Accepted Manuscript

Decision problems for word-hyperbolic semigroups

Alan J. Cain, Markus Pfeiffer

PII: S0021-8693(16)30183-1
DOI: http://dx.doi.org/10.1016/j.jalgebra.2016.07.007
Reference: YJABR 15803

To appear in: *Journal of Algebra*

Received date: 8 September 2015

Decision problems for word-hyperbolic semigroups

Alan J. Cain[a,1,*], Markus Pfeiffer[b,2]

[a] *Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa, 2829–516 Caparica, Portugal*
[b] *School of Mathematics & Statistics, University of St Andrews North Haugh, St Andrews, Fife, KY16 9SX, United Kingdom*

Abstract

This paper studies decision problems for semigroups that are word-hyperbolic in the sense of Duncan & Gilman. A fundamental investigation reveals that the natural definition of a 'word-hyperbolic structure' has to be strengthened slightly in order to define a unique semigroup up to isomorphism. (This does not alter the class of word-hyperbolic semigroups.) The isomorphism problem is proven to be undecidable for word-hyperbolic semigroups (in contrast to the situation for word-hyperbolic groups). It is proved that it is undecidable whether a word-hyperbolic semigroup is automatic, asynchronously automatic, biautomatic, or asynchronously biautomatic. (These properties do not hold in general for word-hyperbolic semigroups.) It is proved that the uniform word problem for word-hyperbolic semigroups is solvable in polynomial time (improving on the previous exponential-time algorithm). Algorithms are presented for deciding whether a word-hyperbolic semigroup is a

*Corresponding author

Email addresses: `a.cain@fct.unl.pt` (Alan J. Cain), `markus.pfeiffer@st-andrews.ac.uk` (Markus Pfeiffer)

[1] Initially supported by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT (Fundação para a Ciência e a Tecnologia) under the project PEst-C/MAT/UI0144/2011 and through an FCT Ciência 2008 fellowship, and later supported by an FCT Investigador advanced fellowship (IF/01622/2013/CP1161/CT0001). This work was partially supported by FCT through the project UID/MAT/00297/2013 (Centro de Matemática e Aplicações). Part of the work described here was carried out during a visit by the first author to the University of St Andrews, which was funded by a London Mathematical Society Research in Pairs Grant (ref. 41410).

[2] Part of the work described here was carried out during a visit by the second author to the Universidade Nova de Lisboa, which was funded by an EPSRC Doctoral Prize 2012.

monoid, a group, a completely simple semigroup, a Clifford semigroup, or a free semigroup.

Keywords: Word-hyperbolic semigroups, decision problems, undecidability, isomorphism problem, context-free languages
2010 MSC: 20M35, 68Q45, 20F67

---

1. Introduction

The concept of word-hyperbolicity in groups, which has grown into one of the most fruitful areas of group theory since the publication of Gromov's seminal paper [1], admits a natural extension to monoids via using Gilman's characterization of word-hyperbolic groups using context-free languages [2], which generalizes directly to semigroups and monoids [3]. Informally, a word-hyperbolic structure for a semigroup consists of a regular language of representatives (not necessarily unique) for the elements of the semigroup, and a context-free language describing the multiplication table of the semigroup in terms of those representatives.

This generalization has led to a substantial amount of research on word-hyperbolic semigroups. For example, there have been studies of the relationship between the classes of rational, automatic, and word-hyperbolic semigroups [4, 5], which has shown that word-hyperbolic semigroups need not be automatic or even asynchronously automatic (in contrast to the situation for groups). Another approach has been to study interesting classes of word-hyperbolic semigroups: for example, Rees matrix semigroups over word-hyperbolic groups are word-hyperbolic [6], and semigroups (and monoids) presented by context-free complete rewriting systems are word-hyperbolic [7].

The computational aspect of word-hyperbolic semigroups has so far received limited attention. The only established result seems to be the solvablity of the word problem [4, Theorem 3.8]. In contrast, automatic semigroups, which generalize automatic groups [8] and whose study was inaugurated by Campbell et al. [9], have been studied from a computational perspective, with both decidability and undecidability results emerging [10, 11, 12].

This paper is devoted to some important decision problems for word-hyperbolic semigroups. Word-hyperbolic structures are not necessarily 'stronger' or 'weaker' computationally than automatic structures. As noted above, word-hyperbolicity does not imply automaticity for semigroups, so one cannot appeal to known results for automatic semigroups. A word-hyperbolic structure encodes the whole multiplication table for the semigroup, not just right-multiplication by generators (as is the case for automatic structures). On the other hand, context-free languages are computationally less pleasant than regular languages. For instance, an intersection of two context-free languages is not in general context-free, and indeed the emptiness of such an intersection cannot be decided algorithmically. Thus, in constructing algorithms for word-hyperbolic semigroups, it is often necessary to proceed via an indirect route, or use some unusual 'trick'.

Two of the most important results in this paper are the undecidability results in Section 4. First, the isomorphism problem for word-hyperbolic semigroups is undecidable, which contrasts the decidability of the isomorphism problem for hyperbolic groups [13, Theorem 1]. Second, it is undecidable whether a word-hyperbolic semigroup is automatic. (As noted above, for semigroups, word-hyperbolicity does not in general imply automaticity.)

Among the positive decidability results, the most important is that the uniform word problem for word-hyperbolic semigroups is soluble in polynomial time (Section 6). As remarked above, the word problem was already known to be solvable, but the previously-known algorithm required time exponential in the lengths of the input words [4, Theorem 3.8].

Some basic properties are then shown to be decidable (Section 7): being a monoid, Green's relations $\mathcal{L}$, $\mathcal{R}$, and $\mathcal{H}$, being a group, and commutativity. These results are not particularly difficult, but are worth noting.

The main body of the paper shows the decidability of more complicated algebraic properties: being completely simple (Section 8), being a Clifford semigroup (Section 9), and being a free semigroup (Section 10).

Before embarking on the discussion of decision problems, it is necessary to make a fundamental study of the notion of word-hyperbolicity, because the natural notion of a word-hyperbolic structure, or more precisely an 'interpretation' of a word-hyperbolic structure, does not determine a unique semigroup up to isomorphism. A slightly strengthened definition is needed, and this is the purpose of the preliminary Section 3.

The paper ends with a list of some open problems (Section 11).

2. Preliminaries

Throughout the paper, we assume basic knowledge of regular language and finite state automata, of context-free languages and pushdown automata, and of rational relations and transducers; see [14] and [15] for background reading. We also assume knowledge of the standard concepts and results about string-rewriting systems and their connection to semigroup presentations; see [16], and [17] for the necessary background.

We denote the empty word (over any alphabet) by ε. For an alphabet A, we denote by A^* the language of all words over A, and by A^+ the language of all non-empty words over A. The length of $u \in A^*$ is denoted $|u|$, and, for any $a \in A$. We denote by u^{rev} the reversal of a word u; that is, if $u = a_1 \cdots a_{n-1} a_n$ then $u^{\mathrm{rev}} = a_n a_{n-1} \cdots a_1$, with $a_i \in A$. We extend this notation to languages: for any language $L \subseteq A^*$, let $L^{\mathrm{rev}} = \{\, w^{\mathrm{rev}} : w \in L \,\}$.

If $\mathcal{R}$ is a relation on A^*, then $\mathcal{R}^\#$ denotes the congruence generated by $\mathcal{R}$. A presentation is a pair $\langle A \,|\, \mathcal{R} \rangle$ that defines [any semigroup isomorphic to] $A^+/\mathcal{R}^\#$.

3. The limits of interpretation

Before developing any algorithms for word-hyperbolic semigroups, we must clarify the relationship between a word-hyperbolic structure (that is, an abstract collection of certain languages) and a semigroup it describes. A similar study grounds the study of decision problems for automatic semigroups by Kambites & Otto [11], and our strategy and choice of terminology closely follows theirs.

Definition 3.1. A *pre-word-hyperbolic structure* Σ consists of:

- a finite alphabet $A(\Sigma)$;
- a regular language $L(\Sigma)$ over $A(\Sigma)$, not including the empty word;
- a context-free language $M(\Sigma)$ over $A(\Sigma) \cup \{\#_1, \#_2\}$, where $\#_1$ and $\#_2$ are new symbols not in $A(\Sigma)$, such that $M(\Sigma) \subseteq L(\Sigma)\#_1 L(\Sigma)\#_2 L(\Sigma)^{\mathrm{rev}}$.

When Σ is clear from the context, we may write A, L, and M instead of $A(\Sigma)$, $L(\Sigma)$, and $M(\Sigma)$, respectively.

The idea is that $A(\Sigma)$ will represent a set of generators for a semigroup, $L(\Sigma)$ will be a language of representatives for the elements of that semigroup, and $M(\Sigma)$ will describe the multiplication table for that semigroup in terms of the representatives in $L(\Sigma)$. However, a 'pre-word-hyperbolic structure' consists only of languages fulfilling certain basic properties: there is no mention of being a structure 'for a semigroup' in the definition. In particular, at this point there is nothing that guarantees $L(\Sigma)$ or $M(\Sigma)$ are non-empty. Or, $L(\Sigma)$ could be $A(\Sigma)^+$ and $M(\Sigma)$ could be the language $u\#_1v\#_2L(\Sigma)$ for some fixed $u, v \in L(\Sigma)$; clearly, $M(\Sigma)$ is very far from describing a multiplication table.

Now, following Kambites & Otto for automatic semigroups [11, § 2.2], let us attempt to turn the abstract pre-word-hyperbolic structure into something that describes a semigroup.

Definition 3.2. An *interpretation* of a pre-word-hyperbolic structure Σ with respect to a semigroup S is a homomorphism $\phi : A(\Sigma)^+ \to S$, with $\phi|_A$ being injective, such that $(L(\Sigma))\phi = S$ and

$$M(\Sigma) = \left\{ u\#_1v\#_2w^{\mathrm{rev}} : u, v, w \in L(\Sigma), (u\phi)(v\phi) = w\phi \right\}.$$

When there is no risk of confusion, denote $u\phi$ by $\overline{u}$ for any $u \in A^+$, and $X\phi$ by $\overline{X}$ for any $X \subseteq A^+$.

If a pre-word-hyperbolic structure Σ admits an interpretation with respect to a semigroup S, then Σ is a *word-hyperbolic structure* for S.

A semigroup is *word-hyperbolic* if it admits a word-hyperbolic structure.

This separation of the notion of a 'pre-word-hyperbolic structure' and an 'interpretation' is simply a reformulation (which is better for discussing decision problems) of the definition of Duncan & Gilman [3, Definitions 3.1 & 3.3], with the additional requirement that $\phi|_A$ is injective. This condition may seem like an odd, technical, restriction, and naïve definition of an interpretation would probably not include it. Yet the restriction is important, because without it the same pre-word-hyperbolic structure could admit interpretations with respect to two non-isomorphic semigroups (see Example 3.3 below). Note, however, that this restriction does not alter the class of semigroups under consideration: as an immediate consequence of Proposition 3.6 below, a semigroup is word-hyperbolic in the Duncan–Gilman sense if and only if it is word-hyperbolic in the sense of Definition 3.2.

Example 3.3. Let Σ be a pre-word-hyperbolic structure with $A = \{a, b, c\}$, $L = A$, and $M = \{ u\#_1v\#_2a^{\mathrm{rev}} : u, v \in L \}$. (Of course, $a^{\mathrm{rev}} = a$ since a is a single letter.)

Let S be the two-element null semigroup $\{0, x\}$, where all products are equal to 0. Let T be the three-element null semigroup $\{0, x, y\}$, again with all products equal to 0.

Define mappings $\phi : A \to S$ and $\psi : A \to T$ by

$$\begin{aligned} a\phi &= 0, & b\phi &= x, & c\phi &= x, \\ a\psi &= 0, & b\psi &= x, & c\psi &= y. \end{aligned}$$

Then $L\phi = S$ and $L\psi = T$. Furthermore,

$$\begin{aligned} M &= \{\, u\#_1 v\#_2 a^{\mathrm{rev}} : u, v \in L \,\} \\ &= \{\, u\#_1 v\#_2 a^{\mathrm{rev}} : u, v \in L, (u\phi)(v\phi) = a\phi \,\} \\ &= \{\, u\#_1 v\#_2 w^{\mathrm{rev}} : u, v, w \in L, (u\phi)(v\phi) = w\phi \,\}, \end{aligned}$$

since all products in S are equal to 0 and a is the unique word in L mapped to 0 by ϕ. Similarly,

$$M = \left\{\, u\#_1 v\#_2 w^{\mathrm{rev}} : u, v, w \in L, (u\psi)(v\psi) = w\psi \,\right\}$$

since all products in T are equal to 0 and a is the unique word in L in mapped to 0 by ψ.

Thus, except that their restrictions to A are non-injective, the maps ϕ and ψ satisfy the definition of interpretations of Σ with respect to the non-isomorphic semigroups S and T respectively.

Hence, without the restriction, it would not to make sense to consider decision problems for general word-hyperbolic semigroups, for it would be illogical to ask for an algorithm that took as input a word-hyperbolic structure and determined some property of 'the' semigroup is describes, since there is no such unique semigroup. However, as we shall now prove, the definition of interpretation in Definition 3.2 suffices to determine a unique semigroup up to isomorphism (see Proposition 3.5 below). We require the following lemma, which we will use again later:

Lemma 3.4. *Let Σ be a word-hyperbolic structure. Then there is a relation $E(\Sigma) \subseteq L(\Sigma) \times L(\Sigma)$, dependent only on Σ, such that the following are equivalent for any words $w, x \in L(\Sigma)$:*

1. $(w, x) \in E(\Sigma)$*;*
2. $w\phi = x\phi$ *for some semigroup S and interpretation $\phi : A(\Sigma)^+ \to S$;*

3. $w\phi = x\phi$ *for any semigroup* S *and interpretation* $\phi : A(\Sigma)^+ \to S$.

Proof. Define

$$E' = \big\{ (w, x) : w \in L(\Sigma), x \in L(\Sigma), |w| \geq |x|, |w| \geq 2, \\ (\exists u, v \in L(\Sigma))(u\#_1 v\#_2 w^{\mathrm{rev}} \in M(\Sigma) \wedge u\#_1 v\#_2 x^{\mathrm{rev}} \in M(\Sigma) \big\}$$

and let

$$E(\Sigma) = \big\{ (a, a) : a \in A(\Sigma) \cap L(\Sigma) \big\} \cup E' \cup (E')^{-1}. \tag{3.1}$$

The aim is now to show that $E(\Sigma)$ has the required properties. Let $w, x \in L(\Sigma)$.

