International Journal of Algebra and Computation

Commutative Semigroups with a Context-Free Word Problem

Peter Kostolányi
Department of Computer Science, Comenius University in Bratislava, Mlynská dolina, 842 48 Bratislava, Slovakia
kostolanyi@fmph.uniba.sk

Received 6 January 2024
Accepted 18 January 2025

Communicated by Mikhail Volkov

The word problem of a finitely generated commutative semigroup is shown to be context-free if and only if this semigroup does not contain the free commutative semigroup on two generators as a subsemigroup, answering a question of T. Brough, A. J. Cain, and M. Pfeiffer. An analogous characterisation is established for monoids as well.

Keywords: Word problem; commutative semigroup; commutative monoid; context-free language; pushdown automaton.

Mathematics Subject Classification 2020: 20M05, 68Q45

1. Introduction

The *word problem* for a semigroup S, finitely generated by A, is usually described as an algorithmic problem, in which one is given two nonempty words u, v over A understood as an alphabet, and the task is to decide whether they evaluate to the same element of S. Almost the same definition of a word problem can be given for finitely generated monoids, except that the input words can now also be empty. Finally, the word problem for a group G, finitely generated by A *as a monoid*, is precisely the same as the word problem for G understood as a monoid; in this case, one often considers the generating set $A = X \cup X^{-1}$, where G is finitely generated by X as a group and $X^{-1} = \{a^{-1} \mid a \in X\}$.

Some of the deepest connections between group theory and formal languages were uncovered by adopting a *language-theoretic viewpoint* on the word problem for groups. The essence of this approach of A. V. Anisimov [1] lies in encoding the word problem of a group G, generated by a finite set A as a monoid, into the language of all $u \in A^*$ evaluating to the identity element of G. This language capturing all the necessary information about the word problem is rational if and only if G is finite, and the classical Muller–Schupp theorem [28,11] says that it is context-free if and only if G is virtually free. See, e.g., [18, Chapter 11] for an exposition.

Many additional results have been obtained in this direction since the seminal work mentioned above – for instance, groups with word problems falling into several other natural classes of languages were studied [4,12,13,15,17,21,23,25,26,36], other problems related to the word problem were considered from a language-theoretic perspective [25,27], and a deeper understanding of the class of context-free groups and the Muller–Schupp theorem was achieved [2,8,9,24].

Given the richness of this theory obtained by viewing the word problem of a group as a formal language, it is a natural endeavour to extend its scope by studying similar questions for finitely generated semigroups and monoids. However, when trying to do so, one is faced with a problem that the language of all words over the alphabet of generators evaluating to the identity element might no longer capture the essential information about the word problem. One thus has to consider a language that instead encodes all *pairs of words* over the alphabet of generators evaluating to the same element of the semigroup or monoid in question.

Such a language can be defined in different ways. Under the most commonly used definition due to A. Duncan and R. H. Gilman [10], it consists of all words $u\#v^R$ such that u, v are nonempty words over A evaluating to the same element of S; here, v^R denotes the reversal of v. The definition is almost the same for monoids, except that the words u, v need not be nonempty. An alternative approach to defining these languages appears in [33,3,5,6] – nevertheless, we stick with the above-described definitions of A. Duncan and R. H. Gilman in this article.

A natural question raised by A. Duncan and R. H. Gilman [10] is whether the characterisation of groups with a context-free word problem, provided by the Muller–Schupp theorem, can be generalised to semigroups or monoids. Although this still seems far from being answered [29], there has been considerable progress in the understanding of word problem languages of semigroups and monoids. Several fundamental properties of semigroups and monoids with a context-free word problem were explored by M. Hoffmann et al. [16] and by T. Brough, A. J. Cain, and M. Pfeiffer [7]. Moreover, D. F. Holt, M. D. Owens, and R. M. Thomas [17] studied the semigroups with a one-counter word problem, and several new results on semigroups with a context-free word problem, as well as on semigroups with word problems from certain abstract families of languages, were recently obtained by C.-F. Nyberg-Brodda [29,30,31].

T. Brough, A. J. Cain, and M. Pfeiffer [7] have asked whether it is possible to characterise the (finitely generated) *commutative* semigroups whose word problem is context-free. We answer this question – and the same question for monoids – in this article.

We show that the word problem of a finitely generated commutative monoid M is context-free if and only if M does not contain a submonoid isomorphic to $(\mathbb{N}\times\mathbb{N}, +)$, i.e., the free commutative monoid on two generators. Similarly, a finitely generated commutative semigroup S has a context-free word problem if and only if S does not contain an isomorphic copy of the free commutative semigroup $((\mathbb{N}\times\mathbb{N})\setminus\{(0,0)\}, +)$ on two generators as a subsemigroup.

These results fit well with the Muller–Schupp theorem, from which it follows that the word problem of a finitely generated *abelian group* G is context-free if and only if G does not contain a subgroup isomorphic to $(\mathbb{Z}\times\mathbb{Z},+)$, i.e., the free abelian group on two generators.

2. Preliminaries

We denote by $\mathbb{N}$ the set of all *nonnegative* integers and for each $n\in\mathbb{N}$, we write $[n]=\{1,\ldots,n\}$. The reader is assumed to be familiar with the basics of formal language theory, as presented, e.g., in [19,20,22]. Alphabets are always understood to be finite, but they are *not* necessarily nonempty. Given an alphabet A, the *free semigroup* on A is denoted by A^+ and the *free monoid* on A by A^*. The *reversal* of a word $w\in A^*$ over some alphabet A is denoted by w^R, the *length* of w by $|w|$, and the *number of occurrences* of a symbol $a\in A$ in w by $|w|_a$. The class of all context-free languages is denoted by $\mathscr{L}(\mathrm{CF})$.

Given an alphabet $A=\{a_1,\ldots,a_n\}$ and a word $w\in A^*$, we write $\Psi(w)$ for the *Parikh vector* [32,22] of w defined by

$$\Psi(w)=(|w|_{a_1},|w|_{a_2},\ldots,|w|_{a_n})\,.$$

We also use this notation when $a_1,\ldots,a_n$ are not explicitly given; in such cases, some arbitrary fixed ordering of symbols is implicitly assumed.

The usual componentwise partial order on $\mathbb{N}^n$ is denoted by $\leq$; we thus have $(p_1,\ldots,p_n)\leq(q_1,\ldots,q_n)$ if and only if $p_k\leq q_k$ for $k=1,\ldots,n$. *Dickson's lemma* – see, e.g., [35, Theorem 5.1] – states that every subset of $\mathbb{N}^n$ has finitely many minimal elements with respect to $\leq$.

Let S be a semigroup finitely generated by a set $A\subseteq S$. Then there is a unique semigroup homomorphism $\nu\colon A^+\to S$ such that $\nu(a)=a$ for all $a\in A$; note that A^+ denotes the *free semigroup* on A, rather than the subsemigroup of S generated by A (that equals S). Given nonempty words $u,v\in A^+$, we write $u=_S v$ if and only if $\nu(u)=\nu(v)$. Observe that the relation $=_S$ is a congruence on A^+. According to the definition of A. Duncan and R. H. Gilman [10], the *word problem* of S with respect to the generating set A is the language

$$\mathrm{WP}_A(S)=\{u\#v^R\mid u,v\in A^+;\ u=_S v\},$$

where $\#\notin A$ is some fixed delimiter symbol.

Similarly, for M a monoid finitely generated by a set $A\subseteq M$, there is a unique monoid homomorphism $\eta\colon A^*\to M$ such that $\eta(a)=a$ for all $a\in A$; given $u,v\in A^*$, we write $u=_M v$ if and only if $\eta(u)=\eta(v)$, the relation $=_M$ being a congruence on A^*. The *word problem* of M with respect to A is defined by

$$\mathrm{WP}_A(M)=\{u\#v^R\mid u,v\in A^*;\ u=_M v\}$$

for a fixed delimiter $\#\notin A$. Note that the word problem of a monoid M considered as a semigroup is different from the language just defined; the correct meaning of $\mathrm{WP}_A(M)$ can, nevertheless, be always understood from the context.

Given two finite generating sets A, B of a monoid M with $\# \notin A \cup B$, clearly

$$\mathrm{WP}_A(M) = h^{-1}(\mathrm{WP}_B(M))$$

for an arbitrary homomorphism $h\colon (A \cup \{\#\})^* \to (B \cup \{\#\})^*$ such that $h(\#) = \#$ and $h(a) \in B^*$ satisfies

$$h(a) =_M a$$

for each $a \in A$. Moreover, existence of at least one homomorphism with this property follows by the fact that B is a generating set of M. Similar observations can be made for semigroups as well.

It follows that if a class of languages $\mathcal{C}$ is closed under inverse homomorphisms, then the word problem $\mathrm{WP}_A(M)$ of a monoid M is in $\mathcal{C}$ either for all finite generating sets $A \subseteq M$, or for no such generating set – and the same holds for word problems of semigroups [16,17]. This means that the property of a word problem being in $\mathcal{C}$ does not depend on the generating set considered, provided $\mathcal{C}$ is closed under inverse homomorphisms. One can thus simply say that "*the* word problem" of a semigroup or a monoid is or is not in $\mathcal{C}$, meaning that this holds with respect to all finite generating sets or with respect to no such set.

As the class of all context-free languages is closed under inverse homomorphisms, the convention described above applies in this setting as well, and it is used throughout this article.

The classical Muller–Schupp theorem [28,11] says that the word problem of a finitely generated group[a] is context-free if and only if the group is virtually free. A. Duncan and R. H. Gilman [10] asked whether this characterisation can be generalised to semigroups. This still remains an open problem, which has a reputation of being relatively hard [29]. To gain at least some insight, the following weaker question was raised by T. Brough, A. J. Cain, and M. Pfeiffer [7].

Question 2.1. Which finitely generated *commutative* semigroups have a context-free word problem?

Note that it is a straightforward consequence of the Muller–Schupp theorem and the Fundamental Theorem of Finitely Generated Abelian Groups that a finitely generated *abelian group* has a context-free word problem if and only if it does not contain a subgroup isomorphic to $\mathbb{Z}\times\mathbb{Z}$, the free abelian group on two generators [7]. In what follows, we answer both Question 2.1 and a similar question for monoids consistently with this observation – we show that the word problem of a finitely generated commutative semigroup (monoid) is context-free if and only if the semigroup (monoid) does not contain the free commutative semigroup (monoid) on two generators as a subsemigroup (submonoid).

[a] Although the word problem is usually defined in a slightly different way in the setting of groups – see the Introduction – the statement remains valid even when one interprets the group as a monoid and uses the definition of the word problem described above [10, Theorem 5.3].

3. Basic Observations

Let us first establish the easier implication of the above anticipated theorem characterising finitely generated commutative monoids with a context-free word problem, which amounts to the following proposition.

Proposition 3.1. *Let M be a finitely generated commutative monoid that contains an isomorphic copy of $(\mathbb{N} \times \mathbb{N}, +)$ as a submonoid. Then the word problem of M is not context-free.*

Proof. Let $a, b \in M$ freely generate a commutative submonoid of M. Let $A \subseteq M$ be a finite generating set of M such that $a, b \in A$. Suppose for contradiction that $\mathrm{WP}_A(M) \in \mathscr{L}(\mathrm{CF})$. Then, by closure of $\mathscr{L}(\mathrm{CF})$ under intersection with a rational language, it follows that

$$\mathrm{WP}_A(M) \cap a^*b^*\#a^*b^* = \{a^i b^j \# a^i b^j \mid i, j \in \mathbb{N}\}$$

is a context-free language, which is clearly false – a contradiction. □

The rest of this section is mostly devoted to gathering observations leading to the converse of the above proposition. The actual main results – i.e., the characterisations of commutative monoids and semigroups with a context-free word problem – are then proved in Section 4.

We start by recording the following simple fact that we use on multiple occasions in this article, and that allows us to infer new pairs of words in the relation $=_M$ from known ones.

