorem 1 as given in [6, Proof of Theorem 2].

All hyperbolic groups are automatic [10, Theorem 3.4.5], but word-hyperbolic semigroups may not even be *asynchronously* automatic [16, Example 7.7]. Even within the smaller class of $U(\mathcal{CF})$ semigroups, one can find semigroups that are not automatic:

Example 4. Let $A = \{a, b, c, d, z\}$, let $\mathcal{R} = \{(ab^\alpha c^\alpha d, z) : \alpha \in \mathbb{N}\}$. Let M be the monoid presented by $\langle A \,|\, \mathcal{R}\rangle$. Then M is $U(\mathcal{CF})$ by Theorem 1, but cannot be automatic [7, Corollary 5.5]. (In fact, it can be shown that M is not even asynchronously automatic.)

Given that $U(\mathcal{CF})$ groups are virtually free and thus automatic, and since the monoid in Example 4 is not cancellative, the following question is natural:

Question 1. Is a cancellative $U(\mathcal{CF})$ semigroup necessarily automatic?

5   Direct products

A direct product of two finitely generated semigroups is not necessarily finitely generated. However, a direct product of two $U(\mathcal{CF})$ semigroups is not necessarily $U(\mathcal{CF})$, even if it is finitely generated: for example, the free monoid of rank 1 is $U(\mathcal{CF})$, but the direct product of two copies of this monoid is the free commutative monoid of rank 2, which is finitely generated but not $U(\mathcal{CF})$.

For a semigroup S, we say that S is *decomposable* if $S^2 = S$. We will show that for a direct product of two $U(\mathcal{CF})$ semigroups to be $U(\mathcal{CF})$, it is necessary and sufficient that one of the factors is finite and decomposable (decomposability being necessary to ensure finite generation). First we establish sufficiency.

Lemma 1. *The classes of $U(\mathcal{CF})$ and $U(\mathcal{DCF})$ semigroups are closed under taking direct product with a finite decomposable semigroup.*

Proof. Let S be a $U(\mathcal{CF})$ semigroup and T a finite decomposable semigroup. Then $S \times T$ is finitely generated ([23, Theorem 8.2]). Let C be a finite generating set for $S \times T$ and let A and B be the projections of C onto S and T respectively. Then A and B are finite generating sets for S and T respectively. Thus there exists a pushdown automaton $\mathcal{A}$ recognising $\mathrm{WP}(S, A)$, which can be modified to give a pushdown automaton $\mathcal{A}'$ recognising $\mathrm{WP}(S \times T, A \times B)$, by processing the symbols from A as usual, while using the states to record the finite information required to check validity of the input on the second tape. Hence $S \times T$ is $U(\mathcal{CF})$. Moreover, if S is $U(\mathcal{DCF})$, then $\mathcal{A}$ can be taken to be deterministic, in which case $\mathcal{A}'$ is also deterministc, so $S \times T$ is $U(\mathcal{DCF})$.   □

Necessity arises from the following language-theoretic result, which encapsulates the idea that context-free languages cannot admit 'cross-dependencies'. For words w, w', we use the notation $w' \sqsubseteq w$ to mean that w' is a subword of w.

Lemma 2. *Let A and B be disjoint alphabets, and let ρ_A, ρ_B be equivalence relations on A^* and B^* respectively with infinitely many equivalence classes. Then the language $L(\rho_A, \rho_B) = \{\, u_1 v_1 u_2 v_2 : (u_1, u_2) \in \rho_A, (v_1, v_2) \in \rho_B \,\}$ is not context-free.*

Proof. Suppose that $L = L(\rho_A, \rho_B)$ is context-free, and let k be the pumping constant for L. Let $\mathcal{E}_A$ be the set of all equivalence classes of ρ_A that contain a word of length at most k, and define $\mathcal{E}_B$ similarly.

Let $w = u_1 v_1 u_2 v_2 \in L$ with $|v_1|, |u_2| > k$. Then we can write $w = pqrst$ where $|qrs| \leq k$, $|qs| \geq 1$ and $pq^i r s^i t \in L$ for all $i \in \mathcal{N}_0$. Due to the form of words in L, q and s must each be a subword of some u_i or v_i. Moreover, the lengths of u_2 and v_1 preclude the possibility that $q \sqsubseteq u_1$ and $s \sqsubseteq u_2$ or $p \sqsubseteq v_1$ and $q \sqsubseteq v_2$. Let $w' = prt = u_1' v_1' u_2' v_2'$. Then we have $u_i' = u_i$ for some $i \in \{1, 2\}$ and $v_j' = v_j$ for some $j \in \{1, 2\}$. Since $w' \in L$, this implies that the equivalence classes of the factors are unchanged between w and w'. By induction, we can repeat this process until we obtain a word $w^\flat = u_1^\flat v_1^\flat u_2^\flat v_2^\flat \in L$ with $|v_1^\flat| \leq k$ or $|u_2^\flat| \leq k$, where the $u_i^\flat$ are in the same ρ_A-equivalence class as the u_i and the $v_i^\flat$