First suppose that (1) holds; that is, that $(w, x) \in E(\Sigma)$. Let ϕ be any interpretation of Σ. Either $w, x \in A(\Sigma) \cap L(\Sigma)$, in which case $w = x$ by (3.1) and so $w\phi = x\phi$, or $(w, x) \in E' \cup (E')^{-1}$. Assume $(w, x) \in E'$; the other case is symmetrical. Then there exist $u, v \in L(\Sigma)$ such that $u\#_1 v\#_2 w^{\mathrm{rev}} \in M(\Sigma)$ and $u\#_1 v\#_2 x^{\mathrm{rev}} \in M(\Sigma)$. Hence $w\phi = (u\phi)(v\phi) = x\phi$ since ϕ is an interpretation of Σ. Hence (1) implies (3).

It is clear that (3) implies (2). Now suppose that (2) holds; that is, that $w\phi = x\phi$ for some interpretation $\phi : A(\Sigma)^+ \to S$. If $|w| = |x| = 1$, then $w, x \in A$ and so $w = x$ since $\phi|_A$ is injective, and so $(w, x) \in E(\Sigma)$ by (3.1). Now suppose that at least one of $|w|$ and $|x|$ is greater than 1. Assume $|w| \geq |x|$; the other case is similar. Since w has at least two letters, the element $w\phi$ is decomposable in S. So there are words $u, v \in L$ with $(u\phi)(v\phi) = w\phi$. Since $w\phi = x\phi$, it also follows that $(u\phi)(v\phi) = x\phi$. Thus the words $u\#_1 v\#_2 w^{\mathrm{rev}}$ and $u\#_1 v\#_1 x^{\mathrm{rev}}$ both lie in M since ϕ is an interpretation of Σ. Hence $(w, x) \in E' \subseteq E(\Sigma)$. Hence (2) implies (1). ☐

Proposition 3.5. *Let* Σ *be a word-hyperbolic structure admitting interpretations* $\phi : A^+ \to S$ *and* $\psi : A^+ \to T$. *Then there is an isomorphism* τ *from* S *to* T *such that* $\phi|_L \tau = \psi|_L$.

Proof. Define maps $\tau : S \to T$ and $\tau' : T \to S$ as follows. For any $s \in S$ let $s\tau$ be $w\psi$, where $w \in L$ is some word with $w\phi = s$, and for any $t \in T$, let $t\tau'$ be $w'\phi$, where $w' \in L$ is some word with $w'\psi = t$. (The words w and w' are guaranteed to exist since ϕ and ψ are surjections.) Let $E(\Sigma)$ be as in Lemma 3.4. To show that τ is well-defined, suppose $w_1, w_2 \in A^*$ are such that $s = w_1\phi = w_2\phi$. Then $(w_1, w_2) \in E(\Sigma)$ by Lemma 3.4 since ϕ is an interpretation, and so $w_1\psi = w_2\psi$ by Lemma 3.4 again since ψ is an interpretation. Hence τ is well-defined. Similarly, τ' is well-defined.

To show that τ is a homomorphism, proceed as follows. Let $r, s \in S$ and choose $u, v, w \in L$ with $u\phi = r$, $v\phi = s$, and $w\phi = rs$. Then $r\tau = u\psi$, $s\tau = v\psi$, and $(rs)\tau = w\psi$, by the definition of τ. Now, $u\#_1 v\#_2 w^{\text{rev}} \in M$ (since ϕ is an interpretation of Σ) and so $(u\psi)(v\psi) = w\psi$ (since ψ is an interpretation of Σ). Thus

$$(r\tau)(s\tau) = (u\psi)(v\psi) = (w\psi) = (rs)\tau$$

and so τ is a homomorphism.

Symmetric reasoning shows that $\tau' : T \to S$ is a homomorphism. The maps τ and τ' are mutually inverse, since if $w \in L$ is such that $w\phi = s$ and $w\psi = t$, then $s\tau = t$ and $\tau' = s$. Thus $\tau : S \to T$ is an isomorphism. By the definition of τ using elements of L, it follows that $\phi|_L\tau = \psi|_L$. □

Proposition 3.5 shows that it makes sense to attempt to solve questions about a semigroup using the word-hyperbolic structure describing it.

The following result shows that the requirement that an interpretation be an injection when restricted to the alphabet A does not restrict the class of word-hyperbolic semigroups:

Proposition 3.6. *Let Σ be a pre-word-hyperbolic structure and let $\phi : A(\Sigma)^+ \to S$ be a map fulfilling the definition of an interpretation for Σ with respect to a semigroup S, except that $\phi|_{A(\Sigma)}$ is not requried to be injective. Then there is a word-hyperbolic structure Π, effectively computable from Σ and $\phi|_{A(\Sigma)}$, with $A(\Pi) \subseteq A(\Sigma)$, admitting an interpretation $\psi : A(\Pi)^+ \to S$.*

Proof. Initially, let $\Pi = \Sigma$. We will modify Π until it has the desired property.

Suppose $\phi|_{A(\Pi)}$ is not injective. Pick $a, b \in A(\Pi)$ with $a\phi = b\phi$. Replace every instance of b by a in words in $L(\Pi)$. (This corresponds to replacing b by a whenever it appears as a label on an edge in a finite automaton recognizing $L(\Pi)$.) Replace every instance of b by a in words in $M(\Pi)$. (This corresponds to replacing b by a whenever it appears as non-terminal in a context-free grammar defining $L(\Pi)$.) Finally, delete b from $A(\Pi)$. Since $a\phi = b\phi$, it follows that Π is a word-hyperbolic structure admitting an interpretation $\phi|_{A(\Pi)^+} : A(\Pi)^+ \to S$ with respect to S.

Since $A(\Pi)$ is finite, we can iterate this process until $\phi|_{A(\Pi)}$ becomes injective. Finally, define $\psi = \phi|_{A(\Pi)^+}$. □

However, although a word-hyperbolic structure determines a unique semigroup, it does not determine a unique interpretation, even up to automorphic permutation. This parallels the situation for automatic semigroups

[11, § 2.2], but is also true in a rather vacuous sense for word-hyperbolic semigroups, for the alphabet $A(\Sigma)$ for a word-hyperbolic structure Σ for a semigroup S may include a symbol c that does not appear in any word in either $L(\Sigma)$ or $M(\Sigma)$. (In this situation, c must represent a redundant generator for S.) For example, let $A(\Sigma) = \{a, b, c\}$, $L(\Sigma) = \{a, b\}^+$, and $M(\Sigma) = \{\, u\#_1 v\#_1 (uv)^{\mathrm{rev}} : u, v \in \{a, b\}^+ \,\}$. Then Σ is a word-hyperbolic structure for the free semigroup F with basis $\{x, y\}$: let $\phi : A(\Sigma)^+ \to F$ be such that $a\phi = x$ and $b\phi = y$; regardless of how $c\phi$ is defined, ϕ is an interpretation of Σ with respect to F.

Less trivial is the following example, showing that non-uniqueness of interpretation can arise even when all the symbols in $A(\Sigma)$ appear in $L(\Sigma)$ and $M(\Sigma)$:

Example 3.7. Let $S = (\{1, 2\} \times \{1, 2, 3\}) \cup \{0_S, 1_S\}$ and define multiplication on S by

$$
\begin{aligned}
(i, \lambda)(j, \mu) &= \begin{cases} 0_S & \text{if } \lambda = j = 1, \\ (i, \mu) & \text{otherwise;} \end{cases} \\
1_S x = x 1_S &= x \quad \text{for all } x \in S; \\
0_S x = x 0_S &= 0_S \quad \text{for all } x \in S.
\end{aligned}
$$

Then S is a monoid. (In fact, S is a monoid formed by adjoining an identity to a 0-Rees matrix semigroup over the trivial group.)

Let $A(\Sigma) = \{a, b, c, d, e, i, z\}$. Let $L(\Sigma) = \{a, b, c, d, ced, dec, i, z\}$. Define

$$
\begin{array}{lllll}
\phi_1 : A(\Sigma)^+ \to S & a\phi_1 = (1, 1), & b\phi_1 = (2, 1), & c\phi_1 = (1, 2), & \\
 & d\phi_1 = (2, 3), & e\phi_1 = (2, 2), & i\phi_1 = 1_S, & z\phi_1 = 0_S; \\
\phi_2 : A(\Sigma)^+ \to S & a\phi_2 = (1, 1), & b\phi_2 = (2, 1), & c\phi_2 = (1, 2), & \\
 & d\phi_2 = (2, 3), & e\phi_2 = (1, 3), & i\phi_1 = 1_S, & z\phi_2 = 0_S.
\end{array}
$$

Notice that the only difference in the definitions of ϕ_1 and ϕ_2 is the image of the symbol e. Furthermore,

$$
\begin{aligned}
(ced)\phi_1 = (1, 2)(2, 2)(2, 3) = (1, 3) = (1, 2)(1, 3)(2, 3) = (ced)\phi_2, \\
(dec)\phi_1 = (2, 3)(2, 2)(1, 2) = (2, 2) = (2, 3)(1, 3)(1, 2) = (dec)\phi_2;
\end{aligned}
$$

thus $\phi_1|_L = \phi_2|_L$ and $L\phi_1 = L\phi_2 = S$.

Define

$$M(\Sigma) = \{\, u\#_1 v\#_2 w^{\text{rev}} : u, v, w \in L(\Sigma), (u\phi_1)(v\phi_1) = w\phi_1 \,\}.$$

Since $L(\Sigma)$ is finite, $M(\Sigma)$ is also finite and thus context-free. So Σ is a word-hyperbolic structure for S and ϕ_1 is an interpretation for Σ with respect to S. Furthermore, since Hence $\phi_1|_L = \phi_2|_L$,

$$M(\Sigma) = \{\, u\#_1 v\#_2 w^{\text{rev}} : u, v, w \in L(\Sigma), (u\phi_2)(v\phi_2) = w\phi_2 \,\},$$

and so ϕ_2 is also an interpretation of Σ with respect to S.

Moreover, there is no automorphism ρ of S such that $\phi_1\rho = \phi_2$. To see this, notice that such a ρ would have to map $e\phi_1 = (2,1)$ to $e\phi_2 = (1,3)$. The map ρ would also preserve $\mathcal{R}$-classes. But the $\mathcal{R}$-class of $(1,3)$ contains the element $(1,1)$, which is not idempotent (since $(1,1)(1,1) = 0$), whereas every element of the $\mathcal{R}$-class of $(2,1)$ is idempotent. So no such map ρ can exist. So the two interpretations are not even equivalent up to automorphic permutation of S.

The crucial point in Example 3.7 is that Proposition 3.5 only guarantees that the *restriction* of two interpretations to L are equivalent up to automorphic permutation. It says nothing about the interpretation maps on the whole of A^+.

The next result essentially shows that word-hyperbolicity is invariant under change of finite generating set. This result, without mention of interpretations, is essentially due to Hoffmann et al. [4, Proposition 4.2]. We state it here using the more precise definitions of the present paper; the proof is a straightforward extention of that of Hoffmann et al.:

Proposition 3.8. *Let S be a word-hyperbolic semigroup. Let $X \subseteq S$ be a finite generating set for S. Then there is a word-hyperbolic structure Σ for S with an interpretation $\phi : A(\Sigma)^+ \to S$ such that $(A(\Sigma))\phi = X$.*

In order to compute with the semigroup described by a word-hyperbolic structure, interpretations must be coded in a finite way.

Definition 3.9. An *assignment of generators* for a word-hyperbolic structure Σ is a map $\alpha : A(\Sigma) \to L(\Sigma)$ with the property that there is some interpretation $\phi : A(\Sigma)^+ \to S$ such that $a\alpha\phi = a\phi$ for all $a \in A$; such an interpretation is said to be *consistent* with α. Two assignments of generators α and β for Σ are *equivalent* if $(a\alpha, a\beta) \in E(\Sigma)$ for all $a \in A(\Sigma)$.

Proposition 3.10. *An assignment of generators for a word-hyperbolic structure is consistent with a unique interpretation (up to automorphic permutation of the semigroup described). Equivalent assignments of generators are consistent with the same interpretation.*

Conversely, every interpretation is consistent with a unique (up to equivalence) assignment of generators.

Proof. Let Σ be a word-hyperbolic structure and $\alpha : A \to L$ an assignment of generators. Then there is an interpretation $\phi : A^+ \to S$ of Σ that is consistent with α; that is, $a\alpha\phi = a\phi$ for all $a \in A$.

Let $\psi : A^+ \to S$ be another interpretation of Σ that is consistent with α; the aim is to show that ϕ and ϕ differ only by an automorphic permutation of S. First, $a\alpha\psi = a\psi$ for all $a \in A$, since ψ is consistent with α. By Proposition 3.5, there is an automorphism τ of S such that $\phi|_L\tau = \psi|_L$, and so $a\phi\tau = a\alpha\phi\tau = a\alpha\psi = a\psi$ for all $a \in A$. Hence ϕ and ψ differ only by the automorphism τ.

Now let $\beta : A \to L$ be an assignment of generators equivalent to α; the aim is to show that β is also consistent with ϕ. Now, $(a\alpha, a\beta) \in E(\Sigma)$ for all $a \in A$ since α and β are equivalent. Thus $a\phi = a\alpha\phi = a\beta\phi$ by Lemma 3.4, and hence β is also consistent with the interpretation ϕ.

Finally, suppose $\gamma : A \to L$ is an assignment of generators consistent with ϕ; the aim is to show α and β are equivalent. Now, $a\alpha\phi = a\phi = a\gamma\phi$ for all $a \in A$ since α and γ are consistent with ϕ. Hence $(a\alpha, a\gamma) \in E(\Sigma)$ for all $a \in A$ by Lemma 3.4. ☐

Definition 3.11. A word-hyperbolic structure Σ is said to be an *interpreted* word-hyperbolic structure if it is equipped with an assigment of generators $\alpha(\Sigma)$.

