Characterizing word problems of groups

Sam A. M. Jones[1] and Richard M. Thomas[2]

[1] School of Mathematics & Computer Science, University of Wolverhampton
Wulfruna Street Wolverhampton WV1 1LY, U.K.
[2] Department of Computer Science, University of Leicester, Leicester LE1 7RH, U.K.

Abstract. The word problem of a finitely generated group is a fundamental notion in group theory; it can be defined as the set of all the words in the generators of the group that represent the identity element of the group. This definition allows us to consider a word problem as a formal language and a rich topic of research concerns the connection between the complexity of this language and the algebraic structure of the corresponding group.

Another interesting problem is that of characterizing which languages are word problems of groups. There is a known necessary and sufficient criterion for a language to be a word problem of a group; however a natural question is what other characterizations there are. In this paper we investigate this question, using sentences expressed in first-order logic where the relations we consider are membership of the language in question and concatenation of words. We choose some natural conditions that apply to word problems and then characterize which sets of these conditions are sufficient to guarantee that the language in question really is the word problem of a group. We finish by investigating the decidability of these conditions for the families of regular and one-counter languages.

1 Introduction

The word problem of a finitely generated group G is a fundamental notion in group theory; it can be defined as the set of all the words in the generators of the group that represent the identity element of G. This definition allows us to consider a word problem as a formal language and a rich topic of research concerns the connection between the complexity of this language and the algebraic structure of the corresponding group. For example, the groups with a regular word problem were classified in [1] and those with a context-free word problem in [11] (modulo a subsequent result in [3]).

We will focus on the one-counter languages in this paper (see Section 2) which are particularly interesting in the context of word problems of groups for the following reason. Herbst showed in [4] that, if $\mathcal{F}$ is a subset of the context-free languages which is a cone (in the sense of [2], i.e. $\mathcal{F}$ is a family of languages closed under homomorphism, inverse homomorphism and intersection with regular languages), then the finitely generated groups whose word problem lies in $\mathcal{F}$ are either those with a regular word problem, those with a one-counter word problem or those with a context-free word problem. He also classified [4] the groups with a one-counter word problem (see also [5, 6]). Note that, in all the cases mentioned here, whether or not the word problem of the group lies in the specified family of languages is independent of the choice of finite generating set for the group (see [5] for example).

Another interesting problem is that of characterizing which languages are word problems of groups. A simple necessary and sufficient criterion for a language to be a word problem was established in [13]. This involves the conjunction of two conditions, universal prefix closure and deletion closure (see Definition 2 below); a natural question is what other such characterizations there are. We investigate this problem, using sentences expressed in first-order logic where the only relations are membership of the language in question and concatenation of words. We choose some natural conditions that hold in all word problems (see Definition 2) and then characterize which sets of these conditions are sufficient to guarantee that the language really is the word problem of a group (see Theorem 25).

We then build on the work in [10] in Section 5 and investigate the decidability of these conditions for the families of regular and one-counter languages, noting that all the properties are decidable for the regular languages but undecidable for the one-counter languages (and hence for the context-free languages as well).

2 Preliminaries

In this section we will survey some concepts, notation and results we need from formal language theory and group theory. For the background material on formal language theory the reader is referred to [2, 7, 8] and, for group theory, to [9, 14].

As usual, we let Σ^* denote the set of all *words*, including the *empty word* ϵ, and Σ^+ denote the set of all non-empty words over the alphabet Σ. If $\alpha \in \Sigma^*$ and $x \in \Sigma$ we let $|\alpha|$ denote the length of α and $|\alpha|_x$ the number of occurrences of x in α. If $n \in \mathbb{N}$ then $\Sigma^{\leqslant n}$ is the set of words in Σ^* of length at most n and $\Sigma^{\geqslant n}$ is the set of words of length at least n.

If $\alpha = \beta\gamma$ for some $\beta, \gamma \in \Sigma^*$ then β is said to be a *prefix* of α and γ is a *suffix* of α; if $\alpha = \beta\gamma\delta$ for some $\beta, \gamma, \delta \in \Sigma^*$ then γ is said to be a *factor* of α. If α is the word $a_1a_2 \ldots\ a_{n-1}a_n$ with $n \geqslant 1$ and $a_i \in \Sigma$ for each i, then the *reversal* α^{rev} of α is the word $a_na_{n-1} \ldots a_2a_1$ (and ϵ^{rev} is defined to be ϵ). For any language L we define L^{rev} to be the language $\{\alpha^{rev} : \alpha \in L\}$.

Given a language L over an alphabet Σ we define the *syntactic congruence* $\approx_L$ to be the congruence on Σ^* defined by:

$$\alpha \approx_L \beta \iff (\gamma\alpha\delta \in L \Leftrightarrow \gamma\beta\delta \in L \text{ for all } \gamma, \delta \in \Sigma^*),$$

and then the *syntactic monoid* M_L of L is the quotient $\Sigma^*/ \approx_L$. If φ is the natural map from Σ^* onto M_L then $L = S\varphi^{-1}$ for some subset S of M_L.

We will be discussing *one-counter languages*, which are the languages accepted by a *one-counter automaton*, i.e. a pushdown automaton where we have only a single stack symbol apart from a special symbol marking the bottom of the stack; these automata are nondeterministic and accept by final state.