Proposition 3.2. *Let M be a finitely generated commutative monoid, $A \subseteq M$ a finite generating set of M, and $u, v, x, y \in A^*$. Then $vx =_M vy$ holds whenever $ux =_M uy$ and $\Psi(u) \leq \Psi(v)$.*

Proof. By commutativity of M and $\Psi(u) \leq \Psi(v)$, surely $v =_M v'u$ for some $v' \in A^*$. As $=_M$ is a congruence, $ux =_M uy$ gives $v'ux =_M v'uy$, and $v =_M v'u$ gives $vx =_M v'ux$ and $v'uy =_M vy$. Thus $vx =_M vy$ by transitivity of $=_M$. □

We now focus on finitely generated commutative monoids that *do not* contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$, and gather several observations concerning their structure. When such a monoid M is generated by a finite set A, every word $w \in A^*$ determines an ideal of M generated by an element $\eta(w) \in M$, to which w evaluates in M. Our aim is to show that the elements $a \in A$ with sufficiently many occurrences in w can be classified into three simple classes based upon how multiplication by a behaves in the ideal of M generated by $\eta(w)$.

Let us first consider the submonoids generated by two elements a, b of a finitely generated commutative monoid M *not* containing an isomorphic copy of the free commutative monoid $(\mathbb{N} \times \mathbb{N}, +)$. We show that in case at least one of the functions

$$(q, s) \mapsto (a^q b^q)\, a^s$$

or

$$(q, t) \mapsto (a^q b^q)\, b^t$$

does not become periodic in the second argument for some sufficiently large fixed q, the element a is "essentially equivalent" to b or to an inverse of b when multiplying elements of the ideal generated by $a^q b^q$.

Proposition 3.3. *Let M be a commutative monoid, generated by a finite set $A \subseteq M$, that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$. Let $a, b \in A$. Then there is $q = q_{a,b} \in \mathbb{N}$ such that* at least one *of the following holds*:

(*i*) $(a^q b^q)\, a^s =_M (a^q b^q)$ *for some* $s \in \mathbb{N} \setminus \{0\}$;
(*ii*) $(a^q b^q)\, b^t =_M (a^q b^q)$ *for some* $t \in \mathbb{N} \setminus \{0\}$;
(*iii*) $(a^q b^q)\, a^s =_M (a^q b^q)\, b^t$ *for some* $s, t \in \mathbb{N} \setminus \{0\}$;
(*iv*) $(a^q b^q)\, a^s b^t =_M (a^q b^q)$ *for some* $s, t \in \mathbb{N} \setminus \{0\}$.

Proof. As the submonoid of M generated by $\{a, b\} \subseteq A$ cannot be isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$, there have to be $i_1, j_1, i_2, j_2 \in \mathbb{N}$ such that $(i_1, j_1) \neq (i_2, j_2)$ and

$$a^{i_1} b^{j_1} =_M a^{i_2} b^{j_2}.$$

For $i = \min\{i_1, i_2\}$ and $j = \min\{j_1, j_2\}$, this rewrites as

$$\left(a^i b^j\right)\, a^{i_1 - i} b^{j_1 - j} =_M \left(a^i b^j\right)\, a^{i_2 - i} b^{j_2 - j}$$

using commutativity of M. Here, at least one of the exponents $i_1 - i$, $j_1 - j$, $i_2 - i$, $j_2 - j$ is positive, while neither $i_1 - i$ and $i_2 - i$, nor $j_1 - j$ and $j_2 - j$ can be both nonzero. Thus, setting

$$q = \max\{i, j\}$$

and using Proposition 3.2, a relation

$$(a^q b^q)\, a^{i_1 - i} b^{j_1 - j} =_M (a^q b^q)\, a^{i_2 - i} b^{j_2 - j}$$

is obtained, which takes one of the forms (*i*) – (*iv*) modulo symmetry of $=_M$. □

By virtue of Rédei's theorem [34] – see also [14,35] – every commutative monoid M generated by a finite set A is actually finitely presented, i.e.,[b]

$$M = \langle A \mid \varrho \rangle = \langle A \mid u_1 = v_1, \ldots, u_n = v_n \rangle$$

for some $u_1, \ldots, u_n, v_1, \ldots, v_n \in A^*$ and $\varrho = \{(u_1, v_1), \ldots, (u_n, v_n)\}$. We then write $\ell(M, A, \varrho)$ for the length of the longest word appearing in the defining relations, i.e.,

$$\ell(M, A, \varrho) = \max\{|u_1|, |v_1|, |u_2|, |v_2|, \ldots, |u_n|, |v_n|\}$$

if $n > 0$ and $\ell(M, A, \varrho) = 0$ otherwise.

Given a commutative monoid M finitely generated by A and $m \in \mathbb{N}$, we call any mapping $\Phi\colon A \to \{0, \ldots, m\}$ an (A, m)-*vector*. Moreover, for every $w \in A^*$, we define $\Phi^{(m)}_{A,w}\colon A \to \{0, \ldots, m\}$ for all $a \in A$ by

$$\Phi^{(m)}_{A,w}(a) = \begin{cases} |w|_a & \text{if } |w|_a \leq m, \\ m & \text{otherwise.} \end{cases}$$

This (A, m)-vector can be viewed as a "capped Parikh vector" of w, in which one is only interested in occurrence numbers of letters up to m, and thus all numbers greater than or equal to m are identified with m.

Conversely, for $A = \{a_1, \ldots, a_k\}$ and an (A, m)-vector Φ, let

$$z_\Phi = a_1^{\Phi(a_1)} a_2^{\Phi(a_2)} \ldots a_k^{\Phi(a_k)} \in A^*;$$

this word corresponds canonically to the (A, m)-vector Φ and satisfies $\Phi^{(m)}_{A,z_\Phi} = \Phi$. Clearly $\Psi(z_\Phi) \leq \Psi(w)$ for all $w \in A^*$ such that $\Phi^{(m)}_{A,w} = \Phi$. Moreover, let

$$V(\Phi) = \{a \in A \mid \Phi(a) = m\}$$

be the alphabet of letters from A, for which the maximum possible value m of the (A, m)-vector Φ is actually attained.

Definition 3.4. Let M be a finitely generated commutative monoid that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$, and $A \subseteq M$ a finite generating set of the monoid M. Then for every $m \in \mathbb{N}$ and (A, m)-vector Φ, let:

(*i*) The set $A_\times(\Phi)$ consist of all $a \in A$ such that $wa^s =_M wa^t$ for some $w \in A^*$ with $\Phi^{(m)}_{A,w} = \Phi$ and some $s, t \in \mathbb{N}$ such that $s > t$;
(*ii*) The set $V_\times(\Phi)$ consist of all $a \in V(\Phi)$ such that $wa^s =_M w$ for some $w \in A^*$ with $\Phi^{(m)}_{A,w} = \Phi$ and some $s \in \mathbb{N} \setminus \{0\}$;
(*iii*) $V_\pm(\Phi) = V(\Phi) \setminus V_\times(\Phi)$.

[b]All presentations in this article should be understood as presentations *of monoids*, as opposed to presentations of commutative monoids. In other words, the relations $ab = ba$ for all $a, b \in A$ have to follow as a consequence of the relations $u_1 = v_1, \ldots, u_n = v_n$.

Some explanation of the above-defined concepts is in order. In what follows, we show that when m is large enough and $\Phi^{(m)}_{A,u} = \Phi^{(m)}_{A,v} = \Phi$ for some $u, v \in A^*$ and some (A, m)-vector Φ, then the ideals of M generated by the images $\eta(u), \eta(v)$ of u, v in M share certain common properties. The (A, m)-vector Φ itself is thus perhaps best thought of as a representation of this class of ideals.

The alphabet $A_\times(\Phi)$ contains all $a \in A$ such that multiplication by a becomes periodic in the ideal generated by the image $\eta(w)$ of some $w \in A^*$ with $\Phi^{(m)}_{A,w} = \Phi$ in M. However, it turns out that the choice of w actually does not matter when m is sufficiently large: we prove in Lemma 3.5 that in this case, the same property always holds for $w = z_\Phi$ – and thus, by virtue of Proposition 3.2, also for all other $w \in A^*$ such that $\Phi^{(m)}_{A,w} = \Phi$. This means that $A_\times(\Phi)$ essentially contains all $a \in A$ such that multiplication by a is periodic in the ideals represented by Φ.

Similarly, the role of $V_\times(\Phi)$ is better understood in the light of its characterisation for sufficiently large m, provided by Lemma 3.5: we show there that $V_\times(\Phi) = A_\times(\Phi) \cap V(\Phi)$ when m is large enough. The set $V_\times(\Phi)$ thus consists of all $a \in A$ such that $\Phi(a) = m$ and multiplication by a is periodic in the ideals represented by Φ. Moreover, we prove in Lemma 3.5 that w can be replaced by z_Φ in the defining relation $wa^s =_M w$ when m is large enough, and s can be replaced by some α that is the same for all (A, m)-vectors Φ.

Finally, $V_\pm(\Phi)$ consists of the remaining $a \in A$ such that $\Phi(a) = m$. For every (A, m)-vector Φ, the set $V(\Phi)$ can thus be written as a disjoint union

$$V(\Phi) = V_\pm(\Phi) \cup V_\times(\Phi).$$

When m is sufficiently large, the set $V_\pm(\Phi)$ contains precisely all $a \in A$ such that $\Phi(a) = m$, and multiplication by a is not periodic in the ideals represented by Φ. As we observe later in this section, $V_\pm(\Phi)$ can be decomposed into two subsets $V_+(\Phi)$ and $V_-(\Phi)$ in this case, such that any two elements from any of these sets behave in "essentially the same way" when used to multiply elements of ideals represented by Φ, while the behaviour of elements of $V_+(\Phi)$ can be seen as "inverse" to the behaviour of elements of $V_-(\Phi)$. One can thus classify the elements of $V_\pm(\Phi)$ as being "positive" or "negative", which is the reason behind the notation $V_\pm(\Phi)$.

Nevertheless, we first need to establish the above-anticipated properties of $A_\times(\Phi)$ and $V_\times(\Phi)$ for sufficiently large m. The $q_{a,b}$ for $a, b \in A$ in the statement of the following lemma are as in Proposition 3.3.

Lemma 3.5. *Let M be a commutative monoid, finitely presented by $M = \langle A \mid \varrho \rangle$ for some $A \subseteq M$ and $\varrho \subseteq A^* \times A^*$, that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$. Then there exists $m \in \mathbb{N}$ and $\alpha \in \mathbb{N} \setminus \{0\}$ with $m \geq q_{a,b}$ for all $a, b \in A$ and $\alpha \geq \ell(M, A, \varrho)$ such that for every (A, m)-vector Φ:*

- *(i)* $V_\times(\Phi) = A_\times(\Phi) \cap V(\Phi)$;
- *(ii) For every $a \in V_\times(\Phi)$, one has $z_\Phi a^\alpha =_M z_\Phi$;*
- *(iii) For every $a \in A_\times(\Phi)$, there are $s, t \in \mathbb{N}$ such that $s > t$ and $z_\Phi a^s =_M z_\Phi a^t$.*

Proof. For every $B = \{b_1, \ldots, b_r\} \subseteq A$ and $n \in \mathbb{N}$, let

$$z_{B,n} = b_1^n b_2^n \ldots b_r^n,$$

and let $B_\times$ consist of all $a \in A$ such that

$$z_{B,n} w a^s =_M z_{B,n} w a^t \tag{3.1}$$

for some $n \in \mathbb{N}$, some $w \in (A \setminus B)^*$, and some $s, t \in \mathbb{N}$ such that $s > t$. For $a \in B_\times$ fixed, let L_a be the language of all $w \in (A \setminus B)^*$ such that (3.1) holds for some $n \in \mathbb{N}$ and $s, t \in \mathbb{N}$ such that $s > t$. The set of minimal Parikh vectors of words from L_a is finite by Dickson's lemma, hence L_a contains a finite sublanguage $L'_a \subseteq L_a$ such that for every $w \in L_a$, there is some $x \in L'_a$ with $\Psi(x) \leq \Psi(w)$.