Just as in the case of automatic semigroups [11, Question 2.5], it is not known whether one can compute an assignment of generators from a word-hyperbolic structure. However, when we know an assignment of generators, we can transform the word-hyperbolic structure to one that includes the generating symbols in the language of normal forms:

Proposition 3.12. *Let Σ be an interpreted word-hyperbolic structure for a semigroup S. Then there is another interpreted word-hyperbolic structure Σ' for S, effectively computable from Σ, such that $A(\Sigma') \subseteq L(\Sigma')$ and $\alpha(\Sigma')$ is the embedding map from $A(\Sigma')$ to $L(\Sigma')$.*

Proof. Let $A(\Sigma')$ be $A(\Sigma)$ and $L(\Sigma') = L(\Sigma) \cup A(\Sigma)$. For brevity, let $\alpha = \alpha(\Sigma)$. For each $a, b, c \in A(\Sigma')$, define the languages:

$$\begin{aligned}
M_a^{(1)} &= \{\, a\#_1 v\#_2 w^{\mathrm{rev}} : (a\alpha)\#_1 v\#_2 w^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_a^{(2)} &= \{\, u\#_1 a\#_2 w^{\mathrm{rev}} : u\#_1 (a\alpha)\#_2 w^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_a^{(3)} &= \{\, u\#_1 v\#_2 a^{\mathrm{rev}} : u\#_1 v\#_2 (a\alpha)^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_{a,b}^{(4)} &= \{\, a\#_1 b\#_2 w^{\mathrm{rev}} : (a\alpha)\#_1 (b\alpha)\#_2 w^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_{a,b}^{(5)} &= \{\, u\#_1 a\#_2 b^{\mathrm{rev}} : u\#_1 (a\alpha)\#_2 (b\alpha)^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_{a,b}^{(6)} &= \{\, a\#_1 v\#_2 b^{\mathrm{rev}} : (a\alpha)\#_1 v\#_2 (b\alpha)^{\mathrm{rev}} \in M(\Sigma) \,\},\\
M_{a,b,c}^{(7)} &= \{\, a\#_1 b\#_2 c^{\mathrm{rev}} : (a\alpha)\#_1 (b\alpha)\#_2 (c\alpha)^{\mathrm{rev}} \in M(\Sigma) \,\}.
\end{aligned}$$

Each of these languages is context-free because each is the intersection of the context-free language $M(\Sigma)$ with a regular language. [Notice that $M_{a,b,c}^{(7)}$ is either empty or a singleton language.]

Now let

$$\begin{aligned}
M(\Sigma') = M(\Sigma) &\cup \bigcup_{a \in A(\Sigma)} \left(M_a^{(1)} \cup M_b^{(2)} \cup M_c^{(3)} \right)\\
&\cup \bigcup_{a,b \in A(\Sigma)} \left(M_{a,b}^{(4)} \cup M_{a,b}^{(5)} \cup M_{a,b}^{(6)} \right)\\
&\cup \bigcup_{a,b,c \in A(\Sigma)} M_{a,b,c}^{(7)};
\end{aligned}$$

notice that $M(\Sigma')$ is also context-free.

Let $\phi : A(\Sigma) \to S$ be an interpretation of Σ. Then, recalling that $A(\Sigma) = A(\Sigma')$,

$$M(\Sigma') = \{\, u\#_1 v\#_2 w^{\mathrm{rev}} : u, v, w \in L(\Sigma'), (u\phi)(v\phi) = w\phi \,\},$$

because u, v, and w range over $L(\Sigma') = L(\Sigma) \cup A(\Sigma)$, and the eight cases that arise depending on whether each word lies in $L(\Sigma)$ or $A(\Sigma)$ correspond to the eight sets $M(\Sigma)$, $M_a^{(1)}$, $M_a^{(2)}$, $M_a^{(3)}$, $M_{a,b}^{(4)}$, $M_{a,b}^{(5)}$, $M_{a,b}^{(6)}$, and $M_{a,b,c}^{(7)}$. [Notice that these sets are not necessarily disjoint, since it is possible that $a\alpha = a$ for some $a \in A$.]

Finally, define $\alpha(\Sigma')$ to be the embedding map from $A(\Sigma')$ to $L(\Sigma')$. This map is an assignment of generators since trivially $((a)\alpha(\Sigma'))\phi = a\phi$ for any interpretation ϕ of Σ'. ☐

In light of Proposition 3.12, we will assume without further comment that an interpreted word-hyperbolic structure Σ has the property that $A(\Sigma) \subseteq L(\Sigma)$ and that $\alpha(\Sigma)$ is the embedding map from $A(\Sigma)$ to $L(\Sigma)$. Notice further that the computational effectiveness aspect of Proposition 3.12 ensures we are free to assume that an interpreted word-hyperbolic structure serving as input to a decision problem has this property.

For automatic semigroups, it is possible to assume that the automatic structure has a further pleasant property, namely that every element of the semigroup is represented by a unique word in the language of representatives [11, Proposition 2.9(iii)]. However, there exist word-hyperbolic semigroups (indeed, word-hyperbolic monoids) that do not admit word-hyperbolic structures where the languages of representatives have this uniqueness property [7, Examples 10 & 11].

4. Isomorphism problem & automaticity

This section proves that the isomorphism problem, and the problem of deciding automaticity, are both undecidable for word-hyperbolic semigroups. Recall that, as noted in the introduction, a word-hyperbolic monoid is not necessarily automatic or asynchronously automatic [4, Example 7.7 et seq.].

We briefly recap the definitions of automaticity and biautomaticity; for further background on automaticity see [9]; for asynchronous automaticity, see [4]; for biautomaticity, see the study by Hoffman & Thomas [18]

Definition 4.1. Let A be an alphabet and let \$ be a new symbol not in A. Define the mapping $\lrcorner^{\$} : A^* \times A^* \to ((A \cup \{\$\}) \times (A \cup \{\$\}))^*$ by

$$(u_1 \cdots u_m, v_1 \cdots v_n) \mapsto \begin{cases} (u_1, v_1) \cdots (u_m, v_n) & \text{if } m = n, \\ (u_1, v_1) \cdots (u_n, v_n)(u_{n+1}, \$) \cdots (u_m, \$) & \text{if } m > n, \\ (u_1, v_1) \cdots (u_m, v_m)(\$, v_{m+1}) \cdots (\$, v_n) & \text{if } m < n, \end{cases}$$

and the mapping ${}^{\$}\lrcorner : A^* \times A^* \to ((A \cup \{\$\}) \times (A \cup \{\$\}))^*$ by

$$(u_1 \cdots u_m, v_1 \cdots v_n) \mapsto \begin{cases} (u_1, v_1) \cdots (u_m, v_n) & \text{if } m = n, \\ (u_1, \$) \cdots (u_{m-n}, \$)(u_{m-n+1}, v_1) \cdots (u_m, v_n) & \text{if } m > n, \\ (\$, v_1) \cdots (\$, v_{n-m})(u_1, v_{n-m+1}) \cdots (u_m, v_n) & \text{if } m < n, \end{cases}$$

where $u_i, v_i \in A$.

Definition 4.2. Let M be a monoid. Let A be a finite alphabet representing a set of generators for M and let $L \subseteq A^*$ be a regular language such that every element of M has at least one representative in L. For each $a \in A \cup \{\varepsilon\}$, define the relations

$$L_a = \{ (u, v) : u, v \in L, \overline{ua} = \overline{v} \}$$
$$_aL = \{ (u, v) : u, v \in L, \overline{au} = \overline{v} \}.$$

The pair (A, L) is an *automatic structure* for M if $L_a^{\$}$ is a regular language over $(A \cup \{\$\}) \times (A \cup \{\$\})$ for all $a \in A \cup \{\varepsilon\}$. A monoid M is *automatic* if it admits an automatic structure with respect to some generating set.

The pair (A, L) is an *asynchronous automatic structure* for M if L_a is a rational relation for all $a \in A \cup \{\varepsilon\}$. A monoid M is *asychronously automatic* if it admits an asynchronous automatic structure with respect to some generating set.

The pair (A, L) is a *biautomatic structure* for M if $L_a^{\$}$, ${}_aL^{\$}$, ${}^{\$}L_a$, and ${}_a^{\$}L$ are regular languages over $(A \cup \{\$\}) \times (A \cup \{\$\})$ for all $a \in A \cup \{\varepsilon\}$. A monoid M is *biautomatic* if it admits a biautomatic structure with respect to some generating set.

The pair (A, L) is an *asynchronous biautomatic structure* for M if L_a and ${}_aL$ are rational relations for all $a \in A \cup \{\varepsilon\}$. A monoid M is *asynchronously biautomatic* if it admits an asynchronous biautomatic structure with respect to some generating set.

Unlike in the situation for groups, biautomaticity and automaticity for semigroups is dependent on the choice of generating set [9, Example 4.5]. However, for monoids, biautomaticity and automaticity are independent of the choice of *semigroup* generating sets [19, Theorem 1.1].

Note that biautomaticity implies automaticity and asynchronous biautomaticity, and both of these properties imply asynchronous automaticity.

Hoffmann & Thomas have made a careful study of biautomaticity for semigroups [18]. They distinguish four notions of biautomaticity for semigroups, which are all equivalent for groups and more generally for cancellative semigroups [18, Theorem 1] but distinct for semigroups [18, Remark 1 & § 4]. In the sense used in this paper, 'biautomaticity' implies *all four* of the Hoffmann–Thomas notions of biautomaticity.

We also recall some less commly-used terms from the theory of rewriting systems; see [16] 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.

The key to encoding undecidability results into decision problems for word-hyperbolic semigroups is the following result, due to the first author and Maltcev:

Theorem 4.3 ([7, Theorem 3.1])**.** *Let $(A, \mathcal{R})$ be a confluent context-free monadic rewriting system where $\mathcal{R}$ does not contain rewriting rules with ε on the right-hand side. Then there is an intepreted word-hyperbolic structure Σ for the semigroup presented by $\langle A \,|\, \mathcal{R}\rangle$ such that $A(\Sigma) = A$ and $L(\Sigma) = A^*$. Furthermore, Σ can be effectively constructed from context-free grammars describing $\mathcal{R}$.*

(The preceding result was originally stated for monoids, allowing $\mathcal{R}$ to contain rules with ε on the right-hand side; it is immediate that it holds in this form for semigroups. The 'effective construction' part follows easily by inspecting the construction of the word-hyperbolic structure in the proof.)

Lemma 4.4. *Let Γ be a context-free grammar over a finite alphabet B. Let x, y, z be new symbols not in B and let $A = B \cup \{x, y, z\}$. Define $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{Z}$ as follows:*

$$\begin{aligned} \mathcal{R}_1 &= \big\{\, xwy \to z : w \in L(\Gamma) \,\big\}, \\ \mathcal{R}_2 &= \big\{\, xwy \to z : w \in B^* \,\big\}, \\ \mathcal{Z} &= \big\{\, az \to z, za \to z : a \in A \,\big\}. \end{aligned}$$

Let S_1 and S_2 be the semigroups presented by, respectively, $\langle A \,|\, \mathcal{R}_1 \cup \mathcal{Z}\rangle$ and $\langle A \,|\, \mathcal{R}_2 \cup \mathcal{Z}\rangle$. Then:

1. *S_1 and S_2 are word-hyperbolic, with effectively computable word-hyperbolic structures.*
2. *$L(\Gamma) = B^*$ if and only if S_1 and S_2 are isomorphic.*
3. *The following are equivalent:*
   (a) *$L(\Gamma)$ is regular;*
   (b) *S_1 is biautomatic;*
   (c) *S_1 is asynchronously biautomatic;*

(d) S_1 *is automatic;*
(e) S_1 *is asynchronously automatic.*

Proof. 1. Notice first that $(A, \mathcal{R}_1 \cup \mathcal{Z})$ is a context-free monadic rewriting system. It is confluent because any rewriting must produce a symbol z, and so the entire word rewrites to z using rewriting rules in $\mathcal{Z}$. Similarly, $(A, \mathcal{R}_2 \cup \mathcal{Z})$ is a confluent context-free monadic rewriting system. By Theorem 4.3, S_1 and S_2 have effectively computable word-hyperbolic structures Σ_1 and Σ_2 such that $A(\Sigma_1) = A$ and $A(\Sigma_2) = A$. Let $\phi_1 : A(\Sigma_1)^+ \to S_1$ and $\phi_2 : A(\Sigma_2)^+ \to S_2$ be interpretations of Σ_1 and Σ_2.

2. Suppose that $L(\Gamma) = B^*$. Then $\mathcal{R}_1 = \mathcal{R}_2$ and so S_1 and S_2 are isomorphic.
Now suppose S_1 and S_2 are isomorphic. Let $\tau : S_1 \to S_2$ be an isomorphism. Now, $z\phi_1$ and $z\phi_2$ are the unique zeroes of S_1 of S_2, so τ must map $z\phi_1$ to $z\phi_2$. Furthermore, for $i \in \{1, 2\}$, the element $x\phi_i$ is the unique indecomposable element of S_i such that there exists an element $q_i \in S_i$ with $q_i \neq z\phi_i$ and $(x\phi_i)q_i = z\phi_i$. Hence τ maps $x\phi_1$ to $x\phi_2$. Similarly, τ maps $y\phi_1$ and $y\phi_2$. Since all elements of $B\phi_1$ and $B\phi_2$ are indecomposable, τ must map $B\phi_1$ to $B\phi_1$ and thus τ restricts to an isomorphism between the free subsemigroups $B^+\phi_1$ and $B^+\phi_2$.
Suppose, with the aim of obtaining a contradiction, that $L(\Gamma) \subsetneq B^*$. Let $u \in B^* \setminus L(\Gamma)$ and let $v \in B^*$ be such that $u\phi_1\tau = v\phi_2$. Then $(x\phi_1)(u\phi_1)(y\phi_1) = (xuy)\phi_1 \neq z\phi_1$ since no rewriting rule in $\mathcal{R}_1 \cup \mathcal{Z}$ can be applied to xuy. But $(x\phi_2\tau)(u\phi_2\tau)(y\phi_2\tau) = (xvy)\phi_2 = z\phi_2 = z\phi_1\tau$ by the rules in $\mathcal{R}_2 \cup \mathcal{Z}$, which contradicts τ being an isomorphism. Hence $L(\Gamma) = B^*$.
Thus $L(\Gamma) = B^*$ if and only if S_1 and S_2 are isomorphic.