A *group* is a set G together with a closed binary operation $*$ which is associative and where there is an identity element $1 = 1_G$ for $*$ and each g in G has an inverse g^{-1}. We often suppress the reference to $*$, simply referring to the group as G and writing gh for $g * h$. If G is a group, Σ is a finite set and $\varphi : \Sigma^* \to G$ is a surjective monoid homomorphism then we refer to Σ as a (monoid) *generating set* for G (via φ). For each $a \in \Sigma$ let $\overline{a}$ be an element of Σ^* such that $\overline{a}\varphi = (a\varphi)^{-1}$. We have that $a_1a_2 \dots a_n = b_1b_2 \dots b_m$ in G (where $a_i, b_j \in \Sigma$) if and only if $a_1a_2 \dots a_n\overline{b_m}\,\overline{b_{m-1}} \dots \overline{b_1}$ represents 1_G; so we can focus on the set of the words in Σ^* representing the identity of G and we refer to this language as the *word problem* $W(G, \Sigma)$ of G with respect to the generating set Σ (via φ).

Remark 1. A group is the syntactic monoid of its word problem; see [5] for example. We will need a more general result here. Let Σ be a finite set, G be a group, $\varphi : \Sigma^* \to G$ be a surjective homomorphism, H be a subgroup of G (i.e. a subset of G that forms a group in its own right) such that there is no non-trivial normal subgroup of G contained in H (i.e. such that $\bigcap\{g^{-1}Hg : g \in G\} = \{1\}$) and $L = H\varphi^{-1}$; then G is the syntactic monoid of L (see [12]). □

3 Properties of word problems

As we said in the introduction, we are interested in determining which sets of properties of languages are sufficient to ensure that a language must be the word problem of a group. Obviously such properties must be ones that are satisfied by word problems; the ones we consider are listed in the following definition:

Definition 2. *The following are potential properties of a language L over an alphabet Σ:*

(UPP) *for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\alpha\beta \in L$;*
if L satisfies (UPP) we say that L has the universal prefix property*;*
(USP) *for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\beta\alpha \in L$;*
if L satisfies (USP) we say that L has the universal suffix property*;*
(UFP) *for all $\alpha \in \Sigma^*$ there exist $\beta, \gamma \in \Sigma^*$ such that $\beta\alpha\gamma \in L$;*
if L satisfies (UFP) we say that L has the universal factor property*;*
(DC) *$\alpha\beta\gamma \in L, \beta \in L \Rightarrow \alpha\gamma \in L$;*
if L satisfies (DC) we say that L is deletion closed*;*
(CRD) *$\alpha\beta \in L, \beta \in L \Rightarrow \alpha \in L$;*
if L satisfies (CRD) we say that L is closed under right deletions*;*
(CLD) *$\alpha\beta \in L, \alpha \in L \Rightarrow \beta \in L$;*
if L satisfies (CLD) we say that L is closed under left deletions*;*
(IC) *$\alpha\beta \in L, \gamma \in L \Rightarrow \alpha\gamma\beta \in L$;*
if L satisfies (IC) we say that L is insertion closed*;*
(CCS) *$\alpha\beta \in L \Rightarrow \beta\alpha \in L$;*
if L satisfies the (CCS) we say that L is closed under cyclic shift*;*
(CC) *$\alpha, \beta \in L \Rightarrow \alpha\beta \in L$;*
if L satisfies (CC) we say that L is closed under concatenation*.* □

It is clear that all the properties in Definition 2 are satisfied by word problems of groups; we will use this fact from now on without further comment. We now introduce a concept that we will call (for the purposes of this paper) "duality".

Remark 3. Suppose we have (as in Definition 2) a sentence σ in first-order logic where the only relations in σ are membership of the language in question and concatenation of words. We can obtain a new sentence σ' by reversing the order of the words in any concatenation in σ (but leaving everything else fixed). For example, if we take the sentence representing the property (UPP), then the only concatenation in the sentence is $\alpha\beta \in L$; we reverse this to get $\beta\alpha \in L$ and we now have the sentence representing (USP). In this sense we say that (USP) is the *dual* of (UPP) (and that (UPP) is the dual of (USP)).

In a similar vein we see that (CRD) is the dual of (CLD) and that the other properties listed in Definition 2 are all self-dual. We sum these facts up in the following tables:

Dual properties	
(UPP)	(USP)
(CLD)	(CRD)

Self dual properties
(UFP) (DC) (IC)
(CCS) (CC)

□

The motivation for introducing this concept is that, when characterizing word problems of groups, we will make extensive use of the following result:

Proposition 4. *If L is a language over some alphabet Σ, $S = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ is a subset of the properties listed in Definition 2, σ_i' is the dual of σ_i for each i and $S' = \{\sigma_1', \sigma_2', \ldots, \sigma_n'\}$, then the following statements are equivalent:*
(i) L is the word problem of a group if and only if it satisfies S.
(ii) L is the word problem of a group if and only if it satisfies S'.

Proof. We will first show that L is a word problem of a group if and only if L^{rev} is a word problem of a group.

