ON GROUPS WITH EDT0L WORD PROBLEM

ALEX BISHOP, MURRAY ELDER, ALEX EVETTS, PAUL GALLOT, AND ALEX LEVINE

Abstract. We prove that the word problem for the infinite cyclic group is not EDT0L, and obtain as a corollary that a finitely generated group with EDT0L word problem must be torsion. In addition, we show that the property of having an EDT0L word problem is invariant under change of generating set, and passing to finitely generated subgroups. This represents significant progress towards the conjecture that all groups with EDT0L word problem are finite (i.e. precisely the groups with regular word problem).

1. Introduction

An interesting problem is to classify finitely generated groups in terms of the language complexity of their *word problem*, the set of all words over generators and their inverses which spell the identity of the group. The formal study of this problem began when Anisimov [1] first observed that the word problem of a group is regular if and only if the group is finite. An influential result of Muller and Schupp [28] later showed that the word problem of a group is context-free if and only if the group is virtually free. Moreover, the following equivalences have been shown: the word problem is one-counter if and only if the group is virtually cyclic [21]; the intersection of finitely many one-counter languages if and only if the group is virtually abelian [22]; blind multicounter if and only if the group is virtually abelian [15,34]; a language accepted by a Petri net if and only if the group is virtually abelian [29]; an NTS language if and only if the group is virtually free [2].

The family of EDT0L (Extended alphabet, Deterministic, Table, 0-interaction, Lindenmayer) languages have recently received a great amount of interest within geometric group theory and related areas [3–10, 23, 26, 27]. In this paper, we also consider the family of *finite-index* EDT0L languages. Figure 1 shows the relative expressive power of these families of languages, where each forwards arrow represents a strict inclusion. In particular, the family of EDT0L languages contains the family of finite-index EDT0L languages as a strict subfamily; and the family of context-free languages is incomparable (in terms of set inclusion) with the families of EDT0L and finite-index EDT0L languages.

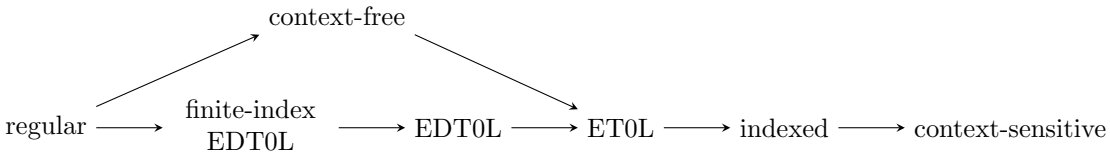

Figure 1. Inclusion diagram of formal languages (see Theorem 15 in [31] and Figure 8 in [17]).

We begin by showing that having an EDT0L word problem is invariant under change of generating set, which is not immediate since EDT0L languages are not closed under inverse monoid homomorphism [13]. From the following proposition, it makes sense to speak of a word problem being EDT0L without the need to specify a generating set. Let $\mathrm{WP}(G, X)$ and $\mathrm{coWP}(G, X)$ denote the word and coword problem, respectively, for a group G with respect to the generating set X.

Proposition A. *Let G be a group with finite monoid generating sets X and Y. If $\mathrm{WP}(G, X)$ (resp. $\mathrm{coWP}(G, X)$) is EDT0L, then $\mathrm{WP}(G, Y)$ (resp. $\mathrm{coWP}(G, Y)$) is EDT0L of finite index. These results also hold if Y instead generates a submonoid of G. In particular, this implies that, if a word problem is EDT0L, then it is EDT0L of finite index; and that having a word problem that is EDT0L of finite index is independent of choice of generating set.*

Date: 20 January 2026.
2020 *Mathematics Subject Classification.* 20F10, 68Q42, 20F65.
Key words and phrases. word problem, EDT0L language, finite-index EDT0L grammar.

The following conjecture was posed by Ciobanu, Ferov and the second author.

Conjecture B (Conjecture 8.2 in [8]). *If G has an EDT0L word problem, then G is finite.*

Towards this conjecture, we have the following results.

Theorem C. *The word problem for the infinite cyclic group $\mathbb{Z}$ is not EDT0L.*

From the above theorem and Proposition A, we then obtain the following theorem.

Theorem D. *If G has an EDT0L word problem, then G is a torsion group.*

Our proof of Theorem C is an expanded version of the proof given by the fourth author [18, Chapter 8] in their PhD thesis. It remains to be shown whether there can exist an infinite torsion group with EDT0L word problem.

A language related to the word problem of $\mathbb{Z}$ with respect to a standard generating set is the *Dyck language* D_1, where

$$D_n = \left\{ w \in \{a_1, b_1, \ldots, a_n, b_n\}^* \;\middle|\; |w|_{a_i} = |w|_{b_i} \text{ and, for each prefix } u \text{ of } w \text{ and each } i,\ |u|_{a_i} \geqslant |u|_{b_i} \right\}$$

where $|\cdot|_x$ counts the instances of the letter x in w. By definition, we see that D_1 is a proper subset of the word problem of $\mathbb{Z}$ (taking a_1 as the generator 1 and b_1 as -1). It was shown in [14, Theorem 9] that the language D_n is not EDT0L for $n \geqslant 8$. This was later strengthened in [33, Theorem 6.3.2] where it was shown that D_1 is not EDT0L. However, this result and the techniques used therein were not helpful in proving Theorem C.

This paper is organised as follows. In Section 2 we fix some terminology about geodesic words. In Section 3 we define the families of EDT0L and finite-index EDT0L languages, and provide some tools for proving that an EDT0L language has finite index. In Section 4 we prove that the class of EDT0L languages is closed under string transduction. In Section 5 we prove that if the word problem of a group is EDT0L, then it is EDT0L of finite index, and obtain Proposition A. In Section 6 we define a *non-permuting, non-erasing, non-branching multiple context-free grammar* to be a multiple context-free grammar that does not permit permutation of variables, does not allow erasure of variables, and restricts productions to maintain a single nonterminal. For ease of reference, we refer to such a grammar as a *restricted multiple context-free grammar* (abbreviated *R-MCFG*) throughout the paper. We prove that every finite-index EDT0L language is the language of an R-MCFG. Finally, in Section 7, we prove Theorems C and D.

1.1. **High-level overview of the proof of Theorem C.** This subsection furnishes a very brief outline of the main ideas of the proof, to motivate the technology required in the sections leading up to and including Section 7. We start the proof of Theorem C by assuming for contradiction that the word problem of $\mathbb{Z}$ with respect to the generators $1, -1$ is generated by a given R-MCFG, and moreover we assume the grammar is in a particular *normal form* (see Definition 6.7). From this grammar, we obtain a bound C related to the difference in the number of 1 versus -1 letters in a tuple words that appear in a step of any derivation (see Lemma 7.3 for details). Using this bound, we construct a word $\mathcal{W}$, which we call the *counterexample word*, having a very specific and controlled form (which uses two parameters m, k coming from the grammar and bound C). An example of $\mathcal{W}$ for $k = 2$, $m = 4$ is shown in Figure 2, where the generator 1 is represented by an up-step $(1, 1)$, and -1 by a down-step $(1, -1)$. The basic

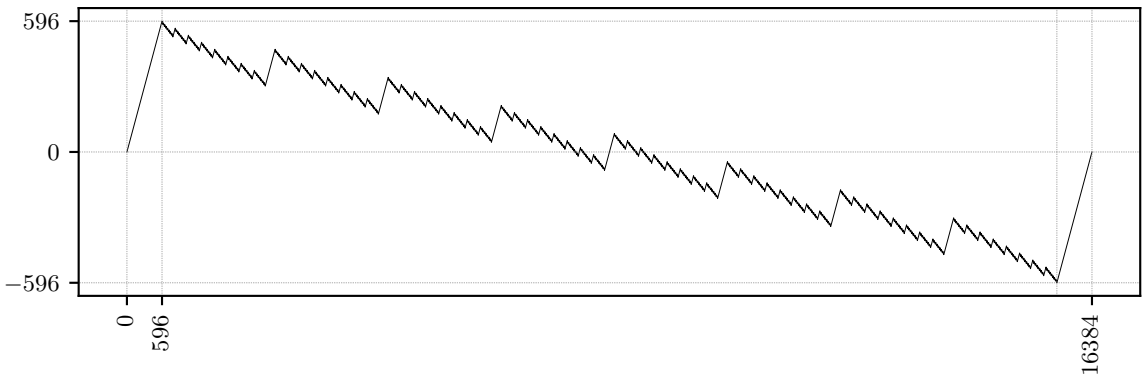

FIGURE 2. The word $\mathcal{W}$ defined in Section 7 for values $k = 2$ and $m = 4$.

idea of the construction of this word is to ensure that it contains "long" up-step factors, having lengths exceeding some bound (depending on C and the grammar) and occurring sufficiently frequently, and all down-step factors of constant size, such that the total difference between the number of up and down steps is zero (so that $\mathcal{W}$ is in the word problem of $\mathbb{Z}$). We show that any derivation of the R-MCFG in normal form generating $\mathcal{W}$ will have the form

$$S(\mathcal{W}) \leftarrow H_\ell(\mathcal{W}, \varepsilon, \ldots, \varepsilon) \leftarrow \cdots \leftarrow H_1(\varepsilon, \ldots, \varepsilon) \leftarrow$$

(for some nonterminals $H_1, \ldots, H_\ell$ and starting nonterminal S; see Definition 6.1) where the words in the tuple $(\varepsilon, \ldots, \varepsilon)$ do not have any long up-step factors, whereas $\mathcal{W}$ does (in particular, its prefix and suffix are such factors). Definition 7.20 defines a property of tuples that is enjoyed by $(\mathcal{W}, \varepsilon, \ldots, \varepsilon)$ and not by $(\varepsilon, \ldots, \varepsilon)$. Specifically, a tuple is said to *have a decomposition* (as in Definition 7.20) if a particular inequality which compares the lengths and occurrences of long up-step factors at the prefix and suffix of coordinates, as well as several other quantities, is satisfied. We then prove (Proposition 7.31) that for each step of a derivation of the R-MCFG, if the target tuple has a decomposition then so must the source tuple. Since $(\mathcal{W}, \varepsilon, \ldots, \varepsilon)$ has a decomposition and $(\varepsilon, \ldots, \varepsilon)$ does not, we have our contradiction.

1.2. **Notation.** We write $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ for the set of nonnegative integers including zero, and $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. Unless otherwise stated, monoid actions are applied on the right: we make this choice to simplify the constructions within the proofs below. An *alphabet* is a finite set. If X is an alphabet, we let X^* denote the set of words (finite length strings) of letters from X, and $X^+ = X^* \setminus \{\varepsilon\}$. We will sometimes speak of the word problem of a monoid. Suppose that M is a monoid with finite generating set X, and that 1_M is the monoid identity. The word problem of (M, X) is given by

$$\mathrm{WP}(M, X) \coloneqq \{w \in X^* \mid \overline{w} = 1_M\}$$

where $\bar{\cdot}\colon X^* \to M$ is the natural homomorphism from the words over X to the monoid.

2. Geodesics in finitely-generated groups

The results in this paper combine topics in formal language theory and geometric group theory. This section is provided for readers who are not familiar with some of the terms and basic notations of geometric group theory which are used in this paper. In particular, this section explains what it means for a word to be *geodesic* in a finitely generated group. Readers who are familiar with this terminology can safely skip this section.

Suppose that G is a group which is generated as a monoid by a finite subset X. That is, we have a surjective homomorphism, which we denote as $\bar{\cdot}\colon X^* \to G$, from the free monoid over X to the group G. Thus, for each element $g \in G$, there exists at least one word $w \in X^*$ such that $\overline{w} = g$. If the word $w \in X^*$ is a minimal-length word for which $\overline{w} = g$, then we say that w is a *geodesic* for g. Notice that each element $g \in G$ must have at least one corresponding geodesic.

For example, suppose that G is the group given by vectors in $\mathbb{Z}^2$ with group operator being vector addition. Such a group is generated by $X = \{x, x^{-1}, y, y^{-1}\}$ where $\overline{x} = (1, 0)$ and $\overline{y} = (0, 1)$. The element $(2, 3)$ can be written as any of the following words over X:

$$xxyyy \qquad yxyxy \qquad y^{-1}xxyyyy \qquad x^{-1}xx^{-1}yxyxyx.$$

Notice here that the word $xxyyy$ and $yxyxy$ are both geodesics.

We have the following properties of geodesics which are used in this paper:

(1) any prefix or suffix of a geodesic is also a geodesic;
(2) the only geodesic for the group identity is the empty word; and
(3) if G is an infinite group, then it has arbitrarily long geodesics.

The above properties are straightforward to prove, and left as an exercise to the reader.