3. Suppose $L(\Gamma)$ is regular. Then $(A, \mathcal{R}_1 \cup \mathcal{Z})$ is a regular monadic rewriting system, and is confluent by the reasoning in the proof of part 1. Let L be the language of normal forms for $(A, \mathcal{R}_1 \cup \mathcal{Z})$; it is easy to see that

$$L = (A - \{z\})^+ - A^* x L(\Gamma) y A^* \cup \{z\},$$

and that

$$\begin{aligned}
L_\varepsilon = {}_\varepsilon L &= \{\, (u,u) : u \in L \,\} \\
L_a &= \{\, (u,ua) : u \in L \,\} \qquad\qquad \text{for all } a \in A - \{y,z\} \\
L_y &= \{\, (u,uy) : u \in L - A^*xL(\Gamma) \,\} \cup \{\, (u,z) : u \in A^*xL(\Gamma) \,\} \\
L_z &= \{\, (u,z) : u \in L \,\}; \\
{}_aL &= \{\, (au,u) : u \in L \,\} \qquad\qquad \text{for all } a \in A - \{x,z\} \\
{}_xL &= \{\, (xu,u) : u \in L - L(\Gamma)yA^* \,\} \cup \{\, (u,z) : u \in L(\Gamma)yA^* \,\} \\
{}_zL &= \{\, (u,z) : u \in L \,\}.
\end{aligned}$$

It is easy to see that $L_a^\$$, ${}^\$L_a$, ${}^\$_aL$ and ${}_aL^\$$ are all regular for all $a \in A \cup \{\varepsilon\}$. Thus (A, L) is a biautomatic structure for S_1. Thus a) implies b).

It is clear that b) implies c) and d), and both c) and d) imply e).

So suppose S_1 is asynchronously automatic. By [4, Proposition 4.1], S_1 admits an asynchronous automatic structure (A, L). If $w \in B^+ - L(\Gamma)$, then xwy is the unique word over A representing $\overline{xwy} \in S_1$ (since no relation in $\mathcal{R}_1 \cup \mathcal{Z}$ can be applied to xwy). Hence $x(B^+ - L(\Gamma))y \subseteq L$. Note also that the language K of words in L representing $\overline{z}$ is regular, since

$$K = \{\, u \in L : (u, z) \in L_\varepsilon \,\}.$$

Since xwy represents $\overline{z}$ if and only if $w \in L(\Gamma)$, it follows that $x(B^+ - L(\Gamma))y = xB^+y - K$. So $x(B^+ - L(\Gamma))y$ is regular, and thus $L(\Gamma) = B^+ - x^{-1}(x(B^+ - L(\Gamma)y)y^{-1}$ is regular. Thus e) implies a). □

The two undecidability results can now be deduced from the preceding lemma:

Theorem 4.5. *The isomorphism problem is undecidable for word-hyperbolic semigroups. That is, there is no algorithm that takes as input two interpreted word-hyperbolic structures Σ_1 and Σ_2 for semigroups S_1 and S_2 and decides whether S_1 and S_2 are isomorphic.*

Proof. Since there is no algorithm that takes a context-free grammar Γ and decides whether $L(\Gamma) = B^*$ by [14, Theorem 8.11], it follows from Lemma 4.4(1,2) that there is no algorithm that takes two interpreted word-hyperbolic structures and decides whether the semigroups they define are isomorphic. □

Theorem 4.6. *It is undecidable whether a word-hyperbolic semigroup is automatic (respectively, asynchronously automatic, biautomatic, asynchronously biautomatic). That is, there is no algorithm that takes as input an interpreted word-hyperbolic structure for a semigroup S decides whether S is automatic (respectively, asynchronously automatic, biautomatic, asynchronously biautomatic).*

Proof. Since there is no algorithm that takes a context-free grammar Γ and decides whether $L(\Gamma)$ is regular [14, Theorem 8.15], it follows from Lemma 4.4(1,3) that there is no algorithm that takes as input an interpreted word-hyperbolic structure and decides whether the semigroup it defines is automatic (respectively, asynchronously automatic, biautomatic, asynchronously biautomatic). ☐

5. Basic calculations

This section notes a few very basic facts about computing with word-hyperbolic structures for semigroups that are used later in the paper.

Lemma 5.1 ([4, Lemma 3.6 & its proof])**.** *There is an algorithm that takes as input a word-hyperbolic structure Σ for a semigroup, with $M(L)$ being specified by a context-free grammar in quadratic Greibach normal form, and two words $p, q \in L(\Sigma)$, and outputs a word $r \in L(\Sigma)$ satisfying $\overline{p}\,\overline{q} = \overline{r}$ with $|r| \leq c(|p| + |q|)$ (where c is a constant dependent only on Σ) in time $\mathcal{O}((|p| + |q|)^5)$.*

(Actually, the appearance of this lemma in [4] allows p or q to be empty and asserts that $|r| \leq c(|p| + |q| + 2)$. To obtain the lemma above, where p and q are non-empty, increase c appropriately. Notice that there may be many possibilities for a word r with $\overline{p}\,\overline{q} = \overline{r}$.)

Lemma 5.2. *There is an algorithm that takes as input a word-hyperbolic structure Σ for a semigroup and three words $p, q, r \in L(\Sigma)$, and decides whether $\overline{p}\,\overline{q} = \overline{r}$ in time $\mathcal{O}((|p| + |q| + |r|)^3)$.*

Proof. The algorithm simply checks whether $p\#_1 q\#_2 r^{\mathrm{rev}} \in M(\Sigma)$, and the membership problem for arbitrary context-free languages is soluble in cubic time [20]. ☐

6. Word problem

This section is dedicated to proving that the uniform word problem for word-hyperbolic semigroups is soluble in polynomial time.

As noted in the introduction, the previously-known algorithm required exponential time [4, Theorem 3.8]. This motivated Hoffmann & Thomas to define a narrower notion of word-hyperbolicity for monoids that still generalizes word-hyperbolicity for groups. By restricting to this version of word-hyperbolicity, one recovers automaticity [5, Theorem 3] and an algorithm that runs in time $\mathcal{O}(n \log n)$, where n is the length of the input words [5, Theorem 2]. Although the algorithm described below is not as efficient as this, the existence of a polynomial-time solution to the word problem for word-hyperbolic monoids (in the original Duncan–Gilman sense) diminishes the appeal of the Hoffmann–Thomas restricted version.

Recall that for a context-free grammar Γ, the *size* of Γ, denoted $|\Gamma|$, is the sum of the lengths of the right-hand sides of the productions in P.

Theorem 6.1. *There is an algorithm that takes as input an interpreted word-hyperbolic structure Σ for a semigroup, where $M(\Sigma)$ is defined by a context-free grammar Γ, and two words $w, w' \in A(\Sigma)^+$ and determines whether $\overline{w} = \overline{w'}$ in time polynomial in $|w| + |w'|$ and $|\Gamma|$. More succinctly, the uniform word problem for word-hyperbolic semigroups is soluble in polynomial time.*

Proof. By interchanging w and w' if necessary, assume that $|w| \geq |w'|$. First, if $|w| = |w'| = 1$, then $w, w' \in A(\Sigma)$ and so (since the interpretation map is injective on $A(\Sigma)$), we have $\overline{w} = \overline{w'}$ if and only if $w = w'$.

So assume $|w| \geq 2$. Factorize w as $w = w^{(1)}w^{(2)}$, where $w^{(1)} = \lfloor |w|/2 \rfloor$. Notice that $\overline{w} = \overline{w'}$ if and only if $\overline{w^{(1)}}\,\overline{w^{(2)}} = \overline{w'}$.

By Lemma 6.2 below, there is an algorithm that takes the three words $w^{(1)}$, $w^{(2)}$, and w', and the word-hyperbolic structure Σ, and yields words $u^{(1)}$, $u^{(2)}$, and u' in $L(\Sigma)$ representing $\overline{w^{(1)}}$, $\overline{w^{(2)}}$, and $\overline{w'}$, of lengths at most $(c+1)|w^{(1)}|^{1+\log(c+1)}$, $(c+1)|w^{(2)}|^{1+\log(c+1)}$ and $(c+1)|w'|^{1+\log(c+1)}$, respectively, where c is a constant dependent only on Σ, in time polynomial in $|w^{(1)}| + |w^{(2)}| + |w'|$ and $|\Gamma|$.

It follows that $\overline{w} = \overline{w'}$ if and only if $\overline{u^{(1)}}\,\overline{u^{(2)}} = \overline{u'}$, and, by Lemma 5.2, this can be checked in time cubic in $|u^{(1)}| + |u^{(2)}| + |u'|$, which, by the bounds on the lengths of $u^{(1)}$, $u^{(2)}$, and u', is still polynomial in the lengths of w and w'. Thus the word problem for the semigroup described by Σ is soluble in polynomial time. ∎

Lemma 6.2. *There is an algorithm that takes as input an interpreted word-hyperbolic structure Σ for a semigroup, where $M(\Sigma)$ is defined by a context-free grammar Γ, and a word $w \in A(\Sigma)^+$ and outputs a word $u \in L(\Sigma)$ with $\overline{w} = \overline{u}$ and $|u| \leq |w|(c+1)|w|^{\log(c+1)}$ (where c is a constant dependent only on Σ), and which takes time polynomial in $|w|$ and $|\Gamma|$.*

Proof. The first step is to convert Γ to a quadratic Greibach normal form grammar, so that Lemma 5.1 can be applied. This takes time $\mathcal{O}(|\Gamma|)^2$ by Lemma 6.3 below.

Suppose $w = w_1 \cdots w_n$, where $w_i \in A \subseteq L$. Therefore $w_1, \ldots, w_n$ is a sequence of words in L whose concatenation represents the same element of the semigroup as w.

For the purposes of this proof, the *total length* of a sequence $s_1, \ldots, s_\ell$ of words in A^* is defined to be the sum of the lengths of the words $|s_1|+\ldots+|s_\ell|$.

Consider the following computation, which will form the iterative step of the algorithm: suppose there is a sequence of words $s_1, \ldots, s_\ell$, each lying in $L(\Sigma)$ and each of length at most t. Notice that ℓt is an upper bound for the total length of this sequence. For $i = 1, \ldots, \lfloor \ell/2 \rfloor$, apply Lemma 5.1 to compute a word $s'_i \in L(\Sigma)$ representing $\overline{s_{2i-1}s_{2i}}$ of length at most $c(|s_{2i-1}|+|s_{2i}|) \leq 2ct$. For each $i = 1, \ldots, \lfloor \ell/2 \rfloor$, this takes $\mathcal{O}((|s_{2i-1}|+|s_{2i}|)^5)$ time, which is at worst $\mathcal{O}((2t)^5)$ time. Therefore the total time used is at most $\mathcal{O}(\lfloor \ell/2 \rfloor (2t)^5)$, which is certainly no worse than time $\mathcal{O}((\ell t)^5)$. That is, the total time used is at worst quintic in the upper bound of the total length of the original sequence.

If ℓ is odd, set $s'_{\lceil \ell/2 \rceil}$ to be s_ℓ. (If ℓ is even, $\lceil \ell/2 \rceil = \lfloor \ell/2 \rfloor$, so $s'_{\lceil \ell/2 \rceil}$ has already been computed.) This is purely notational; no extra computation is done.

The result of this computation is a sequence of $\lceil \ell/2 \rceil$ words, each of length at most $2ct$, whose concatenation represents the same element of the semigroup as the concatenation of the original sequence. The total length of the result is at most $(c+1)\ell t$; that is, at most $c+1$ times the total length of the previous sequence.

Apply this computation iteratively, starting with the sequence $w_1, \ldots, w_n$ and continuing until a sequence with only one element results. Since each iteration takes a sequence with ℓ terms to one with $\lceil \ell/2 \rceil$ terms, there are at most $\lceil \log n \rceil$ iterations. The first iteration of this computation, applied to a sequence whose total length is at most n, completes in time $\mathcal{O}(n^5)$, yielding a sequence of total length at most $n(c+1)$; the next iteration completes in time $\mathcal{O}((n(c+1))^5)$, yielding a sequence of total length at most $n(c+1)^2$. In general the i-th iteration completes in time at most $\mathcal{O}((n(c+1)^{i-1})^5)$, yielding a sequence of total length at most $n(c+1)^i$. So the $\lceil \log n \rceil$ iterations together complete in time at most $\mathcal{O}((1 + \log n)(n(c+1)^{1+\log n})^5)$, since $\lceil \log n \rceil \leq 1 + \log n$. (Informally, each iteration yields a sequence of roughly half as many words in $L(\Sigma)$ labelling a sequence of arcs that each span a subword twice as long as the corresponding terms in the preceding sequence.)

Applying exponent and logarithm laws,

$$\begin{aligned} n(c+1)^{1+\log n} &= n(c+1)(c+1)^{\log n} \\ &= n(c+1)n^{\log(c+1)} \\ &= (c+1)n^{1+\log(c+1)}, \end{aligned}$$

and so, since c is a constant, the algorithm completes in time

$$\mathcal{O}(n^{5+5\log(c+1)} \log n),$$

yielding a word in $L(\Sigma)$ of length at most $n(c+1)n^{\log(c+1)}$. □

Lemma 6.3. *There is an algorithm that takes as input an ε-free context-free grammar Γ and outputs a quadratic Greibach normal form grammar Γ_{G}, taking time $\mathcal{O}(|\Gamma|^2)$.*

Proof. The strategy is to follow the construction used by Blum & Koch [21, Paragraph following Theorem 2.1] and note the time complexity at each stage.

The first step is to convert Γ to an extended Chomsky normal form grammar Γ_{EC}; this takes time $\mathcal{O}(|\Sigma|)$ by inspection of the usual construction (see, for example, [14, Proof of Theorem 4.5], ignoring the removal of unit productions), and $|\Gamma_{\mathrm{EC}}|$ is at most a constant multiple of $|\Gamma|$.

The next step is Blum & Koch's own construction [21, p.116] to convert Γ_{EC} to an quadratic Greibach normal form grammar Γ_{eg}. This involves first constructing auxiliary grammars Γ_X for all X in $N - \{S\}$; by inspection this takes time $\mathcal{O}(|\Gamma_{\mathrm{EC}}|)$ for each X, and thus $\mathcal{O}(|\Gamma_{\mathrm{EC}}|^2)$ time in total, and the grammars Γ_X have size at most $3|\Gamma_{\mathrm{EC}}|$. The final construction of the quadratic Greibach normal form grammar Γ_{G} from Γ_{EC} and the various Γ_X thus takes time $\mathcal{O}(|\Gamma_{\mathrm{EC}}|^2)$.

Since $|\Gamma_{\mathrm{EC}}|$ is at most a constant multiple of $|\Gamma|$, the construction of Γ_{G} takes time $\mathcal{O}(|\Gamma|^2)$. □

Interestingly, although Theorem 6.1 gives a polynomial-time algorithm for the word problem for word-hyperbolic monoids, the proof does not give a bound on the exponent of the polynomial, because the constant c of Lemma 5.1 is dependent on the word-hyperbolic structure Σ. There is thus an open question: does such a bound actually exist? or can the word problem for hyperbolic semigroups be arbitrarily hard within the class of polynomial-time problems?