are in the same ρ_B-equivalence class as the v_i. Hence our original word w had either $u_i \in C$ for some $C \in \mathcal{E}_A$ or $v_i \in D$ for some $D \in \mathcal{E}_B$. But $\mathcal{E}_A$ and $\mathcal{E}_B$ are both finite, and so L cannot contain all words of the form $u_1 v_1 u_2 v_2$ with $(u_1, u_2) \in \rho_A$ and $(v_1, v_2) \in \rho_B$. Hence L is not context-free. $\square$

The preceding lemma is immediately applicable only to monoids.

Lemma 3. *The direct product of two infinite monoids cannot be $U(\mathcal{CF})$.*

Proof. Let $S = \langle A \rangle$ and $T = \langle B \rangle$ be infinite monoids. Then the relations $\rho_A = \iota(S, A)$ and $\rho_B = \iota(T, B)$ both have infinitely many equivalence classes. Moreover, the language $L = \mathrm{WP}(S \times T, A \cup B) \cap A^* B^* \# A^* B^*$ has as a homomorphic image the language $L(\rho_A, \rho_B)$ defined in Lemma 2. Since the class of context-free languages is closed under homorphisms and intersection with regular sets, this implies that $S \times T$ is not $U(\mathcal{CF})$.

Thus if $S \times T$ is $U(\mathcal{CF})$, then at least one of S or T is finite. $\square$

In order to extend Lemma 3 to all semigroups, we first establish the following fact (which is clear for monoids, where direct factors are submonoids).

Lemma 4. *The class of $U(\mathcal{CF})$ semigroups is closed under taking direct factors.*

Proof. Assume that $S \times T$ is $U(\mathcal{CF})$. In particular, $S \times T$ is finitely generated. By [23, Theorem 2.1], S and T are finitely generated, and $S^2 = S$ and $T^2 = T$.

Let $C = \{c_1, \ldots, c_k\}$ be a finite generating set for T. Since $T^2 = T$, we can choose a factorization $c_i = c_{i\zeta} u_i$ for each $c_i \in C$. Construct a labelled digraph with vertex set C and an edge from c_i to $c_{i\zeta}$ labelled by u_i for each $c_i \in C$. Since this digraph is finite, it must contain a circuit. Fix some vertex c on that circuit and let w be the concatenation in reverse order of the labels on the edges around the circuit. Then $cw = c$.

Let A be a finite generating set for $S \times T$ and let B be a finite generating set for S. Then $X = A \cup (B \times \{c, w\})$ is a finite generating set for $S \times T$. Let R be the regular language $(B \times \{c\})(B \times \{w\})^* \# (B \times \{w\})^* (B \times \{c\})$. Let $L = \mathrm{WP}(S \times Y, X) \cap R$. Then

$$\begin{aligned} &(b_1, c)(b_2, w) \cdots (b_m, w) \# (b'_n, w) \cdots (b'_2, w)(b_1, c) \in L \\ \iff &(b_1 b_2 \cdots b_m, cw^{m-1}) =_{S \times T} (b'_1 b'_2 \cdots b'_n, cw^{n-1}) \\ \iff &(b_1 b_2 \cdots b_m, c) =_{S \times T} (b'_1 b'_2 \cdots b'_n, c) \\ \iff &b_1 b_2 \cdots b_m =_S b'_1 b'_2 \cdots b'_n. \end{aligned} \tag{3}$$

Define a homomorphism

$$\pi : \big((B \times \{c, w\}) \cup \{\#\}\big) \to \big(B \cup \{\#\}\big), \qquad (b, \_) \mapsto b, \quad \# \mapsto \#.$$

Then (3) shows that $L\pi = \mathrm{WP}(S, B)$. Since the class of context-free languages is closed under homomorphism [20, Corollary to Theorem 6.2], S is a $U(\mathcal{CF})$ semigroup. $\square$

Theorem 3. *The direct product of two semigroups is $U(\mathcal{CF})$ if and only if it is finite or one of the factors is $U(\mathcal{CF})$ and the other factor is finite and decomposable.*

Proof. Sufficiency was already established in Lemma 1.