3. EDT0L and EDT0L of finite index

In this section we provide background on the family of EDT0L languages and the subfamily of EDT0L languages of finite index. Our aim here is to prove Lemma 3.6 due to Latteux [25], which provides a method to show that a given language is EDT0L of finite index. We begin this section by defining the class of EDT0L grammars as follows. We note that our definition of EDT0L grammar differs from their original definition given in [32, Definition 1 and 2], in particular, we make additional assumptions of our *tables* in (4) of Definition 3.1. However, it can be seen from [11, Lemma 2.2] that our definition produces the same class of languages to that of [32, Definition 1 and 2].

Definition 3.1. An *EDT0L grammar* is a 4-tuple $E = (\Sigma, V, I, H)$ where

(1) Σ is an alphabet of *terminal letters*;
(2) V is an alphabet of *nonterminal letters* which is disjoint from Σ;
(3) $I \in V$ is an *initial symbol*; and
(4) $H \subset \mathrm{End}((V \cup \Sigma)^*)$ is a finite set of monoid endomorphisms such that $\sigma \cdot h = \sigma$ for each $h \in H$ and each $\sigma \in \Sigma$. The endomorphisms $h \in H$ are called *tables*.

The language generated by such a grammar E is given by

$$L(E) \coloneqq \{I \cdot \alpha \mid \alpha \in H^*\} \cap \Sigma^*$$

where $I \cdot \alpha$ denotes the (right) action of $\overline{\alpha} \in \mathrm{End}((V \cup \Sigma)^*)$ on the word I. That is, $L(E)$ is the set of words in Σ^* that can be obtained by applying any sequence of maps from H to I.

For example, the language

$$L = \{a^{n_1}ba^{n_2}ba^{n_3}b \cdots a^{n_k}b \mid n_i, k \in \mathbb{N} \text{ with } 0 \leqslant n_1 \leqslant n_2 \leqslant \cdots \leqslant n_k\}$$

is EDT0L [8, Proposition 7] as it can be generated using the grammar $E = (\Sigma, V, I, H)$ where

(1) $\Sigma = \{a, b\}$;
(2) $V = \{I, A\}$; and
(3) $H = \{h_1, h_2, h_3\}$ with

$$v \cdot h_1 = \begin{cases} IAb & \text{if } v = I, \\ v & \text{otherwise,} \end{cases} \qquad v \cdot h_2 = \begin{cases} Aa & \text{if } v = A, \\ v & \text{otherwise,} \end{cases} \qquad v \cdot h_3 = \varepsilon$$

for each $v \in V$.

In particular, the word $abaaab \in L$ can be generated as $I \cdot (h_1h_2h_2h_1h_2h_3)$.

Suppose that $E = (\Sigma, V, I, H)$ is an EDT0L grammar. Then, we call $w \in (\Sigma \cup V)^*$ a *sentential form* if there exists some sequence of tables $\alpha \in H^*$ such that $w = I \cdot \alpha$. For example, in the grammar given above, the word $AabAaab$ is a sentential form which can be generated as $I \cdot (h_1h_2h_1)$. Notice that the initial word I and every word generated by the grammar are examples of sentential forms.

Given a word $w \in X^*$ over a finite alphabet X, we write $|w|$ for its length. For each letter $x \in X$, we write $|w|_x$ to denote the number of instances of the letter x in w. Moreover, for any subset $V \subset X$, we write $|w|_V$ to denote $\sum_{v \in V} |w|_v$. We now define the EDT0L languages of finite index as follows.

Definition 3.2 (See [31])**.** An EDT0L grammar $E = (\Sigma, V, I, H)$ is of *index* n if $|I \cdot \alpha|_V \leqslant n$ for each monoid endomorphism $\alpha \in H^*$. That is, if there are at most n nonterminals in any sentential form. An EDT0L grammar is said to be of finite index if it is an EDT0L grammar of index n for some $n \in \mathbb{N}$. A language is *EDT0L of finite index* if it can be generated by an EDT0L grammar of finite index.

From the formal grammar description of regular languages, we have the following result.

Lemma 3.3. *Regular languages are EDT0L of index 1.*

In order to prove the results in this section, we define the family of $LULT$ grammars as follows.

Definition 3.4. An EDT0L grammar $E = (\Sigma, V, I, H)$ is $LULT$ (which stands for *EDT0**L**-système **ult**ralinéaire*, cf. [25, p. 361]) if for each word $w \in L(E)$, there exists some $\alpha \in H^*$ with $w = I \cdot \alpha$ such that for each factorisation $\alpha = \alpha_1\alpha_2$ with $\alpha_1, \alpha_2 \in H^*$, and each $v \in V$, either $|I \cdot \alpha_1|_v \leqslant 1$ or $|v \cdot \alpha_2| \leqslant 1$.

In the next result we show that if a language is generated by a LULT grammar, then the language is EDT0L of finite index. Throughout the proof, we write $f \colon A \rightharpoonup B$ to denote a partial function from A to B, and we write $\mathrm{dom}(f) \subseteq A$ for its domain. That is, $f \colon A \rightharpoonup B$ is a function from some subset of A to a subset of B. Moreover, we write $f = \emptyset$ if $\mathrm{dom}(f) = \emptyset$.

Lemma 3.5 (Latteux [24])**.** *If $E = (\Sigma, V, I, H)$ is an EDT0L grammar which is $LULT$, then the language $L(E)$ is EDT0L of finite index.*

Proof. Let $E = (\Sigma, V, I, H)$ be a LULT grammar as in the lemma statement. In this proof, we construct an EDT0L grammar $E' = (\Sigma, V', I', H')$ with index $|V| + 1$ such that $L(E) = L(E')$. In particular, we first construct the grammar E', then prove that it generates precisely the language $L(E)$.

The construction given in this proof is quite involved. In order to assist the reader's comprehension, we provide an informal overview as follows. From the definition of LULT grammars, we see that if $w \in \Sigma^*$ is generated by the grammar E, then there must exist some sequence of tables $\alpha \in H^*$ with $w = I \cdot \alpha$ such that for each factorisation $\alpha = \alpha_1\alpha_2$, and each nonterminal $v \in V$, we have either $|v \cdot \alpha_1|_v \leqslant 1$

or $|v \cdot \alpha_2| \leqslant 1$. Notice that we do not assume that α is uniquely defined. The idea of our construction is to produce a grammar E' which only considers such sequences of tables. In particular, we construct our grammar E' such that each sentential form $u \in (V' \cup \Sigma)^*$ of E' is either some *dead-end* word (see case 2 in part 1.1 of this proof) which cannot lead to an accepted word, or corresponds to some word $u' = I \cdot \alpha_1$ of E (as before) where the nonterminals v with $|u'|_v \geqslant 2$ have been replaced with the words $v \cdot \alpha_2 \in \Sigma^*$. Notice then that the remaining nonterminals in u each appear at most once, thus resulting in our grammar having finite index as desired. In order to correctly emulate such productions of the grammar E, for technical reasons, each of the nonterminals of E' must store information about which nonterminals are present in the sentential form: to do so, we introduce the following nonterminals. Their properties are explained in more detail at the end of part 1.1 of this proof.

1. Nonterminals:

We begin our construction by defining our set of nonterminals V' as

$$V' = \{\mathfrak{d}, I'\} \cup \{X_{a,A,f} \mid a \in A \subseteq V \text{ and } f \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\}) \text{ with } \operatorname{dom}(f) \cap A = \emptyset\} \\ \cup \{Y_{A,f} \mid A \subseteq V \text{ and } f \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\}) \text{ with } \operatorname{dom}(f) \cap A = \emptyset\}.$$

We immediately notice that this set of nonterminals is finite, in particular, we have

$$|V'| \leqslant 2 + \left(|V| \cdot 2^{|V|} \cdot (|\Sigma| + 2)^{|V|}\right) + \left(2^{|V|} \cdot (|\Sigma| + 2)^{|V|}\right).$$

We note here that the symbol I' is the starting symbol of the new grammar, and that $\mathfrak{d}$ is an additional nonterminal which we call the *dead-end symbol*. In our construction, we ensure that each table maps $\mathfrak{d}$ to itself. Thus, once $\mathfrak{d}$ enters a sentential form, there is no way to continue to generate a word in the language associated to the grammar. We describe the remaining symbols of V' in detail in part 1.1.

Informally, the nonterminals $X_{a,A,f}$ correspond to instances of the nonterminal $a \in A$ within sentential forms w of E, where each nonterminal v that appear in w either has the property that (1) it occurs at most once in w and $v \in A$, or (2) it goes on to produce the word $f(v) \in \Sigma^*$. Similarly, the nonterminals $Y_{A,f}$ are included to describe such a sentential form: these nonterminals are required for the case where we are representing a sentential form w of E where each nonterminal in w occurs more than once.

1.1. Properties of sentential forms of E':

In our construction we ensure that, for each sequence of tables of the form $\alpha \in (H')^*$, the resulting sentential form $w = I' \cdot \alpha \in (\Sigma \cup V')^*$ is in one of the following four forms:

(1) *initial:* $w = I'$, that is, w contains only the starting symbol;
(2) *dead-end:* $\mathfrak{d}$ is the only nonterminal in w and $|w|_{\mathfrak{d}} = 1$;
(3) *production:* there exists some $A \subseteq V$ and $f \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ with $\operatorname{dom}(f) \cap A = \emptyset$ such that
$$\{X_{a,A,f} \mid a \in A\} \cup \{Y_{A,f}\}$$
are the only nonterminals which appear in w, each such nonterminal appears exactly once in w, and the nonterminal $Y_{A,f}$ appears as the last letter of w; or
(4) *final:* $w \in \Sigma^*$, that is, the word contains no nonterminals.

Notice that, with such a property, the grammar would be EDT0L of index $|V| + 1$.

Notice that if a word w is a production as in case 3, as described above, then $w = w'Y_{\emptyset,\emptyset}$ would imply that $w' \in \Sigma^*$. This is clear as, from the definition of the normal form, $Y_{\emptyset,\emptyset}$ would be the only nonterminal which may appear in such a normal form, i.e., there are no nonterminals of the form $X_{a,\emptyset,f}$.

Suppose that $\beta_1, \beta_2 \in (H')^*$ with $I' \cdot (\beta_1\beta_2) \in \Sigma^*$, and that $w = I' \cdot \beta_1 \in (\Sigma \cup V)^*$ is a word as in case 3, as above, with $A \subseteq V$ and $f \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$. In our construction, these sequences $\beta_1, \beta_2 \in (H')^*$ correspond to some sequences $\alpha_1, \alpha_2 \in H^*$ in the grammar E for which

- $I \cdot \alpha_1 \in (\Sigma \cup \operatorname{dom}(f) \cup A)^*$;
- for each $a \in A$, we have $|I \cdot \alpha_1|_a \leqslant 1$; and
- for each $b \in \operatorname{dom}(f)$, we have $|b \cdot \alpha_2| \leqslant 1$, in particular, we have $b \cdot \alpha_2 = f(b)$.

Compare the above with the definition of LULT grammars as in Definition 3.4. In the remainder of this proof, we construct the tables H' of E' such that the above properties hold and $L(E) = L(E')$.

2. Table t_{init}:

We begin the productions of our grammar E' with a table $t_{\text{init}} \in H'$ defined as

$$v \cdot t_{\text{init}} = \begin{cases} X_{I,\{I\},\emptyset}\, Y_{\{I\},\emptyset} & \text{if } v = I', \\ v & \text{otherwise.} \end{cases}$$

Notice that applying this table preserves the property of a word belonging to one of the four forms described in part 1.1 of this proof. In the remainder of this proof, we notice that in each production of E', we may assume that t_{init} is the first table we apply, and that it is applied exactly once.

3. Table t_{end}:

We end the production of our grammar with the table $t_{\text{end}} \in H'$ defined as

$$v \cdot t_{\text{end}} = \begin{cases} v & \text{if } v \in \Sigma, \\ \mathfrak{d} & \text{if } v = Y_{A,f} \text{ where either } A \neq \emptyset \text{ or } f \neq \emptyset, \\ \mathfrak{d} & \text{if } v \in \{I', \mathfrak{d}\}, \\ \varepsilon & \text{otherwise.} \end{cases}$$

Notice that applying this table preserves the property of a word being in one of the four forms described in part 1.1 of this proof. Moreover, we notice that given a word $w \in (\Sigma \cup V')^*$ in one of the forms described in part 1.1 of this proof, that $w \cdot t_{\text{end}} \in \Sigma^*$ if and only if either $w \in \Sigma^*$ or $w = w' Y_{\emptyset,\emptyset}$ in which case we would have $w' \in \Sigma^*$ (see part 1.1 of this proof).