For each $x \in L'_a$, let

$$n(B, a, x) = \min\{n \in \mathbb{N} \mid z_{B,n} x a^s =_M z_{B,n} x a^t \text{ for some } s > t\} \tag{3.2}$$

and

$$t(B, a, x) = \min\{t \in \mathbb{N} \mid z_{B,n(B,a,x)} x a^s =_M z_{B,n(B,a,x)} x a^t \text{ for some } s > t\}. \tag{3.3}$$

Moreover, let $s(B, a, x) \in \mathbb{N}$ be such that $s(B, a, x) > t(B, a, x)$ and

$$z_{B,n(B,a,x)} x a^{s(B,a,x)} =_M z_{B,n(B,a,x)} x a^{t(B,a,x)}. \tag{3.4}$$

Define

$$m_B = \max\{n(B, a, x) \mid a \in B_\times;\ x \in L'_a\} + \max\{t(B, a, x) \mid a \in B_\times;\ x \in L'_a\} \tag{3.5}$$

(with maximum of the empty set understood as zero) and

$$m = \max\left(\{m_B \mid B \subseteq A\} \cup \{q_{a,b} \mid a, b \in A\}\right). \tag{3.6}$$

Let us now consider an arbitrary (A, m)-vector Φ and any $a \in A_\times(\Phi)$, so that Proposition 3.2 gives $a \in B_\times$ for $B = V(\Phi)$. Moreover, let $A \setminus B = \{c_1, \ldots, c_r\}$ and

$$w = c_1^{\Phi(c_1)} c_2^{\Phi(c_2)} \ldots c_r^{\Phi(c_r)}.$$

Then, using the notation of the previous paragraph, $w \in L_a$, and there has to be some $x \in L'_a$ such that $\Psi(x) \leq \Psi(w)$. By definition of the language L'_a, it follows that

$$z_{B,n} x a^s =_M z_{B,n} x a^t$$

holds for some $n \in \mathbb{N}$ and $s, t \in \mathbb{N}$ with $s > t$, so that by (3.2), (3.3), and (3.4) we obtain

$$z_{B,n(B,a,x)} x a^{s(B,a,x)} =_M z_{B,n(B,a,x)} x a^{t(B,a,x)},$$

and by (3.5) and (3.6) together with Proposition 3.2, we get

$$z_{B,m-t(B,a,x)} x a^{s(B,a,x)} =_M z_{B,m-t(B,a,x)} x a^{t(B,a,x)}.$$

As we know that $\Psi(x) \leq \Psi(w)$, it follows by Proposition 3.2 that for $s = s(B, a, x)$ and $t = t(B, a, x)$, we also have

$$z_{B,m-t}wa^s =_M z_{B,m-t}wa^t.$$

Moreover, as clearly $\Psi(z_{B,m-t}) \leq \Psi(z_{B,m})$ and $\Psi(z_{B,m-t}a^t) \leq \Psi(z_{B,m})$ for every $a \in V(\Phi) = B$, using Proposition 3.2 we obtain (with w, s, t as above)

$$z_{B,m}wa^s =_M z_{B,m}wa^t$$

for all $a \in A_\times(\Phi)$ and

$$z_{B,m}wa^{s-t} =_M z_{B,m}w$$

for all $a \in A_\times(\Phi) \cap V(\Phi)$. As in addition clearly $\Psi(z_{B,m}w) = \Psi(z_\Phi)$, we finally obtain

$$z_\Phi a^s =_M z_\Phi a^t \tag{3.7}$$

for all $a \in A_\times(\Phi)$ and some $s, t \in \mathbb{N}$ such that $s > t$, and

$$z_\Phi a^q =_M z_\Phi \tag{3.8}$$

for all $a \in A_\times(\Phi) \cap V(\Phi)$ and some $q \in \mathbb{N} \setminus \{0\}$. For each $a \in A_\times(\Phi) \cap V(\Phi)$, let us denote the smallest such q by $q(\Phi, a)$.

Now, (*iii*) follows directly by (3.7). Furthermore, the inclusion

$$V_\times(\Phi) \subseteq A_\times(\Phi) \cap V(\Phi)$$

is evident, and (3.8) gives the opposite inclusion

$$V_\times(\Phi) \supseteq A_\times(\Phi) \cap V(\Phi),$$

proving (*i*). Finally, let α denote the least common multiple of $\max\{\ell(M, A, \varrho), 1\}$ and of $q(\Phi, a)$ over all[c] (A, m)-vectors Φ and all $a \in V_\times(\Phi)$. Then obviously

$$z_\Phi a^\alpha =_M z_\Phi$$

for all (A, m)-vectors Φ and $a \in V_\times(\Phi)$, and (*ii*) is proved as well. At the same time, the definition of m implies $m \geq q_{a,b}$ for all $a, b \in A$ and the definition of α implies $\alpha \geq \ell(M, A, \varrho)$. □

Recall once again that every finitely generated commutative monoid is finitely presented thanks to Rédei's theorem. This means that Lemma 3.5 actually applies to all *finitely generated* commutative monoids not containing a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$. The statement of the following corollary makes sense as a result.

[c]The set of all (A, m)-vectors for fixed A and m is clearly finite.

Corollary 3.6. *Let M be a finitely generated commutative monoid that does not contain a submonoid isomorphic to $(\mathbb{N}\times\mathbb{N},+)$, and $A\subseteq M$ a finite generating set of M. Let m,α be as in Lemma 3.5. Then*:

(*i*) *For every $w\in A^*$ and $a\in A_\times(\Phi^{(m)}_{A,w})$, there are $s,t\in\mathbb{N}$ such that $s>t$ and $wa^s=_M wa^t$;*
(*ii*) *For every $w\in A^*$ and $a\in V_\times(\Phi^{(m)}_{A,w})$, one has $wa^\alpha=_M w$.*

Proof. Follows by Lemma 3.5, by Proposition 3.2, and by noting that every $w\in A^*$ clearly satisfies the inequality $\Psi\left(z_{\Phi^{(m)}_{A,w}}\right)\leq\Psi(w)$. □

We are now ready to prove that when M and m are as above and Φ is an (A,m)-vector, then $V_\pm(\Phi)$ can be decomposed into two disjoint sets $V_+(\Phi)$ and $V_-(\Phi)$ such that each of them contains elements behaving "essentially in the same way" when multiplying elements of the ideals of M captured by Φ, while elements of $V_+(\Phi)$ behave "inversely" to the elements of $V_-(\Phi)$. More precisely, this means that there is an exponent $\beta(\Phi,a)\in\mathbb{N}\setminus\{0\}$ for each $a\in V_\pm(\Phi)$ such that one has $z_\Phi a^{\beta(\Phi,a)}=_M z_\Phi b^{\beta(\Phi,b)}$ for all $a,b\in V_+(\Phi)$, as well as for all $a,b\in V_-(\Phi)$ – while on the other hand, one has $z_\Phi a^{\beta(\Phi,a)}b^{\beta(\Phi,b)}=_M z_\Phi$ for all $a\in V_+(\Phi)$ and $b\in V_-(\Phi)$. Here, the exponents $\beta(\Phi,a)$ can be thought of as "exchange rates" between particular elements of $V_\pm(\Phi)$.

In fact, we actually prove a slightly more general statement, in which the word z_Φ is replaced by $z_{\Phi'}$ for an arbitrary (A,m)-vector Φ' such that $V_\pm(\Phi')\subseteq V_\pm(\Phi)$ and $a,b\in V_\pm(\Phi')$. This generalisation is essential for our later construction of a pushdown automaton recognising the word problem of M.

Similarly as above, Rédei's theorem guarantees that the following lemma applies to all *finitely generated* commutative monoids not containing the free commutative monoid on two generators as a submonoid.

Lemma 3.7. *Let M be a commutative monoid, finitely presented by $M=\langle A\mid\varrho\rangle$ for some $A\subseteq M$ and $\varrho\subseteq A^*\times A^*$, that does not contain a submonoid isomorphic to $(\mathbb{N}\times\mathbb{N},+)$. Let m,α be as in Lemma 3.5, and Φ an (A,m)-vector. Then $V_\pm(\Phi)$ can be written as a disjoint union $V_\pm(\Phi)=V_+(\Phi)\cup V_-(\Phi)$ such that the following holds for some $\beta(\Phi,a)\in\mathbb{N}\setminus\{0\}$ satisfying $\beta(\Phi,a)\geq\ell(M,A,\varrho)$ for each $a\in V_\pm(\Phi)$, and for all (A,m)-vectors Φ' with $V_\pm(\Phi')\subseteq V_\pm(\Phi)$:*

(*i*) *For every $a,b\in V_+(\Phi)\cap V_\pm(\Phi')$, one has*
$$z_{\Phi'}a^{\beta(\Phi,a)}=_M z_{\Phi'}b^{\beta(\Phi,b)}.$$
(*ii*) *For every $a,b\in V_-(\Phi)\cap V_\pm(\Phi')$, one has*
$$z_{\Phi'}a^{\beta(\Phi,a)}=_M z_{\Phi'}b^{\beta(\Phi,b)}.$$
(*iii*) *For every $a\in V_+(\Phi)\cap V_\pm(\Phi')$ and $b\in V_-(\Phi)\cap V_\pm(\Phi')$, one has*
$$z_{\Phi'}a^{\beta(\Phi,a)}b^{\beta(\Phi,b)}=_M z_{\Phi'}.$$

Proof. Given $a, b \in A$, let us write aEb if

$$(a^m b^m)\, a^s =_M (a^m b^m)\, b^t \tag{3.9}$$

for some $s, t \in \mathbb{N} \setminus \{0\}$, and aIb if

$$(a^m b^m)\, a^s b^t =_M (a^m b^m) \tag{3.10}$$

for some $s, t \in \mathbb{N} \setminus \{0\}$. This gives us two binary relations E and I on A. Let us begin by collecting a few basic properties of these two relations.

Claim 1. Every pair of elements $a, b \in V_{\pm}(\Phi)$ is in *precisely one* of the relations E and I.

Proof. Consider an arbitrary (A, m)-vector Φ and the set $V_{\pm}(\Phi)$. Let $a, b \in V_{\pm}(\Phi)$, and $q = q_{a,b}$ be as in Proposition 3.3, so that $m \geq q$. An equality $(a^q b^q)\, a^s =_M (a^q b^q)$ for $s \in \mathbb{N} \setminus \{0\}$ would imply $z_\Phi a^s =_M z_\Phi$ by Proposition 3.2, and in the same way, an equality $(a^q b^q)\, b^t =_M (a^q b^q)$ would imply $z_\Phi b^t =_M z_\Phi$. In both cases, we would have $a \in V_\times(\Phi)$ or $b \in V_\times(\Phi)$, contradicting $a, b \in V_{\pm}(\Phi) = V(\Phi) \setminus V_\times(\Phi)$. As a result, it follows by Proposition 3.3 that aEb or aIb has to hold.

Observe that by Proposition 3.2,

$$z_\Phi a^s =_M z_\Phi b^t \tag{3.11}$$

holds for some $s, t \in \mathbb{N} \setminus \{0\}$ whenever aEb and

$$z_\Phi a^p b^r =_M z_\Phi \tag{3.12}$$

holds for some $p, r \in \mathbb{N} \setminus \{0\}$ whenever aIb. Moreover, the equalities (3.11) and (3.12) cannot hold simultaneously for $a, b \in V_{\pm}(\Phi)$, as otherwise we obtain

$$z_\Phi a^{prst} =_M z_\Phi,$$

contradicting the assumption $a \in V_{\pm}(\Phi)$. It thus follows that every pair of elements $a, b \in V_{\pm}(\Phi)$ indeed is in precisely one of the relations E and I. □

Claim 2. The restriction of E to $V_{\pm}(\Phi) \times V_{\pm}(\Phi)$ is an equivalence relation.