If $\Sigma = \{a_1, a_2, \ldots, a_n\}$ and $\varphi : \Sigma^* \to G$ is a a surjective homomorphism from Σ^* onto a group G then we define a new homomorphism $\theta : \Sigma^* \to G$ by $a\theta = (a\varphi)^{-1}$ for all $a \in \Sigma$. If L is the word problem of G then, since $(g_1 \ldots\ g_n)^{-1} = g_n^{-1} \ldots\ g_1^{-1}$ in G and $\alpha\varphi = 1$ if and only if $(\alpha\varphi)^{-1} = 1$, we see that L^{rev} is also the word problem of G via the homomorphism θ. Applying the argument again shows that, if L^{rev} is the word problem of a group, then $L = (L^{rev})^{rev}$ is also the word problem of a group.

The result now follows from the observation that L satisfies the properties in S if and only if L^{rev} satisfies the properties in S'. □

4 Characterizing word problems

The following result from [13] is the starting point for our investigations in this paper:

Proposition 5. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UPP) and (DC).* □

Using Remark 3 and Proposition 4 we immediately have:

Corollary 6. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (USP) and (DC).* □

We note the following:

Proposition 7. *If a language L over an alphabet Σ satisfies properties (CCS) and (UFP) then it satisfies property (UPP).*

Proof. If $\alpha \in \Sigma^*$ then we there exist $\beta, \gamma \in \Sigma^*$ such that $\beta\alpha\gamma \in L$ by (UFP). Then $\alpha\gamma\beta \in L$ by (CCS) and so there exists $\delta = \gamma\beta$ with $\alpha\delta \in L$ as required. □

Using Remark 3 we immediately have:

Corollary 8. *If a language L over an alphabet Σ satisfies properties (CCS) and (UFP) then it satisfies property (USP).* □

Given Propositions 5 and 7, we have the following immediate consequence:

Corollary 9. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (DC), (CCS) and (UFP).* □

We next note the following:

Proposition 10. *If a language L over an alphabet Σ satisfies properties (CCS) and (CRD) then it satisfies property (DC).*

Proof. If $\alpha\beta\gamma \in L$ and $\beta \in L$ then we apply (CCS), (CRD), (CCS) in turn to get that $\gamma\alpha\beta \in L$, $\gamma\alpha \in L$, and then $\alpha\gamma \in L$ as required. □

Given Propositions 5 and 10, we have the following:

Corollary 11. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UPP), (CCS) and (CRD).* □

By Remark 3 and Proposition 10, we have the following:

Corollary 12. *If a language L over an alphabet Σ satisfies properties (CCS) and (CLD) then it satisfies property (DC).* □

Given Proposition 5 and Corollary 12 we have:

Corollary 13. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UPP), (CCS) and (CLD).* □

Given Propositions 5, 7 and 10 we have:

Corollary 14. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UFP), (CCS) and (CRD).* □

In a similar vein, Propositions 5, 7 and 12 give:

Corollary 15. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UFP), (CCS) and (CLD).* □

Given Corollaries 6 and 12 we have:

Corollary 16. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (CCS), (USP) and (CLD).* □

In a similar way, Corollaries 6 and 10 give:

Corollary 17. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (CCS), (USP) and (CRD).* □

Another such result is the following:

Proposition 18. *If a language L over an alphabet Σ satisfies properties (UPP), (IC) and and (CRD) then it satisfies property (DC).*

Proof. Assume that L satisfies (UPP), (IC) and (CRD); we want to show that L satisfies (DC).

So assume that $\alpha\beta\gamma \in L$ and $\beta \in L$. By (UPP) there exists $\delta \in \Sigma^*$ such that $\alpha\gamma\delta \in L$. Since $\beta \in L$ we have by (IC) that $\alpha\beta\gamma\delta \in L$. Since $\alpha\gamma\delta \in L$ and $\alpha\beta\gamma \in L$, (IC) also gives us that $\alpha\gamma(\alpha\beta\gamma)\delta \in L$. Since $\alpha\gamma\alpha\beta\gamma\delta \in L$ and $\alpha\beta\gamma\delta \in L$, (CRD) gives that $\alpha\gamma \in L$ as required. □

Given Propositions 5 and 18 we have another characterization of word problems as follows:

Corollary 19. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (UPP), (IC) and (CRD).*

Given Proposition 18 we can apply Remark 3 to deduce:

Proposition 20. *If a language L over an alphabet Σ satisfies properties (USP), (IC) and and (CLD) then it satisfies property (DC).*

Given Propositions 6 and 20 we have another characterization as follows:

Corollary 21. *A language L over an alphabet Σ is the word problem of a group if and only if it satisfies properties (USP), (IC) and (CLD).*

Remark 22. For the convenience of the reader we show the implications between the conditions listed in Definition 2 which we have established in this section by means of the following diagrams:

$$(\mathrm{UPP}) \Longleftarrow \boxed{\begin{matrix}(\mathrm{CCS})\\ \\ (\mathrm{UFP})\end{matrix}} \Longrightarrow (\mathrm{USP}) \qquad \boxed{\begin{matrix}(\mathrm{CCS})\\ \\ (\mathrm{CRD})\end{matrix}} \Longrightarrow (\mathrm{DC}) \Longleftarrow \boxed{\begin{matrix}(\mathrm{CCS})\\ \\ (\mathrm{CLD})\end{matrix}}$$

$$\boxed{\begin{matrix}(\mathrm{UPP})\\ (\mathrm{IC})\\ (\mathrm{CRD})\end{matrix}} \Longrightarrow (\mathrm{DC}) \Longleftarrow \boxed{\begin{matrix}(\mathrm{USP})\\ (\mathrm{IC})\\ (\mathrm{CLD})\end{matrix}}$$