4. Tables $t_{h,B,g}$:

Let $h \in H$ be a table from the grammar E, let $B \subseteq V$ be a subset of nonterminals of E, and let $g \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ be a partial function with $\operatorname{dom}(g) \cap B = \emptyset$. We then introduce a new table $t_{h,B,g} \in H'$. Such a table corresponds to an application of the table h of E, and is intended to produce words as in case 3 of part 1.1 which end in the nonterminal $Y_{B,g}$ (with some edge cases for invalid productions).

We begin our construction of this table by specifying that

- $\sigma \cdot t_{h,B,g} = \sigma$ for each $\sigma \in \Sigma$;
- $\mathfrak{d} \cdot t_{h,B,g} = \mathfrak{d}$; and
- $I' \cdot t_{h,B,g} = \mathfrak{d}$.

Thus, we see that if $w \in (\Sigma \cup V')^*$ is a word as in case 1, 2 or 4 as in part 1.1 of this proof, then $w \cdot t_{h,B,g}$ is also a word of such a form. Thus, it only remains to specify the table such that we understand how it acts on word in the form described in case 3 of part 1.1.

Let $A \subseteq V$ and $f \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ with $\operatorname{dom}(f) \cap A = \emptyset$; and let w be a word as in case 3 of part 1.1 which ends in the nonterminal $Y_{A,f}$. In the remainder of this part, we describe the word $w \cdot t_{h,B,g}$, in particular, we specify the action of the table on the nonterminals of the form $X_{a,A,f}$ and $Y_{A,f}$.

4.1. Notation. For each $C \subseteq V$ and each partial function $k \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ with $\operatorname{dom}(k) \cap C = \emptyset$, we define a monoid homomorphism $\operatorname{Label}_{C,k} \colon (\Sigma \cup C \cup \operatorname{dom}(k))^* \to (\Sigma \cup V')^*$ such that

$$\operatorname{Label}_{C,k}(v) \coloneqq \begin{cases} v & \text{if } v \in \Sigma \\ X_{v,C,k} & \text{if } v \in A \\ k(v) & \text{if } v \in \operatorname{dom}(k) \end{cases}$$

for each $v \in \Sigma \cup C \cup \operatorname{dom}(k)$. From our choice of C and k, we see that $\operatorname{Label}_{C,k}$ is well-defined.

Let $k \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ be a partial function as above, then we write $\widetilde{k} \in \operatorname{End}(\Sigma \cup V)^*$ for the table defined such that

$$v \cdot \widetilde{k} \coloneqq \begin{cases} k(v) & \text{if } v \in \operatorname{dom}(k) \\ v & \text{otherwise} \end{cases}$$

for each $v \in (\Sigma \cup V)$.

4.2. Intention. Our intention is for our construction to have the following property. If there exists some $\beta_1, \beta_2 \in (H')^*$ such that $w = I' \cdot \beta_1$ and $I' \cdot (\beta_1 \, t_{h,B,g} \, \beta_2) \in \Sigma^*$, then there exists some $\alpha_1, \alpha_2 \in H^*$ where

(I1) $I' \cdot (\beta_1 \, t_{h,B,g} \cdot \beta_2) = I \cdot (\alpha_1 \, h \, \alpha_2)$;
(I2) $I' \cdot \beta_1 = \operatorname{Label}_{A,f}(I \cdot \alpha_1) \, Y_{A,f}$ is a word as in case 3 of part 1.1; and
(I3) $I' \cdot (\beta_1 \, t_{h,B,g}) = \operatorname{Label}_{B,g}(I \cdot \alpha_1 h) \, Y_{B,g}$ is a word as in case 3 of part 1.1.

Thus, the table $t_{t,B,g}$ of E' behaves similarly to the table h of E except it performs additional replacements, i.e. the replacements given by f and g which are completely specified by the nonterminals used in the sentential form. Thus, our reason for including the nonterminals $Y_{A,f}$ and $Y_{B,g}$ in these words is so that this information is always present in our sentential form.

4.3. Technical details. We label the elements of A as $A = \{x_1, x_2, \ldots, x_k\} \subseteq V$. We then define two sets $B^{(1)}, B^{(\geqslant 2)} \subseteq V$ as

$$B^{(1)} \coloneqq \left\{ v \in V \;\middle|\; |(x_1 x_2 \cdots x_k) \cdot h|_v = 1 \right\} \qquad \text{and} \qquad B^{(\geqslant 2)} \coloneqq \left\{ v \in V \;\middle|\; |(x_1 x_2 \cdots x_k) \cdot h|_v \geqslant 2 \right\}$$

where $h \in H$ is the table as in $t_{h,B,g}$. We are then interested in the case where the sets $B^{(1)}$ and $B^{(\geqslant 2)}$ satisfy the following 4 additional properties:

(T1) $B^{(\geqslant 2)} \subseteq \operatorname{dom}(g)$;
(T2) $B = B^{(1)} \setminus \operatorname{dom}(g)$;
(T3) for each $a \in A$, we have $a \cdot h \in (\Sigma \cup \operatorname{dom}(g) \cup B)^*$; and
(T4) for each $v \in \operatorname{dom}(f)$, we have $v \cdot h \in (\Sigma \cup \operatorname{dom}(g))^*$ with $f(v) = \tilde{g}(v \cdot h)$.

If any of these above properties (T1–4) are violated, then we define

$$X_{a,A,f} \cdot t_{h,B,g} = \varepsilon \qquad \text{and} \qquad Y_{A,f} \cdot t_{h,B,g} = \mathfrak{d}$$

for each $a \in A$, which produces a dead-end word as in case 2 of part 1.1. Otherwise, if properties (T1–4) are satisfied, then we define

$$X_{a,A,f} \cdot t_{h,B,g} = \operatorname{Label}_{B,g}(a \cdot h) \qquad \text{and} \qquad Y_{A,f} \cdot t_{h,B,g} = Y_{B,g} \tag{1}$$

for each $a \in A$. From the definition of $B^{(1)}$, $B^{(\geqslant 2)}$ and properties (T1–4), we see that the applications of $\operatorname{Label}_{B,g}$, as above in (1), are well-defined.

Notice that we may then completely describe the action of the tables $t_{h,B,g}$ by repeating the above construction for any valid values of A and f.

4.4. Property. Suppose that $w \in (\Sigma \cup V')^*$ is a word in the form described in case 3 of part 1.1. Then, from the construction of $B^{(1)}$ and properties T1 and T2, as above, we see that $w \cdot t_{h,B,g}$ is either a dead-end word as in case 2 of part 1.1, or a production as in case 3 of part 1.1. Further, suppose additionally that there exists some word $u \in (\Sigma \cup V)^*$ such that $w = \operatorname{Label}_{A,f}(u)\, Y_{A,f}$. Then either $w \cdot t_{h,B,g}$ is a dead-end word; otherwise it follows from property T4 that $w \cdot t_{h,B,g} = \operatorname{Label}_{B,g}(u \cdot h)\, Y_{B,g}$.

5. Soundness and Completeness:

Suppose that $w \in L(E)$, then from the definition of LULT grammars, we see that there must exist some sequence of tables $\alpha = h_1 h_2 \cdots h_k \in H^*$ such that for each factorisation $\alpha = \alpha_1 \alpha_2$ with $\alpha_1, \alpha_2 \in H^*$, and each nonterminal $v \in V$, either $|I \cdot \alpha_1|_v \leqslant 1$ or $|v \cdot \alpha_2| \leqslant 1$. For each $i \in \{1, 2, \ldots, k-1\}$, let

$$C_i = \left\{ v \in V \;\middle|\; |I \cdot (h_1 h_2 \cdots h_k)|_v \geqslant 1 \right\}.$$

We then see that

$$I \cdot (h_1 h_2 \cdots h_i) \in (\Sigma \cup C_i)^*$$

for each $i \in \{1, 2, \ldots, k-1\}$. From our choice of sequence $\alpha = h_1 h_2 \cdots h_k$, we see that we can partition each set C_i into two disjoint subsets

$$B_i = \left\{ v \in V \;\middle|\; |I \cdot (h_1 h_2 \cdots h_i)|_v = 1 \text{ and } |v \cdot (h_{i+1} h_{i+1} \cdots h_k)| \geqslant 2 \right\} \subseteq C_i \text{ and}$$
$$C_i^{(*)} = \left\{ v \in V \;\middle|\; |I \cdot (h_1 h_2 \cdots h_i)|_v \geqslant 1 \text{ and } |v \cdot (h_{i+1} h_{i+2} \cdots h_k)| \leqslant 1 \right\} \subseteq C_i.$$

In particular, the two sets correspond to the two cases in the definition of LULT grammars.

Notice then that, for each $i \in \{1, 2, \ldots, k-1\}$, we can define a partial function $g_i \colon V \rightharpoonup (\Sigma \cup \{\varepsilon\})$ with $\operatorname{dom}(g_i) = C_i^{(*)}$ such that $g_i(v) = v \cdot (h_{i+1} h_{i+2} \cdots h_k)$ for each $v \in C_i^{(*)}$.

We then see that the word w is generated by the grammar E' as

$$w = I' \cdot \left(t_{\text{init}}\; t_{h_1,B_1,g_1}\; t_{h_2,B_2,g_2}\; \cdots t_{h_{k-1},B_{k-1},g_{k-1}}\; t_{h_k,\emptyset,\emptyset}\; t_{\text{end}} \right).$$

Thus, we see that $L(E) \subseteq L(E')$.

Suppose then that $w \in L(E')$ where $w = I' \cdot (\tau_1 \tau_2 \cdots \tau_k)$ with each $\tau_i \in H'$. Then, from the definitions of the tables in H', we may assume without loss of generality that

- $k \geqslant 3$ with $\tau_1 = t_{\text{init}}$ and $\tau_k = t_{\text{end}}$; and
- for each $i \in \{2, 3, \ldots, k-1\}$, the table t_i is of the form $\tau_i = t_{h_i,B_i,g_i}$ where each $h_i \in H$.

From our construction, we then see that $w = I \cdot (h_2 h_3 \cdots h_{k-1})$. Thus, we have $L(E') \subseteq L(E)$.

We thus conclude that $L(E) = L(E')$ where E' is an EDT0L grammar of index $|V| + 1$. □

Using Lemma 3.5, we have the following result.

Lemma 3.6 (Proposition 25 in [25]). *Let $L \subseteq \Sigma^*$ and $c \notin \Sigma$. If the language*

$$L \uparrow c^* \coloneqq \left\{ c^{n_0} w_1 c^{n_1} w_2 c^{n_2} \cdots w_k c^{n_k} \in (\Sigma \cup \{c\})^* \;\middle|\; \begin{array}{r} w = w_1 w_2 \cdots w_k \in L \text{ and} \\ n_0, n_1, n_2, \ldots, n_k \in \mathbb{N} \end{array} \right\}$$

is EDT0L, then L is EDT0L of finite index.

Proof. Suppose that $E = (\Sigma \cup \{c\}, V, I, H)$ is an EDT0L grammar for the language $L \uparrow c^*$. We then introduce a monoid homomorphism $h\colon (\Sigma \cup \{c\} \cup V)^* \to (\Sigma \cup V)^*$ such that

$$h(c) = \varepsilon \quad \text{and} \quad h(v) = v \text{ for each } v \neq c.$$

From this, we define an EDT0L grammar $E' = (\Sigma, V, I, H')$ where $H' = \{th \mid t \in H\}$. Notice that since c is a terminal letter of the grammar E, deleting the letter c after every application of a table is equivalent to removing all instances of c at the end of a production. Thus, we have $L(E') = L(E) \cdot h = L$. It only remains to be shown that E' is a EDT0L language of finite index. We demonstrate this by proving that E' is a LULT grammar, after which we apply Lemma 3.5 to obtain our result.

To simplify this proof, we define a (length-preserving) monoid homomorphism $\varphi\colon H^* \to (H')^*$ which we define such that $\varphi(t) = th \in H'$ for each $t \in H$. Notice also that $H' = \varphi(H)$. Moreover, since c is a terminal letter, we see that $w \cdot \varphi(\alpha) = w \cdot (\alpha h)$ for each $\alpha \in H^*$ and each $w \in (\Sigma \cup \{c\} \cup V)^*$, that is, removing the letter c after every application of a table is equivalent to removing every c only at the end.

Suppose we are given a word $w' = w_1 w_2 \cdots w_k \in L(E')$, then it follows that

$$w = w_1 c w_2 c^2 w_3 c^3 \cdots w_k c^k \in L \uparrow c^*.$$

We then see that there exists some $\alpha = h_1 h_2 \cdots h_k \in H^*$ with $w = I \cdot \alpha$ and thus $w' = I \cdot (\alpha h) = I \cdot \varphi(\alpha)$. We now show that the sequence $\alpha' = h'_1 h'_2 \cdots h'_k \in H'$, where each $h'_i = \varphi(h_i)$, is a choice of a sequence of tables which generates w' in E' and satisfies the constraints of a LULT grammar (see Definition 3.4).