Conversely suppose that $S \times T$ is $U(\mathcal{CF})$. Let C be a finite generating set for $S \times T$ with the projection of C onto the first component being A and the projection onto the second component B. By Lemma 4, S and T are both $U(\mathcal{CF})$. Let $A_1 = A \times \{1\}$, $B_1 = \{1\} \times B$, and $C_1 = A_1 \cup B_1 \cup C$. We will describe a pushdown automaton $\mathcal{P}$ recognising $\mathrm{WP}(S^1 \times T^1, C_1)$. This automaton is defined in terms of pushdown automata $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, recognising $\mathrm{WP}(S, A)$, $\mathrm{WP}(T, B)$ and $\mathrm{WP}(S \times T, C)$ respectively.

On input $(x, y) \in C$, the automaton $\mathcal{P}$ behaves as a 'delayed' version of $\mathcal{C}$, storing the input symbol in the state and then (except in the start state, which has no stored symbol) simulating $\mathcal{C}$ on input of the current stored symbol. The automaton may guess at any point that the input is complete, and process the stored symbol from the current state as an ϵ-transition. In this case we move to a state with no stored symbol and accepting no further input, which is a final state if and only if it is a final state in $\mathcal{C}$. Thus on input in $(C \cup \{\#\})^*$, $\mathcal{P}$ behaves exactly like $\mathcal{C}$ but 'one step behind', and so the sublanguage of $(C \cup \{\#\})^*$ accepted by $\mathcal{P}$ is $\mathrm{WP}(S \times T, C)$.

In order to work with input from $A_1 \cup B_1$ we choose, for all $x, x' \in A$ and $y, y' \in B$, representatives $w_{x,x',y}$ and $w_{x,y,y'}$ in C for the elements (xx', y) and (x, yy') of $S \times T$.

Now, if the automaton $\mathcal{P}$ reads the symbol $(x', 1)$ in a state with stored symbol (x, y), it simulates reading all but the final symbol of $w_{x,x',y}$ in $\mathcal{C}$ from the current state, and stores the final symbol in the last state of this computation. Symmetrically, the same occurs when we replace $(x', 1)$ by $(1, y')$ and $w_{x,x',y}$ by $w_{x,y,y'}$. Thus on input $u\#v$ from $CC_1^* \# CC_1^*$, the automaton is able to simulate processing in $\mathcal{C}$ some $u'\#v'$ such that $u =_{S \times T} u'$ and $v =_{S \times T} v'$.

Finally, on input from A_1 or B_1 in the start state, the automaton guesses whether the remaining (non-$\#$) input will be in A_1^* resp. B_1^*. If it guesses yes, it moves to a copy of the appropriate automaton $\mathcal{A}$ resp. $\mathcal{B}$, treating input $(x, 1)$ as x and $(1, y)$ as y. Thus the sublanguage of $(A_1 \cup \{\#\})^*$ recognised by $\mathcal{P}$ is $\mathrm{WP}(S \times \{1\}, A_1)$, while the sublanguage of $(B_1 \cup \{\#\})^*$ recognised is $\mathrm{WP}(\{1\} \times T, B_1)$. If, on the other hand, the automaton guesses no, we describe what happens on input from A_1, the other case being symmetric. Supposing the input is $(x, 1)$, the automaton guesses which $y \in B$ will be read next, and stores this guess in the state, along with the symbol (x, y). States with a stored guess $y \in B$ operate as usual, except on input of the form (x, y). On such input, the automaton deletes the 'guess' y and otherwise operates as if the input were $(x, 1)$, since it already simulated reading y earlier. (If $x = 1$, then we simply delete the guess and otherwise do nothing.) The automaton must similarly make a guess on input from A_1 or B_1 in a state with stored symbol $\#$. Since $(x, 1)w(x', y) = (x, y)w(x', 1)$ for $w \in A_1^*$, the automaton $\mathcal{P}$ is now able to simulate reading a corresponding word in C^* for any input not in $(A_1 \cup \{\#\})^* \cup (B_1 \cup \{\#\})^*$.

Combined with the fact that $\mathcal{P}$ can also simulate the automata $\mathcal{A}$ and $\mathcal{B}$ on appropriate inputs, this establishes that $\mathcal{P}$ recognises $\mathrm{WP}(S^1 \times T^1, C_1)$.

Thus $S^1 \times T^1$ is $U(\mathcal{CF})$, and so by Lemma 3, without loss of generality we can assume T^1 is finite. Moreover, S^1 is $U(\mathcal{CF})$, and hence so is S, by Proposition 1.2. By [23, Theorem 8.1], if S is infinite then T must also be decomposable, since $S \times T$ is finitely generated.   □

6   Free products

Theorem 4. *The class of $U(\mathcal{CF})$ semigroups is closed under taking semigroup free products and under taking free factors.*

Proof. Let S and T be $U(\mathcal{CF})$ semigroups. Let A_S and A_T be finite generating sets for S and T, respectively, and for $X \in \{S, T\}$, let $\mathcal{P}_X$ be a pushdown automaton recognizing $\mathrm{WP}(X, A_X)$ accepting by final state, Assume that in $\mathcal{P}_X$, the initial stack content is only a stack bottom symbol $\bot_X$, which is never never popped or pushed.