□

We now establish a result that will be crucial in establishing the minimality of certain sets of conditions from Definition 2 when characterizing word problems of groups:

Proposition 23. *There are languages L_1, L_2, L_3, L_4, L_5 and L_6 that satisfy respectively the following specified subsets of the set of the properties listed in Definition 2:*

	(UPP)	*(DC)*	*(CCS)*	*(UFP)*	*(CRD)*	*(IC)*	*(CC)*	*(CLD)*	*(USP)*
L_1	*Yes*	*No*	*Yes*	*Yes*	*No*	*Yes*	*Yes*	*No*	*Yes*
L_2	*No*	*Yes*	*Yes*	*No*	*Yes*	*No*	*No*	*Yes*	*No*
L_3	*Yes*	*No*	*No*	*Yes*	*Yes*	*No*	*Yes*	*Yes*	*Yes*
L_4	*No*	*Yes*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*
L_5	*No*	*Yes*	*Yes*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*No*
L_6	*Yes*	*No*	*No*	*Yes*	*No*	*Yes*	*Yes*	*Yes*	*No*

Proof. Let $\Sigma = \{a, b\}$, $n \geqslant 1$, $L_1 = \Sigma^{\geqslant n}$ and $L_2 = \Sigma^{\leqslant n}$. We see that L_1 satisfies (UPP), (USP), (UFP), (CCS), (CC) and (IC) but not (DC), (CLD) or (CRD). On the other hand, L_2 satisfies (DC), (CRD), (CLD) and (CCS) but not (UPP), (USP), (UFP), (CC) or (IC).

Let Ω be a finite set, G be a group, $\varphi : \Omega^* \to G$ be a surjective homomorphism, H be a non-trivial subgroup of G such that there is no non-trivial normal subgroup of G contained in H and $L_3 = H\varphi^{-1}$. By Remark 1, we see that G is the syntactic monoid of L_3. If L_3 were the word problem of a group then every element of L_3 would represent the identity in its syntactic monoid G, contradicting the fact that H is non-trivial.

Despite L_3 not being a word problem, it does satisfy some of the properties in Definition 2. For example, it satisfies (UPP) (and hence (UFP) as well): if $\alpha \in \Omega^*$ choose $g \in G$ such that $(\alpha\varphi)g \in H$ and then $\beta \in \Omega^*$ with $\beta\varphi = g$; since $(\alpha\beta)\varphi \in H$ we have that $\alpha\beta \in L_3$. A similar argument shows that L_3 also satisfies (USP).

L_3 also satisfies (CRD): if $\alpha\beta \in L$ and $\beta \in L$ then $(\alpha\varphi)(\beta\varphi) \in H$ and $\beta\varphi \in H$, so that $\alpha\varphi = (\alpha\varphi)(\beta\varphi)(\beta\varphi)^{-1} \in H$, and so $\alpha \in L$. Similarly L_3 satisfies (CLD). It is clear that L_3 satisfies (CC): if $\alpha, \beta \in L_3$ then $\alpha\varphi \in H$ and $\beta\varphi \in H$, so that $(\alpha\beta)\varphi = (\alpha\varphi)(\beta\varphi) \in H$ and hence $\alpha\beta \in L_3$. Given that L_3 is not the word problem of a group it cannot satisfy (DC) by Proposition 5, (CCS) by Corollary 14 or (IC) by Corollary 19.

For our next language we consider the bicyclic monoid B with the (monoid) presentation $\langle a, b : ab = 1\rangle$. We let $\Sigma = \{a, b\}$ and let L_4 consist of all those words in Σ^* that represent the identity element of B; more formally, we have the natural homomorphism $\theta : \Sigma^* \to B$ and we let $L_4 = \{1\}\theta^{-1}$.

Each element of B is represented by a word of the form $b^i a^j$ (were $i, j \geqslant 0$) and we have that $(b^i a^j)\theta = (b^k a^\ell)\theta$ if and only if $i = k$ and $j = \ell$. If we consider the complete (i.e. the confluent and terminating) string rewriting system $\mathcal{R}$ over Σ where the only rule is $ab \to \epsilon$, we see that $\mathcal{R}$ reduces any word α in Σ^* to the word β of the form $b^i a^j$ that represents the same element of B as α (i.e. to the word β such that $\beta\theta = \alpha\theta$).

It is clear that L_4 satisfies (DC) (and hence (CRD) and (CLD) as well): if $\alpha\beta\gamma \in L_4$ and $\beta \in L_4$ then $(\alpha\beta\gamma)\theta = \beta\theta = 1$ and then

$$(\alpha\gamma)\theta = (\alpha\theta)(\gamma\theta) = (\alpha\theta)(\beta\theta)(\gamma\theta) = (\alpha\beta\gamma)\theta = 1,$$

so that $\alpha\gamma \in L_4$. Similarly L_4 satisfies (IC): if $\alpha\beta \in L_4$ and $\gamma \in L_4$ then $(\alpha\beta)\theta = \gamma\theta = 1$ and so

$$(\alpha\gamma\beta)\theta = (\alpha\theta)(\gamma\theta)(\beta\theta) = (\alpha\theta)(\beta\theta) = (\alpha\beta)\theta = 1,$$

and so $\alpha\gamma\beta \in L_4$. Given that L_4 satisfies (IC) it clearly satisfies (CC) as well.