The algorithm described in Lemma 6.2 is not particularly novel. It is similar in outline to that described by Hoffmann & Thomas [5, Lemma 11] for their restricted notion of word-hyperbolicity in monoids. However, the proof that it takes time polynomial in the lengths of the input words *is* new.

Hoffmann & Thomas describe their algorithm in recursive terms: to find a word in $L(\Sigma)$ representing the same element as $w \in A^*$, factor w as $w'w''$, where the lengths of w' and w'' differ by at most 1, recursively compute representatives p' and p'' in $L(\Sigma)$ of $\overline{w'}$ and $\overline{w''}$, then compute a representative for $\overline{w}$ using p' and p''. This last step they prove to take linear time (recall that this only applies for their restricted notion of word-hyperbolicity) and to yield a word of length at most $|p'| + |p''| + 1$, which shows that the whole algorithm takes time $\mathcal{O}(n \log n)$. However, this recursive, 'top-down' view of the algorithm obscures the fact that the overall strategy can be made to work even for monoids that are word-hyperbolic in the general Duncan–Gilman sense. It is through the iterative, 'bottom-up' view of the algorithm presented above that it becomes apparent that the length increase of Lemma 5.1 remains under control through the $\log n$ iterations.

7. Deciding basic properties

This section shows that certain basic properties are effectively decidable for word-hyperbolic semigroups. First, being a monoid is decidable:

Algorithm 7.1.

Input: An interpreted word-hyperbolic structure Σ for a semigroup.

Output: If the semigroup is a monoid (that is, contains a two-sided identity), output *Yes* and a word in $L(\Sigma)$ representing the identity; otherwise output *No*.

Method:

1. For each $a \in A$, construct the context-free language

$$\begin{aligned} I_a &= \{\, i \in L : a\#_1 i \#_2 a \in M \,\} \\ &= (a\#_1)^{-1}\big(M \cap a\#_1 L \#_2 a\big)(\#_2 a)^{-1} \end{aligned} \tag{7.1}$$

   and check that it is non-empty. If any of these checks fail, halt and output *No*.
2. For each $a \in A$, choose some $i_a \in I_a$.
3. Iterate the following step for each $a \in A$. For each $b \in A$, if $\overline{i_a}\,\overline{b} = \overline{b}\,\overline{i_a} = \overline{b}$, halt and output *Yes* and i_a.
4. Halt and output *No*.

Proposition 7.2. *Algorithm 7.1 outputs* Yes *and i if and only if the semigroup defined by Σ is a monoid with identity $\overline{i}$.*

Proof. Suppose first that Algorithm 7.1 halts with output *Yes* and i. Then by step 3, $\overline{i}\,\overline{b} = \overline{b}\,\overline{i} = \overline{b}$ for all $b \in A$. Since $\overline{A}$ generates S, it follows that $s\overline{i} = \overline{i}s = s$ for all $s \in S$ and hence $\overline{i}$ is an identity for S.

Suppose now that S is a monoid with identity e. Then there is some word $w \in L$ with $\overline{w} = e$. For every $a \in A$, $\overline{a}e = \overline{a}$, and so $a\#_1 w \#_2 a \in M$. Thus $w \in I_a$ for all $a \in A$ and so each I_a is non-empty. Thus the checks in step 1 succeed and the algorithm proceeds to step 2.

Suppose that $w = w_1 \cdots w_n$, where $w_j \in A$ for each $j = 1, \ldots, n$. Then

$$\begin{aligned} e = \overline{w} &= \overline{w_1 \cdots w_{n-1}}\,\overline{w_n} \\ &= \overline{w_1 \cdots w_{n-1}}\,\overline{w_n}\,\overline{i_{w_n}} && \text{(by the choice of } i_{w_n} \in I_{w_n}) \\ &= e\overline{i_{w_n}} \\ &= \overline{i_{w_n}} && \text{(since } e \text{ is an identity for } S). \end{aligned}$$

Hence i_{w_n} represents the identity e and so $\overline{i_{w_n}}\,\overline{b} = \overline{b}\,\overline{i_{w_n}} = \overline{b}$. Thus at least one of the i_a chosen in step 2 passes the test of step 3 (which guarantees that it represents an identity since $\overline{A}$ generates S) and so the algorithm halts at step 3 and outputs *Yes* and a word i_a representing the identity. ☐

Question 7.3. Is there an algorithm that takes as input an interpreted word-hyperbolic structure and determines whether the semigroup it defines contains a zero?

Notice that this cannot be decided using a procedure like Algorithm 7.1, or at least not obviously, because the natural analogue of I_a is

$$Z_a = \{\, z \in L : a\#_1 z \#_2 z^{\mathrm{rev}} \in M \,\}$$

and if one tried to construct effectively this language, one would naturally consider the intersection of the context-free language M and the context-free language $\{\, a\#_1 z\#_2 z^{\mathrm{rev}} : z \in L \,\}$. However, testing the emptiness of an intersection of context-free languages is in general undecidable. So it seems that using Z_a would, at minimum, require some additional insight into the kind of context-free languages that can appear as M.

Notice that commutativity is very easy to decide for a word-hyperbolic semigroup; one needs to check only that $\overline{ab} = \overline{ba}$ for all symbols $a, b \in A(\Sigma)$. This is simply a matter of performing a bounded number of multiplications and checks using Lemmata 5.1 and 5.2.

We now turn to Green's relations, which form the foundation of the structure theory of semigroups; see [22, Chapter 2] for background. Green's relation $\mathcal{L}$ is decidable for automatic semigroups; in contrast, Green's relation $\mathcal{R}$ is undecidable, as a corollary of the fact that right-invertibility is undecidable in automatic monoids [11, Theorem 5.1]. In contrast, $\mathcal{R}$ and $\mathcal{L}$ are both decidable for word-hyperbolic semigroups, as a consequence of $M(\Sigma)$ describing the entire multiplication table.

Proposition 7.4. *There is an algorithm that takes as input an interpreted word-hyperbolic structure Σ and two words $w, w' \in L(\Sigma)$ and decides whether the elements represented by w and w' are:*

1. *$\mathcal{R}$-related,*
2. *$\mathcal{L}$-related,*
3. *$\mathcal{H}$-related.*

Proof. Let S be the semigroup described by Σ. The elements $\overline{w}$ and $\overline{w'}$ are $\mathcal{R}$-related if and only if there exist $s, t \in S^1$ such that $\overline{w}s = \overline{w'}$ and $\overline{w'}t = \overline{w}$. That is, $\overline{w} \mathrel{\mathcal{R}} \overline{w'}$ if and only if either $\overline{w} = \overline{w'}$, or there exist $s, t \in S$ with $\overline{w}s = \overline{w'}$ and $\overline{w'}t = \overline{w}$. The possibility that $\overline{w} = \overline{w'}$ can be checked algorithmically by Theorem 6.1. The existence of an element $s \in S$ such that $\overline{w}s = \overline{w'}$ is equivalent to the non-emptiness of the language

$$\{\, v \in L : w\#_1 v \#_2 (w')^{\mathrm{rev}} \in M \,\}.$$

This context-free language can be effectively constructed and its non-emptiness effectively decided. Similarly, it is possible to decide whether there is an element $t \in S$ such that $\overline{w'}t = \overline{w}$. Hence it is possible to decide whether $\overline{w} \mathrel{\mathcal{R}} \overline{w'}$.

Similarly, one can effectively decide whether $\overline{w} \mathrel{\mathcal{L}} \overline{w'}$. Since $\overline{w} \mathrel{\mathcal{H}} \overline{w'}$ if and only if $\overline{w} \mathrel{\mathcal{R}} \overline{w'}$ and $\overline{w} \mathrel{\mathcal{L}} \overline{w'}$, whether $\overline{w}$ and $\overline{w'}$ are $\mathcal{H}$-related is effectively decidable. □

Corollary 7.5. *There is an algorithm that takes as input an interpreted word-hyperbolic structure and decides whether the semigroup it describes is a group.*

Proof. Suppose the input word-hyperbolic structure is Σ and that it describes a semigroup S. Apply Algorithm 7.1. If S is not a monoid, it cannot be a group. Otherwise we know that S is a monoid and we have a word $i \in L(\Sigma)$ that represents its identity. For each $a \in A(\Sigma)$, check whether $\overline{a} \mathrel{\mathcal{R}} \overline{i}$ and $\overline{a} \mathrel{\mathcal{L}} \overline{i}$: if all these checks succeed, then every generator is both right-and left-invertible, and so S is a group; if any fail, there is some generator that is either not right- or not left-invertible and so S cannot be a group. Hence it is decidable whether Σ describes a group. □

Question 7.6. Are Green's relations $\mathcal{D}$ and $\mathcal{J}$ decidable for word-hyperbolic semigroups?

Note that $\mathcal{D}$ and $\mathcal{J}$ are both undecidable for automatic semigroups [12, Theorems 4.1 & 4.3].

8. Being completely simple

This section shows that it is decidable whether a word-hyperbolic semigroup is completely simple. This is particularly useful because a completely simple semigroup is word-hyperbolic if and only if its Cayley graph is a hyperbolic metric space [6, Theorem 4.1], generalizing the equivalence for groups of these properties for groups.

Definition 8.1. Let S be a semigroup, I and Λ be index sets, and P be a $\Lambda \times I$ matrix over S whose (λ, i)-th element is $p_{\lambda,i}$. The Rees matrix semigroup $\mathcal{M}[S; I, \Lambda; P]$ is defined to be the set $I \times S \times \Lambda$ with multiplication

$$(i, g, \lambda)(j, h, \mu) = (i, gp_{\lambda,j}h, \mu).$$

Recall that a semigroup is completely simple if it has no proper two-sided ideals, is not the two-element null semigroup, and contains a primitive idempotent (that is, an idempotent e such that, for all idempotents f, we have $ef = fe = f \implies e = f$). The version of the celebrated Rees theorem due to Suschkewitsch [22, Theorem 3.3.1] shows that all completely simple semigroups are isomorphic to a semigroup $\mathcal{M}[G; I, \Lambda; P]$, where G is a group and I and Λ are index sets. When the completely simple semigroup is finitely generated, the index sets I and Λ are finite. (See [22, § 3.3] for further background related to completely simple semigroups.)

Let A be an alphabet representing a generating set for a completely simple semigroup $\mathcal{M}[G; I, \Lambda; P]$. Define maps $\upsilon : A \to I$ and $\xi : A \to \Lambda$ by letting $a\upsilon$ and $a\xi$ be such that $\overline{a} \in \{a\upsilon\} \times G \times \{a\xi\}$. For the purposes of this paper, we call the pair of maps (υ, ξ) the *species* of the completely simple semigroup. We first of all prove that it is decidable whether a word-hyperbolic semigroup is a completely simple semigroup of a particular species.

Algorithm 8.2.

Input: An interpreted word-hyperbolic structure Σ, two finite sets I and Λ, and two surjective maps $\upsilon : A(\Sigma) \to I$ and $\xi : A(\Sigma) \to \Lambda$.

Output: If Σ describes a completely simple semigroup of species (υ, ξ), output *Yes*; otherwise output *No*.

Method: At various points in the algorithm, checks are made. If any of these checks fail, the algorithm halts and outputs *No*.

1. For each $i \in I$ and $\lambda \in \Lambda$, construct the regular language

$$L_{i,\lambda} = \{\, a_1 \cdots a_n \in L : a_i \in A, a_1\upsilon = i, a_n\xi = \lambda \,\}.$$

   Check that each $L_{i,\lambda}$ is non-empty.
2. For each $i, j \in I$ and $\lambda, \mu \in \Lambda$, construct the context-free language

$$\{\, u\#_1 v\#_2 w^{\mathrm{rev}} \in M : u \in L_{i,\lambda}, v \in L_{j,\mu}, w \in L - L_{i,\mu} \,\}, \tag{8.1}$$

   and check that it is empty.
3. For each $i \in I$ and $\lambda \in \Lambda$, choose a word $w_{i,\lambda} \in L_{i,\lambda}$ and construct the context-free language

$$I_{i,\lambda} = \{\, u \in L_{i,\lambda} : w_{i,\lambda}\#_1 u\#_2 w_{i,\lambda}^{\mathrm{rev}} \in M \,\}.$$

   Check that each $I_{i,\lambda}$ is non-empty.

4. For each $i \in I$ and $\lambda \in \Lambda$, choose a word $u_{i,\lambda} \in I_{i,\lambda}$.
5. For each $a \in A$, $i \in I$, and $\lambda \in \Lambda$, check that $\overline{u_{av,\lambda}}\,\overline{a} = \overline{a}$ and $\overline{a}\,\overline{u_{i,a\xi}} = \overline{a}$.
6. For each $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, calculate a word $h_{i,a,\mu,\lambda} \in L$ such that $\overline{h_{i,a,\mu,\lambda}} = \overline{u_{i,\mu}}\,\overline{a}\,\overline{u_{i,\lambda}}$.
7. For each $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, check that $\overline{h_{i,a,\mu,\lambda}}\,\overline{u_{i,a\xi}} = \overline{u_{i,\mu}}\,\overline{a}$.
8. For each $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, check that

$$\overline{u_{i,\lambda}}\,\overline{h_{i,a,\mu,\lambda}} = \overline{h_{i,a,\mu,\lambda}}\,\overline{u_{i,\lambda}} = \overline{h_{i,a,\mu,\lambda}}.$$

9. For each $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, construct the context-free language

$$V_{i,a,\mu,\lambda} = \{\, v \in L : h_{i,a,\mu,\lambda}\#_1 v \#_2 u_{i,\lambda} \in M \,\}$$

and check that it is non-empty.
10. For each $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, choose some $v_{i,a,\mu,\lambda} \in V_{i,a,\mu,\lambda}$ and check that $\overline{a}\,\overline{h_{i,a,\mu,\lambda}} = \overline{u_{i,\lambda}}$.
11. Halt and output *Yes*.

Lemmata 8.3 and 8.4 show that this algorithm works.

Lemma 8.3. *If Algorithm 8.2 outputs* Yes*, the semigroup defined by the word-hyperbolic structure* Σ *is a completely simple semigroup of species* (υ, ξ).

Proof. Let S be the semigroup defined by the input word-hyperbolic structure Σ. Suppose the algorithm output *Yes*. Then all the checks in steps 1–10 must succeed.

For each $i \in I$ and $\lambda \in \Lambda$, let $T_{i,\lambda} = \overline{L_{i,\lambda}}$. By the definition of $L_{i,\lambda}$, for each $a \in A$, the word a lies in $L_{av,a\lambda}$. By the check in step 1, each $T_{i,\lambda}$ is non-empty.