Proof. Reflexivity and symmetry of E are evident. For transitivity, note that for each $a, b, c \in V_{\pm}(\Phi)$,

$$z_\Phi a^s =_M z_\Phi b^t \qquad \text{and} \qquad z_\Phi b^p =_M z_\Phi c^r$$

for $p, r, s, t \in \mathbb{N} \setminus \{0\}$ give

$$z_\Phi a^{ps} =_M z_\Phi c^{rt},$$

which, in the same way as in the proof of Claim 1, implies that aIc cannot hold. We can thus conclude that aEc. □

Claim 3. If $aEbIcEd$ for some $a, b, c, d \in V_{\pm}(\Phi)$, then aId.

Proof. By Claim 1, we would have aEd otherwise. This would imply bEc by transitivity and symmetry of E, contradicting bIc. □

Claim 4. If $aIbIc$ for some $a, b, c \in V_{\pm}(\Phi)$, then aEc.

Proof. The assumption $aIbIc$ for $a, b, c \in V_{\pm}(\Phi)$ means that

$$z_\Phi a^s b^t =_M z_\Phi \qquad \text{and} \qquad z_\Phi b^p c^r =_M z_\Phi$$

for some $p, r, s, t \in \mathbb{N} \setminus \{0\}$. We thus get

$$z_\Phi a^{sp} =_M z_\Phi a^{sp} b^{tp} c^{tr} =_M z_\Phi c^{tr},$$

and aIc cannot hold for the same reason as in the proof of Claim 1. As a result, we obtain aEc by Claim 1. □

We are now prepared to define the sets $V_+(\Phi)$ and $V_-(\Phi)$. If $V_{\pm}(\Phi)$ is empty, take $V_+(\Phi) = V_-(\Phi) = \emptyset$, and we are done. Otherwise take an arbitrary element $a_\Phi \in V_{\pm}(\Phi)$, and set

$$V_+(\Phi) = \{b \in V_{\pm}(\Phi) \mid a_\Phi E b\} \qquad \text{and} \qquad V_-(\Phi) = V_{\pm}(\Phi) \setminus V_+(\Phi).$$

It follows by Claims 1 to 4 that aEb holds for all $a, b \in V_+(\Phi)$ and all $a, b \in V_-(\Phi)$, while aIb holds for all $a \in V_+(\Phi)$ and $b \in V_-(\Phi)$. Thus by (3.11) and (3.12), there are exponents $s(b), t(b) \in \mathbb{N} \setminus \{0\}$ for each $b \in V_{\pm}(\Phi) \setminus \{a_\Phi\}$ such that

$$z_\Phi a_\Phi^{s(b)} =_M z_\Phi b^{t(b)}$$

holds for all $b \in V_+(\Phi) \setminus \{a_\Phi\}$ and

$$z_\Phi a_\Phi^{s(b)} b^{t(b)} =_M z_\Phi$$

holds for all $b \in V_-(\Phi)$. Let $\gamma(\Phi, a_\Phi)$ be the least common multiple of $s(b)$ over all $b \in V_{\pm}(\Phi) \setminus \{a_\Phi\}$, and for all $b \in V_{\pm}(\Phi) \setminus \{a_\Phi\}$, let

$$\gamma(\Phi, b) = \frac{t(b)\,\gamma(\Phi, a_\Phi)}{s(b)}.$$

Then clearly

$$z_\Phi a_\Phi^{\gamma(\Phi, a_\Phi)} =_M z_\Phi b^{\gamma(\Phi, b)}$$

holds for all $b \in V_+(\Phi) \setminus \{a_\Phi\}$, and transitivity of $=_M$ gives

$$z_\Phi a^{\gamma(\Phi, a)} =_M z_\Phi b^{\gamma(\Phi, b)} \tag{3.13}$$

for all $a, b \in V_+(\Phi)$. Similarly,

$$z_\Phi a_\Phi^{\gamma(\Phi, a_\Phi)} b^{\gamma(\Phi, b)} =_M z_\Phi$$

has to hold for all $b \in V_-(\Phi)$, and $=_M$ being a congruence together with (3.13) gives

$$z_\Phi a^{\gamma(\Phi,a)} b^{\gamma(\Phi,b)} =_M z_\Phi \tag{3.14}$$

for all $a \in V_+(\Phi)$ and $b \in V_-(\Phi)$. Finally, for all $a, b \in V_-(\Phi)$, we have

$$z_\Phi a^{\gamma(\Phi,a)} =_M z_\Phi a^{\gamma(\Phi,a)} a_\Phi^{\gamma(\Phi,a_\Phi)} b^{\gamma(\Phi,b)} =_M z_\Phi b^{\gamma(\Phi,b)}.$$

Transitivity of $=_M$ thus implies

$$z_\Phi a^{\gamma(\Phi,a)} =_M z_\Phi b^{\gamma(\Phi,b)} \tag{3.15}$$

for all $a, b \in V_-(\Phi)$.

To conclude the proof, recall that (3.9) or (3.10) has to hold for each $a, b \in V_\pm(\Phi)$ and some $s, t \in \mathbb{N} \setminus \{0\}$ by virtue of Claim 1. Let σ be the least common multiple of such $s, t \in \mathbb{N} \setminus \{0\}$ over all $a, b \in V_\pm(\Phi)$ and of $\max\{\ell(M, A, \varrho), 1\}$. For all $a, b \in V_+(\Phi)$, we have aEb, so (3.9) gives

$$(a^m b^m)\, a^{\sigma\,\gamma(\Phi,a)} =_M (a^m b^m)\, b^\tau$$

for some $\tau \in \mathbb{N} \setminus \{0\}$. By the preceding equality, Proposition 3.2, and (3.13), we have both

$$z_\Phi a^{\sigma\,\gamma(\Phi,a)} =_M z_\Phi b^\tau \qquad \text{and} \qquad z_\Phi a^{\sigma\,\gamma(\Phi,a)} =_M z_\Phi b^{\sigma\,\gamma(\Phi,b)},$$

which in turn implies

$$z_\Phi b^\tau =_M z_\Phi b^{\sigma\,\gamma(\Phi,b)}.$$

As $b \in V_\pm(\Phi)$, necessarily $\tau = \sigma\,\gamma(\Phi, b)$. We thus obtain the equality

$$(a^m b^m)\, a^{\sigma\,\gamma(\Phi,a)} =_M (a^m b^m)\, b^{\sigma\,\gamma(\Phi,b)}. \tag{3.16}$$

The same reasoning using (3.15) instead of (3.13) can be applied to prove that

$$(a^m b^m)\, a^{\sigma\,\gamma(\Phi,a)} =_M (a^m b^m)\, b^{\sigma\,\gamma(\Phi,b)} \tag{3.17}$$

holds for all $a, b \in V_-(\Phi)$. Finally, given $a \in V_+(\Phi)$ and $b \in V_-(\Phi)$, we have aIb, so that (3.10) gives

$$(a^m b^m)\, a^{\sigma\,\gamma(\Phi,a)} b^\tau =_M (a^m b^m)$$

for some $\tau \in \mathbb{N} \setminus \{0\}$. Together with Proposition 3.2 and (3.14), this implies that we have both

$$z_\Phi a^{\sigma\,\gamma(\Phi,a)} b^\tau =_M z_\Phi \qquad \text{and} \qquad z_\Phi a^{\sigma\,\gamma(\Phi,a)} b^{\sigma\,\gamma(\Phi,b)} =_M z_\Phi.$$

Hence

$$z_\Phi a^{\sigma\,\gamma(\Phi,a)} b^\tau =_M z_\Phi a^{\sigma\,\gamma(\Phi,a)} b^{\sigma\,\gamma(\Phi,b)},$$

and as $b \in V_\pm(\Phi)$, surely $\tau = \sigma\,\gamma(\Phi, b)$. As a consequence, we obtain

$$(a^m b^m)\, a^{\sigma\,\gamma(\Phi,a)} b^{\sigma\,\gamma(\Phi,b)} =_M (a^m b^m). \tag{3.18}$$

It now suffices to take

$$\beta(\Phi, a) = \sigma\,\gamma(\Phi, a)$$

for each $a \in V_{\pm}(\Phi)$. Then, given an (A, m)-vector Φ' with $V_{\pm}(\Phi') \subseteq V_{\pm}(\Phi)$, the equality (3.16) together with Proposition 3.2 implies

$$z_{\Phi'} a^{\beta(\Phi,a)} =_M z_{\Phi'} b^{\beta(\Phi,b)}$$

for all $a, b \in V_{+}(\Phi) \cap V_{\pm}(\Phi')$, proving (i). Similarly, (3.17) gives

$$z_{\Phi'} a^{\beta(\Phi,a)} =_M z_{\Phi'} b^{\beta(\Phi,b)}$$

for all $a, b \in V_{-}(\Phi) \cap V_{\pm}(\Phi')$, proving (ii). Finally, (3.18) gives

$$z_{\Phi'} a^{\beta(\Phi,a)} b^{\beta(\Phi,b)} =_M z_{\Phi'}$$

for every $a \in V_{+}(\Phi) \cap V_{\pm}(\Phi')$ and $b \in V_{-}(\Phi) \cap V_{\pm}(\Phi')$, hence (iii) is proved as well. At the same time, all exponents $\beta(\Phi, a) \geq 1$ with $a \in V_{\pm}(\Phi)$ are divisible by σ, and σ is divisible by $\max\{\ell(M, A, \varrho), 1\}$. Thus $\beta(\Phi, a) \geq \ell(M, A, \varrho)$ for all $a \in V_{\pm}(\Phi)$. □

We now prove that in case two words $u, v \in A^*$ evaluate to the same element of a commutative monoid M finitely generated by A as above, the sets $A_{\times}(\Phi^{(m)}_{A,u})$, $A_{\times}(\Phi^{(m)}_{A,v})$ corresponding to the respective (A, m)-vectors coincide.

Proposition 3.8. *Let M be a commutative monoid, generated by a finite set $A \subseteq M$, that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$; let m be as in Lemma 3.5. Then*

$$A_{\times}(\Phi^{(m)}_{A,u}) = A_{\times}(\Phi^{(m)}_{A,v})$$

for all $u, v \in A^*$ such that $u =_M v$.

Proof. If $a \in A_{\times}(\Phi^{(m)}_{A,u})$, then Corollary 3.6 implies that $ua^s =_M ua^t$ for some $s, t \in \mathbb{N}$ with $s > t$, and thus

$$va^s =_M ua^s =_M ua^t =_M va^t,$$

so that $a \in A_{\times}(\Phi^{(m)}_{A,v})$. By interchanging the roles of the words u, v, one obtains the remaining inclusion as well. □

Let us finally prove that with M as above and for $w \in A^*$ fixed, one can find an (A, m)-vector Φ such that $V_{\pm}(\Phi^{(m)}_{A,x})$ is contained in $V_{\pm}(\Phi)$ for all $x \in A^*$ that evaluate to the same element of M as w. This means that by Lemma 3.7, we can classify the elements of the alphabet $V_{\pm}(\Phi^{(m)}_{A,x})$ consistently for all such x as "positive" or "negative" based on whether they belong to $V_{+}(\Phi)$ or to $V_{-}(\Phi)$. This observation is essential for our later construction of a pushdown automaton recognising the language $\mathrm{WP}_A(M)$ for an arbitrary finitely generated monoid M not containing the free commutative monoid on two generators as a submonoid.

Lemma 3.9. *Let M be a commutative monoid, generated by a finite set $A \subseteq M$, that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$; let m be as in Lemma 3.5. Then for every $w \in A^*$, there exists an (A, m)-vector Φ such that*

$$V_\pm(\Phi^{(m)}_{A,x}) \subseteq V_\pm(\Phi)$$

for all $x \in A^*$ satisfying $x =_M w$.