We also have that L_4 satisfies (UFP): if $\alpha = a^{i_1}b^{j_1} \ldots\ a^{i_n}b^{j_n}$ let

$$J = i_1 + \ldots + i_n \quad \text{and} \quad I = j_1 + \ldots + j_n;$$

then $a^I \alpha b^J$ reduces in $\mathcal{R}$ to ϵ and so $a^I \alpha b^J \in L_4$. However, L_4 does not satisfy (UPP): if we let $\alpha = b$ then there is no word β such that $\alpha\beta \in L$ as any word in L can be reduced to ϵ through repeated uses of the rewriting rule $ab \to \epsilon$ and no word starting in b can be so reduced. A similar argument shows that no word ending in a can be so reduced and so L_4 does not satisfy (USP). The fact that no word starting in b can belong to L_4 also shows that L_4 does not satisfy (CCS) (since $ab \in L_4$ but $ba \notin L_4$).

We next consider $L_5 = \emptyset$. It is clear that L_5 satisfies (DC), (CCS), (CRD), (CLD), (IC) and (CC) but not (UPP), (USP) or (UFP).

Lastly we let $\Sigma = \{a, b\}$ and let $L_6 = \{\epsilon\} \cup \Sigma^*\{a\}$. It is clear that satisfies (UPP) and (UFP) but not (USP). L_6 also satisfies (IC) and (CC) but not (CCS). Lastly L_6 satisfies (CLD) but not (CRD) or (DC). □

As we said above, the languages specified in Proposition 23 will be useful in establishing the minimality of certain sets of conditions from Definition 2. We can now show that the characterizations we have obtained so far are all minimal, in that no proper subset of any of the specified eleven sets of properties is sufficient to ensure that the language in question is a word problem:

Proposition 24. *For any non-empty proper subset S of any of the sets*

$$\begin{array}{cc}
\{\textit{(UPP), (DC)}\}, & \{\textit{(DC), (CCS), (UFP)}\}, \\
\{\textit{(UPP), (CCS), (CRD)}\}, & \{\textit{(CCS), (CRD), (UFP)}\} \\
\{\textit{(UPP), (IC), (CRD)}\}, & \{\textit{(USP), (DC)}\}, \\
\{\textit{(IC), (CLD), (USP)}\}, & \{\textit{(UPP), (CCS), (CLD)}\}, \\
\{\textit{(USP), (CCS), (CLD)}\}, & \{\textit{(USP), (CCS), (CRD)}\}, \\
\textit{or} & \{\textit{(UFP), (CCS), (CLD)}\}
\end{array}$$

there is a language satisfying all the conditions in S which is not a word problem of a group.

Proof. Throughout this proof we will refer to the six languages L_1, L_2, L_3, L_4, L_5 and L_6 introduced in Proposition 23.

To eliminate proper subsets of . . .	. . . we consider . . .
$\{(\mathrm{UPP}), (\mathrm{DC})\}$	L_1 and L_2.
$\{(\mathrm{DC}), (\mathrm{CCS}), (\mathrm{UFP})\}$	L_1, L_2 and L_4.
$\{(\mathrm{UPP}), (\mathrm{CCS}), (\mathrm{CRD})\}$	L_1, L_2 and L_3.
$\{(\mathrm{CCS}), (\mathrm{CRD}), (\mathrm{UFP})\}$	L_1, L_2 and L_3.
$\{(\mathrm{UPP}), (\mathrm{IC}), (\mathrm{CRD})\}$	L_1, L_3 and L_4.
$\{(\mathrm{USP}), (\mathrm{DC})\}$	L_1 and L_2.
$\{(\mathrm{IC}), (\mathrm{CLD}), (\mathrm{USP})\}$	L_1, L_3 and L_4.
$\{(\mathrm{UPP}), (\mathrm{CCS}), (\mathrm{CLD})\}$	L_1, L_2 and L_3.
$\{(\mathrm{USP}), (\mathrm{CCS}), (\mathrm{CLD})\}$	L_1, L_2 and L_3.
$\{(\mathrm{USP}), (\mathrm{CCS}), (\mathrm{CRD})\}$	L_1, L_2 and L_3.
$\{(\mathrm{UFP}), (\mathrm{CCS}), (\mathrm{CLD})\}$	L_1, L_2 and L_4.

For each maximal proper subset S of one the eleven sets we have given a language satisfying all the properties in S which is not the word problem of a group. □

Theorem 25. *The sets of properties listed in Proposition 24 are precisely those subsets S of the set of properties listed in Definition 2 such that satisfying the conditions in S is sufficient for a language L to be the word problem of a group but such that no proper subset of S has this property.*

Proof. To start with, notice that the empty set L_5 is not a characterisation and satisfies all of the properties except (UPP), (USP) and (UFP); so any characterisation must contain at least one of these three properties. Next we note that, if a language satisfies (CCS) and one of (UPP), (USP) and (UFP), then it satisfies all of them; so, in the first instance, we will consider languages which do not satisfy (CCS).

Note, also, that each of (USP) and (UPP) imply (UFP) so, when considering languages which satisfy two of (USP), (UPP) and (UFP), there is in fact only one pair to consider (taking minimality into account), namely (UPP) and (USP). The result of these considerations is that we have five cases to consider (with respect to minimal characterizations):

- Case 1. We specify (UPP) but not (CCS).
- Case 2. We specify (USP) but not (CCS).
- Case 3. We specify (UFP) but not (CCS).
- Case 4. We specify (UPP) and (USP) but not (CCS).
- Case 5. We specify (CCS) and one of (UPP), (USP) and (UFP).