By the check in step 2, for all $i, j \in I$ and $\lambda, \mu \in \Lambda$, there do not exist $u \in L_{i,\lambda}$, $v \in L_{j,\mu}$, $w \in L - L_{i,\mu}$ with $\overline{u}\,\overline{v} = \overline{w}$. That is,

$$T_{i,\lambda}T_{j,\mu} \subseteq T_{i,\mu} \text{ for all } i, j \in I \text{ and } \lambda, \mu \in \Lambda. \tag{8.2}$$

In particular, $T_{i,\lambda}T_{i,\lambda} \subseteq T_{i,\lambda}$ and so each $T_{i,\lambda}$ is a subsemigroup of S.

In each $T_{i,\lambda}$, there is some element that stabilizes some other element $\overline{w_{i,\lambda}}$ on the right (that is, that right-multiplies $\overline{w_{i,\lambda}}$ like an identity) by the check in step 3. In step 4, $u_{i,\lambda}$ is chosen to be such an element. Let $e_{i,\lambda} = \overline{u_{i,\lambda}}$.

By the check in step 5,

$$e_{av,\lambda}\overline{a} = \overline{a} \text{ and } \overline{a}e_{i,a\xi} = \overline{a} \text{ for all } i \in I \text{ and } \lambda \in \Lambda. \tag{8.3}$$

In step 6, $h_{i,a,\mu,\lambda}$ is calculated for all $i \in I$, $\lambda, \mu \in \Lambda$, $a \in A$ so that

$$\overline{h_{i,a,\mu,\lambda}} = e_{i,\mu}\overline{a}e_{i,\lambda}. \tag{8.4}$$

By (8.2), $h_{i,a,\mu,\lambda} \in L_{i,\lambda}$. By the check in step 7,

$$\overline{h_{i,a,\mu,\lambda}}e_{i,a\xi} = e_{i,\mu}\overline{a} \text{ for all } i \in I,\ \lambda, \mu \in \Lambda,\ a \in A. \tag{8.5}$$

Let $i \in I$ and $\lambda \in \Lambda$. Let $t \in T_{i,\lambda}$. Then $t = \overline{a_1}\,\overline{a_2}\cdots\overline{a_n}$ for some $a_k \in A$. Since $a_1a_2\cdots a_n \in L_{i,\lambda}$, $a_1\upsilon = i$ and $a_n\xi = \lambda$. Then

$$\begin{aligned}
&\overline{a_1}\,\overline{a_2}\,\overline{a_3}\cdots\overline{a_n} \\
&= e_{i,\lambda}\overline{a_1}\,\overline{a_2}\,\overline{a_3}\cdots\overline{a_n}e_{i,\lambda} && \text{[by (8.3), since } a_1\upsilon = i \text{ and } a_n\xi = \lambda] \\
&= \overline{h_{i,a_1,\lambda,\lambda}}e_{i,a_1\xi}\overline{a_2}\,\overline{a_3}\cdots\overline{a_n}e_{i,\lambda} && \text{[by (8.5)]} \\
&= \overline{h_{i,a_1,\lambda,\lambda}}\,\overline{h_{i,a_2,a_1\xi,\lambda}}e_{i,a_2\xi}\overline{a_3}\cdots\overline{a_n}e_{i,\lambda} && \text{[by (8.5)]} \\
&= \overline{h_{i,a_1,\lambda,\lambda}}\,\overline{h_{i,a_2,a_1\xi,\lambda}}\,\overline{h_{i,a_3,a_2\xi,\lambda}}e_{i,a_3\xi}\cdots\overline{a_n}e_{i,\lambda} && \text{[by (8.5)]} \\
&\quad\vdots \\
&= \overline{h_{i,a_1,\lambda,\lambda}}\,\overline{h_{i,a_2,a_1\xi,\lambda}}\,\overline{h_{i,a_3,a_2\xi,\lambda}}\cdots e_{i,a_{n-1}\xi}\overline{a_n}e_{i,\lambda} \\
& && \text{[by repeated use of (8.5)]} \\
&= \overline{h_{i,a_1,\lambda,\lambda}}\,\overline{h_{i,a_2,a_1\xi,\lambda}}\,\overline{h_{i,a_3,a_2\xi,\lambda}}\cdots\overline{h_{i,a_n,a_{n-1}\xi,\lambda}} && \text{[by (8.4)]}
\end{aligned}$$

Therefore the subsemigroup $T_{i,\lambda}$ is generated by the set of elements $H_{i,\lambda} = \{\overline{h_{i,a,\mu,\lambda}} : a \in A, \mu \in \Lambda\}$.

By the check in step 8, for all $i \in I$, $\lambda \in \Lambda$, and $h \in H_{i,\lambda}$, we have $he_{i,\lambda} = e_{i,\lambda}h = h$. Since $H_{i,\lambda}$ generates $T_{i,\lambda}$, it follows that $e_{i,\lambda}$ is an identity for $T_{i,\lambda}$. So each $T_{i,\lambda}$ is a submonoid of S with identity $e_{i,\lambda}$. In particular, each $e_{i,\lambda}$ is idempotent.

Let $i \in I$ and $\lambda \in \Lambda$. By the check in step 9, every element $h \in H_{i,\lambda}$ has a right inverse h' in $T_{i,\lambda}$. By the check in step 10, $h'h = e_{i,\lambda}$ and so h' is also a left-inverse for h in $T_{i,\lambda}$. Thus every generator in $H_{i,\lambda}$ is both right- and left-invertible. Hence every element of $T_{i,\lambda}$ is both right- and left-invertible and so $T_{i,\lambda}$ is a subgroup of S.

Since S is the union of the various $T_{i,\lambda}$, the semigroup S is regular and the $e_{i,\lambda}$ are the only idempotents in S. Thus by (8.2), distinct idempotents cannot be related by the idempotent ordering. Hence all idempotents of S are primitive. Since S does not contain a zero (since it is the union of the $T_{i,\lambda}$ and (8.2) holds), it is completely simple by [22, Theorem 3.3.3]. ☐

Lemma 8.4. *If semigroup defined by the word-hyperbolic structure Σ is a completely simple semigroup of species (υ, ξ), then Algorithm 8.2 outputs* Yes.

Proof. Suppose the semigroup S defined by the word-hyperbolic structure Σ is a completely simple semigroup, with $S = \mathcal{M}[G; I, \Lambda; P]$. For all $i \in I$ and $\lambda \in \Lambda$, let $e_{i,\lambda}$ be the identity of the subgroup $T_{i,\lambda} = \{i\} \times G \times \{\lambda\}$; that is, $e_{i,\lambda} = (i, p_{\lambda,i}^{-1}, \lambda)$. For each $a \in A$, the element $\overline{a}$ has the form $(a\upsilon, g_a, a\xi)$ for some $g_a \in G$.

By the definition of multiplication in S, the word $a_1 \cdots a_n \in L$ represents an element of $T_{i,\lambda}$ if and only if $a_1\upsilon = i$ and $a_n\xi = \lambda$. Hence each $L_{i,\lambda}$ must be the preimage of $T_{i,\lambda}$ and map surjectively onto $T_{i,\lambda}$. In particular, $L_{i,\lambda}$ must be non-empty and so the checks in step 1 succeed.

For any $i, j \in I$ and $\lambda, \mu \in \Lambda$, we have $T_{i,\lambda}T_{j,\mu} \subseteq T_{i,\mu}$. Hence if $u \in L_{i,\lambda}$, $v \in L_{j,\mu}$, and $w \in L$ are such that $\overline{u}\,\overline{v} = \overline{w}$, then $w \in L_{i,\mu}$. Thus the language (8.1) is empty for all $i, j \in I$ and $\lambda, \mu \in \Lambda$. Hence all the checks in step 2 succeed.

For any $i \in I$ and $\lambda \in \Lambda$, if $w_{i,\lambda} \in L_{i,\lambda}$, then $\overline{w_{i,\lambda}} \in T_{i,\lambda}$. Since $T_{i,\lambda}$ is a subgroup, $\overline{w_{i,\lambda}}e_{i,\lambda} = \overline{w_{i,\lambda}}$, and $e_{i,\lambda}$ is the unique element of $T_{i,\lambda}$ that stabilizes $\overline{w_{i,\lambda}}$ on the right. Thus the language $I_{i,\lambda}$ is non-empty, and consists of words representing $e_{i,\lambda}$. Hence the checks in step 3 succeed, and the words $u_{i,\lambda}$ chosen in step 4 are such that $\overline{u_{i,\lambda}} = e_{i,\lambda}$.

In a completely simple semigroup, each idempotent is a left identity within its own $\mathcal{R}$-class and $e_{i,a\xi}$ is a right identity within its own $\mathcal{L}$-class [22, Proposition 2.3.3]. Hence for each $a \in A$, $i \in I$, and $\lambda \in \Lambda$, we have $e_{a\upsilon,\lambda}\overline{a} = \overline{a}$ and $\overline{a}e_{i,a\xi} = \overline{a}$. Thus the checks in step 5 succeed.

For all $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$,

$$\begin{aligned}
&\overline{h_{i,a,\mu,\lambda}}\,\overline{u_{i,a\xi}} \\
&= e_{i,\mu}\overline{a}e_{i,\lambda}e_{i,a\xi} \\
&= e_{i,\mu}(a\upsilon, g_a, a\xi)(i, p_{\lambda,i}^{-1}, \lambda)(i, p_{a\xi,i}^{-1}, a\xi) \\
&= e_{i,\mu}(a\upsilon, g_a p_{a\xi,i} p_{\lambda,i}^{-1} p_{\lambda,i} p_{a\xi,i}^{-1}, a\xi) \\
&= e_{i,\mu}(a\upsilon, g_a, a\xi) \\
&= e_{i,\mu}\overline{a}.
\end{aligned}$$

Thus all the checks in step 7 succeed.

For all $a \in A$, $i \in I$, and $\lambda, \mu \in \Lambda$, the element $\overline{h_{i,a,\mu,\lambda}}$ lies in the subgroup $T_{i,\lambda}$, whose identity is $e_{i,\lambda}$. Hence all the checks in step 8 succeed. Since all elements of this subgroup are right-invertible, each language $V_{i,a,\mu,\lambda}$ is non-empty; hence all the checks in step 9 succeed. Finally, since a right inverse is also a left inverse in a group, all the checks in step 10 succeed. Therefore the algorithm reaches step 10 and halts with output *Yes*. □

Theorem 8.5. *There is an algorithm that takes as input an interpreted word-hyperbolic structure* Σ *for a semigroup and decides whether it is a completely simple semigroup.*

Proof. We prove that this problem can be reduced to the problem of deciding whether the semigroup defined by an interpreted word-hyperbolic structure Σ is a completely simple semigroup of a particular species $(\upsilon : A(\Sigma) \to I, \xi : A(\Sigma) \to \Lambda)$.

Let S be the semigroup specified by Σ. Then S is finitely generated. Thus we need only consider the problem of deciding whether S is a finitely generated completely simple semigroup. By the definition of multiplication in a completely simple semigroup (viewed as a Rees matrix semigroup), the leftmost generator in a product determines its $\mathcal{R}$-class (that is, the I-component of the product) and the rightmost generator in a product determines its $\mathcal{L}$-class (that is, the Λ-component of the product). Thus there must be at least one generator in each $\mathcal{R}$- and $\mathcal{L}$- class, and hence if S is an $I \times \Lambda$ Rees matrix semigroup, both $|I|$ and $|\Lambda|$ cannot exceed $|A(\Sigma)|$.

Thus it is suffices to decide whether S is an $I \times \Lambda$ completely simple semigroup for some fixed choice of I and Λ, for one can simply test the finitely many possibilities for index sets I and Λ no larger than $A(\Sigma)$.

One can restrict further, and ask whether S is completely semigroup of some particular species $(\upsilon : A(\Sigma) \to I, \xi : A(\Sigma) \to \Lambda)$, for there are a bounded number of possibilities for the maps surjective υ and ξ, so it suffices to test each one. □

It is a natural to ask whether Theorem 8.5 can be generalized from 'completely simple' to 'completely 0-simple' (see [22, § 3.3]. This question seems to be related to, and may indeed depend upon, deciding if a word-hyperbolic semigroup contains a zero (Question 7.3).

Question 8.6. Is there an algorithm that takes as input an interpreted word-hyperbolic structure and determines whether the semigroup it defines is completely 0-simple?

9. Being a Clifford semigroup

This section is dedicated to showing that being a Clifford semigroup is decidable for word-hyperbolic semigroups. We recall here the definition of a Clifford semigroup; see [22, § 4.2] for further background.

Definition 9.1. Let Y be a [meet] semilattice and let $\{ G_\alpha : \alpha \in Y \}$ be a collection of disjoint groups with, for all $\alpha, \beta \in Y$ such that $\alpha \geq \beta$, a homomorphism $\phi_{\alpha,\beta} : G_\alpha \to G_\beta$ satisfying the following conditions:

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

$$\phi_{\alpha,\gamma} = \phi_{\alpha,\beta}\phi_{\beta,\gamma}. \tag{9.1}$$

The set of elements of the *Clifford semigroup* $\mathcal{S}[Y; G_\alpha; \phi_{\alpha,\beta}]$ is the union of the disjoint groups G_α. The product of the elements s and t of S, where $s \in G_\alpha$ and $t \in G_\beta$, is

$$(s\phi_{\alpha,\alpha\wedge\beta})(t\phi_{\beta,\alpha\wedge\beta}), \tag{9.2}$$

which lies in the group $G_{\alpha\wedge\beta}$. [The meet of α and β is denoted $\alpha \wedge \beta$.]

Note that if $\mathcal{S}[Y; G_\alpha; \phi_{\alpha,\beta}]$ is finitely generated, the semilattice Y must be finitely generated and thus finite.

Let A be an alphabet representing a generating set for a Clifford semigroup $\mathcal{S}[Y; G_\alpha; \phi_{\alpha,\beta}]$. Define a map $\xi : A \to Y$ by letting $a\xi$ be such that $\overline{a} \in G_{a\xi}$. For the purposes of this paper, we call this map $\xi : A \to Y$ the *species* of the Clifford semigroup. [Notice that the map ξ extends to a unique homomorphism $\xi : A^+ \to Y$.] We first of all prove that it is decidable whether a word-hyperbolic semigroup is a Clifford semigroup of a particular species.

Algorithm 9.2.

Input: An interpreted word-hyperbolic structure Σ and a map $\xi : A \to Y$.

Output: If Σ describes a Clifford semigroup of species $\xi : A \to Y$, output *Yes*; otherwise output *No*.