Suppose for contradiction that there is a factorisation $\alpha' = \alpha'_1 \alpha'_2$ and a nonterminal $v \in V$ such that both $|I \cdot \alpha'_1|_v \geqslant 2$ and $|v \cdot \alpha'_2| \geqslant 2$. Let $\alpha = \alpha_1 \alpha_2$ be the unique factorisation of $\alpha \in H^*$ with $\alpha'_1 = \varphi(\alpha_1)$ and $\alpha'_2 = \varphi(\alpha_2)$. We then have the follow two observations:

- Since $|I \cdot \alpha'_1|_v \geqslant 2$, from the definition of φ, we see that $|I \cdot \alpha|_v \geqslant 2$, that is, $I \cdot \alpha_1$ contains at least 2 distinct instances of the variable v.
- Since $|v \cdot \alpha'_2| \geqslant 2$, from the definition of the word w and the map φ, we see that $v \cdot \alpha_2$ must contain a factor of the form $\sigma_1 c^m \sigma_2$ where $\sigma_1, \sigma_2 \in \Sigma$ and $m \in \mathbb{N}_+$.

From these observations, we see that the word $w = I \cdot (\alpha_1 \alpha_2)$ must contain two distinct factors of the form $\sigma_1 c^m \sigma_2$. This contradicts our choice of word w. Hence, we conclude that either $|I \cdot \alpha'_1|_v \leqslant 1$ or $|v \cdot \alpha'_2| \leqslant 1$ holds. From this, we then see that E' is a LULT grammar.

From Lemma 3.5, we conclude that the language L is EDT0L of finite index. □

4. EDT0L is closed under application of string transducers

In this section, we furnish a proof that the family of EDT0L languages is closed under mapping by a *string transducer*, also known as a *deterministic finite-state transducer*, or a *deterministic generalised sequential machine (deterministic gsm)*. We begin with the following definition.

Definition 4.1. A (deterministic) *string transducer* is a tuple $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ where

- Γ and Σ are the *input* and *output alphabets*, respectively;
- Q is a finite set of *states*;
- $A \subseteq Q$ is a finite set of *accepting states*;
- $q_0 \in Q$ is the *initial state*; and
- $\delta\colon \Gamma \times Q \to \Sigma^* \times Q$ is a *transition function*.

Given a language $L \subseteq \Gamma^*$, we may then define the language $M(L) \subseteq \Sigma^*$ as

$$M(L) = \left\{ u_1 u_2 \cdots u_k \in \Sigma^* \;\middle|\; \begin{array}{r} \text{there exists some word } w = w_1 w_2 \cdots w_k \in L \subseteq \Gamma^* \\ \text{such that } \delta(w_i, q_{i-1}) = (u_i, q_i) \text{ for each } i \in \{1, 2, \ldots, k\} \\ \text{where } q_0 \text{ is the initial state, and } q_1, q_2, \ldots, q_k \in Q \text{ with } q_k \in A \end{array} \right\}.$$

We then say that $M(L)$ is the image of L under mapping by the string transducer M.

Example 4.2. Here is a simple example of a string transducer, which computes the successor of a non-negative integer. A non-negative integer is represented as a word of the form $w\$$ where $w \in \{0,1\}^*$ is the minimum-length binary encoding of the integer (with the least significant digit appearing first), and \$ is an end-of-input symbol. For example, we encode the numbers 0, 4, 10 and 11 as \$, 001\$, 0101\$ and 1101\$, respectively. Notice that 100\$ would not be a valid string as it does not represent the number 1 with minimal length. The string transducer $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ where $\Gamma = \Sigma = \{0, 1, \$\}$, $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $A = \{q_4\}$, and $\delta\colon \Gamma \times Q \to \Sigma^* \times Q$ is described by

$$\begin{aligned}
&\delta(0,q_0) = (1,q_1), \quad \delta(0,q_1) = (0,q_1), \quad \delta(0,q_2) = (0,q_1), \quad \delta(0,q_3) = (0,q_3), \quad \delta(0,q_4) = (0,q_3),\\
&\delta(1,q_0) = (0,q_0), \quad \delta(1,q_1) = (1,q_2), \quad \delta(1,q_2) = (1,q_2), \quad \delta(1,q_3) = (1,q_3), \quad \delta(1,q_4) = (1,q_3),\\
&\delta(\$,q_0) = (1\$,q_4), \quad \delta(\$,q_1) = (\$,q_3), \quad \delta(\$,q_2) = (\$,q_4), \quad \delta(\$,q_3) = (\$,q_3), \quad \delta(\$,q_4) = (\$,q_3)
\end{aligned}$$

computes the successor of a given number. For example, if \$, 001\$, 0101\$ and 1101\$ are given to the transducer, then it will output the words 1\$, 101\$, 1101\$ and 00101\$, respectively. We can represent the string transducer by the graph in Figure 3 where the vertices are given by the state set Q, and for each transition $\delta(u, q) = (v, q')$ there is a labelled edge of the form $q \to^{u/v} q'$.

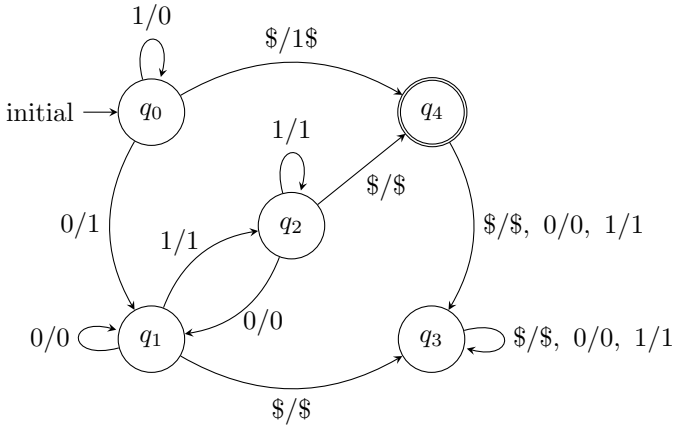

FIGURE 3. Add one to a number encoded in binary with an end marker.

Note that the state

- q_0 corresponds to prefixes of the form 1^n for some $n \in \mathbb{N}$;
- q_1 corresponds to prefixes of the form $w0$ where $w \in \{0,1\}^*$;
- q_2 corresponds to prefixes which can be written as $w0v1$ where $w, v \in \{0,1\}^*$;
- q_3 corresponds to invalid input sequences; and
- q_4 corresponds to valid sequences of the form $w\$$ where $w \in \{0,1\}^*$.

For the interested reader, further examples of string transducers abound in the literature on self-similar groups, see for example [20].

The following definition provides some useful notation when working with string transducers.

Definition 4.3. Let $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ be a string transducer. Then, for each pair of states $q, q' \in Q$, and words $w = w_1 w_2 \cdots w_k \in \Gamma^*$ and $w' \in \Sigma^*$, we write $q \to^{(w,w')} q'$ if there is a path from state q to q' which rewrites the word w to w'; that is, if there is a sequence of states $q_1, q_2, \ldots, q_{k+1} \in Q$ such that

- $q = q_1$ and $q' = q_{k+1}$; and
- $\delta(w_i, q_i) = (u_i, q_{i+1})$ for each $i \in \{1, 2, \ldots, k\}$ where $w' = u_1 u_2 \cdots u_k$.

Notice then that

$$M(L) = \{w' \in \Sigma^* \mid q_0 \to^{(w,w')} q \text{ where } w \in L \text{ and } q \in A\}$$

for each $L \subseteq \Gamma^*$.

We then have the following.

Lemma 4.4 (Corollary 4.7 in [16])**.** *The family of EDT0L languages is closed under applying a string transducer. That is, if L is an EDT0L language, and M is a string transducer, then $M(L)$ is also an EDT0L language.*

Proof. Let L be an EDT0L language with EDT0L grammar $E = (\Gamma, V, I, H)$, and $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ be a string transducer. In this proof, we construct an EDT0L grammar $E' = (\Sigma, V', I', H')$ for the language $M(L)$, thus showing that $M(L)$ is EDT0L.

In this paragraph we give an informal description of our construction. We construct our grammar E' such that it contains a nonterminal for every choice of nonterminal $v \in V$ of E, and every choice of states $q, q' \in Q$ of M. That is, E' will have nonterminals of the form $X_{v,q,q'}$ where $v \in V$ and $q, q' \in Q$. We construct E' such that it has the property that, if

$$w' = w'_0 X_{v_1,q_1,q'_1} w'_1 X_{v_2,q_2,q'_2} w'_2 \cdots X_{v_k,q_k,q'_k} w'_k \tag{2}$$

is a sentential form that appears in some production of E' where each $w'_i \in \Sigma^*$, then there exists some sentential form

$$w = w_0 v_1 w_1 v_2 w_2 \cdots v_k w_k$$

which appears in a production of E where each $w_i \in \Gamma^*$ such that

- $q_0 \to^{(w_0,w'_0)} q_1$,
- $q_i \to^{(w_i,w'_i)} q_{i+1}$ for each $i \in \{1, 2, \ldots, k-1\}$ and
- $q_k \to^{(w_k,w'_k)} q'$ for some $q' \in A$.

That is, the words w'_i in (2) correspond to words in Γ^* which have been rewritten by M, and the nonterminals X_{v_i,q_i,q'_i} are placeholders for words which are yet to be rewritten by M. Moreover, the states q_i, q'_i are included to ensure that the sentential form w' in (2) corresponds to a path in the automaton M. We achieve this by constructing the tables of E' such that, for every replacement

$$v \cdot h = w_0 u_1 w_1 u_2 w_2 \cdots u_k w_k, \tag{3}$$

where $v \in V$, $h \in H$, each $w_i \in \Gamma^*$ and each $u_i \in V$, we have a table $h' \in H'$ with the property that

$$X_{v,q,q'} \cdot h' = w'_0 X_{u_1,q_1,q'_1} w'_1 X_{u_2,q_2,q'_2} w'_2 \cdots X_{u_k,q_k,q'_k} w'_k \tag{4}$$

where

- $q \to^{(w_0,w'_0)} q_1$,
- $q_i \to^{(w_i,w'_i)} q_{i+1}$ for each $i \in \{1, 2, \ldots, k-1\}$ and
- $q_k \to^{(w_k,w'_k)} q'$.

Thus, the application of the table h' as in (4) corresponds to the application of h in (3) where each word in Γ^* has been rewritten by M, and we are representing a path in M from q to q'. It then follows that the language of all words in $M(L)$ corresponds to the union of the languages obtained from EDT0L grammars with initial nonterminals of the form $X_{I,q_0,a}$ where $a \in A$: we construct our grammar to correspond to this union by introducing a new start nonterminal I' along with some additional tables $t_{\text{init},a} \in H'$. We finish this informal description of our construction by noting that there may be cases where there are no choices of states $q_i, q'_i \in Q$ for which we can define an expansion as in (4), thus we introduce an additional nonterminal $\mathfrak{d}$ which we call a *dead-end symbol* which stops the production in such situations.

1. Nonterminals:

The set of nonterminals of E' is given as

$$V' = \{X_{v,q,q'} \mid v \in V \text{ and } q, q' \in Q\} \cup \{I', \mathfrak{d}\}.$$

Here, the nonterminals $X_{v,q,q'}$ of E' correspond to nonterminals $v \in V$ of E which generate words which are read by the string transducer M during a path from state q to q'; the nonterminals I' and $\mathfrak{d}$ are disjoint symbols which are used as the starting symbol and a *dead-end symbol* respectively. In our grammar, we ensure that every table maps $\mathfrak{d}$ to itself. Thus, if a $\mathfrak{d}$ appear in a sentential form, then there is no way of removing it to continue to generate a word in the language.

2. Initial tables $t_{\text{init,a}} \in H'$:

For each accepting state $a \in A$ of the string transducer M, we introduce a table $t_{\text{init},a} \in H'$ such that

$$v \cdot t_{\text{init},a} = \begin{cases} v & \text{if } v \in \Sigma, \\ X_{I,q_0,a} & \text{if } v = I', \\ \mathfrak{d} & \text{otherwise.} \end{cases}$$

(Note that we write $v \cdot t_{\text{init},a} = v$ for $v \in \Sigma$ only to comply with Definition 3.1, in our grammar to produce a word in the language such tables will only be applied to the sentential form I'.)

The remaining tables in H' are modified versions of tables in H, described as follows.