Construct a new pushdown automaton $\mathcal{Q}$ recognizing words over $A_S \cup A_T \cup \{\#\}$, functioning as follows. First, $\mathcal{Q}$ will recognize words in $(A_S \cup A_T)^+ \#(A_S \cup A_T)^+$; since this is a regular language, assume without loss that the input is in this form. When $\mathcal{Q}$ begins, it reads a symbol from A_X (for some $X \in \{S, T\}$). It pushes $\bot_X$ onto its stack and begins to simulate $\mathcal{P}_X$. Whenever it is simulating $\mathcal{P}_X$ and reads a symbol from A_Y, where $Y \neq X$, it pushes the current state of $\mathcal{P}_X$ onto the stack, then pushes $\bot_Y$ onto the stack and begins to simulate $\mathcal{P}_Y$. These alternating simulations of $\mathcal{P}_S$ and $\mathcal{P}_T$ continue until the $\#$ is encountered.

On reading the symbol $\#$, the automaton $\mathcal{Q}$ continues to simulate whichever $\mathcal{P}_X$ it was currently simulating. After this point, whenever it is simulating $\mathcal{P}_X$ (for some $X \in \{S, T\}$) and reads a symbol from A_Y, where $Y \neq X$, how it proceeds depends on whether the currently-simulated $\mathcal{P}_X$ is in an accept state:

- If it is in accept state, $\mathcal{Q}$ pops symbols from its stack until it encounters $\bot_X$, which it pops, then pops the state of $\mathcal{P}_Y$, restores the simulation of $\mathcal{P}_Y$ from this state (and with the stack contents down to the symbol $\bot_Y$), and simulates $\mathcal{P}_Y$ on reading $\#$ and then on reading the symbol just read by $\mathcal{Q}$. (If after popping $\bot_X$ the stack of $\mathcal{Q}$ is empty, it fails.)
- If it is not in an accept state, $\mathcal{Q}$ fails.

These alternating simulations of $\mathcal{P}_S$ and $\mathcal{P}_T$ continue until the end of the input unless $\mathcal{Q}$ fails before then. At this point $\mathcal{Q}$ accepts if the currently-simulated $\mathcal{P}_X$ is in an accept state, and if the stack only contains symbols from the stack alphabet B_X plus a single symbol $\bot_X$.

It follows from the above description that $\mathcal{Q}$ recognizes strings of the form

$$u_1 u_2 \cdots u_k \# v_k^{\mathrm{rev}} \cdots v_2^{\mathrm{rev}} v_1^{\mathrm{rev}}, \tag{4}$$

where $u_i \# v_i^{\mathrm{rev}} \in L(\mathcal{P}_{X(i)})$ and either $X(2j) = S$ and $X(2j + 1) = T$, or else $X(2j) = T$ and $X(2j + 1) = S$. Thus $\mathcal{Q}$ recognizes strings (4) such that

$$u_1 u_2 \cdots u_k =_{S*T} v_1 v_2 \cdots v_k,$$

and the u_i and v_i are either both in A_X^+ or both in A_Y^+ for alternating i. Thus $\mathcal{Q}$ recognizes $\mathrm{WP}(S * T, A_X \cup A_Y)$.

The free factors of a finitely generated free product are themselves finitely generated, so closure under free factors follows from Proposition 1.2.   □

Notice that the strategy of the proof of Theorem 4 cannot be applied to show that the class of $U(\mathcal{DCF})$ semigroups is closed under taking free products. The problem is in the very last step: after the automaton has read its last input symbols from some A_X, it cannot deterministically check that the stack only contains symbols from the stack alphabet B_X plus a single symbol $\perp_X$. Therefore the following question remains open:

Question 2. Is the class of $U(\mathcal{DCF})$ semigroups closed under forming free products?