Proof. Let $x_1, \ldots, x_n \in A^*$ be words such that $x_k =_M w$ for $k = 1, \ldots, n$, and for all $x \in A^*$ satisfying $x =_M w$, one has

$$V_\pm(\Phi^{(m)}_{A,x}) \subseteq \bigcup_{k=1}^{n} V_\pm(\Phi^{(m)}_{A,x_k}) =: V_\pm. \tag{3.19}$$

Let $V_\pm = \{a_1, \ldots, a_r\}$, and for $k = 1, \ldots, n$ and $j = 0, \ldots, r$, let

$$y^{(j)}_k = x_k a_1^m \ldots a_j^m.$$

For $j = 0, \ldots, r$, obviously

$$y^{(j)}_1 =_M y^{(j)}_2 =_M \ldots =_M y^{(j)}_n. \tag{3.20}$$

Moreover, for each $j \in [r]$, the letter a_j is contained in $V_\pm(\Phi^{(m)}_{A,x_{i(j)}})$ for some $i(j) \in [n]$, so that

$$\Phi^{(m)}_{A,y^{(j)}_{i(j)}} = \Phi^{(m)}_{A,y^{(j-1)}_{i(j)}}. \tag{3.21}$$

Let $y = y^{(r)}_1$ and $\Phi = \Phi^{(m)}_{A,y}$. By (3.20), (3.21), $x_1 =_M \ldots =_M x_n =_M w$, and Proposition 3.8, it follows that

$$A_\times(\Phi) = A_\times(\Phi^{(m)}_{A,x})$$

for every $x \in A^*$ such that $x =_M w$. In particular,

$$V_\pm \cap A_\times(\Phi) = \emptyset, \tag{3.22}$$

as otherwise there would be $a \in A$ and $k \in [n]$ such that

$$a \in V_\pm(\Phi^{(m)}_{A,x_k}) \cap A_\times(\Phi^{(m)}_{A,x_k}) = V_\pm(\Phi^{(m)}_{A,x_k}) \cap V(\Phi^{(m)}_{A,x_k}) \cap A_\times(\Phi^{(m)}_{A,x_k}),$$

which, by Lemma 3.5, rewrites as

$$a \in V_\pm(\Phi^{(m)}_{A,x_k}) \cap V_\times(\Phi^{(m)}_{A,x_k}) = \emptyset.$$

Finally, again by Lemma 3.5,

$$V_\pm(\Phi) = V(\Phi) \setminus V_\times(\Phi) = V(\Phi) \setminus (A_\times(\Phi) \cap V(\Phi)) = V(\Phi) \setminus A_\times(\Phi). \tag{3.23}$$

At the same time, as $\Phi = \Phi^{(m)}_{A,y}$, we clearly have $V_\pm \subseteq V(\Phi)$, so (3.22) and (3.23) give

$$V_\pm \subseteq V_\pm(\Phi).$$

Thus by (3.19), indeed $V_\pm(\Phi^{(m)}_{A,x}) \subseteq V_\pm(\Phi)$ for all $x \in A^*$ satisfying $x =_M w$. □

4. The Main Results

We now proceed towards the proof of the remaining implication of the characterisation of finitely generated commutative monoids with a context-free word problem: if a commutative monoid M, finitely generated by A, does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$, then $\mathrm{WP}_A(M) \in \mathscr{L}(\mathrm{CF})$.

To show that $\mathrm{WP}_A(M) \in \mathscr{L}(\mathrm{CF})$, it suffices to construct a pushdown automaton that recognises a word $u\#v^R$ with $u, v \in A^*$ if and only if u and v belong to the same congruence class of $=_M$. One might attempt to construct this automaton such that it first saves the prefix u on its pushdown store and then compares this information with the suffix v^R to see whether $u =_M v$ or not. However, such a straightforward approach might not always be possible. Consider, for instance, the monoid $M = \langle a, b \mid ab = ba, ab = 1\rangle$ isomorphic to $\mathbb{Z}$. Then each word $a^i b^j \# a^k$, for some nonnegative integers $i, j, k \geq 0$ such that $i \geq j$, should be recognised by the pushdown automaton if and only if $i - j = k$. If $a^i b^j$ was stored on the pushdown upon reading the delimiter $\#$, then this information could hardly be used to decide the equality $i - j = k$, as one only has access to the top of the pushdown store. It seems that one should maintain some information "equivalent" to a^{i-j} on the pushdown instead.

To make this idea work over any finitely generated commutative monoid M not containing the free commutative monoid on two generators, let $m \in \mathbb{N}$ be as in Lemma 3.5, and recall that given any $u \in A^*$, Lemma 3.9 guarantees existence of an (A, m)-vector Φ such that $V_\pm(\Phi^{(m)}_{A,x}) \subseteq V_\pm(\Phi)$ for all $x \in A^*$ satisfying $x =_M u$. Let

$$A^\Phi = \{w \in A^* \mid V_\pm(\Phi^{(m)}_{A,w}) \subseteq V_\pm(\Phi)\};$$

all $x \in A^*$ such that $x =_M u$ are then contained in this set. We now introduce an equivalence relation $\sim_\Phi$ on A^Φ that refines $=_M$ on this subset of A^* – i.e., words $x, y \in A^\Phi$ satisfying $x \sim_\Phi y$ always represent the same element of M. This is done such that the language $\mathrm{WP}_A(M)$ can be recognised by a pushdown automaton that first nondeterministically guesses an (A, m)-vector Φ and whose *configurations* upon reading the delimiter $\#$ subsequently *correspond to equivalence classes of* $\sim_\Phi$.

Let M be a commutative monoid, generated by a finite set $A \subseteq M$, that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$. Let m and α be as in Lemma 3.5. Let Φ be an (A, m)-vector, and $\beta(\Phi, a)$ for each $a \in V_\pm(\Phi)$ be as in Lemma 3.7. Moreover, let the decomposition $V_\pm(\Phi) = V_+(\Phi) \cup V_-(\Phi)$ be given as in Lemma 3.7 as well. For each $w \in A^\Phi$ and $\Phi' = \Phi^{(m)}_{A,w}$, let

$$f_+(w) = \sum_{a \in V_+(\Phi) \cap V_\pm(\Phi')} \left\lfloor \frac{|w|_a - m}{\beta(\Phi, a)} \right\rfloor$$

and

$$f_-(w) = \sum_{a \in V_-(\Phi) \cap V_\pm(\Phi')} \left\lfloor \frac{|w|_a - m}{\beta(\Phi, a)} \right\rfloor.$$

Definition 4.1. Let the notation be as above. Given $x, y \in A^\Phi$, we write $x \sim_\Phi y$ if and only if all of the following conditions are satisfied:

(i) $\Phi^{(m)}_{A,x} = \Phi^{(m)}_{A,y} =: \Phi'$;
(ii) $f_+(x) - f_-(x) = f_+(y) - f_-(y)$;
(iii) $(|x|_a - m) \bmod \beta(\Phi, a) = (|y|_a - m) \bmod \beta(\Phi, a)$ for each $a \in V_\pm(\Phi')$;
(iv) $(|x|_a - m) \bmod \alpha = (|y|_a - m) \bmod \alpha$ for each $a \in V_\times(\Phi')$.

An explanation of the definition given above is in order. Lemma 3.7 implies that if there are $\Phi'(c)$ occurrences of each $c \in A$ fixed in a word $w \in A^\Phi$ with $\Phi^{(m)}_{A,w} = \Phi'$, then one may replace every other $\beta(\Phi, a)$ occurrences of $a \in V_+(\Phi) \cap V_\pm(\Phi')$ in w by $\beta(\Phi, b)$ occurrences of any other letter $b \in V_+(\Phi) \cap V_\pm(\Phi')$, and *vice versa* – the word still represents the same element of M after such a replacement. A similar replacement can be done with any two letters $a, b \in V_-(\Phi) \cap V_\pm(\Phi')$. Moreover, one may replace every other $\beta(\Phi, a)$ occurrences of $a \in V_+(\Phi) \cap V_\pm(\Phi')$ together with $\beta(\Phi, b)$ occurrences of $b \in V_-(\Phi) \cap V_\pm(\Phi')$ by the empty word, as well as newly introduce $\beta(\Phi, a)$ occurrences of a and $\beta(\Phi, b)$ occurrences of b – every $\beta(\Phi, a)$ occurrences of a and $\beta(\Phi, b)$ occurrences of b, beyond the first m occurrences of both, are "inverse to each other". Finally, one is free to delete every other α occurrences of $a \in V_\times(\Phi')$, as well as to newly introduce such occurrences.

The idea of Definition 4.1 is to identify *precisely* the words from A^Φ that can be obtained one from each other by a sequence of above-described replacements. First of all, for words $x, y \in A^\Phi$ to satisfy $x \sim_\Phi y$, the (A, m)-vectors $\Phi^{(m)}_{A,x}$ and $\Phi^{(m)}_{A,y}$ have to be the same (A, m)-vector Φ' – this is expressed by condition (i). Next, given the set $V(\Phi')$ of all $a \in A$ appearing at least m times in any of the words x, y, one may count for each $a \in V_+(\Phi) \cap V_\pm(\Phi')$ how many disjoint selections of $\beta(\Phi, a)$ occurrences of a can one make in these words, ignoring some fixed m occurrences of a in each word. Summing over all $a \in V_+(\Phi) \cap V_\pm(\Phi')$, one obtains the values $f_+(x)$ and $f_+(y)$, and via the same procedure for symbols from $V_-(\Phi) \cap V_\pm(\Phi')$, one obtains $f_-(x)$ and $f_-(y)$. As $\beta(\Phi, a)$ occurrences of a and $\beta(\Phi, b)$ occurrences of b are "equivalent" if a, b are both in $V_+(\Phi) \cap V_\pm(\Phi')$ or in $V_-(\Phi) \cap V_\pm(\Phi')$, and "inverse" to each other if $a \in V_+(\Phi) \cap V_\pm(\Phi')$ and $b \in V_-(\Phi) \cap V_\pm(\Phi')$, this leads to the condition (ii). Furthermore, the numbers of occurrences of particular symbols that remain after performing the described selections have to be the same in both words, which is expressed by the condition (iii). Finally, provided m occurrences of $a \in V_\times(\Phi')$ are fixed, the number of remaining occurrences of a modulo α has to be the same in both words, which is condition (iv).

Example 4.2. Consider the monoid $M = \langle a, b \mid ab = ba, ab = 1 \rangle$ isomorphic to $\mathbb{Z}$ and $A = \{a, b\}$. One can take $m = 0$, so there is only one (A, m)-vector $\Phi = (0, 0)$. Let $V_+(\Phi) = \{a\}$, $V_-(\Phi) = \{b\}$, and $\beta(\Phi, a) = \beta(\Phi, b) = 2$. Then $f_+(w) = \lfloor |w|_a/2 \rfloor$ and $f_-(w) = \lfloor |w|_b/2 \rfloor$ for all $w \in A^*$. Words $x, y \in A^\Phi = A^*$ thus satisfy $x \sim_\Phi y$ if and only if $\lfloor |x|_a/2 \rfloor - \lfloor |x|_b/2 \rfloor = \lfloor |y|_a/2 \rfloor - \lfloor |y|_b/2 \rfloor$, $|x|_a \equiv |y|_a \pmod 2$, and $|x|_b \equiv |y|_b \pmod 2$.