3. Tables $t_{h,r} \in H'$:

For each $h \in H$ and each $v \in V$, we have

$$v \cdot h = w_0 x_1 w_1 x_2 w_2 \cdots x_{k_v} w_{k_v}$$

for some $k_v \in \mathbb{N}$ where each $w_i \in \Sigma^*$ and each $x_i \in V$. Then for each pair of states $q, q' \in Q$, we define a finite set $C_{h,v,q,q'} \subset (\Sigma \cup V')^*$ as

$$C_{h,v,q,q'} = \left\{ u_0 \, X_{x_1,q_1,q_1'} \, u_1 \, X_{x_2,q_2,q_2'} \, u_2 \cdots X_{x_{k_v},q_{k_v},q_{k_v}'} \, u_{k_v} \;\middle|\; \begin{array}{l} q \to^{(w_0,u_0)} q_1, \\ q_i' \to^{(w_i,u_i)} q_{i+1} \text{ for each } 1 \leqslant i < k_v, \\ \text{and } q_{k_v}' \to^{(w_{k_v},u_{k_v})} q' \end{array} \right\}$$

Notice that each set $C_{h,v,q,q'}$ is finite as there are only finitely many choices for each state $q_i' \in Q$; the state q_1 is completely determined from q and w_0; for each $i \geqslant 1$, that state q_{i+1} is completely determined from q_i' and w_i; and the words $u_i \in \Sigma^*$ are completely determined from the state q and states q_i.

From the sets $C_{h,v,q,q'}$, we define a set of functions $R_h \subset ((\Sigma \cup V')^* \cup \{\mathfrak{d}\})^{V \times Q \times Q}$ as

$$R_h = \left\{ r \colon V \times Q \times Q \to ((\Sigma \cup V')^* \cup \{\mathfrak{d}\}) \;\middle|\; \begin{array}{r} \text{where } r(v,q,q') \in C_{h,v,q,q'} \cup \{\mathfrak{d}\} \\ \text{for each } v \in V \text{ and } q, q' \in Q \end{array} \right\}.$$

Notice that R_h is finite, in particular,

$$|R_h| \leqslant \prod_{v \in V} \prod_{q \in Q} \prod_{q' \in Q} (|C_{h,v,q,q'}| + 1).$$

Notice that we added $\mathfrak{d}$ as an option in R_h so that there is always a choice for the value of $r(v,q,q')$, in particular, so that there is a choice when $C_{h,v,q,q'}$ is empty.

For each $r \in R_h$, we then define a table $t_{h,r} \in H'$ such that

$$X_{v,q,q'} \cdot t_{h,r} = r(v,q,q')$$

for each $v \in V$, and each $q, q' \in Q$.

4. Soundness and completeness:

Suppose that $w \in L(E')$, then there must exist some $\tau_1, \tau_2, \ldots, \tau_k \in H'$ such that

$$w = I' \cdot (\tau_1 \tau_2 \cdots \tau_k).$$

From the definition of the initial tables $t_{\text{init},a}$, we see that

- $\tau_1 = t_{\text{init},a}$ for some $a \in A$; and
- $\tau_i = t_{h_i,r_i}$ for each $i \neq 1$.

From the construction of the tables of the form $t_{h,r}$, the word

$$u = I \cdot (h_2 h_3 \cdots h_k) \in L(E)$$

has the property that $q_0 \to^{(u,w)} a$, and thus $w \in M(L(E))$. Hence, $M(L(E)) \subseteq L(E')$.

Suppose that $w \in M(L(E))$, then there exists some word $u \in L(E)$ and an accepting state $a \in A$ such that $q_0 \to^{(u,w)} a$. Thus, there exists a sequence $h_1 h_2 \cdots h_k \in H^*$ such that

$$u = I \cdot (h_1 h_2 \cdots h_k) \in L(E).$$

From the construction of our tables, we then see that

$$w = I' \cdot (t_{\text{init},a} t_{h_1,r_1} t_{h_2,r_2} \cdots t_{h_k,r_k})$$

for some choice of functions $r_1, r_2, \ldots, r_k$. Hence, we see that $L(E') \subseteq M(L(E))$.

We now conclude that $L(E') = M(L(E))$, and thus the family of EDT0L languages is closed under mapping by string transducer. □

The following lemma, which makes use of string transducers, will be used in the next section. Recall that a subset $W \subset \Gamma^+$ is an *antichain* with respect to prefix order if for each choice of words $u, v \in W$, the word u is not a proper prefix of v.

Lemma 4.5. *Let $W \subset \Gamma^+$ be a finite antichain with respect to prefix order. For each word $w \in W$, we fix a word $x_w \in \Sigma^*$. Define a map $f \colon \mathcal{P}(\Gamma^*) \to \mathcal{P}(\Sigma^*)$ as*

$$f(L) = \{x_{w_1} x_{w_2} \cdots x_{w_k} \in \Sigma^* \mid w_1 w_2 \cdots w_k \in L \text{ where each } w_i \in W\}.$$

Then, there is a string transducer $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ such that $f(L) = f(L \cap W^*) = M(L)$.

Proof. Firstly we notice that if $W = \emptyset$, then $f(L) = \emptyset$ for each language $L \subseteq \Gamma^*$. In this case, any such string transducer with $A = \emptyset$ satisfies the lemma statement. Thus, in the remainder of this proof, we assume that $W \neq \emptyset$.

Let $w \in \Gamma^*$ be a word for which $w \in W^*$. Then, since W is a finite antichain with respect to the prefix order, there is a unique factorisation of w as $w = w_1 w_2 \cdots w_k$ where each $w_i \in W$.

We construct a string transducer $M = (\Gamma, \Sigma, Q, A, q_0, \delta)$ as follows. For each proper prefix $u \in \Gamma^*$ of a word $w \in W$, we introduce a state $q_u \in Q$. The initial state is $q_0 = q_\varepsilon$, and the set of accepting states is $A = \{q_\varepsilon\}$. Further, our automaton has one additional state q_{fail} which is a fail state; that is,

$$\delta(g, q_{\text{fail}}) = (\varepsilon, q_{\text{fail}})$$

for each $g \in \Gamma$. We then specify the remaining transitions as follows.

For each state q_u with $u \in \Gamma^*$, and each $g \in \Gamma$, we define the transition

$$\delta(g, q_u) = \begin{cases} (x_w, q_\varepsilon) & \text{if } w = ug \in W, \\ (\varepsilon, q_{ug}) & \text{if } ug \text{ is a proper prefix of some } w \in W, \\ (\varepsilon, q_{\text{fail}}) & \text{otherwise.} \end{cases}$$

We then see that the string transducer M is now completely specified. It is clear from our construction that $f(L) = M(L)$ for each $L \subseteq \Gamma^*$. □

5. EDT0L word problem

A standard approach to showing that having word problem in a given formal class $\mathcal{C}$ is invariant under change of finite generating set is to rely on the fact that $\mathcal{C}$ is closed under inverse monoid homomorphism. For example, this holds when $\mathcal{C}$ is any class in the Chomsky hierarchy, indexed, ET0L, and finite-index EDT0L (for the class of finite-index EDT0L languages, see [31, Theorem 5]).

As noted in the introduction, EDT0L languages are not closed under taking inverse monoid homomorphism. In particular, the language $L_1 = \{a^{2^n} \mid n \in \mathbb{N}\}$ can easily be shown to be EDT0L, however, the language $L_2 = \{w \in \{a, b\}^* \mid |w| = 2^n \text{ for some } n \in \mathbb{N}\}$, which can be written as an inverse monoid homomorphism of L_1, is known to not be EDT0L (see either the proof of Theorem 1 on p. 22 of [30], or Corollary 2 on p. 22 of [12]). Instead, we prove Proposition A via the following steps.

Recall from Section 2 that if G is infinite then it has geodesics of arbitrary length.

Lemma 5.1. *Suppose that G is an infinite group with finite monoid generating set X. Fix a finite number of words $u_1, u_2, \ldots, u_k \in X^*$. Then there exists a choice of non-empty words $w_1, w_2, \ldots, w_k \in X^* \setminus \{\varepsilon\}$, such that each $\overline{w_i} = 1$ and*

$$W = \{w_1 u_1, w_2 u_2, \ldots, w_k u_k\}$$

is an antichain in the sense of Lemma 4.5; that is, for each choice of words $x, y \in W$, the word x is not a proper prefix of y.

Proof. We begin by constructing the words $w_1, w_2, \ldots, w_k$ as follows.

Let $\alpha_1 \in X$ be a nontrivial generator, that is, $\overline{\alpha_1} \neq 1$; then let $\beta_1 \in X^*$ be a geodesic with $\overline{\alpha_1 \beta_1} = 1$ (Notice that if X is a symmetric generating set, then we may choose $\beta_1 = \alpha_1^{-1}$). From this selection, we then define $w_1 = \alpha_1 \beta_1$. We now choose the words $w_2, w_3, \ldots, w_k$ sequentially as follows.

For each $i \geqslant 2$, we choose a geodesic $\alpha_i \in X^*$ with length $|\alpha_i| = |\alpha_{i-1}\beta_{i-1}| + 1$. We then choose a geodesic $\beta_i \in X^*$ such that $\overline{\alpha_i \beta_i} = 1$ (Notice that if X is a symmetric generating set, then we may choose $\beta_i = \alpha_i^{-1}$). Then, define $w_i = \alpha_i \beta_i$.

We have now selected a sequence of words $w_1, w_2, \ldots, w_k$. For each word w_i, let γ_i denote the longest prefix which is a geodesic. We then notice that

$$|\gamma_{i-1}| < |w_{i-1}| < |\alpha_i| \leqslant |\gamma_i|$$

for each $i \in \{2, 3, \ldots, k\}$. Thus, $|\gamma_1| < |\gamma_2| < \cdots < |\gamma_k|$ and $|\gamma_i| < |w_i u_i|$ for each i.

We now see that, if $w_i u_i$ is a proper prefix of some word $v \in X^*$, then γ_i is also the longest prefix of v which is a geodesic. Hence, we conclude that the set

$$W = \{w_1 u_1, w_2 u_2, \ldots, w_k u_k\}$$

is an antichain as required. □

Recall from the introduction that the word and coword problem for a group G with respect to a generating set X are denoted as

$$\mathrm{WP}(G,X) \coloneqq \{w \in X^* \mid \overline{w} = 1\} \qquad \text{and} \qquad \mathrm{coWP}(G,X) \coloneqq X^* \setminus \mathrm{WP}(G,X).$$

We have the following result for groups with EDT0L (co-)word problem.

Proposition 5.2. *Let G be a group with finite monoid generating set X. If the word problem $\mathrm{WP}(G,X)$ is EDT0L, then it is EDT0L of finite index. Moreover, if the coword problem $\mathrm{coWP}(G,X)$ is EDT0L, then it is EDT0L of finite index.*

Proof. Notice that if G is finite, then this result follows from the fact that the word problem of a finite group is a regular language (see [1, Theorem 1]) and Lemma 3.3. Thus, in the remainder of this proof we may assume that G is an infinite group.

Let c be a letter which is disjoint from the alphabet $X = \{x_1, x_2, \ldots, x_n\}$. From Lemma 5.1, there exists a choice of words $w_1, w_2, \ldots, w_n, w_{n+1} \in X^*$ such that

$$W = \{w_1x_1, w_2x_2, \ldots, w_nx_n, w_{n+1}\}$$

is an antichain with respect to the prefix order where each $\overline{w_i} = 1$.

We define a map $f\colon \mathcal{P}(X^*) \to \mathcal{P}((X \cup \{c\})^*)$ as

$$f(L) = \left\{ y_{i_1}y_{i_2}\cdots y_{i_k} \in (X \cup \{c\})^* \;\middle|\; \begin{array}{l} (w_{i_1}u_{i_1})(w_{i_2}u_{i_2})\cdots(w_{i_k}u_{i_k}) \in L \\ \text{where each } i_j \in \{1,2,\ldots,n+1\} \end{array} \right\}$$

where $u_i = y_i = x_i$ for each $i \in \{1,2,\ldots,n\}$, $u_{n+1} = \varepsilon$, and $y_{n+1} = c$. From Lemma 4.5, f is a mapping by string transducer.