Let us consider Case 1 where we specify (UPP) but not (CCS), (USP) or (UFP). Since (UPP) and (DC) is already a characterization by Proposition 5 there is no minimal characterization properly containing both of these properties; so we will exclude (DC). Since (IC) implies (CC) we do not include both of these; so we are looking at subsets of (UPP), (CLD), (CRD) and (CC) or of (UPP), (CLD), (CRD) and (IC). With regards to (UPP), (CLD), (CRD) and (CC), the language L_3 satisfies all these conditions, and so no subset of this is sufficient for a characterization.

Let us now consider (UPP), (CLD), (CRD) and (IC). Considering L_3 again we see that (IC) must be included. If we only have (UPP) and (IC) then this is not sufficient as is demonstrated by L_1. If we add (CLD) to (UPP) and (IC) we see that this is not a characterization as witnessed by L_6. If we add (CRD) to (UPP) and (IC) we have a characterization by Corollary 19, and this is minimal by Proposition 24.

Case 2 is the dual of Case 1 (in the sense of Remark 3). Using Proposition 4 we see that the only minimal sets of conditions here are $\{(\mathrm{USP}), (\mathrm{DC})\}$ and $\{(\mathrm{USP}), (\mathrm{IC}), (\mathrm{CLD})\}$.

Case 3 cannot give rise to any characterizations as witnessed by L_4 which satisfies all the properties in Definition 2 except (UPP), (USP) and (CCS).

Let us now consider Case 4 where we specify (UPP) and (USP) but not (CCS) or (UFP). Again, using Proposition 5, we can exclude (DC) if we are considering minimal characterizations. Again, since (IC) implies (CC), we do not include both of these properties; so we are looking at subsets of (UPP), (USP), (CLD), (CRD) and (CC) or of (UPP), (USP), (CLD), (CRD) and (IC). With regards to (UPP), (USP), (CLD), (CRD) and (CC), the language L_3 satisfies all these conditions, and so no subset of this particular set is sufficient for a characterization.

Now consider (UPP), (USP), (CLD), (CRD) and (IC). Given L_3 we see that (IC) must be included. If we only have (UPP), (USP) and (IC) this is not sufficient as demonstrated by L_1. If we add (CRD) to (UPP), (USP) and (IC) then we have a proper superset of $\{(\mathrm{UPP}), (\mathrm{IC}), (\mathrm{CRD})\}$ which is a characterization as above, and, if we add (CLD) to (UPP), (USP) and (IC) then we have a proper superset of $\{(\mathrm{USP}), (\mathrm{IC}), (\mathrm{CLD})\}$ which is also a characterization; so no new minimal characterizations arise here.

Lastly consider Case 5. We first consider the case where we have (CCS) and (UPP). Again, by minimality, we can assume that (DC) is excluded.

Given Proposition 24, if we include (CRD), then we have a minimal characterization by Corollary 11 and, if we include (CLD), then we have a minimal characterization by Corollary 15. We must include one of these, however, as L_1 satisfies (UPP), (CCS), (IC) and (CC).

We next turn to the case where we specify (CCS) and (USP). This is the dual of the case where we specify (CCS) and (UPP) and so we get the minimal characterizations $\{(\mathrm{CCS}), (\mathrm{USP}), (\mathrm{CRD})\}$ and $\{(\mathrm{CCS}), (\mathrm{USP}), (\mathrm{CLD})\}$ here.

Lastly we look at the case where we specify (CCS) and (UFP). Given that (UPP), (USP) and (UFP) are all equivalent in the presence of (CCS), we get (using Proposition 24) the minimal characterizations $\{(\mathrm{CCS}), (\mathrm{UFP}), (\mathrm{CRD})\}$ and $\{(\mathrm{CCS}), (\mathrm{UFP}), (\mathrm{CLD})\}$. The only other possibility would be to include (DC) as, unlike (UPP) and (USP), (DC) is not sufficient to guarantee a word problem when taken in conjunction with (UFP) as witnessed by L_4. The set $\{(\mathrm{CCS}), (\mathrm{UFP}), (\mathrm{DC})\}$ is a characterization by Corollary 9 and is minimal by Proposition 24; so this is our last possibility (as we clearly cannot take any set properly containing it and preserve minimality). □

5 Decidability results

We now investigate the decidability of the properties listed in Definition 2. It is reasonably clear that these are all decidable for regular languages, i.e. given a finite automaton M we can decide whether or not $L(M)$ satisfies the property in question.

One possible approach for regular languages involves considering the syntactic monoid of $L(M)$. If $L = L(M) \subseteq \Sigma^*$ then we know that M_L is finite, that $L = S\varphi^{-1}$ for some $S \subseteq M$ (where φ is the natural map from Σ^* onto M_L) and that we can explicitly construct M_L and S from M. Given this (for example), (UPP) is equivalent to the sentence $\forall x \in M_L \, \exists y \in M_L : xy \in S$, which is decidable as M_L is finite. The decidability of the other properties listed in Definition 2 for regular languages can all be established in the same way.