Proposition 4.3. *With definitions and notation as above, $\sim_\Phi$ is an equivalence relation on A^Φ and $x \sim_\Phi y$ implies $x =_M y$ for all $x, y \in A^\Phi$.*

Proof. It is obvious that $\sim_\Phi$ is an equivalence relation on A^Φ. Consider any $x, y \in A^\Phi$ such that $x \sim_\Phi y$. Then $\Phi^{(m)}_{A,x} = \Phi^{(m)}_{A,y} = \Phi'$ by the condition (i) of Definition 4.1, which in particular implies $|x|_a = |y|_a$ for all $a \in A \setminus V(\Phi')$. Moreover, the condition (iv) of Definition 4.1 means that removing α occurrences of every symbol $a \in V_\times(\Phi')$ in both words x, y as many times as possible while retaining the inequalities $|x|_a \geq m$ and $|y|_a \geq m$ yields words $x', y' \in A^\Phi$ such that $\Phi^{(m)}_{A,x'} = \Phi^{(m)}_{A,y'} = \Phi'$ and $x' \sim_\Phi x \sim_\Phi y \sim_\Phi y'$, while at the same time it follows by Lemma 3.5 and Proposition 3.2 used together with the commutativity of M that

$$x =_M x' \qquad \text{and} \qquad y' =_M y. \tag{4.1}$$

At the same time, we have $|x'|_a = |y'|_a$ for each $a \in V_\times(\Phi') \cup (A \setminus V(\Phi'))$.

Let us finally denote the common value of $f_+(x') - f_-(x')$ and $f_+(y') - f_-(y')$ by d. If $d > 0$, there surely exists at least one $a_+ \in V_+(\Phi) \cap V_\pm(\Phi')$, and if $d < 0$, there has to be at least one $a_- \in V_-(\Phi) \cap V_\pm(\Phi')$. Delete precisely

$$\beta(\Phi, a) \left\lfloor \frac{|x'|_a - m}{\beta(\Phi, a)} \right\rfloor$$

occurrences of each $a \in V_\pm(\Phi')$ in x'. If $d > 0$, add precisely $d\beta(\Phi, a_+)$ occurrences of a_+ to x' and if $d < 0$, add precisely $-d\beta(\Phi, a_-)$ occurrences of a_-. Call the resulting word x''. Similarly, delete precisely

$$\beta(\Phi, a) \left\lfloor \frac{|y'|_a - m}{\beta(\Phi, a)} \right\rfloor$$

occurrences of each $a \in V_\pm(\Phi')$ in y'. If $d > 0$, add $d\beta(\Phi, a_+)$ occurrences of a_+ to y' and if $d < 0$, add precisely $-d\beta(\Phi, a_-)$ occurrences of a_-. Call the resulting word y''. Lemma 3.7 together with Proposition 3.2 and the commutativity of M gives $x' =_M x''$ and $y'' =_M y'$. At the same time, obviously $|x''|_a = |y''|_a$ for all $a \in V_\pm(\Phi') \cup V_\times(\Phi') \cup (A \setminus V(\Phi')) = A$, so that $\Psi(x'') = \Psi(y'')$, and commutativity of M implies $x'' =_M y''$. Thus, by (4.1), we finally obtain

$$x =_M x' =_M x'' =_M y'' =_M y' =_M y,$$

completing the proof. □

We are now prepared to show that when a finitely generated commutative monoid M does not contain the free commutative monoid on two generators as a submonoid, the word problem of M can always be recognised by a pushdown automaton, and is thus context-free.

Let the notation be the same as above in this section. The pushdown automaton constructed in the proof of the following proposition nondeterministically guesses an (A, m)-vector Φ at the beginning of its run. Configurations of this automaton upon reading the delimiter symbol # then correspond to equivalence classes of the relation $\sim_\Phi$ on A^Φ.

We now describe how such a configuration corresponding to the equivalence class $[u]$ of $\sim_\Phi$ containing a word $u \in A^\Phi$ looks like.

The following information is stored *in a state* of the automaton in a configuration corresponding to $[u]$; a *finite state set* is obviously sufficient to do so:

- The (A,m)-vector $\Phi^{(m)}_{A,u} =: \overline{\Phi}$; indeed, note that there are only finitely many different (A,m)-vectors, and storing them in a state is thus possible. By Definition 4.1, one also has $\Phi^{(m)}_{A,x} = \overline{\Phi}$ for every $x \in [u]$.
- The value $(|u|_a - m) \bmod \beta(\Phi, a) =: r_a$ for every $a \in V_\pm(\overline{\Phi})$. By Definition 4.1, one also necessarily has $(|x|_a - m) \bmod \beta(\Phi, a) = r_a$ for every $x \in [u]$ and each $a \in V_\pm(\overline{\Phi})$.
- The value $(|u|_a - m) \bmod \alpha =: s_a$ for each $a \in V_\times(\overline{\Phi})$. By Definition 4.1, one also always has $(|x|_a - m) \bmod \alpha = s_a$ for every $x \in [u]$ and each $a \in V_\times(\overline{\Phi})$.
- The sign $\operatorname{sgn}(f_+(u) - f_-(u)) =: \sigma$. By Definition 4.1, one also always has $\operatorname{sgn}(f_+(x) - f_-(x)) = \sigma$ for every $x \in [u]$.

Moreover, the following is stored *on the pushdown*:

- The absolute value $|f_+(u) - f_-(u)| =: d_+$ in unary, i.e., the word $\dashv Z^{d_+}$ for some pushdown symbol Z and the bottom-of-pushdown symbol $\dashv$. By Definition 4.1, one also always necessarily has $|f_+(x) - f_-(x)| = d_+$ for every $x \in [u]$.

Note that the configuration of the automaton corresponding to the equivalence class $[u]$ is unambiguously determined by any $x \in [u]$ and, in particular, by u itself. On the other hand, every configuration unambiguously determines the corresponding equivalence class $[u]$; this configuration is then "shared by all $x \in [u]$", so the word u itself is not unambiguously determined.

We can now finally proceed to the actual main results of this article. The construction of a pushdown automaton in the proof of the following proposition is largely presented in terms of its configurations taking the form described above.

Proposition 4.4. *Let M be a finitely generated commutative monoid that does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$. The word problem of M is then context-free.*

Proof. Let A, m, α, and $\beta(\Phi, a)$ for every (A,m)-vector Φ and $a \in V_\pm(\Phi)$ be as above. Let $\varrho \subseteq A^* \times A^*$ be a finite relation such that $M = \langle A \mid \varrho \rangle$, while $\alpha \geq \ell(M, A, \varrho)$ and $\beta(\Phi, a) \geq \ell(M, A, \varrho)$ for every (A,m)-vector Φ and $a \in V_\pm(\Phi)$. We describe a construction of a pushdown automaton $\mathcal{A}$ recognising the language $\mathrm{WP}_A(M)$. As the class $\mathscr{L}(\mathrm{CF})$ is closed under intersection with a rational language and the language $A^*\#A^*$ with $\# \notin A$ is rational, we assume without loss of generality that every input word of $\mathcal{A}$ takes the form $u\#v^R$ for some $u, v \in A^*$.

High-Level Description of $\mathcal{A}$. We construct the automaton $\mathcal{A}$ such that its run on a word $u\#v^R$ consists of the following four phases:

(1) The automaton nondeterministically guesses two (A,m)-vectors Φ' and Φ such that $\Phi^{(m)}_{A,u} = \Phi'$ and $V_{\pm}(\Phi^{(m)}_{A,x}) \subseteq V_{\pm}(\Phi)$ for all $x \in A^*$ satisfying $x =_M u$. (Existence of such Φ is guaranteed by Lemma 3.9.) The automaton actually only chooses among the pairs of (A,m)-vectors (Φ',Φ) satisfying $V_{\pm}(\Phi') \subseteq V_{\pm}(\Phi)$, as this has to be the case whenever the guess is correct. Note that the set of all such (Φ',Φ) is finite, so that the guess amounts to a single nondeterministic choice among finitely many transitions, while the chosen (A,m)-vectors Φ',Φ can be stored in a state.
(2) The prefix $u\#$ of the input word is read by $\mathcal{A}$, and the initial guess of Φ' is verified. In case this turns out to be incorrect, the corresponding run of $\mathcal{A}$ rejects the input. Otherwise surely $u \in A^{\Phi}$ – regardless of whether the initial guess of Φ was correct or not – and the automaton happens to be in a configuration corresponding to the equivalence class $[u]$ of $\sim_{\Phi}$.
(3) In case the input was not rejected so far, this configuration is nondeterministically transformed to a configuration corresponding to $[x]$ for some $x \in A^{\Phi}$ such that $x =_M u$. This is done in a way that in case the initial guess of Φ was correct – and every $x \in A^*$ satisfying $x =_M u$ is thus in A^{Φ} – then configurations corresponding to all possible $[x]$ for such x can indeed be reached; one can thus also obtain all configurations corresponding to $[x]$ with $x \in A^*$ such that $x =_M u$. The aim in this step is to nondeterministically guess a configuration corresponding to $[x]$ such that $[x] = [v]$.
(4) Going through the suffix v^R, the guess of the equivalence class from the preceding step is verified, i.e., the automaton checks whether indeed $[x] = [v]$. If so, the run of $\mathcal{A}$ accepts; otherwise it rejects.

Correctness of $\mathcal{A}$. It is easy to see that a pushdown automaton $\mathcal{A}$ conforming to the high-level specification above necessarily recognises the language $\mathrm{WP}_A(M)$. Indeed, if $u\#v^R \in \mathrm{WP}_A(M)$, we have $u =_M v$. By Lemma 3.9, the automaton $\mathcal{A}$ guesses Φ' and Φ such that $\Phi^{(m)}_{A,u} = \Phi'$ and $V_{\pm}(\Phi^{(m)}_{A,x}) \subseteq V_{\pm}(\Phi)$ for all $x \in A^*$ with $x =_M u$ in at least one of its runs. After reading $u\#$ and arriving at a configuration corresponding to $[u]$, one can transform $[u]$ to $[v]$ and finally verify the guess of $[v]$ while reading the suffix. If, on the contrary, $u\#v^R \notin \mathrm{WP}_A(M)$, then either the initial guess of Φ' is not correct, or the configuration corresponding to $[u]$ cannot be transformed to a one corresponding to $[v]$ since $v \neq_M u$. The input word is rejected by the run in both cases, hence it is rejected by all runs of $\mathcal{A}$.

Note in particular that the initial nondeterministic guess of Φ does not have to be verified – the input might even get accepted for some wrong guesses. In case both Φ' and Φ are guessed incorrectly, the corresponding run rejects in Step 2. If Φ' is guessed correctly and Φ with $V_{\pm}(\Phi') \subseteq V_{\pm}(\Phi)$ is guessed incorrectly, then one obtains a smaller set of equivalence classes reachable in Step 3. However, if a con-

figuration corresponding to an equivalence class $[x]$ is obtained in Step 3, then it is correct – in the sense that $x =_M u$ – regardless of the correctness of the guessed (A, m)-vector Φ. This means that exclusively words from $\mathrm{WP}_A(M)$ can be accepted, regardless of the initial guess of Φ; if, however, Φ was guessed correctly, then *all* words from $\mathrm{WP}_A(M)$ are accepted by at least one run.

Detailed Description of Phases (1), (2). We now describe a run of $\mathcal{A}$ in more detail. At its beginning, $\mathcal{A}$ performs a nondeterministic guess of (A, m)-vectors Φ' and Φ such that $V_{\pm}(\Phi') \subseteq V_{\pm}(\Phi)$; this amounts to choosing a pair (Φ', Φ) from a fixed finite set, and can thus be implemented directly in the transition function. The guessed (A, m)-vectors Φ' and Φ are stored in a state of the automaton.

The automaton $\mathcal{A}$ next goes through the prefix $u\#$ of the input word $u\#v^R$. During this pass through $u\#$, it gradually builds a configuration corresponding to the equivalence class $[u]$ of $\sim_\Phi$ (in case the initial guess of Φ' was correct), and at the same time it verifies the guess of Φ'.