Method: At various points in the algorithm, checks are made. If any of these checks fail, the algorithm halts and outputs *No*.

1. For each $\alpha \in Y$, construct the regular language

$$L_\alpha = \{ w \in L : w\xi = \alpha \}.$$

(These languages are regular since L is regular, Y is finite, and the map $\xi : A \to Y$ is known.) Check that each L_α is non-empty.

2. For each $\alpha, \beta \in Y$, construct the context-free language
$$\{ u\#_1 v\#_2 w^{\mathrm{rev}} \in M : u \in L_\alpha, v \in L_\beta, w \in L - L_{\alpha\wedge\beta} \} \tag{9.3}$$
and check that it is empty.
3. For each $\alpha \in Y$, choose some word $w_\alpha \in L_\alpha$ and construct the context-free language
$$I_\alpha = \{ i \in L_\alpha : w_\alpha \#_1 i \#_2 w_\alpha^{\mathrm{rev}} \in M \}$$
and check that I_α is non-empty.
4. For each $\alpha \in Y$, pick some $i_\alpha \in I_\alpha$ and check that for all $\alpha, \beta \in Y$, $\overline{i_\alpha}\,\overline{i_\beta} = \overline{i_{\alpha\wedge\beta}}$.
5. For each $a \in A$, check that $\overline{i_{a\xi}}\,\overline{a} = \overline{a}\,\overline{i_{a\xi}} = \overline{a}$. For each $\alpha \in Y$ and $a \in A$ check that $\overline{a}\,\overline{i_\alpha} = \overline{i_\alpha}\,\overline{a}$.
6. For each $\alpha \in Y$ and $a \in A$ such that $a\xi \geq \alpha$, construct the context-free language
$$V_{\alpha,a} = \{ v \in L_\alpha : a\#_1 v\#_2 i_\alpha \in M \}$$
and check that $V_{\alpha,a}$ is non-empty.
7. For each $\alpha \in Y$ and $a \in A$ such that $a\xi \geq \alpha$, pick some $v_{\alpha,a} \in V_{\alpha,a}$ and check that $\overline{v_{\alpha,a}}\,\overline{a} = \overline{i_\alpha}$.
8. Halt and output *Yes*.

Lemmata 9.3 and 9.4 show that this algorithm works.

Lemma 9.3. *If Algorithm 9.2 outputs* Yes*, the semigroup described by the word-hyperbolic structure* Σ *is a Clifford semigroup of species* $\xi : A \to Y$*.*

Proof. Let S be the semigroup defined by the input word-hyperbolic structure Σ. Suppose the algorithm output *Yes*. Then all the checks in steps 1–7 must succeed.

For each $\alpha \in Y$, let $T_\alpha = \overline{L_\alpha}$. By the check in step 1, all T_α are non-empty.

By the check in step 2, for every $\alpha, \beta \in Y$, there do not exist $u \in L_\alpha$, $v \in L_\beta$, $w \in L - L_{\alpha\wedge\beta}$ with $\overline{u}\,\overline{v} = \overline{w}$. That is, $T_\alpha T_\beta \subseteq T_{\alpha\wedge\beta}$. In particular, $T_\alpha T_\alpha \subseteq T_\alpha$ and so each T_α is a subsemigroup of S.

In each T_α, there is some element that right-multiplies some other element like an identity by the check in step 3.

For each $\alpha \in Y$, the word i_α represents an element e_α, and the set of elements $E = \{ e_\alpha : \alpha \in Y \}$ forms a subsemigroup isomorphic to the semilattice Y by the check in step 4.

By the checks in step 5, for each $a \in A$, the element $e_{a\xi}$ (which, like $\overline{a}$, lies in $T_{a\xi}$)) acts like an identity on $\overline{a}$ (that is, $e_{a\xi}\overline{a} = \overline{a}e_{a\xi} = \overline{a}$), and every element e_α commutes with $\overline{a}$.

Let $\alpha \in Y$ and $t \in T_\alpha$. Then $t = \overline{a_1}\,\overline{a_2} \cdots \overline{a_n}$ for some $a_i \in A$ with $(a_1 a_2 \cdots a_n)\xi = \alpha$. Then

$$\begin{aligned}
&\overline{a_1}\,\overline{a_2} \cdots \overline{a_n} \\
&= e_{a_1\xi}\overline{a_1}e_{a_2\xi}\overline{a_2} \cdots e_{a_n\xi}\overline{a_n} && \text{[by the check in step 6]} \\
&= e_{a_1\xi}e_{a_2\xi} \cdots e_{a_n\xi}\overline{a_1}\,\overline{a_2} \cdots \overline{a_n} && \text{[by the check in step 6]} \\
&= e_{(a_1\xi)\wedge(a_2\xi)\wedge\cdots\wedge(a_n\xi)}\overline{a_1}\,\overline{a_2} \cdots \overline{a_n} && \text{[by the isomorphism of } E \text{ and } Y] \\
&= e_{(a_1a_2\cdots a_n)\xi}\alpha\overline{a_1}\,\overline{a_2} \cdots \overline{a_n} && \text{[by the extension of } \xi \text{ to } A^+] \\
&= e_\alpha\overline{a_1}\,\overline{a_2} \cdots \overline{a_n}.
\end{aligned}$$

Thus $t = e_\alpha t$. Similarly $te_\alpha = t$. Hence e_α is an identity for T_α.

For each $\alpha \in Y$ and $a \in A$ with $a\xi \geq \alpha$, there is an element $\overline{v_{\alpha,a}} \in T_\alpha$ such that $\overline{v_{\alpha,a}}\,\overline{a} = \overline{a}\,\overline{v_{\alpha,a}} = e_\alpha$ by the checks in steps 6 and 7. Since T_α is generated by elements $\overline{a}$ such that $a\xi \geq \alpha$, it follows that T_α is a subgroup of S.

Since L is the union of the various L_α, the semigroup S is the union of the various subgroups T_α. In particular, S is regular. Furthermore, the only idempotents in S are the identities of these subgroups; that is, the elements e_α. Since every e_α commutes with every element of $\overline{A}$, it follows that all idempotents of S are central. Hence S is a regular semigroup in which the idempotents are central, and thus is a Clifford semigroup by [22, Theorem 4.2.1]. □

Lemma 9.4. *If the semigroup defined by the word-hyperbolic structure Σ is a Clifford semigroup of species $\xi : A \to Y$, then Algorithm 9.2 outputs* Yes.

Proof. Suppose the semigroup S defined by the word-hyperbolic structure $(A, L, M(L))$ is a Clifford semigroup, with $S = \mathcal{S}[Y; G_\alpha; \phi_{\alpha,\beta}]$. For each $\alpha \in Y$, let e_α be the identity of G_α. The language L_α clearly consists of exactly those words in L that map onto G_α, so L_α is non-empty. Hence the checks in step 1 succeed.

By the definition of multiplication in a Clifford semigroup, $G_\alpha G_\beta \subseteq G_{\alpha\wedge\beta}$. Hence if $u \in L_\alpha$, $v \in L_\beta$, and $w \in L$ are such that $\overline{u}\,\overline{v} = \overline{w}$, then $w \in L_{\alpha\wedge\beta}$.

Thus the language (9.3) is empty for all $\alpha, \beta \in Y$. Hence all the checks in step 2 succeed.

Let $\alpha \in Y$. For any $w_\alpha \in L_\alpha$, the element $\overline{w_\alpha}$ lies in the subgroup G_α. Thus the language I_α consists of precisely the words that represent elements of G_α that right-multiply w_α like an identity. Since G_α is a subgroup, every element of I_α represents e_α. Since there must be at least one such representative, I_α is non-empty. Thus every check in step 3 succeeds.

The identities e_α form a subsemigroup isomorphic to the semilattice Y by the definition of multiplication in a Clifford semigroup. Thus every check in step 4 succeeds.

Furthermore, every e_α is idempotent and thus central in S by [22, Theorem 4.2.1], and so every check in step 5 succeeds.

Let $\alpha \in Y$ and $a \in A$ be such that $a\xi \geq \alpha$. Let $v_{\alpha,a}$ be the word representing $(\overline{a}\phi_{a\xi,\alpha})^{-1}$. Then

$$\overline{u_a}\,\overline{v_{\alpha,a}} = \overline{a}(\overline{a}\phi_{a\xi,\alpha})^{-1} = (\overline{a}\phi_{a\xi,\alpha})(\overline{a}\phi_{a\xi,\alpha})^{-1} = e_\alpha.$$

Hence $v_{\alpha,a} \in V_{\alpha,a}$ and so all the checks in step 6 succeed. Similarly $\overline{v_{\alpha,a}}\,\overline{u_a}$ and so all the checks in step 7 succeed.

Therefore the algorithm reaches step 8 and halts with output *Yes*. □

Theorem 9.5. *There is an algorithm that takes as input an interpreted word-hyperbolic structure Σ for a semigroup and decides whether it is a Clifford semigroup.*

Proof. We prove that this problem can be reduced to the problem of deciding whether the semigroup defined by an interpreted word-hyperbolic structure Σ is a Clifford semigroup with a particular species $\xi : A(\Sigma) \to Y$.

Let S be the semigroup specified by Σ. Then S is finitely generated. Thus we need only consider the problem of deciding whether S is a finitely generated Clifford semigroup, whose corresponding semilattice must therefore also be finitely generated. A finitely generated semilattice is finite.

So if S is a Clifford semigroup $\mathcal{S}[Y; G_\alpha; \phi_{\alpha,\beta}]$, the semilattice Y must be a homomorphic image of the free semilattice of rank $|A(\Sigma)|$, which has $2^{|A(\Sigma)|} - 1$ elements. Thus it is suffices to decide whether S is a Clifford semigroup for some fixed semilattice Y, for one can simply test the finitely many possibilities for Y.

One can restrict further, and ask whether S is a Clifford semigroup with some fixed semilattice Y and some particular placement of generators into the semilattice of groups. (That is, with knowledge of in which group G_α each generator $\overline{a}$ putatively lies, described by a map $\xi : A(\Sigma) \to Y$. Of course, it is necessary that $\operatorname{im}\xi$ generates Y.) There are a bounded number of possibilities for the map ξ, so it suffices to test each one. $\square$

10. Being free

This section shows that it is decidable whether a word-hyperbolic semigroup is free. The following technical lemma, which is possibly of independent interest, is necessary.

Lemma 10.1. *There is an algorithm that takes as input an alphabet A, a symbol $\#_2$ not in A, and a context-free grammar Γ defining a context-free language $L(\Gamma)$ that is a subset of $A^*\#_2A^*$, and decides whether $L(\Gamma)$ contains a word $x\#_2w^{\mathrm{rev}}$ where $x \neq w$.*

Proof. Suppose $\Gamma = (N, A \cup \{\#_2\}, P, O)$. [Here, N is the set of non-terminal symbols, $A \cup \{\#_2\}$ is of course the set of terminal symbols, P the set of productions, and $O \in N$ is the start symbol.] Since $L(\Gamma)$ does not contain the empty word (since every word in $L(\Gamma)$ lies in $A^*\#_2A^*$), assume without loss that Γ contains no useless symbols or unit productions [14, Theorem 4.4].

Let

$$N_\# = \{\, M \in N : (\exists p, q \in A^*)(M \Rightarrow^* p\#_2q) \,\}.$$

Notice that if $M \to p$ is a production in P and $M \in N - N_\#$, then every non-terminal symbol appearing in p also lies in $N - N_\#$. [This relies on there being no useless symbols in Γ, which means that every other non-terminal in P derives some terminal word.] For this reason, it is easy to compute $N_\#$.

Suppose that $M \Rightarrow^* uMv$ for some $M \in N - N_\#$ and $u, v \in (A \cup \{\#_2\})^*$. Then u and v cannot contain $\#_2$ since $M \in N - N_\#$. Since there are no unit productions in P, at least one of u and v is not the empty word. Since M is not a useless symbol, it appears in some derivation of a word $w\#_2x^{\mathrm{rev}} \in L(\Gamma)$. Pumping the derivation $M \Rightarrow^* uMv$ yields a word $w'\#_2(x')^{\mathrm{rev}}$ where exactly one of $w' = w$ or $x' = x$ holds, since the extra inserted u and v cannot be on opposite sides of the symbol $\#_2$ since $M \in N - N_\#$. Hence either $w \neq x$ or $w' \neq x'$. Hence in this case $L(\Gamma)$ does contain a word of the given form.

Since it is easy to check whether there is a non-terminal $M \in N - N_\#$ with $M \Rightarrow^* uMv$, we can assume that no such non-terminal exists. Therefore any non-terminal $M \in N - N_\#$ derives only finitely many words (since any derivation starting at M can only involve non-terminals in $N - N_\#$ and by assumption no such non-terminal can appear twice in a given derivation). These words can be effectively enumerated. Let $M \in N - N_\#$ and let $w_1, \ldots, w_n$ be all the words that M derives. Replacing a production $S \to pMq$ by the productions $S \to pw_1q$, $S \to pw_2q$, $\ldots$, $S \to pw_nq$ does not alter $L(\Gamma)$. Iterating this process, we eventually obtain a grammar Γ where no non-terminal symbol in $N - N_\#$ appears on the right-hand side of a production. Thus all symbols in $N - N_\#$ can be eliminated and we now have a grammar Γ with $N = N_\#$.

Every production is now of the form $M \to pSq$ or $M \to p\#_2q$, where $p, q \in A^*$ and $S \in N$. [There can be only one non-terminal on the right-hand side of each production, since otherwise some terminal word would contain two symbols $\#_2$, which is impossible.]

We are now going to iteratively define a map $\phi : N \to \mathrm{FG}(A)$, where $\mathrm{FG}(A)$ denotes the free group on A, which we will identify with the set of reduced words on $A \cup A^{-1}$. First, define $O\phi = \varepsilon$. Now, iterate through the productions as follows. Choose some production $M \to pSq^{\mathrm{rev}}$ such that $M\phi$ is already defined. Let $z = p^{-1}(M\phi)q \in \mathrm{FG}(A)$. If $S\phi$ is undefined, set $S\phi = z$. If $S\phi$ is defined, check that $S\phi$ and z are equal; if they are not, halt: $L(\Gamma)$ does contain words $w\#_2x^{\mathrm{rev}}$ with $w \neq x$.