Notice that $f(\mathrm{WP}(G,X)) = \mathrm{WP}(G,X) \uparrow c^*$ and $f(\mathrm{coWP}(G,X)) = \mathrm{coWP}(G,X) \uparrow c^*$. In particular, given some word $v = v_1v_2\cdots v_k \in X^*$, if $v \in \mathrm{WP}(G,X)$ then

$$v' \coloneqq (w_{n+1})^{i_1}v_1(w_{n+1})^{i_2}v_2(w_{n+1})^{i_3}v_3\cdots(w_{n+1})^{i_k}v_k(w_{n+1})^{i_{k+1}} \in \mathrm{WP}(G,X)$$

for each $i_j \in \mathbb{N}$ where $w_{n+1} \in X^*$ is the word as in the set W given above. Then $f(v')$ is the word $c^{i_1}v_1c^{i_2}v_2\cdots c^{i_k}v_kc^{i_{k+1}} \in \mathrm{WP}(G,X) \uparrow c^*$. Hence, we can construct any word in $\mathrm{WP}(G,X) \uparrow c^*$ in this way. A similar statement also holds for $\mathrm{coWP}(G,X)$ and $\mathrm{coWP}(G,X) \uparrow c^*$.

From Lemma 4.4, we see that $\mathrm{WP}(G,X) \uparrow c^*$ and $\mathrm{coWP}(G,X) \uparrow c^*$ are EDT0L languages. Thus, from Lemma 3.6, we conclude that $\mathrm{WP}(G,X)$ and $\mathrm{coWP}(G,X)$ are both EDT0L of finite index. □

We then obtain the following result as a corollary to the above Proposition.

Proposition A. *Let G be a group with finite monoid generating sets X and Y. If $\mathrm{WP}(G,X)$ (resp. $\mathrm{coWP}(G,X)$) is EDT0L, then $\mathrm{WP}(G,Y)$ (resp. $\mathrm{coWP}(G,Y)$) is EDT0L of finite index. These results also hold if Y instead generates a submonoid of G. In particular, this implies that, if a word problem is EDT0L, then it is EDT0L of finite index; and that having a word problem that is EDT0L of finite index is independent of choice of generating set.*

Proof. Let X be a finite monoid generating set for G, let Y be a finite monoid generating set for a submonoid of G, and suppose that $\mathrm{WP}(G,X)$ (resp. $\mathrm{coWP}(G,X)$) is EDT0L. Then by Proposition 5.2, $\mathrm{WP}(G,X)$ (resp. $\mathrm{coWP}(G,X)$) is EDT0L of finite index. Let $\psi\colon Y^* \to X^*$ be a monoid homomorphism such that $\psi(y) =_G y$ for each $y \in Y$. Then $\mathrm{WP}(G,Y) = \psi^{-1}(\mathrm{WP}(G,X))$ (resp. $\mathrm{coWP}(G,Y) = \psi^{-1}(\mathrm{coWP}(G,X))$). From [31, Theorem 5], it is known that the class of EDT0L languages of finite index is closed under inverse monoid homomorphisms, thus we conclude that the language $\mathrm{WP}(G,Y)$ (resp. $\mathrm{coWP}(G,Y)$) is EDT0L of finite index, so in particular the language is EDT0L. □

6. Non-branching multiple context-free languages

In this section, we introduce non-branching multiple context-free grammars, and define what it means for such a grammar to be non-permuting and non-erasing. We then show that every finite-index EDT0L language can be generated by a non-branching multiple context-free grammar which is both non-permuting and non-erasing. We conclude this section by providing a normal form for grammars of this type which is then used throughout the proof of our main result in Section 7.

Definition 6.1 (Non-branching multiple context-free grammar)**.** Let Σ be an alphabet, and Q be a finite set of symbols called *nonterminals* where each $H \in Q$ has a *rank* some positive integer. We define an *initiating rule* to be an expression of the form

$$H(u_1, u_2, \ldots, u_k) \leftarrow$$

where $H \in Q$ has rank k, and each $u_i \in \Sigma^*$. A *propagating rule* is an expression of the form

$$K(u_1, u_2, \ldots, u_m) \leftarrow H(x_1, x_2, \ldots, x_n)$$

where

- $K \in Q$ has rank m,
- $H \in Q$ has rank n,
- $x_1, x_2, \ldots, x_n$ are abstract variables,
- each $u_i \in (\Sigma \cup \{x_1, x_2, \ldots, x_n\})^*$, and
- for each $i \in \{1, 2, \ldots, n\}$, the word $u_1 u_2 \cdots u_m$ contains at most one instance of x_i.

A *non-branching multiple context-free grammar* is a tuple $M = (\Sigma, Q, S, P)$ where Σ is an alphabet, Q is a finite set of nonterminals, $S \in Q$ is the *starting nonterminal* which has rank 1, and P is a finite set of rewrite rules each of which is either initiating or propagating.

A word $w \in \Sigma^*$ is *generated* by the grammar if there is a sequence of rules from M starting with an initiating rule, then followed by propagating rules, and finishing at $S(w)$, that is, if there exists a sequence

$$S(w) \leftarrow H_1(v_{1,1}, v_{1,2}, \ldots, v_{1,m_1}) \leftarrow H_2(v_{2,1}, v_{2,2}, \ldots, v_{2,m_2}) \leftarrow \cdots \leftarrow H_k(v_{k,1}, v_{k,2}, \ldots, v_{k,m_k}) \leftarrow$$

where each $H_i \in Q$ with $H_k(v_{k,1}, v_{k,2}, \ldots, v_{k,m_k}) \leftarrow$ initiating, each $v_{i,j} \in \Sigma^*$, and each replacement follows from P. We refer to the symbols x_i used to define a propagating rule as the *variables* of the rule. We refer to a sequence, as above, as a *derivation* of the word w. The language *generated* by M is the set of all words generated by M, that is, the set of all words which have a derivation as above.

A non-branching multiple context-free grammar is *non-permuting* if in each propagating replacement rule, the variables x_i which appear in $u_1 u_2 \cdots u_m$, appear in the same order as on the right-hand side of the rule. Moreover, a grammar is *non-erasing* if for each propagating replacement rule, each variable x_i which appears in the right-hand side also appears in the word $u_1 u_2 \cdots u_m$. Notice then that if a non-branching multiple context-free grammar is both non-permuting and non-erasing, then we have

$$u_1 u_2 \cdots u_m \in \Sigma^* x_1 \Sigma^* x_2 \Sigma^* \cdots x_n \Sigma^*$$

for each propagating rule.

As noted in the introduction, we call a non-permuting non-erasing non-branching multiple context-free grammar a *restricted multiple context-free grammar*, abbreviated as R-MCFG.

Proposition 6.2. *Let L be EDT0L of finite index. Then L is generated by an R-MCFG.*

Proof. Let $E = (\Sigma, V, I, H)$ be an EDT0L grammar of index n. In this proof, we construct an R-MCFG $M = (\Sigma, Q, S, P)$.

For each sequence $v_1 v_2 \cdots v_k \in V^*$ with length $k \leqslant n$, we introduce a nonterminal $H_{v_1 v_2 \cdots v_k} \in Q$ with rank $k+1$ to our non-branching multiple context-free grammar. Notice then that there are finitely many such nonterminals, in particular, $|Q| \leqslant (|V| + 1)^n$.

In our construction, a configuration of the form

$$H_{v_1 v_2 \cdots v_k}(u_0, u_1, u_2, \ldots, u_k),$$

where each $u_i \in \Sigma^*$, corresponds to a sentential form $u_0 v_1 u_1 v_2 u_2 \ldots v_k u_k$ generated by the EDT0L grammar E. Thus, the starting nonterminal of our non-branching multiple context-free grammar is $S = H_\varepsilon$.

We begin our production by introducing an initiating rule of the form

$$H_I(\varepsilon, \varepsilon) \leftarrow$$

which corresponds to a word containing only the starting symbol I of E. For each table $h \in H$ of our EDT0L grammar, we introduce a monoid endomorphism

$$\overline{h} \colon (\Sigma \cup V \cup \{x_0, x_1, x_2, \ldots, x_n\})^* \to (\Sigma \cup V \cup \{x_0, x_1, x_2, \ldots, x_n\})^*$$

such that $\overline{h}(u) = h(u)$ for each $u \in \Sigma \cup V$, and $\overline{h}(x_i) = x_i$ for each $i \in \{0, 1, 2, \ldots, n\}$. We note here that these additional disjoint symbols $x_0, x_1, \ldots, x_n$ will correspond to words in Σ^*.

For each nonterminal $H_{v_1 v_2 \cdots v_k} \in Q$, we consider the word

$$W_{h,H,v_1 v_2 \cdots v_k} \coloneqq \overline{h}(x_0 v_1 x_1 v_2 x_2 v_3 x_3 \cdots v_k x_k) \in (\Sigma \cup V \cup \{x_0, x_1, x_2, \ldots, x_k\})^*.$$

Notice that $W_{h,H,v_1 v_2 \cdots v_k}$ contains exactly one instance of each symbol $x_0, x_2, \ldots, x_k$, and that these symbols appear in the same order in which they were given to the map $\overline{h}$.

If the word $W_{h,H,v_1v_2\cdots v_k}$ contains at most n instances of letters in V, then we may decompose it as

$$W_{h,H,v_1v_2\cdots v_k} = w_0v_1'w_1v_2'w_2\cdots v_m'w_m$$

where $m \leqslant n$, each $w_i \in (\Sigma \cup \{x_0, x_1, x_2, \ldots, x_k\})^*$, and each $v_i' \in V$. We introduce a propagating rule

$$H_{v_1'v_2'\cdots v_m'}(w_0, w_1, \ldots, w_m) \leftarrow H_{v_1v_2\cdots v_k}(x_1, x_2, \ldots, x_k).$$

Notice that this new rule is non-erasing and non-permuting from our earlier observations on $W_{h,H,v_1v_2\cdots v_k}$.

From our construction, we see that the non-branching multiple context-free language described in this proof is both non-permuting and non-erasing, so is an R-MCFG, and generates the finite-index EDT0L language $L(E)$. □

Remark 6.3. The converse of Proposition 6.2 also holds, that is, every R-MCFG produces an EDT0L language of finite index. In particular, given an R-MCFG one may construct an equivalent EDT0L language with nonterminals of the form $X_{H,k}$ where H is a predicate of the R-MCFG and $k \in \{0, 1, \ldots, r_H\}$ where r_H is the rank of H. One would then construct the EDT0L grammar such that productions of the R-MCFG of the form $H(u_1, u_2, \ldots, u_k)$ correspond to sentential forms $X_{H,0}\, u_1\, X_{H,1}\, u_2\, X_{H,2} \cdots u_k\, X_{H,k}$ of the EDT0L grammar. We do not prove this result as it is not required by the results in this paper.

We now describe three types of replacement rules for non-branching multiple context-free grammars as follows. These types of rules will be used to describe a normal form for R-MCFGs. We begin with insertions rules as follows.

Definition 6.4. A replacement rule is a *left* or *right insertion* if it is of the form

$$K(x_1, x_2, \ldots, x_{i-1}, \sigma x_i, x_{i+1}, \ldots, x_k) \leftarrow H(x_1, x_2, \ldots, x_k)$$

or

$$K(x_1, x_2, \ldots, x_{i-1}, x_i\sigma, x_{i+1}, \ldots, x_k) \leftarrow H(x_1, x_2, \ldots, x_k),$$

respectively, for some $\sigma \in \Sigma$ and $i \in \{1, 2, \ldots, k\}$.

We then require a type of replacement rule which allows us to move the variables x_i around. We do so using merge rules defined as follows.

Definition 6.5. A replacement rule is a *left* or *right merge* if it is of the form

$$K(x_1, x_2, \ldots, x_{i-1}, x_ix_{i+1}, \varepsilon, x_{i+2}, \ldots, x_k) \leftarrow H(x_1, x_2, \ldots, x_k)$$

or

$$K(x_1, x_2, \ldots, x_{i-1}, \varepsilon, x_ix_{i+1}, x_{i+2}, \ldots, x_k) \leftarrow H(x_1, x_2, \ldots, x_k),$$

respectively, for some $i \in \{1, 2, \ldots, k-1\}$.

Further, we require a type of replacement rule which generate words as follows.

Definition 6.6. An *accepting replacement rule* is one of the form

$$S(x_1x_2\cdots x_k) \leftarrow H(x_1, x_2, \ldots, x_k)$$

where S is the starting nonterminal of the non-branching multiple context-free grammar.

From these types of replacement rules as described in the previous definitions, we may now define normal forms for R-MCFGs as follows.

Definition 6.7. An R-MCFG $M = (\Sigma, Q, S, P)$ is in *normal form* if there exists some $k \geqslant 2$ such that every nonterminal $H \in Q$, except for the starting nonterminal S, has rank k, and each replacement rule in P is either initiating of the form

$$H(\underbrace{\varepsilon, \varepsilon, \varepsilon, \varepsilon, \ldots, \varepsilon}_{k \text{ components}}) \leftarrow,$$

a left or right insertion (as in Definition 6.4), a left or right merge (as in Definition 6.5), or an accepting rule (as in Definition 6.6). Moreover, every production of the grammar has the form

$$S(w) \leftarrow H(w, \varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow \cdots \leftarrow K(\varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow$$

where $w \in \Sigma^*$ and $H, K \in Q$.