Theorem 5. *The class of $U(\mathcal{CF})$ monoids is closed under taking monoid free products and free factors.*

Proof (Sketch proof). It is easy to see that the construction of the $\mathcal{Q}$ from the proof of Theorem 4 can be adapted to the case of monoid free products. Using the notation from that proof, one observes that for $X \in \{S, T\}$ the language of words over A_X representing the identity of X is a context-free language K_X. Then one first modifies $\mathcal{Q}$ to accept # (that is, the empty word, followed by #, followed by the empty word), then modifies $\mathcal{Q}$ so that it can non-deterministically read a string from either K_X at any point (including while reading another string from K_Y for $Y \in \{S, T\}$, so that such strings can be 'nested').   □

7   Strong semilattices

We recall the definition of a strong semilattice of semigroups here, and refer the reader to [21, Sect. 4.1] for further background reading:

Let Y be a semilattice. Recall that the meet of $\alpha, \beta \in Y$ is denoted $\alpha \wedge \beta$. For each $\alpha \in Y$, let S_α be a semigroup. For $\alpha \geq \beta$, let $\phi_{\alpha,\beta} : S_\alpha \to S_\beta$ be a homomorphism such that

1. For each $\alpha \in Y$, the homomorphism $\phi_{\alpha,\alpha}$ is the identity mapping.
2. For all $\alpha, \beta, \gamma \in Y$ with $\alpha \geq \beta \geq \gamma$, $\phi_{\alpha,\beta}\phi_{\beta,\gamma} = \phi_{\alpha,\gamma}$.

The *strong semilattice of semigroups* $S = \mathcal{S}[Y; S_\alpha; \phi_{\alpha,\beta}]$ consists of the disjoint union $\bigcup_{\alpha \in Y} S_\alpha$ with the following multiplication: if $x \in S_\alpha$ and $y \in S_\beta$, then

$$xy = (x\phi_{\alpha,\alpha \wedge \beta})(y\phi_{\beta,\alpha \wedge \beta}).$$

Theorem 6. *Let $\mathfrak{C}$ be a class of languages closed under finite union, inverse gsm-mappings and intersection with regular languages (in particular, the class $\mathcal{CF}$). A strong semilattice of semigroups is $U(\mathfrak{C})$ if and only if it is finitely generated and all the semigroups in its semilattice are $U(\mathfrak{C})$.*

Proof. Let $S = \mathcal{S}[Y; S_\alpha; \phi_{\alpha,\beta}]$ be a strong semilattice of semigroups. If S is $U(\mathfrak{C})$, then it is finitely generated by some set A. It follows from the definition of a strong semilattice of semigroups that Y is generated by those $\alpha \in Y$ such that $A \cap S_\alpha \neq \emptyset$, and so is finite, and that each S_α is finitely generated by elements of the form $a\phi_{\beta,\alpha}$ where $a \in S_\beta$ and $\beta \geq \alpha$. Moreover, all the S_α are $U(\mathfrak{C})$, since they are finitely generated subsemigroups of S.

Conversely, suppose that Y is finite and each S_α is $U(\mathfrak{C})$. For each $\alpha \in Y$, let A_α be a finite generating set for S_α and $A'_\alpha = \bigcup_{\beta \geq \alpha} A_\alpha$. Let $A = \bigcup_{\alpha \in Y} A_\alpha$. Define homomorphisms $\phi_\alpha : (A'_\alpha)^* \to A^*_\alpha$ by $x \mapsto x\phi_{\beta,\alpha}$ for $x \in A_\beta$.

We can view $\mathrm{WP}(S, A)$ as the union of its restrictions to each S_α: that is, as the union of the languages $L_\alpha = \{\, u\#v^{\mathrm{rev}} \in \mathrm{WP}(S,A) : u, v \in S_\alpha \,\}$. In turn, each L_α can be expressed as $L'_\alpha \cap R_\alpha$, where $L'_\alpha = \{\, u\#v^{\mathrm{rev}} : u, v \in A^*, u\phi_\alpha =_{S_\alpha} v\phi_\alpha \,\}$ and $R_\alpha = \{\, u\#v^{\mathrm{rev}} : u, v \in A^*, u, v \in S_\alpha \,\}$. Note that $u\#v^{\mathrm{rev}} \in L'_\alpha$ implies $u, v \in S_\beta$ for some $\beta \geq \alpha$, since otherwise ϕ_α is not defined. We have $L'_\alpha, R_\alpha \subseteq (A'_\alpha)^*$ for all $\alpha \in Y$.

Defining $R'_\alpha = \{\, w \in (A'_\alpha)^* : w \in S_\alpha \,\}$, we have $R_\alpha = R'_\alpha \# R'_\alpha$ (since membership of w in S_α depends only on the content of w). The language R'_α is recognised by a finite automaton consisting of the semilattice Y with an adjoined top element $\top$ as the start state, and final state α. The transition function is given by the meet operation: $(\top, x) \mapsto \gamma$ and $(\beta, x) \mapsto \beta \wedge \gamma$ for $x \in A_\gamma$. A word w is accepted by this automaton if and only if the meet of all γ such that w contains a symbol in A_γ is α. Thus R'_α is regular, and hence so is R_α, as a concatenation of regular languages.