To build the said configuration, the automaton initialises the (A, m)-vector $\overline{\Phi}$ to $\mathbf{0}\colon A \to \{0, \ldots, m\}$ such that $\mathbf{0}(a) = 0$ for each $a \in A$ (if the guess of Φ' was correct, $\overline{\Phi}$ has to eventually become Φ' after reading u). Moreover, it sets to 0 the values r_a for all $a \in V_{\pm}(\Phi')$, the values s_a for all $a \in V_{\times}(\Phi')$, and the sign σ. The value d_+ is initialised to zero as well, so that the pushdown store only contains the bottom-of-pushdown symbol $\dashv$ at the beginning of the run. The automaton then goes through u and upon each of its symbols a, the following is performed:

- If $\overline{\Phi}(a) < \Phi'(a) \leq m$, increase $\overline{\Phi}(a)$ by one.
- If $\overline{\Phi}(a) = \Phi'(a) < m$, reject.
- If $\overline{\Phi}(a) = \Phi'(a) = m$, $a \in V_{\pm}(\Phi')$, and $r_a < \beta(\Phi, a) - 1$, increase r_a by one.
- If $\overline{\Phi}(a) = \Phi'(a) = m$, $a \in V_{\pm}(\Phi')$, and $r_a = \beta(\Phi, a) - 1$, set r_a to 0. Moreover:
  a) If $\sigma = 0$, set σ to 1 if $a \in V_+(\Phi)$ and to -1 if $a \in V_-(\Phi)$; in both cases push one symbol Z, so that d_+ increases by one.
  b) If $\sigma = 1$, push Z (i.e., increase d_+) if $a \in V_+(\Phi)$, and pop Z (i.e., decrease d_+) if $a \in V_-(\Phi)$. If $\dashv$ appears on the top of the pushdown in the latter case (i.e., d_+ was decreased to 0), set σ to 0.
  c) If $\sigma = -1$, push Z (i.e., increase d_+) if $a \in V_-(\Phi)$, and pop Z (i.e., decrease d_+) if $a \in V_+(\Phi)$. If $\dashv$ appears on the top of the pushdown in the latter case (i.e., d_+ was decreased to 0), set σ to 0.
- If $\overline{\Phi}(a) = \Phi'(a) = m$, $a \in V_{\times}(\Phi')$, and $s_a < \alpha - 1$, increase s_a by one.
- If $\overline{\Phi}(a) = \Phi'(a) = m$, $a \in V_{\times}(\Phi')$, and $s_a = \alpha - 1$, set s_a to 0.

After reading the delimiter $\#$, the automaton rejects in case $\overline{\Phi} \neq \Phi'$. If on the other hand $\overline{\Phi} = \Phi'$, the initial guess of Φ' was necessarily correct, and the automaton is in a valid configuration corresponding to the equivalence class $[u]$ of $\sim_\Phi$.

Detailed Description of Phase (3). By Proposition 3.8, $A_\times(\Phi^{(m)}_{A,x}) = A_\times(\Phi')$ for every $x \in A^*$ such that $x =_M u$. Let us denote this common alphabet by $A_\times$ and let $A_\pm := A \setminus A_\times$; as there are only finitely many (A,m)-vectors, the automaton can store both alphabets for all Φ' in a state. Lemma 3.5 gives $V_\times(\Phi^{(m)}_{A,x}) = V(\Phi^{(m)}_{A,x}) \cap A_\times$ and $V_\pm(\Phi^{(m)}_{A,x}) = V(\Phi^{(m)}_{A,x}) \cap A_\pm$ for every word $x \in A^*$ satisfying $x =_M u$.

Provided the guess of Φ' was verified as correct and $\mathcal{A}$ is in a valid configuration corresponding to $[u]$ after reading #, the automaton nondeterministically transforms this configuration to some of the configurations corresponding to $[x]$ with $x \in A^\Phi$ such that $x =_M u$. To describe how precisely is this done, interpret ϱ as a rewriting system (A, ϱ), so that for $w, w' \in A^*$ we write $w \longrightarrow w'$ if and only if $w = w_1 y w_2$ and $w' = w_1 y' w_2$ for some $w_1, w_2, y, y' \in A^*$ such that $(y, y') \in \varrho$. If we denote by $\longleftrightarrow$ the symmetric closure of $\longrightarrow$ and by $\overset{*}{\longleftrightarrow}$ the reflexive and transitive closure of $\longleftrightarrow$, then $\overset{*}{\longleftrightarrow}$ is the same as the congruence $=_M$.

After reading #, the automaton $\mathcal{A}$ is allowed to make several nondeterministic steps, in which it reads nothing from the input, and in which the current configuration corresponding to some $[x_1]$ can be transformed to a configuration corresponding to $[x_2]$ for an arbitrary $x_2 \in A^\Phi$ such that

$$x_1' \longleftrightarrow x_2$$

for some $x_1' \in [x_1]$. Put differently, the equivalence class $[x_1]$ of $\sim_\Phi$ is interpreted as an *arbitrary* word $x_1' \in [x_1]$, and this word is rewritten, according to $\longleftrightarrow$, to some other word x_2 – in case $x_2 \in A^\Phi$, the automaton finds itself in a configuration corresponding to the equivalence class $[x_2]$. As always $x_1 =_M x_1' =_M x_2$ and $x_2 \in A^\Phi$, the configurations reachable by a *sequence* of such steps all correspond to $[x]$ for some $x \in A^\Phi$ satisfying $x =_M u$; on the other hand, as $\overset{*}{\longleftrightarrow}$ equals $=_M$, *all* configurations corresponding to $[x]$ for $x \in A^*$ such that $x =_M u$ are reachable by at least one sequence of steps in case Φ was guessed correctly.

Let us now take a closer look at one such step, in which a configuration corresponding to $[x_1]$ is transformed to a one for $[x_2]$ such that $x_2 \in A^\Phi$ and $x_1' \longleftrightarrow x_2$ for some $x_1' \in [x_1]$, and clarify how this can be implemented by the pushdown automaton $\mathcal{A}$. What needs to be done is to nondeterministically choose some $(y, y') \in \varrho \cup \varrho^{-1}$ and some $x_1' \in [x_1]$ containing the factor y, to replace this factor by y' and obtain a configuration corresponding to the equivalence class of the resulting word. We perform this transformation modulo commutativity, essentially dealing with Parikh vectors of the words x_1', y, y' only.

Observe that for every $x_1' \in [x_1]$ and for the corresponding configuration given by $\overline{\Phi} = \Phi^{(m)}_{A,x_1}$, the values r_a for $a \in V_\pm(\overline{\Phi})$, the values s_a for $a \in V_\times(\overline{\Phi})$, the sign σ, and the value d_+, the set $V_\pm(\overline{\Phi})$ further decomposes as $V_\pm(\overline{\Phi}) = X_\pm \cup Y_\pm$, where $X_\pm$ consists of all $a \in V_\pm(\overline{\Phi})$ such that $|x_1'|_a = m + r_a$ and $Y_\pm$ consists of all $a \in V_\pm(\overline{\Phi})$ such that $|x_1'|_a \geq m + \beta(\Phi, a) + r_a$. Similarly, the alphabet $V_\times(\overline{\Phi})$ decomposes as $V_\times(\overline{\Phi}) = X_\times \cup Y_\times$, where $X_\times$ contains precisely all $a \in V_\times(\overline{\Phi})$ such that $|x_1'|_a = m + s_a$ and $Y_\times$ contains all $a \in V_\times(\overline{\Phi})$ with $|x_1'|_a \geq m + \alpha + s_a$.

Moreover, it is not hard to see that a pair of decompositions $V_\pm(\overline{\Phi}) = X_\pm \cup Y_\pm$ and $V_\times(\overline{\Phi}) = X_\times \cup Y_\times$ corresponds in this way to at least one $x_1' \in [x_1]$ if and only if the following conditions are satisfied:

(*i*) If $\sigma = 0$ and $Y_\pm \neq \emptyset$, then both $Y_\pm \cap V_+(\Phi) \neq \emptyset$ and $Y_\pm \cap V_-(\Phi) \neq \emptyset$.
(*ii*) If $\sigma = 1$, then $Y_\pm \cap V_+(\Phi) \neq \emptyset$; if moreover $Y_\pm \cap V_-(\Phi) = \emptyset$, then $d_+ \geq |Y_\pm|$.
(*iii*) If $\sigma = -1$, then $Y_\pm \cap V_-(\Phi) \neq \emptyset$; if moreover $Y_\pm \cap V_+(\Phi) = \emptyset$, then $d_+ \geq |Y_\pm|$.

The automaton $\mathcal{A}$ nondeterministically guesses a pair of decompositions $V_\pm(\overline{\Phi}) = X_\pm \cup Y_\pm$ and $V_\times(\overline{\Phi}) = X_\times \cup Y_\times$ such that the above conditions are satisfied. Note that the inequality $d_+ \geq |Y_\pm|$ in (*ii*) and (*iii*) can indeed be checked by a pushdown automaton by popping at most some constant number of symbols Z from the pushdown, finding out whether the bottom $\dashv$ was or was not reached, and pushing the removed symbols Z back to the pushdown. The remaining checks may be performed using finite information that can be stored in states of the automaton. Moreover, $\mathcal{A}$ nondeterministically chooses some $(y, y') \in \varrho \cup \varrho^{-1}$.

For each $a \in A$, the automaton $\mathcal{A}$ first modifies the configuration corresponding to $[x_1]$ by "decreasing the number of occurrences" of a by $|y|_a$. This has a slightly different meaning depending on which symbol a we currently consider:

- If $a \in A \setminus V(\overline{\Phi})$, then the value $\overline{\Phi}(a)$ is decreased by $|y|_a$ provided the result is nonnegative; should the result be negative, the rewriting step is impossible to perform.
- If $a \in X_\pm$, then r_a is decreased by $|y|_a$ in case $|y|_a \leq r_a$ holds. If on the other hand $|y|_a > r_a$ and $k := |y|_a - r_a$, then r_a is decreased to 0 and the value $\overline{\Phi}(a)$ is decreased by k provided the result is nonnegative; should the result be negative, the rewriting step cannot be performed.
- If $a \in Y_\pm$, then r_a is again decreased by $|y|_a$ in case $|y|_a \leq r_a$ holds. If $|y|_a > r_a$ and $k := |y|_a - r_a$, then r_a is decreased to 0 and new $\beta(\Phi, a)$ occurrences of a are "found" on the pushdown; k of them that correspond to the remaining occurrences of a in y are removed, and the rest is stored as the new value of r_a. More precisely, if $a \in V_+(\Phi)$, then Z is popped if $\sigma = 1$ and pushed if $\sigma \in \{0, -1\}$; in case the original value of σ was 0, it is changed to -1, and in case the last occurrence of Z on the pushdown was popped, σ is changed to 0. Similarly, if $a \in V_-(\Phi)$, then Z is popped if $\sigma = -1$ and pushed if $\sigma \in \{0, 1\}$; in case the original value of σ was 0, it is changed to 1, and in case the last occurrence of Z on the pushdown was popped, σ is changed to 0. The value r_a is set to $\beta(\Phi, a) - k$ in any case. Note that $\beta(\Phi, a) \geq \ell(M, A, \varrho) \geq k > 0$, so this can always be done.
- If $a \in X_\times$, then s_a is decreased by $|y|_a$ in case $|y|_a \leq s_a$. If $|y|_a > s_a$ and $k := |y|_a - s_a$, then s_a is set to 0 and $\overline{\Phi}(a)$ is decreased by k provided the result is nonnegative; should the result be negative, the rewriting step cannot be performed.

- If $a \in Y_\times$, then s_a is again decreased by $|y|_a$ in case $|y|_a \leq s_a$. If $|y|_a > s_a$ and $k := |y|_a - s_a$, then s_a is set to $\alpha - k$, which can be performed since $\alpha \geq \ell(M, A, \varrho) \geq k > 0$.