To see this, suppose $S\phi = z$ and consider the sequence of productions that gave us the original value of $S\phi$:

$$O \to u_1S_1v_1^{\mathrm{rev}}, S_1 \to u_2S_2v_2^{\mathrm{rev}}, \ldots, S_k \to u_kSv_k^{\mathrm{rev}},$$

which implies that $S\phi = (u_1u_2\cdots u_k)^{-1}v_1v_2\cdots v_k$, and the sequence that gave us $M\phi$:

$$O \to p_1M_1q_1^{\mathrm{rev}}, M_1 \to p_2M_2q_2^{\mathrm{rev}}, \ldots, M_k \to p_lMq_l^{\mathrm{rev}},$$

which implies that $M\phi = (p_1p_2\cdots p_l)^{-1}q_1q_2\cdots q_l$. Choose $r, s \in A^*$ such that $S \Rightarrow^* r\#_2s^{\mathrm{rev}}$. Then $L(\Gamma)$ contains both both $u_1\cdots u_kr\#_2s^{\mathrm{rev}}v_k^{\mathrm{rev}}\cdots v_1^{\mathrm{rev}}$ and (recalling that $M \to pSq^{\mathrm{rev}}$ is a production) $p_1\cdots p_lpr\#_2s^{\mathrm{rev}}q^{\mathrm{rev}}q_l^{\mathrm{rev}}\cdots q_1$. Suppose $u_1\cdots u_kr = v_1\cdots v_ks$ and $p_1\cdots p_lpr = q_1\cdots q_lqs$. Then $S\phi = (u_1\cdots u_k)^{-1}v_1\cdots v_k = rs^{-1} = (p_1\cdots p_lp)^{-1}q_1\cdots q_lq = p^{-1}(M\phi)q = z$, which is a contradiction.

Once we have iterated through all the productions of the form $M \to pSq^{\mathrm{rev}}$, iterate through the productions of the form $M \to p\#_2q^{\mathrm{rev}}$, and check that $p^{-1}(M\phi)q$. If this check fails, halt: $L(\Gamma)$ does contain words $w\#_2 x^{\mathrm{rev}}$ with $w \neq x$; the proof of this is very similar to the previous paragraph.

Finally, notice that if the iteration through all the productions completes with all the checks succeeding, a simple induction on derivations, using the values of $M\phi$, shows that all words $w\#_2 x^{\mathrm{rev}} \in L(\Gamma)$ are such that $w = x$. □

Algorithm 10.2.

Input: An interpreted word-hyperbolic structure Σ.

Output: If Σ describes a free semigroup, output *Yes*; otherwise output *No*.

Method:

1. For each $a \in A$, iterate the following:
   (a) Construct the context-free language
   $$D_a = \{\, uv : u\#_1 v\#_2 a^{\mathrm{rev}} \in M \,\}.$$
   (b) Check whether D_a is empty. If it is empty, proceed to the next interation. If it is non-empty, choose some word $d_a \in D_a$. If d_a contains the letter a, halt and output *No*. If d_a does not contain the letter a, define the rational relations
   $$\begin{aligned}\mathcal{Q}_L &= \big(\{(a, d_a)\} \cup \{\,(b, b) : b \in A - \{a\}\,\}\big)^+ \\ \mathcal{Q}_M &= \big(\{(a, d_a)\} \cup \{\,(b, b) : b \in A - \{a\}\,\}\big)^+ \#_1 \\ &\qquad \big(\{(a, d_a)\} \cup \{\,(b, b) : b \in A - \{a\}\,\}\big)^+ \#_2 \\ &\qquad\quad \big(\{(a, d_a^{\mathrm{rev}})\} \cup \{\,(b, b) : b \in A - \{a\}\,\}\big)^+.\end{aligned}$$
   Modify Σ as follows: replace A by $A - \{a\}$; replace L by $L \circ \mathcal{Q}_L$; and replace M by $M \circ \mathcal{Q}_M$, and proceed to the next iteration.
2. If $L \neq A^+$, halt and output *No*.
3. Define the rational relation
   $$\mathcal{P} = \{\,(a, a) : a \in A\} \cup \{(\#_1, \varepsilon), (\#_2, \#_2)\,\}.$$
   Let $N = M \circ \mathcal{P}$. Using the method of Lemma 10.1, check whether N contains any word of the form $x\#_2 w^{\mathrm{rev}}$ with $x \neq w$. If so, halt and ouput *No*. Otherwise, halt and output *Yes*.

Lemmata 10.3 to 10.5 show that this algorithm works.

Lemma 10.3. *If Σ is a word-hyperbolic structure for a semigroup S, then the replacement Σ produced in step 1(b) is also a word-hyperbolic structure for a semigroup S.*

Proof. If the language D_a is non-empty, then any word $w \in D_a$ is such that $\overline{w} = \overline{a}$. In particular, $\overline{d_a} = \overline{a}$. Furthermore, since $d_a \in (A - \{a\})^*$, we see that $\overline{a}$ is a redundant generator. The rational relation $\mathcal{Q}_L$ relates any word in A^+ to the corresponding word in $(A - \{a\})^+$ with all instances of the symbol a replaced by the word d_a. The rational relation $\mathcal{Q}_M$ relates any word in $A^+\#_1A^+\#_2A^+$ to the corresponding word in $(A-\{a\})^+\#_1(A-\{a\})^+\#_2(A-\{a\})^+$ with all instances of the symbol a *before* $\#_2$ replaced by the word d_a and all instances of the symbol a *after* $\#_2$ replaced by the word d_a^{rev}. Hence

$$M \circ \mathcal{Q}_M \subseteq (L \circ \mathcal{Q}_L)\#_1(L \circ \mathcal{Q}_L)\#_2(L \circ \mathcal{Q}_L)^{\mathrm{rev}}.$$

Since application of rational relations preserves regularity and context-freedom, $L \circ \mathcal{Q}_L$ is regular and $M \circ \mathcal{Q}_M$ is context-free. Finally, since $\overline{a} = \overline{d_a}$, we see that $L \circ \mathcal{Q}_L$ maps onto S, and similarly $M \circ \mathcal{Q}_M$ describes the multiplication of elements of S in terms of representatives in L. □

Lemma 10.4. *If Algorithm 10.2 outputs* Yes*, the semigroup defined by the word-hyperbolic structure Σ is a free semigroup.*

Proof. The algorithm can only halt with output *Yes* in step 3, so the algorithm must pass step 2 as well. Hence the language of representatives is A^+. Let S be the semigroup defined by L and let $\phi : A^+ \to S$ be an interpretation.

Suppose for *reductio ad absurdum* that ϕ is not injective. Then there are distinct words $u, v \in A^*$ such that $u\phi = v\phi$. Since $\phi|_A$ is injective by definition, at least one of u and v has length 2 or more. Interchanging u and v if necessary, assume $|u| \geq 2$. So $u = u'u''$, where u' and u'' are both non-empty. Since $L = A^+$, we have $u', u'' \in L$ and so $u'\#_1u''\#_2v^{\mathrm{rev}} \in M$. Hence $u\#_2v^{\mathrm{rev}} \in N$. But since the algorithm outputs *Yes* at step 3, there is no word $x\#_2w^{\mathrm{rev}} \in M$ with $x \neq w$. This is a contradiction and so ϕ is injective.

So $\phi : A^+ \to S$ is an isomorphism and so S is free. □

Lemma 10.5. *If the word-hyperbolic structure Σ defines a free semigroup, Algorithm 10.2 outputs* Yes.

Proof. Let B^+ be the semigroup defined by Σ. Let $\phi : A^+ \to B^+$ be an interpretation. Since elements of B are indecomposable, $B \subseteq A\phi$.

In step 1, the algorithm iterates through each $a \in A$. For each $a \in B\phi^{-1} \subseteq A$, since $a\phi$ is indecomposable, the language D_a is empty and the algorithm moves to the next iteration.

Let $a \in A - B\phi^{-1}$. Then $a\phi$ has length (in B^+) at least two and so is decomposable. Hence there exist $u, v \in L$ such that $uv \in D_a$. Furthermore, since $u\phi$ and $v\phi$ must be shorter (in B^+) than $a\phi$, neither u nor v can include the letter a. Hence the replacement of Σ described in step 1(b) takes place. Since this occurs for all $a \in A - B\phi^{-1}$, at the end of step 1 we have a word-hyperbolic structure Σ with $A = B\phi^{-1}$. Since $\phi|_A$ is injective, $\phi|_A$ must be a bijection from A to B. Hence the homomorphism $\phi : A^+ \to B^+$ must be an isomorphism, and so $L = A^+$; thus the check in step 2 is successful. Therefore

$$M = \{\, u\#_1 v\#_2 (uv)^{\mathrm{rev}} : u, v \in A^+ \,\}$$

and so

$$M \circ \mathcal{P} = \{\, w\#_2 w^{\mathrm{rev}} : w \in A^+ \,\}.$$

Thus the check in step 3 is successful and the algorithm terminates with output *Yes*. □

Thus, from Lemmata 10.4 and 10.5, we obtain the decidability of freedom for word-hyperbolic semigroups:

Theorem 10.6. *There is an algorithm that takes as input an interpreted word-hyperbolic structure Σ for a semigroup and decides whether it is a free semigroup.*

11. Open problems

This conclusing section lists some important question regarding decision problems for word-hyperbolic semigroups.

Question 11.1. Is there an algorithm that takes as input an interpreted word-hyperbolic structure for a semigroup and decides whether that semigroup is (a) regular, (b) inverse?

Whether these properties are decidable for automatic semigroups is currently unknown.

Question 11.2. Is there an algorithm that takes as input an interpreted word-hyperbolic structure for a semigroup and decides whether that semigroup is left-/right-/two-sided-cancellative?

Cancellativity and left-cancellativity are undecidable for automatic semigroups [10]. Right-cancellativity is, however, decidable [11, Corollary 3.3].

Question 11.3. Is there an algorithm that takes as input an interpreted word-hyperbolic structure for a semigroup and decides whether that semigroup is finite?

The equivalent question for automatic semigroups is easy: one takes an automatic structure, effectively computes an automatic structure with uniqueness, and checks whether its regular language of representatives is finite. However, this approach cannot be used for word-hyperbolic semigroups, because there exist word-hyperbolic semigroups that do not admit word-hyperbolic structures with uniqueness indeed, they may not even admit regular languages of unique normal forms [7, Examples 10 & 11].

Question 11.4. Is there an algorithm that takes as input an interpreted word-hyperbolic structure for a semigroup and decides whether that semigroup admits a word-hyperbolic structure with uniqueness? (That is, where the language of representatives maps bijectively onto the semigroup.) If so, it is possible to compute a word-hyperbolic structure with uniqueness in this case?

[1] M. Gromov, Hyperbolic groups, in: S. M. Gersten (Ed.), Essays in group theory, Vol. 8 of Math. Sci. Res. Inst. Publ., Springer, New York, 1987, pp. 75–263.

[2] R. H. Gilman, On the definition of word hyperbolic groups, Mathematische Zeitschrift 242 (3) (2002) 529–541. doi:10.1007/s002090100356.

[3] A. Duncan, R. H. Gilman, Word hyperbolic semigroups, Mathematical Proceedings of the Cambridge Philosophical Society 136 (3) (1999) 513–524. doi:10.1017/S0305004103007497.

[4] M. Hoffmann, D. Kuske, F. Otto, R. M. Thomas, Some relatives of automatic and hyperbolic groups, in: G. M. Gomes, J.-É. Pin, P. V. Silva (Eds.), Semigroups, algorithms, automata and languages, World Scientific, 2002, p. 379–406. doi:10.1142/9789812776884_0016.

[5] M. Hoffmann, R. M. Thomas, Notions of automaticity in semigroups, Semigroup Forum 66 (3) (2002) 337–367. doi:10.1007/s002330010161.

[6] J. Fountain, M. Kambites, Hyperbolic groups and completely simple semigroups, in: I. M. Araújo, M. J. Branco, V. H. Fernandes, G. M. Gomes (Eds.), Semigroups and Languages, World Scientific, 2004, p. 106–132. doi:10.1142/9789812702616_0007.

[7] A. J. Cain, V. Maltcev, Context-free rewriting systems and word-hyperbolic structures with uniqueness, Internat. J. Algebra Comput. 22 (7). doi:10.1142/S0218196712500610.

[8] D. B. Epstein, J. W. Cannon, D. F. Holt, S. V. Levy, M. S. Paterson, W. P. Thurston, Word Processing in Groups, Jones & Bartlett, 1992.

[9] C. M. Campbell, E. F. Robertson, N. Ruškuc, R. M. Thomas, Automatic semigroups, Theoretical Computer Science 250 (1-2) (2001) 365–391. doi:10.1016/S0304-3975(99)00151-6.

[10] A. J. Cain, Cancellativity is undecidable for automatic semigroups, Q. J. Math. 57 (3) (2006) 285–295. doi:10.1093/qmath/hai023.

[11] M. Kambites, F. Otto, Uniform decision problems for automatic semigroups, Journal of Algebra 303 (2) (2006) 789–809. doi:10.1016/j.jalgebra.2005.11.028.

[12] F. Otto, Some decidability and undecidability results on Green's relations for automatic monoids, Semigroup Forum 75 (3) (2007) 520–535. doi:10.1007/s00233-007-0713-y.

[13] F. Dahmani, V. Guirardel, The isomorphism problem for all hyperbolic groups, Geometric and Functional Analysis 21 (2) (2011) 223–300. doi:10.1007/s00039-011-0120-0.

[14] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 1st Edition, Addison–Wesley, 1979.

[15] J. Berstel, Transductions and Context-Free Languages, no. 38 in Leitfäden der Angewandten Mathematik und Mechanik, B. G. Teubner, 1979.

[16] R. V. Book, F. Otto, String Rewriting Systems, Texts and monographs in computer science, Springer, 1993.

[17] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1999.

[18] M. Hoffmann, R. M. Thomas, Biautomatic semigroups, in: M. Liskiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory, no. 3623 in Lecture Notes in Computer Sciene, Springer, 2005, pp. 56–67. doi:10.1007/11537311_6.

[19] A. J. Duncan, E. F. Robertson, N. Ruškuc, Automatic monoids and change of generators, Mathematical Proceedings of the Cambridge Philosophical Society 127 (3) (1999) 403–409. doi:10.1017/S0305004199003722.

[20] S. L. Graham, M. Harrison, W. L. Ruzzo, An improved context-free recognizer, ACM Transactions on Programming Languages and Systems 2 (3) (1980) 415–462. doi:10.1145/357103.357112.

[21] N. Blum, R. Koch, Greibach normal form transformation revisited, Information and Computation 150 (1) (1999) 112–118. doi:10.1006/inco.1998.2772.

[22] J. M. Howie, Fundamentals of Semigroup Theory, no. 12 in London Mathematical Society Monographs: New Series, Clarendon Press, Oxford University Press, 1995.