Now choose a homomorphism $\psi_\alpha : (A'_\alpha)^* \to A^*_\alpha$ defined by $x \mapsto w_x$ such that $w_x =_S x\phi_\alpha$. Let $W = \{\, w_x : w \in A^*_\alpha \,\}$ and $M = \mathrm{WP}(S_\alpha, A_\alpha) \cap W^*$. Then $M \in \mathfrak{C}$, and L'_α is the inverse image of M under the gsm-mapping from $(A'_\alpha)^* \# (A'_\alpha)^*$ to $(A_\alpha)^* \# (A_\alpha)^*$ that preserves $\#$ and maps all symbols in x before the $\#$ to $x\psi_\alpha$ and all symbols x after the $\#$ to $(x\psi_\alpha)^{\mathrm{rev}}$. Since $\mathfrak{C}$ is closed under inverse gsm-mappings, L'_α is thus in $\mathfrak{C}$. In turn, L_α is in $\mathfrak{C}$, hence so is $\mathrm{WP}(S, A)$, as the union of the finitely many L_α. $\qquad\square$

The class $\mathcal{DCF}$ is not closed under finite union [20, Theorem 10.5(b)]. We conjecture that a finitely generated strong semilattice of $U(\mathcal{DCF})$ semigroups need not be $U(\mathcal{DCF})$. Let $Y = \{\alpha, \beta\}$ be a two-element semilattice with $\alpha > \beta$. Let S_α be the free group generated by $\{x, y\}$ and let S_β be $\mathbb{Z}$ (under $+$). Define $\phi_{\alpha,\beta}$ to be the homomorphism extending $x \mapsto 1$, $y \mapsto -1$. Both S_α and S_β are virtually free groups and so $U(\mathcal{DCF})$, but the word problem of $\mathcal{S}[Y; \{S_\alpha, S_\beta\}; \phi_{\alpha,\beta}]$ does not appear to be deterministic context-free, for checking equality in S_α seems to require computing reduced words on the stack, while checking equality in $\mathbb{Z}$ seems to require using the stack as a counter, and there is no way to know in advance which is required.

8   Rees matrix semigroups

Let us recall the definition of a Rees matrix semigroup. Let S be a semigroup, let I and Λ be abstract index sets, and let $P \in \mathrm{Mat}_{\Lambda \times I}(S)$ (that is, P is a $\Lambda \times I$

matrix with entries from S). Denote the (λ, i)-th entry of P by $p_{\lambda i}$. The *Rees matrix semigroup* over S with sandwich matrix P, denoted $\mathcal{M}[S; I, \Lambda; P]$, is the set $I \times S \times \Lambda$ with multiplication defined by

$$(i, x, \lambda)(j, y, \mu) = (i, x p_{\lambda j} y, \mu).$$

This construction is important because it arises in the classification of completely simple semigroups as Rees matrix semigroups over groups; see [21, Sect. 3.2–3.3].

Hoffmann et al. showed that a completely simple semigroup is $U(\mathcal{CF})$ if and only if it is isomorphic to a Rees matrix semigroup over a finitely generated virtually free group [15, Theorem 2]; their proof depends on virtually free groups having *deterministic* context-free word problem. The following theorem generalizes Hoffmann et al.'s characterization to Rees matrix semigroups over arbitrary semigroups. See [12,20] for background on inverse gsm-mappings.

Theorem 7. *Let $\mathfrak{C}$ be a class of languages closed under inverse gsm-mappings and intersection with regular languages (in particular, $\mathcal{CF}$ or $\mathcal{DCF}$). Then a finitely generated Rees matrix semigroup over a semigroup S is $U(\mathfrak{C})$ if and only if S is $U(\mathfrak{C})$.*

Proof. Let $M = \mathcal{M}[S; I, \Lambda; P]$ be a Rees matrix semigroup and let $\mathfrak{C}$ be as in the statement of the theorem. If M is $U(\mathfrak{C})$, then it must be finitely generated, hence S is also finitely generated and thus $U(\mathfrak{C})$.

Conversely, suppose that S is $U(\mathfrak{C})$ and M is finitely generated by $B \subseteq I \times S \times \Lambda$, and let A be the projection of B onto S. For each $i \in I$ and $\lambda \in \Lambda$, choose a word $w_{\lambda i} \in A^*$ representing $p_{\lambda i}$. Let W be the (finite) set of all the $w_{\lambda i}$. Let $L = \mathrm{WP}(S, A) \cap (AW)^* A \# A(WA)^*$, which is in $\mathfrak{C}$, as the intersection of a language in $\mathfrak{C}$ with a regular language. We will define a gsm-mapping Φ such that $\mathrm{WP}(M, B)$ is the inverse image of L under Φ.