When we consider the corresponding questions for one-counter languages then, as in [10], we will need the idea of a *counter machine.* There are several ways of describing these machines and we give one possibility here, following the approach taken in [10]. For the convenience of the reader we will reproduce the basic definitions and notation from [10] here.

A counter machine M (as distinct from a one-counter automaton) is a two-tape machine. The first tape is the input tape; it is read only and the head can only move to the right. The second tape is a stack: whenever we move left, M erases the symbol it moved away from. There is only one stack symbol, a say. Intuitively M can only store a natural number (so that we can think of M as having an input tape and a counter). As we will see, the stack is never empty.

More formally, a *counter machine* is a sextuple $M = (Q, \Sigma, a, \delta, q_0, q_f)$ where Q is a finite set of states containing two distinguished states, q_0, the start state, and q_f, the final state. The input alphabet Σ is a finite set of symbols such that $a \notin \Sigma$. A *configuration* of M is a word of the form qa^n where $q \in Q$ and $n > 0$ (where the current state is q and the current stack contents are a^n).

We take C to be $\{1, 2, 3, 5, 7, \frac{1}{2}, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}\}$; there is no particular significance in our choice of 2, 3, 5 and 7, in that any four pair-wise coprime natural numbers would suffice. The transition relation δ is a function from $(Q - \{q_f\}) \times \Sigma \times C$ to $(Q-\{q_0\}) \times (Q-\{q_0\})$; the fact that δ is a function means that M is deterministic. M starts with just a on its stack (i.e. with the counter set to 1) and must set its counter to 1 again before entering q_f.

A move (p, b, x, q, r) in δ is interpreted as follows. If M is in state p reading an input b and if the result of multiplying the current value n of the counter (i.e. we have a^n on the stack) by the value x is an integer, then we set the counter to xn and move to state q; if xn is not an integer then the counter remains set at n and M moves to state r. We write $pa^n \vdash qa^{xn}$ or $pa^n \vdash ra^n$ as appropriate.

Given a Turing Machine, one can effectively construct a counter machine accepting the same language (see [7] for example). We now turn to the computations of a counter machine:

Definition 26. *Let M be a counter machine. A* valid computation *of M is a word $C_0 C_1 \ldots C_n \in (Q \cup \{a\})^*$ where the C_i are configurations of M and*

$$C_0 = q_0 a \vdash C_1 \vdash \ldots \vdash C_n = q_f a;$$

other elements of $(Q \cup \{a\})^*$ are said to be invalid computations. □

In any valid computation, any configuration qa^n will have $n = 2^b 3^c 5^d 7^e$ for some $b, c, d, e \geqslant 0$. Multiplying by 2, 3, 5 or 7 increases b, c, d or e respectively by 1 and multiplying by $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{5}$ or $\frac{1}{7}$ (if possible) decreases b, c, d or e by 1; so we effectively have four counters each of which can be increased or decreased. The fact that we can only multiply by x if nx is an integer effectively says that we can test each counter individually for zero (e.g. if $n = 2^b 3^c 5^d 7^e$ and we want to multiply by $\frac{1}{2}$ then we must have that $b > 0$).

We will need the following result from [10]:

Proposition 27. *If $M = (Q, \Sigma, a, \delta, q_0, q_f)$ is a counter machine then the following language is a one-counter language:*

$K = \{qa^n pa^j$: the following conditions hold:
if (q, b, k, p, r) is a quintuple of δ and kn is an integer then $kn \neq j$;
if (q, b, k, p, r) is a quintuple of δ and kn is not an integer then $j \neq n\}$

In [10] it was shown that the properties (UPP) and (DC) were undecidable for one-counter languages. Our aim here is to extend this result to the other properties listed in Definition 2. We will need the following technical result:

Proposition 28. *The following problem is undecidable:*

Input: a one-counter automaton M with input alphabet Σ of size at least two such that either $L(M) = \Sigma^*$ or $L(M) = \Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$ for some word α such that α has length at least two and contains at least two different symbols.

Output: "yes" if $L(M) = \Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$;
"no" if $L(M) = \Sigma^*$.

Proof. Our aim is to describe a language L over an alphabet Σ which is closed under taking factors and which does not include a valid computation of a counter machine M (when reading a specified input β) as a factor. This way L will be equal to either Σ^* or $\Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$, where α is the computation path of M accepting β, depending on whether or not M accepts β.

Since we want L to be closed under taking factors we need to ensure that no factor of a word in L is a valid computation of M. We do this by checking that, whenever an initial configuration occurs in α, a valid computation does not follow. Formally, we will consider the following three languages:

(i) $L_1 = \Sigma^* - \Sigma^*\{q_0 a\}\Sigma^*$. This is the set of all words in Σ^* which do not contain the unique initial configuration of M.
(ii) $L_2 = \Sigma^* - \Sigma^*\{q_f a\}\Sigma^*$. This is the set of all words which do not contain the unique halting configuration of M.

(iii) L_3. The set of all words which are invalid as computations of M after every instance of the unique initial configuration of M (i.e. words which do not contain a factor consisting of the unique initial configuration of M followed by a valid computation path ending in the unique halting configuration of M).

L_1 and L_2 are regular and so one-counter; we now show that L_3 is one-counter.