The use of the term 'normal form' in the above definition is justified by the following lemma.

Lemma 6.8. *Suppose that $L \subseteq \Sigma^*$ is the language of an R-MCFG. Then, L can be generated by an R-MCFG in normal form.*

Proof. Let $M = (\Sigma, Q, S, P)$ be an R-MCFG. Let $k \in \mathbb{N}$ be the smallest value for which both $k \geqslant 2$ and for each replacement rule

$$H(w_1, w_2, \ldots, w_\ell) \leftarrow \qquad \text{and} \qquad H(w_1, w_2, \ldots, w_n) \leftarrow K(x_1, x_2, \ldots, x_m)$$

in P, we have $k \geqslant \max\{\ell, m, n\}$. Such a constant k exists as there are finitely many replacement rules.

In this proof, we construct three R-MCFGs M', M'' and M'''. At the end of this proof, we have a grammar M''' in normal form which generates the same language as M.

The techniques used in this proof are quite straightforward. In step 1, we ensure that all nonterminals have the same rank by simply padding each predicate with empty components, as required; in step 2, we unfold each production of the grammar into a sequence of rules each of the form Definitions 6.4, 6.5 and 6.6 or a trivial rule of the form

$$H(x_1, x_2, \cdots, x_k) \leftarrow K(x_1, x_2, \ldots, x_k);$$

then finally, in step 3, we remove these additional trivial rules to produce a grammar in normal form.

Step 1: We construct an R-MCFG, denoted as $M' = (\Sigma, Q', S', P')$, from M as follows.

For each nonterminal $A \in Q$ of the grammar M, we introduce the nonterminals

$$A_0, A_1, A_2, \ldots, A_k \in Q'.$$

In our construction of M', we ensure that if

$$A_i(w_1, w_2, \ldots, w_k)$$

appears in a derivation, then $w_j = \varepsilon$ for each $j > i$. Thus, the subscript of these nonterminals count the number of non-empty components. Moreover, we introduce the nonterminals

$$S', F' \in Q'$$

to our grammar M'. These additional nonterminals are used in the grammar M' as follows.

The grammar M' has the replacement rules

$$S'(x_1 x_2 \ldots x_k) \leftarrow S_1(x_1, x_2, \ldots, x_k) \qquad \text{and} \qquad F'(\underbrace{\varepsilon, \varepsilon, \varepsilon, \varepsilon, \ldots, \varepsilon}_{k \text{ components.}}) \leftarrow$$

For each replacement rule

$$H(w_1, w_2, \ldots, w_\ell) \leftarrow$$

in P, we introduce a replacement rule

$$H_\ell(w_1 x_1, w_2 x_2, \ldots, w_\ell x_\ell, x_{\ell+1}, \ldots, x_k) \leftarrow F'(x_1, x_2, \ldots, x_k)$$

to P'. Moreover, for each replacement rule of the form

$$H(w_1, w_2, \ldots, w_n) \leftarrow K(x_1, x_2, \ldots, x_m)$$

in P, we introduce a replacement rule

$$H_n(w_1, w_2, \ldots, (w_n x_{m+1} x_{m+2} \cdots x_k), \varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow K_m(x_1, x_2, \ldots, x_k)$$

to P'. This completes our construction of M'.

Properties of M': We see that M' generates exactly the same language as M and that all derivations in the grammar M' have the form

$$S'(w) \leftarrow S_1(w, \varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow \cdots \leftarrow F'(\varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow .$$

The nonterminal S' only appears as the leftmost nonterminal of a derivation, and each nonterminal $A \in Q' \setminus \{S'\}$ has rank k.

Step 2: We construct an R-MCFG, which we denote as $M'' = (\Sigma, Q'', S', P'')$, from M' by unfolding each replacement of M' into a sequence of more basic replacements as follows.

The nonterminals of the grammar M'' contain the nonterminals of M', that is, $Q' \subseteq Q''$. In this stage of the proof, we show how to decompose the replacement rules of M' into sequences of finitely many replacement rules, each of which of the form as described in Definition 6.7, or of the form

$$H(x_1, x_2, \ldots, x_k) \leftarrow K(x_1, x_2, \ldots, x_k). \tag{5}$$

In Step 3 of this proof, we complete our construction by removing the replacement rules of the form (5).

Firstly, our grammar M'' contains the replacement rules

$$S'(x_1 x_2 \cdots x_k) \leftarrow S_1(x_1, x_2, \ldots, x_k) \qquad \text{and} \qquad F'(x_1, x_2, \ldots, x_k) \leftarrow$$

which are both a part of the grammar M'.

Suppose that the grammar M' has a replacement rule $p \in P'$ of the form

$$p\colon A(w_1, w_2, \ldots, w_k) \leftarrow B(x_1, x_2, \ldots, x_k)$$

where each

$$w_i = t_i\, z_{i,1}\, u[z_{i,1}]\, z_{i,2}\, u[z_{i,2}] \cdots z_{i,\ell_i}\, u[z_{i,\ell_i}]$$

for some $\ell_i \in \mathbb{N}$, each $z_{i,j} \in \{x_1, x_2, \ldots, x_k\}$, $t_i \in \Sigma^*$ and each $u[x_i] \in \Sigma^*$.

We then introduce a nonterminal of the form

$$C_{p,n,m} \in Q''$$

for each $n \in \{1, 2, \ldots, k\}$ and each $m \in \{0, 1, 2, \ldots, |u(x_n)|\}$. Let each

$$u[x_n] = u_{n,1} u_{n,2} \cdots u_{n,|u[x_n]|} \in \Sigma^*.$$

We then introduce replacement rules to M'' as follows.

- $C_{p,1,0}(x_1, x_2, \ldots, x_k) \leftarrow B(x_1, x_2, \ldots, x_k)$;
- for each $n \in \{1, 2, \ldots, k\}$ and each $m \in \big\{0, 1, 2, \ldots, \big|u[x_n]\big| - 1\big\}$, we have
$$C_{p,n,m+1}(x_1, x_2, \ldots, x_{n-1}, (x_n u_{n,m+1}), x_{n+1}, \ldots, x_k) \leftarrow C_{p,n,m}(x_1, x_2, \ldots, x_k);$$
- for each $n \in \{1, 2, \ldots, k-1\}$, we have
$$C_{p,n+1,0}(x_1, x_2, \ldots, x_k) \leftarrow C_{p,n,|u[x_n]|}(x_1, x_2, \ldots, x_k).$$

Notice then that we have the following production

$$C_{p,k,|u[x_k]|}(x_1 u[x_1], x_2 u[x_2], \ldots, x_k u[x_k]) \leftarrow \cdots \leftarrow B(x_1, x_2, \ldots, x_k)$$

in the grammar M''.

We now introduce a finite number of nonterminals of the form

$$D_{p,\vec{v}} \in Q''$$

where $\vec{v} = (v_1, v_2, \ldots, v_k) \in (\Sigma^*)^k$ is a vector with each

$$v_i \in \{x_1, x_2, \ldots, x_k\}^* \qquad \text{and} \qquad v_1 v_2 \cdots v_k = x_1 x_2 \cdots x_k.$$

For the grammar M', we then introduce all replacements of the form

$$D_{p,(x_1,x_2,\ldots,x_k)}(x_1, x_2, \ldots, x_k) \leftarrow C_{p,k,|u[x_k]|}(x_1, x_2, \ldots, x_k);$$

$$D_{p,(v_1,v_2,\ldots,v_{i-1},v_i v_{i+1},\varepsilon,v_{i+2},\ldots,v_k)}(x_1, x_2, \ldots, x_{i-1}, x_i x_{i+1}, \varepsilon, x_{i+2}, \ldots, x_k) \leftarrow D_{p,(v_1,v_2,\ldots,v_k)}(x_1, x_2, \ldots, x_k)$$

for each $\vec{v} = (v_1, v_2, \ldots, v_k)$ as before, and each $i \in \{1, 2, \ldots, k-1\}$; and

$$D_{p,(v_1,v_2,\ldots,v_{i-1},\varepsilon,v_i v_{i+1},v_{i+2},\ldots,v_k)}(x_1, x_2, \ldots, x_{i-1}, \varepsilon, x_i x_{i+1}, x_{i+2}, \ldots, x_k) \leftarrow D_{p,(v_1,v_2,\ldots,v_k)}(x_1, x_2, \ldots, x_k)$$

for each $\vec{v} = (v_1, v_2, \ldots, v_k)$ as before, and each $i \in \{1, 2, \ldots, k-1\}$.

Notice then that

$$D_{p,\vec{v}}(v_1, v_2, \ldots, v_k) \leftarrow^* C_{p,k,|u[x_k]|}(x_1, x_2, \ldots, x_k)$$

for each $D_{p,\vec{v}}$ as above.

Let

$$\vec{Z} = ((z_{1,1} z_{1,2} \cdots z_{2,\ell_1}),\ (z_{2,1} z_{2,2} \cdots z_{2,\ell_2}),\ \ldots,\ (z_{k,1} z_{k,2} \cdots z_{k,\ell_k})).$$

We then see that in the grammar M'', we have

$$D_{p,\vec{Z}}(w'_1, w'_2, \ldots, w'_k) \leftarrow \cdots \leftarrow B(x_1, x_2, \ldots, x_k)$$

where each

$$w'_i = z_{i,1}\, u(z_{i,1})\, z_{i,2}\, u(z_{i,2}) \cdots z_{i,\ell_i}\, u(z_{i,\ell_i}),$$

that is, $w_i = t_i w'_i$ where $t_i \in \Sigma^*$. We then introduce a nonterminal of the form

$$G_{p,n,m} \in Q''$$

for each $n \in \{1, 2, \ldots, k\}$ and each $m \in \{0, 1, 2, \ldots, |t_i|\}$. Let each

$$t_i = t_{i,1} t_{i,2} \cdots t_{i,|t_i|} \in \Sigma^*.$$

We then introduce replacement rules to M'' as follows.

- $G_{p,1,|t_1|}(x_1, x_2, \ldots, x_k) \leftarrow D_{p,\vec{Z}}(x_1, x_2, \ldots, x_k)$;
- for each $n \in \{1, 2, \ldots, k\}$ and each $m \in \{1, 2, \ldots, |t_n|\}$, we have
$$G_{p,n,m-1}(x_1, x_2, \ldots, x_{n-1}, (t_{n,m}x_n), x_{n+1}, \ldots, x_k) \leftarrow G_{p,n,m}(x_1, x_2, \ldots, x_k);$$
- for each $n \in \{1, 2, \ldots, k-1\}$, we have
$$G_{p,n+1,|t_{n+1}|}(x_1, x_2, \ldots, x_k) \leftarrow G_{p,n,0}(x_1, x_2, \ldots, x_k); \text{ and}$$
- $A(x_1, x_2, \ldots, x_k) \leftarrow G_{p,k,0}(x_1, x_2, \ldots, x_k)$.

We then see that in the grammar M'' we now have the derivation
$$A(w_1, w_2, \ldots, w_k) \leftarrow \cdots \leftarrow B(x_1, x_2, \ldots, x_k)$$
where each rule in the sequence is either of the form as described in Definition 6.7, or the form as in (5).

Properties of M'': Observe that the grammar M'' generates exactly the same language as M', and thus as M. Moreover, the only thing preventing M'' from being in normal form is that it contains rules of the form given in (5) which we remove in the following step.

Step 3:

Suppose that M'' contains a replacement rule as in (5), that is, a replacement rule of the form
$$p\colon H(x_1, x_2, \ldots, x_k) \leftarrow K(x_1, x_2, \ldots, x_k).$$
Then, we modify M'' by
- removing the nonterminal K from Q'';
- removing the replacement rule p from P''; and
- replacing each instance of the nonterminal K with H in each rule contained in P''.

Notice that after this modification, the number of rules of the form (5) is reduced by one, and that the grammar generates the same language. Moreover, we notice by induction that we can remove all such rules. We refer to the resulting grammar as $M''' = (\Sigma, Q''', S', P''')$.

Conclusion:

From the properties of the grammar M'' in step 2, and the procedure described in step 3, we see that we may generate a grammar M''' in normal form for the language L. □

7. Proof of the main theorem

Our aim in this section is to prove Theorem C. From Proposition A, it suffices to prove this theorem for a specific generating set. Thus, we introduce the following language.

Definition 7.1. Let $\Sigma = \{a, b\}$, and let $\Delta\colon \Sigma^* \to \mathbb{Z}$ to be the monoid homomorphism defined such that $\Delta(a) = 1$ and $\Delta(b) = -1$. Let $L \subseteq \Sigma^*$ as
$$L = \{w \in \Sigma^* \mid \Delta(w) = 0\}$$
represent the word problem for $\mathbb{Z}$ with respect to the generating set $X = \{1, -1\}$.