First, define a gsm-mapping $\phi : B^* \to A^*$ by

$$(i_1, x_1, \lambda_1) \dots (i_m, x_m, \lambda_m) \mapsto x_1 w_{\lambda_1 i_2} x_2 \dots w_{\lambda_{m-1} i_m} x_m.$$

Then for $w = (i_1, x_1, \lambda_1) \dots (i_m, x_m, \lambda_m)$ we have $w =_M (i(w), w\phi, \lambda(w))$, where $i(w) := i_1$ and $\lambda(w) := \lambda_m$.

Now extend ϕ to a gsm-mapping $\Phi : (B \cup \{\#\})^* \to (A \cup \{\#\})^*$ as follows: For $u, v \in B^*$ and $w \in (B \cup \{\#\})^*$, let $(u \# v^{\mathrm{rev}})\Phi = u\phi \#(v\phi)^{\mathrm{rev}} c$, where $c = \varepsilon$ if $i(u) = i(v)$ and $\lambda(u) = \lambda(v)$, and $c = \#$ otherwise. (Since I and Λ are finite, the computation of c can be done by storing $i(u)$ and $\lambda(u)$ in the state and then checking against $\lambda(v)$ and $i(v)$.) Let $(u \# v^{\mathrm{rev}} \# w)\Phi = u\phi \#(v\phi)^{\mathrm{rev}} \#$ (achieved by storing in the state whether $\#$ has already been seen).

The preimage of L in $(B \cup \{\#\})^*$ under Φ consists of all words of the form $u \# v^{\mathrm{rev}}$ with $u, v \in B^*$ such that $i(u) = i(v)$, $\lambda(u) = \lambda(v)$ and $u\phi \#(v\phi)^{\mathrm{rev}} \in \mathrm{WP}(S, A)$. But this is exactly all $u \# v^{\mathrm{rev}}$ such that $u =_M v$, so $L\Phi^{-1} = \mathrm{WP}(M, B)$. Hence M is $U(\mathfrak{C})$, since its word problem is obtained from a language in $\mathfrak{C}$ by an inverse gsm-mapping. $\qquad\square$

The fact that every completely regular semigroup is isomorphic to a semilattice (not necessarily strong) of completely simple semigroups [21, Theorem 4.1.3] raises the following question:

Question 3. Which completely regular semigroups are $U(\mathcal{CF})$?

9 Bruck–Reilly extensions

Let M be a monoid with presentation $\langle A \,|\, \mathcal{R} \rangle$ and $\phi : M \to M$ an endomorphism. The *Bruck–Reilly extension* $\mathrm{BR}(M, \phi)$ of M by ϕ is the monoid with presentation $\langle A, b, c \,|\, \mathcal{R}, bc = 1, ba = (a\phi)b, ac = c(a\phi) \quad (a \in A) \rangle$. This is an analogue for monoids of the notion of HNN-extensions for groups.

If ϕ is the identity endomorphism, then $\mathrm{BR}(M, \phi)$ is isomorphic to the direct product of M with the bicyclic monoid generated by $\{b, c\}$. Thus by Lemma 3 the class of $U(\mathcal{CF})$ semigroups is not closed under Bruck–Reilly extensions. However, we can establish a necessary and sufficient condition for $\mathrm{BR}(M, \phi)$ to be $U(\mathcal{CF})$, though we omit the proof here.

Theorem 8. *Let M be a monoid and $\phi : M \to M$ an endomorphism. Then $\mathrm{BR}(M, \phi)$ is $U(\mathcal{CF})$ if and only if M is $U(\mathcal{CF})$ and $\operatorname{im} \phi^n$ is finite for some n.*

10 Further open problems

Question 4. Does every cancellative $U(\mathcal{CF})$ semigroup have deterministic context-free word problem?

Question 5. Is it possible to characterize the commutative (respectively, cancellative, inverse) $U(\mathcal{CF})$ semigroups?

The previous two questions are motivated by the group case, since the classes of $U(\mathcal{CF})$ and $U(\mathcal{DCF})$ groups coincide and are precisely the virtually free groups. In particular, the abelian $U(\mathcal{CF})$ groups are thus either finite or of the form $\mathbb{Z} \times F$, where F is finite and abelian.

Question 6. Does there exist an infinite periodic $U(\mathcal{CF})$ semigroup?