The machine accepting L_3 operates as follows: it scans its input doing nothing until it reads the unique initial configuration of M. At this point the machine changes state and attempts to detect an invalid computation step of M (as in Proposition 27). If the machine does not find a factor which is an invalid computation step of M before reading the unique halting configuration of M then the machine scans the rest of its input, doing nothing, and rejects. If it does find a factor which is an invalid computation step of M then the machine continues to scan its input until it finds another instance of the unique initial configuration and then repeats the process, accepting if and only if, after every instance of the initial configuration, we do not reach the halting configuration without finding an invalid computation step first. If, at any point, the machine finds another instance of the initial configuration before an instance of the halting configuration then the machine resets its state and attempts again to find an invalid computation step of M starting at the most recent initial configuration read.

So $L = L_1 \cup L_2 \cup L_3$ is a language one-counter as the family of one-counter languages is closed under union. Now $L = \Sigma^*$ if and only if M rejects β and $L = \Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$ (for suitable α) if and only if M accepts β. So, if we could distinguish between Σ^* and $\Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$ for one-counter languages, then we could solve the halting problem, a contradiction. □

The condition in Proposition 28 that α can be assumed to have length greater than two and to consist of at least two symbols is included only to facilitate the undecidability results that follow. In a similar manner we can establish:

Proposition 29. *The following problem is undecidable:*

Input: *a one-counter automaton M with input alphabet Σ of size at least two such that either $L(M) = \Sigma^*$ or $L(M) = \Sigma^* - \{\alpha\}$ for some α such that α has length at least two and contains at least two different symbols.*

Output: *"yes" if $L(M) = \Sigma^* - \{\alpha\}$;*
"no" if $L(M) = \Sigma^*$.

Having established Propositions 28 and 29 we can now prove our result:

Theorem 30. *All the properties listed in Definition 2 are undecidable for one-counter languages.*

Proof. Σ^* satisfies all the properties in Definition 2 but $K = \Sigma^* - \Sigma^*\{\alpha\}\Sigma^*$ (where α has length at least two and contains two different symbols) does not satisfy any of the conditions (UPP), (USP), (UFP), (IC), (CCS), (CC). These are reasonably clear. The word α is not a prefix, suffix or factor of any word in K, and so K does not satisfy (UPP), (USP) or (UFP). If $\alpha = \beta\gamma$ with $\beta \neq \epsilon \neq \gamma$ then $\beta, \gamma \in K$ but $\beta\gamma \notin K$; so K does not satisfy (IC) or (CC).

Given that α can be assumed to have two distinct symbols, we can write α in the form $a\delta b\zeta$ for some $a, b \in \Sigma$ with $a \neq b$ and $\delta, \zeta \in \Sigma^*$; if K satisfied (CCS) then, as $b\zeta a\delta \in K$, we would have that $\alpha = a\delta b\zeta \in K$, a contradiction. So all these conditions must be undecidable by Proposition 28.

The remaining properties are (DC), (RDC) and (LDC). If we could decide these then we would be able to distinguish between Σ^* (which satisfies all three properties) and $\Sigma^* - \{\alpha\}$ (which doesn't satisfy any of them; for example, for any character x in Σ, $\alpha x \in \Sigma^* - \{\alpha\}$ and $x \in \Sigma^* - \{\alpha\}$ but deleting x from αx yields α which is not a member of $\Sigma^* - \{\alpha\}$), contradicting Proposition 29. □

Acknowledgments

The supportive comments from the referees and their suggestions about the presentation of the material were very welcome; we are grateful to them for their careful reading of the paper.

This paper was completed whilst the second author was on study leave from the University of Leicester and he would like to acknowledge the help and support of the university in this respect. The second author would also like to thank Hilary Craig for all her help and encouragement.

References

1. Anisimov, V.A.: The group languages. Kibernetika 4, 18–24 (1971)
2. Berstel, J.: Transductions and context-free languages. Teubner (1979)
3. Dunwoody, M.J.: The accessibility of finitely presented groups. Invent. Math. 81, 449–457 (1985)
4. Herbst, T.: On a subclass of context-free groups. RAIRO Inform. Théor. Appl. 25, 255–272 (1991)
5. Herbst, T., Thomas, R.M.: Group presentations, formal languages and characterizations of one-counter groups. Theoret. Comput. Sci. 112, 187–213 (1993)
6. Holt, D.F., Owens, M.D., Thomas, R.M.: Groups and semigroups with a one-counter word problem. J. Aust. Math. Soc. 85, 197–209 (2008)
7. Hopcroft, J.E., Ullman, J.D.: Introduction to automata theory, languages and computation. Addison-Wesley (1979)
8. Ito, M.: Algebraic theory of automata & languages. World Scientific Press (2004)
9. Johnson, D.L.: Presentations of groups. Cambridge University Press (1990)
10. Jones, S.A.M., Thomas, R.M.: Formal languages, word problems of groups and decidability. In: Abdulla, P.A., Potapov, I. (eds.) Reachability Problems. pp. 146–158. Lecture Notes in Computer Science 8169, Springer (2013)
11. Muller, D.E., Schupp, P.E.: Groups, the theory of ends, and context-free languages. J. Comput. System Sci. 26, 295–310 (1983)
12. Parkes, D.W., Thomas, R.M.: Syntactic monoids and word problems. Arab. J. Sci. Eng. 25, 81–94 (2000)
13. Parkes, D.W., Thomas, R.M.: Groups with context-free reduced word problem. Comm. Algebra 30, 3143–3156 (2002)
14. Rotman, J.L.: An introduction to the theory of groups. Springer-Verlag (1995)