Once this "deletion" of y is finished, one has to "insert the new factor" y' and update the configuration accordingly. This is done in a similar manner as while reading the prefix u – for each letter a of y', the automaton does the following:

- If $\overline{\Phi}(a) < m$, increase $\overline{\Phi}(a)$ by one. In case the new value happens to be m, set $r_a = 0$ if $a \in A_\pm$ and $s_a = 0$ if $a \in A_\times$. If $a \in A_\pm$ and $a \notin V_\pm(\Phi)$, reject – the initial guess of the (A, m)-vector Φ was incorrect.
- If $\overline{\Phi}(a) = m$, $a \in A_\pm$, and $r_a < \beta(\Phi, a) - 1$, increase r_a by one.
- If $\overline{\Phi}(a) = m$, $a \in A_\pm$, and $r_a = \beta(\Phi, a) - 1$, set r_a to 0. Moreover:
  a) If $\sigma = 0$, then set σ to 1 if $a \in V_+(\Phi)$ and to -1 if $a \in V_-(\Phi)$; in both cases push one symbol Z, so that d_+ increases by one.
  b) If $\sigma = 1$, then push Z (i.e., increase d_+) if $a \in V_+(\Phi)$, and pop Z (i.e., decrease d_+) if $a \in V_-(\Phi)$. If the symbol $\dashv$ finds itself on the top of the pushdown in the latter case (i.e., d_+ was decreased to 0), set the sign σ to 0.
  c) If $\sigma = -1$, then push Z (i.e., increase d_+) if $a \in V_-(\Phi)$, and pop Z (i.e., decrease d_+) if $a \in V_+(\Phi)$. If the symbol $\dashv$ appears on the top of the pushdown in the latter case (i.e., d_+ was decreased to 0), set the sign σ to 0.
- If $\overline{\Phi}(a) = m$, $a \in A_\times$, and $s_a < \alpha - 1$, increase s_a by one.
- If $\overline{\Phi}(a) = m$, $a \in A_\times$, and $s_a = \alpha - 1$, set s_a to 0.

If the input was not rejected by $\mathcal{A}$ during the above procedure, $V_\pm(\overline{\Phi}) \subseteq V_\pm(\Phi)$ clearly has to hold for the resulting (A, m)-vector $\overline{\Phi}$. The automaton thus is in a valid configuration corresponding to an equivalence class $[x_2]$ of $\sim_\Phi$ for some $x_2 \in A^\Phi$ such that there is $x_1' \in [x_1]$ satisfying $x_1' \longleftrightarrow x_2$. Moreover, configurations corresponding to all such $[x_2]$ can obviously be reached.

Detailed Description of Phase (4). It remains to describe the last part of the run of $\mathcal{A}$, in which the automaton starts in a configuration corresponding to $[x]$ for some $x \in A^\Phi$ satisfying $x =_M u$, and subsequently goes through the suffix v^R to decide whether $[x] = [v]$. This check is largely "inverse" to the procedure performed while reading the prefix u. For each symbol a of v^R, the automaton updates its configuration in the following way:

- If $\overline{\Phi}(a) = 0$, reject.
- If $0 < \overline{\Phi}(a) < m$, decrease $\overline{\Phi}(a)$ by one.
- If $\overline{\Phi}(a) = m$, $a \in A_\pm$, and $r_a > 0$, decrease r_a by one.
- If $\overline{\Phi}(a) = m$, $a \in A_\pm$, and $r_a = 0$, nondeterministically either decrease $\overline{\Phi}(a)$ by one, or set r_a to $\beta(\Phi, a) - 1$ and perform the following:

a) If $\sigma = 0$, then set σ to -1 if $a \in V_+(\Phi)$ and to 1 if $a \in V_-(\Phi)$; in both cases push one symbol Z, so that d_+ increases by one.
b) If $\sigma = 1$, then pop Z (i.e., decrease d_+) if $a \in V_+(\Phi)$, and push Z (i.e., increase d_+) if $a \in V_-(\Phi)$. If $\dashv$ appears on the top of the pushdown in the former case (i.e., d_+ was decreased to 0), set σ to 0.
c) If $\sigma = -1$, then pop Z (i.e., decrease d_+) if $a \in V_-(\Phi)$, and push Z (i.e., increase d_+) if $a \in V_+(\Phi)$. If $\dashv$ appears on the top of the pushdown in the former case (i.e., d_+ was decreased to 0), set σ to 0.

- If $\overline{\Phi}(a) = m$, $a \in A_\times$, and $s_a > 0$, decrease s_a by one.
- If $\overline{\Phi}(a) = m$, $a \in A_\times$, and $s_a = 0$, nondeterministically either decrease $\overline{\Phi}(a)$ by one, or set s_a to $\alpha - 1$.

It is clear that $[x] = [v]$ if and only if after reading the suffix v^R, a configuration can be reached in which $\overline{\Phi} = \mathbf{0}$ and only the bottom-of-pushdown symbol $\dashv$ is left on the pushdown. The automaton accepts the input in this case.

This finishes the description of the pushdown automaton $\mathcal{A}$; finiteness of the state set of $\mathcal{A}$ should be evident. □

The proposition established completes the characterisation of finitely generated commutative *monoids* with a context-free word problem – the first main result of this article, which we now state explicitly.

Theorem 4.5. *Let M be a finitely generated commutative monoid. Then the word problem of M is context-free if and only if M does not contain a submonoid isomorphic to $(\mathbb{N} \times \mathbb{N}, +)$, the free commutative monoid on two generators.*

Proof. Follows directly by Proposition 3.1 and Proposition 4.4. □

We can now also easily establish the second main result of this article – the characterisation of finitely generated commutative *semigroups* with a context-free word problem. One of the implications of this characterisation is obtained in a similar way as in Proposition 3.1 for monoids. The remaining implication – that would be harder to prove on its own – follows as a direct consequence of the characterisation for monoids established above.

Theorem 4.6. *Let S be a finitely generated commutative semigroup. Then the word problem of S is context-free if and only if S does not contain a subsemigroup isomorphic to $((\mathbb{N} \times \mathbb{N}) \setminus \{(0,0)\}, +)$, the free commutative semigroup on two generators.*

Proof. If S contains a commutative subsemigroup freely generated by $a, b \in S$ and $A \subseteq S$ is a finite generating set of S such that $a, b \in A$, then similarly as in the proof of Proposition 3.1 for monoids, the language

$$\mathrm{WP}_A(S) \cap a^+b^+\#a^+b^+ = \{a^ib^j\#a^ib^j \mid i, j \in \mathbb{N} \setminus \{0\}\}$$

is not context-free, thus $\mathrm{WP}_A(S)$ cannot be context-free either.

Conversely, if the semigroup S does not contain a subsemigroup isomorphic to $((\mathbb{N} \times \mathbb{N}) \setminus \{(0,0)\}, +)$, then S^1 cannot contain a submonoid isomorphic to the free commutative monoid $(\mathbb{N}\times\mathbb{N}, +)$, as the only idempotent of this monoid is its identity element. Any finite generating set $A \subseteq S$ of the semigroup S also is a generating set of the monoid S^1. It follows by Theorem 4.5 that the language $\mathrm{WP}_A(S^1)$ is context-free, hence the language

$$\mathrm{WP}_A(S) = \mathrm{WP}_A(S^1) \cap A^+ \# A^+$$

is context-free as well. □

Acknowledgements

I would like to thank the anonymous reviewer for the useful comments and suggestions that significantly improved the presentation of this article.

Bibliography

[1] A. V. Anisimov. Group languages. *Kibernetika*, 4:18–24, 1971.
[2] Y. Antolín. On Cayley graphs of virtually free groups. *Groups Complex. Cryptol.*, 3(2):301–327, 2011.
[3] T. Brough. Inverse semigroups with rational word problem are finite. Available at `https://arxiv.org/abs/1311.3955`, 2013.
[4] T. Brough. Groups with poly-context-free word problem. *Groups Complex. Cryptol.*, 6(1):9–29, 2014.
[5] T. Brough. Word problem languages for free inverse monoids. In *Descriptional Complexity of Formal Systems, DCFS 2018*, pages 24–36. Springer, 2018.
[6] T. Brough and A. J. Cain. A language hierarchy of binary relations. *Inf. Comput.*, 275:104607, 2020.
[7] T. Brough, A. J. Cain, and M. Pfeiffer. Context-free word problem semigroups. In *Developments in Language Theory, DLT 2019*, pages 292–305, 2019.
[8] V. Diekert and A. Weiß. Context-free groups and their structure trees. *Int. J. Algebra Comput.*, 23(3):611–642, 2013.
[9] V. Diekert and A. Weiß. Context-free groups and Bass-Serre theory. In J. González-Meneses, M. Lustig, and E. Ventura, editors, *Algorithmic and Geometric Topics Around Free Groups and Automorphisms*, chapter 2, pages 43–110. Birkhäuser, 2017.
[10] A. Duncan and R. H. Gilman. Word hyperbolic semigroups. *Math. Proc. Camb. Philos. Soc.*, 136(3):513–524, 2004.
[11] M. J. Dunwoody. The accessibility of finitely presented groups. *Invent. Math.*, 81:449–457, 1985.
[12] M. Elder, M. Kambites, and G. Ostheimer. On groups and counter automata. *Int. J. Algebra Comput.*, 18(8):1345–1364, 2008.
[13] R. Gilman and M. Shapiro. On groups whose word problem is solved by a nested stack automaton. Available at `https://arxiv.org/abs/math/9812028`, 1998.
[14] P. A. Grillet. A short proof of Rédei's theorem. *Semigr. Forum*, 46:126–127, 1993.
[15] T. Herbst. On a subclass of context-free groups. *RAIRO Inform. Théor. Appl.*, 25(3):255–272, 1991.

[16] M. Hoffmann, D. F. Holt, M. D. Owens, and R. M. Thomas. Semigroups with a context-free word problem. In *Developments in Language Theory, DLT 2012*, pages 97–108, 2012.
[17] D. F. Holt, M. D. Owens, and R. M. Thomas. Groups and semigroups with a one-counter word problem. *J. Aust. Math. Soc.*, 85(2):197–209, 2008.
[18] D. F. Holt, S. Rees, and C. E. Röver. *Groups, Languages and Automata*. Cambridge University Press, 2017.
[19] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
[20] J. M. Howie. *Automata and Languages*. Clarendon Press, 1991.
[21] M. Kambites. Word problems recognisable by deterministic blind monoid automata. *Theor. Comput. Sci.*, 362(1–3):232–237, 2006.
[22] D. C. Kozen. *Automata and Computability*. Springer, 1997.
[23] R. P. Kropholler and D. Spriano. Closure properties in the class of multiple context-free groups. *Groups Complex. Cryptol.*, 11(1):1–15, 2019.
[24] D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: The group case. *Ann. Pure Appl. Log.*, 131(1–3):263–286, 2005.
[25] S. R. Lakin and R. M. Thomas. Context-sensitive decision problems in groups. In *Developments in Language Theory, DLT 2004*, pages 296–307. Springer, 2004.
[26] S. R. Lakin and R. M. Thomas. Space complexity and word problems of groups. *Groups Complex. Cryptol.*, 1(2):261–273, 2009.
[27] A. Levine. Subsets of groups with context-free preimages. Available at `https://arxiv.org/abs/2312.04191`, 2023.
[28] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. Syst. Sci.*, 26(3):295–310, 1983.
[29] C.-F. Nyberg-Brodda. On the word problem for special monoids. *Semigr. Forum*, 105:295–327, 2022.
[30] C.-F. Nyberg-Brodda. Non-finitely generated maximal subgroups of context-free monoids. *J. Algebra*, 616:227–238, 2023.
[31] C.-F. Nyberg-Brodda. On the word problem for free products of semigroups and monoids. *J. Algebra*, 622:721–741, 2023.
[32] R. J. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.
[33] M. J. Pfeiffer. *Adventures in Applying Iteration Lemmas*. PhD thesis, University of St Andrews, 2013.
[34] L. Rédei. *The Theory of Finitely Generated Commutative Semigroups*. Pergamon Press, 1965.
[35] J. C. Rosales and P. A. García-Sánchez. *Finitely Generated Commutative Monoids*. Nova Science Publishers, 1999.
[36] S. Salvati. MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is captured by the IO and OI hierarchies. *J. Comput. Syst. Sci.*, 81(7):1252–1277, 2015.