References

1. Book, R.V., Otto, F.: String Rewriting Systems. Texts and Monographs in Computer Science, Springer (1993)
2. Brough, T.: Groups with poly-context-free word problem. Groups Complexity Cryptology **6**(1), 9–29 (January 2014). https://doi.org/10.1515/gcc-2014-0002
3. Brough, T.: Word problem languages for free inverse monoids. In: Konstantinidis, S., Pighizzini, G. (eds.) Descriptional Complexity of Formal Systems. pp. 24–36. No. 10952 in Lecture Notes in Computer Science, Springer (2018). https://doi.org/10.1007/978-3-319-94631-3_3
4. Brough, T., Cain, A.J.: A language hierarchy of binary relations (2018)
5. Brough, T.R.: Groups with Poly-Context-Free Word Problem. Ph.d. thesis, University of Warwick (2010)
6. Cain, A.J., Maltcev, V.: Context-free rewriting systems and word-hyperbolic structures with uniqueness. International Journal of Algebra and Computation **22**(7) (November 2012). https://doi.org/10.1142/S0218196712500610
7. Campbell, C.M., Robertson, E.F., Ruškuc, N., Thomas, R.M.: Automatic semigroups. Theoretical Computer Science **250**(1-2), 365–391 (January 2001). https://doi.org/10.1016/S0304-3975(99)00151-6
8. Duncan, A., Gilman, R.H.: Word hyperbolic semigroups. Mathematical Proceedings of the Cambridge Philosophical Society **136**(3), 513–524 (1999). https://doi.org/10.1017/S0305004103007497
9. Dunwoody, M.J.: The accessibility of finitely presented groups. Inventiones Mathematicae **81**(3), 449–457 (October 1985). https://doi.org/10.1007/BF01388581
10. Epstein, D.B.A., Cannon, J.W., Holt, D.F., Levy, S.V.F., Paterson, M.S., Thurston, W.P.: Word Processing in Groups. Jones & Bartlett, Boston, MA (1992)
11. Gilman, R.H.: On the definition of word hyperbolic groups. Mathematische Zeitschrift **242**(3), 529–541 (April 2002). https://doi.org/10.1007/s002090100356
12. Ginsburg, S., Greibach, S.: Deterministic context free languages. Information and Control **9**(6), 620–648 (December 1966). https://doi.org/10.1016/S0019-9958(66)80019-0
13. Gromov, M.: Hyperbolic Groups. In: Gersten, S.M. (ed.) Essays in Group Theory, pp. 75–263. No. 8 in Mathematical Sciences Research Institute Publications, Springer-Verlag (1987)
14. Herbst, T., Thomas, R.M.: Group presentations, formal languages and characterizations of one-counter groups. Theoretical Computer Science **112**(2), 187–213 (May 1993). https://doi.org/10.1016/0304-3975(93)90018-O
15. Hoffmann, M., Holt, D.F., Owens, M.D., Thomas, R.M.: Semigroups with a context-free word problem. In: Developments in Language Theory. pp. 97–108. Springer (2012). https://doi.org/10.1007/978-3-642-31653-1_10
16. Hoffmann, M., Kuske, D., Otto, F., Thomas, R.M.: Some relatives of automatic and hyperbolic groups. In: Gomes, G.M.S., Pin, J.É., Silva, P.V. (eds.) Semigroups, algorithms, automata and languages. p. 379–406. World Scientific, River Edge, NJ (2002). https://doi.org/10.1142/9789812776884_0016
17. Holt, D.F., Owens, M.D., Thomas, R.M.: Groups and semigroups with a one-counter word problem. Journal of the Australian Mathematical Society **85**(02), 197 (October 2008). https://doi.org/10.1017/S1446788708000864
18. Holt, D.F., Rees, S., Röver, C.E., Thomas, R.M.: Groups with context-free co-word problem. Journal of the London Mathematical Society **71**(3), 643–657 (1999). https://doi.org/10.1112/S002461070500654X

19. Holt, D.F., Röver, C.E.: Groups with indexed co-word problem. International Journal of Algebra and Computation **16**(5), 985–1014 (October 2006). https://doi.org/10.1142/S0218196706003359
20. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison–Wesley, Reading, MA, 1st edn. (1979)
21. Howie, J.M.: Fundamentals of Semigroup Theory. No. 12 in London Mathematical Society Monographs: New Series, Clarendon Press, Oxford University Press, New York (1995)
22. Muller, D.E., Schupp, P.E.: Groups, the theory of ends, and context-free languages. Journal of Computer and System Sciences **26**(3), 295–310 (June 1983). https://doi.org/10.1016/0022-0000(83)90003-X
23. Robertson, E.F., Ruškuc, N., Wiegold, J.: Generators and relations of direct products of semigroups. Transactions of the American Mathematical Society **350**(07), 2665–2686 (July 1998). https://doi.org/10.1090/S0002-9947-98-02074-1