The remainder of this section is organised as follows. We begin by introducing the constant C (in Corollary 7.4) and the *counterexample word* $\mathcal{W}$ (see Definition 7.5 in Section 7.1). We note here that the word $\mathcal{W}$ belongs to the language L, as in Definition 7.1, as is used to show that L is not EDT0L by contradiction. Sections 7.2 and 7.3 introduce several constants and functions on words which are used throughout the proofs of technical lemmas in this section. These constants correspond to the lengths of certain factors of $\mathcal{W}$. Our motivation for introducing these constants and functions is so that we can define and study the properties of *n-decompositions*. In Section 7.4, we introduce the important concept of an *n-decomposition* (in Definition 7.20), we then spend the rest of this subsection proving technical lemmas which are used later in this section. In Section 7.5, we prove three propositions (Propositions 7.28, 7.29 and 7.30) which are used to show that, in some way, having a decomposition would be maintained during the production of $\mathcal{W}$ of an R-MCFG for L (from this we obtain a contradiction). These three propositions are summarised and generalised in Proposition 7.31 in Section 7.6. In Section 7.7, we use Proposition 7.31 to prove Theorem C, then obtain Theorem D as a corollary to Theorem C.

In the remainder of this paper L, Σ, Δ, a and b shall refer to the values as in Definition 7.1. To simplify the proofs in this section, we extend $\Delta\colon \Sigma^* \to \mathbb{Z}$ to the domain of all k-tuples as follows.

Definition 7.2. Let $\vec{w} = (w_1, w_2, \ldots, w_k) \in (\Sigma^*)^k$ be a tuple of words, then $\Delta(\vec{w}) \coloneqq \Delta(w_1 w_2 \cdots w_k)$.

From Propositions 5.2 and 6.2 and Lemma 6.8, we see that in order to prove Theorem C, it is sufficient to show that L cannot be generated by an R-MCFG in normal form. We have the following lemma and corollary which place restrictions on the structure of such a grammar.

Lemma 7.3. *If L is generated by some R-MCFG $M = (\Sigma, Q, S, P)$, then for each nonterminal $H \in Q$, there exists a constant $C_H \in \mathbb{Z}$ such that if*

$$S(w) \leftarrow \cdots \leftarrow H(u_1, u_2, \ldots, u_k) \leftarrow \cdots \leftarrow$$

is a derivation in the grammar M, then $\Delta(u_1 u_2 \cdots u_k) = C_H$.

For example, we have $C_S = 0$ where S is the start non-terminal of the grammar.

Proof. Let some nonterminal $H \in Q$ be chosen and suppose that there exists at least one derivation of the form

$$S(g) \leftarrow \cdots \leftarrow H(v_1, v_2, \ldots, v_k) \leftarrow \cdots \leftarrow \tag{6}$$

where $g, v_1, v_2, \ldots, v_k \in \Sigma^*$. To simplify the explanations in this proof, we write $\beta \in P^*$ for the sequence of replacement rules which generate the configuration $H(v_1, v_2, \ldots, v_k)$, and $\alpha \in P^*$ for the sequence of replacement rules which generates $S(g)$ from $H(v_1, v_2, \ldots, v_k)$. We then write

$$S(g) \leftarrow^{\alpha} H(v_1, v_2, \ldots, v_k) \leftarrow^{\beta}$$

to denote the sequence in (6).

From the definition of an R-MCFG, we then see that there exist words $m_0, m_1, m_2, \ldots, m_k \in \Sigma^*$ such that

$$S(m_0 p_1 m_1 p_2 m_2 \cdots p_k m_k) \leftarrow^{\alpha} H(p_1, p_2, \ldots, p_k)$$

for each $p_1, p_2, \ldots, p_k \in \Sigma^*$. In particular, if g and v_i are as in (6), then $g = m_0 v_1 m_1 v_2 m_2 \cdots v_k m_k$.

Suppose that $u_1, u_2, \ldots, u_k \in \Sigma^*$ are words such that there exists some derivation

$$S(w) \leftarrow \cdots \leftarrow H(u_1, u_2, \ldots, u_k) \leftarrow \cdots \leftarrow$$

in the grammar M, and write $\gamma \in P^*$ for the sequence of replacements which generates $H(u_1, u_2, \ldots, u_k)$. That is,

$$H(u_1, u_2, \ldots, u_k) \leftarrow^{\gamma} .$$

We then see that

$$S(m_0 u_1 m_1 u_2 m_2 \cdots u_k m_k) \leftarrow^{\alpha} H(u_1, u_2, \ldots, u_k) \leftarrow^{\gamma}$$

is a derivation in the grammar L and thus

$$m_0 u_1 m_1 u_2 m_2 \cdots u_k m_k \in L.$$

Thus, from the definition of the language L, we see that

$$\Delta(m_0 u_1 m_1 u_2 m_2 \cdots u_k m_k) = 0.$$

Since Δ is a monoid homomorphism onto an abelian group, we then see that

$$\Delta(u_1 u_2 \cdots u_k) + \Delta(m_0 m_1 \cdots m_k) = 0.$$

We thus conclude by setting $C_H = -\Delta(m_0 m_1 \cdots m_k)$. □

From this lemma, we have the following immediate corollary.

Corollary 7.4. *Let L be generated by some R-MCFG $M = (\Sigma, Q, S, P)$. Then there exists a constant $C = C_M \geqslant 1$ such that, if*

$$S(w) \leftarrow \cdots \leftarrow H(u_1, u_2, \ldots, u_k) \leftarrow \cdots \leftarrow$$

is a derivation in the grammar M, then $|\Delta(u_1 u_2 \cdots u_k)| \leqslant C$.

Proof. For each nonterminal $H \in Q$, let $C_H \in \mathbb{Z}$ be the constant as in Lemma 7.3. We then see that our result follows with the constant $C_M = \max\{|C_H| \mid H \in Q\}$ which is well-defined since Q is finite. □

7.1. **Counterexample word.** In the remainder of this section, suppose for contradiction that there exists an R-MCFG $M = (\Sigma, Q, S, P)$ in normal form recognising L. Moreover, write C for the constant in Corollary 7.4, and write k for the rank of the nonterminals in M that are not the starting nonterminal.

We show that such a grammar M cannot exist by constructing a word $\mathcal{W} \in L$ which cannot be generated by any derivation in M.

Definition 7.5. Let $m = 24k + 2C + 5$. We define the word $\mathcal{W} \in \Sigma^*$ as

$$\mathcal{W} = a^{m^{2k}}\, \mathcal{T}_{2k}\, a^{m^{2k}}$$

where $\mathcal{T}_{2k}$ is defined recursively such that

$$\mathcal{T}_0 = b^2 \quad \text{and} \quad \mathcal{T}_n = a^{m^n} (\mathcal{T}_{n-1})^{2m} a^{m^n}$$

for each $n \in \{1, 2, 3, \dots, 2k\}$.

Remark 7.6. Figure 2 illustrates the word $\mathcal{W}$ for the values $m = 4$, $k = 2$ with the letter a represented by an up-step $(1, 1)$ and b by a down-step $(1, -1)$. Note that Definition 7.5 requires $m = 24k + 2C + 5$ so $m = 4$ is not a realistic value, but larger values of m would make the figure difficult to draw, and one can already start to see the general behaviour with these values. Here is a calculation of the word in this case. We have

$$\mathcal{T}_0 = b^2, \quad \mathcal{T}_1 = a^4 b^{16} a^4, \quad \mathcal{T}_2 = a^{16}\big(a^4 b^{16} a^4\big)^8 a^{16}, \quad \mathcal{T}_3 = a^{64}\big(a^{16}\big(a^4 b^{16} a^4\big)^8 a^{16}\big)^8 a^{64},$$

$$\mathcal{T}_4 = a^{256}\big(a^{64}\big(a^{16}\big(a^4 b^{16} a^4\big)^8 a^{16}\big)^8 a^{64}\big)^8 a^{256},$$

so

$$\mathcal{W} = a^{512}\big(a^{64}\big(a^{16}\big(a^4 b^{16} a^4\big)^8 a^{16}\big)^8 a^{64}\big)^8 a^{512}.$$

Notice that in the word $\mathcal{W}$, every factor of b is of length exactly 16.

In the remainder of this section, m denotes the constant in Definition 7.5. We then have the following observation of the words $\mathcal{W}$ and $\mathcal{T}_n$ from Definition 7.5.

Lemma 7.7. *For each $n \in \{0, 1, 2, 3, \dots, 2k\}$, we have $\Delta(\mathcal{T}_n) = -2m^n$ with*

$$|\mathcal{T}_n|_a = 2(2m)^n - 2m^n \quad \textit{and} \quad |\mathcal{T}_n|_b = 2(2m)^n.$$

Moreover, we have $\Delta(\mathcal{W}) = 0$ and thus $\mathcal{W} \in L$.

Proof. From the recursive definition of $\mathcal{T}_n$ in Definition 7.5, we have

$$\Delta(\mathcal{T}_0) = -2 \quad \text{and} \quad \Delta(\mathcal{T}_n) = 2m\,\Delta(\mathcal{T}_{n-1}) + 2\,m^n$$

for each $n \geqslant 1$. From this recurrence, we find that

$$\Delta(\mathcal{T}_n) = -2\,m^n$$

for each $n \geqslant 0$. Thus, $\Delta(\mathcal{T}_{2k}) = -2m^{2k}$. Again from Definition 7.5, we then have

$$\Delta(\mathcal{W}) = \Delta(\mathcal{T}_{2k}) + 2\,m^{2k} = 0.$$

From the recursive definition of $\mathcal{T}_n$, we also see that

$$|\mathcal{T}_0|_b = 2 \quad \text{and} \quad |\mathcal{T}_n|_b = 2m\,|\mathcal{T}_{n-1}|_b.$$

Thus, we see that $|\mathcal{T}_n|_b = 2(2m)^n$ for each $n \in \{1, 2, 3, \dots, 2k\}$; and we find that

$$|\mathcal{T}_n|_a = |\mathcal{T}_n|_b + \Delta(\mathcal{T}_n) = 2(2m)^n - 2m^n$$

as desired. □

7.2. **Constants.** We now introduce some constants which are used in the proofs below.

Definition 7.8. For $n \in \{1, 2, 3, \dots, 2k\}$, we define

$$\Lambda_n = m^{2k+1-n} \qquad \text{and} \qquad B_n = \frac{m^{2k+2-n} - m}{m - 1}.$$

Moreover, for each $n \in \{1, 2, 3, \dots, 2k\}$, we define $\sigma_n = 2\Lambda_n + 2B_n$.

The constants are chosen so that they satisfy the following relations.

Lemma 7.9. *For each $n \in \{1, 2, 3, \dots, 2k\}$, we have $12\Lambda_n > 6B_n \geqslant 6\Lambda_n > \sigma_n > 0$.*

Proof. We observe that $\Lambda_n, B_n > 0$ for each $n \in \{1, 2, 3, \dots, 2k\}$, and that

$$B_n = \frac{m^{2k+2-n} - m}{m-1} = m + m^2 + \cdots + m^{2k+1-n} \qquad \text{and} \qquad \Lambda_n = m^{2k+1-n}.$$

We thus see that $B_n \geqslant \Lambda_n$. Moreover, since $m \geqslant 2$, we see that $2\Lambda_n > B_n$.

Then, since $\sigma_n = 2\Lambda_n + 2B_n$, we see that

$$\sigma_n < 2\Lambda_n + 4\Lambda_n = 6\Lambda_n.$$

Thus, we have our desired inequalities. □

7.3. **Functions on words and word weights.** In this section, we require a function to extract the a^* factors which appear as part of the prefix and suffix of a given word in Σ^*. To accomplish this, we introduce the following function.

Definition 7.10. We define $\mathrm{Affix}\colon \Sigma^* \times \{\triangleleft, \triangleright\} \to \Sigma^*$ such that for each word $w \in \Sigma^*$, the word $\mathrm{Affix}(w, \triangleleft)$ is the longest prefix of w of the form a^*; and $\mathrm{Affix}(w, \triangleright)$ is the longest suffix of w of the form a^*.

For example, if $w = aaababbabbba$ then $\mathrm{Affix}(w, \triangleleft) = aaa$ and $\mathrm{Affix}(w, \triangleright) = a$. In the next lemma we compute Affix for some words of particular interest.

Lemma 7.11. *For each $n \in \{1, 2, 3, \dots, 2k\}$ and each $s \in \{\triangleleft, \triangleright\}$, we have*

$$|\mathrm{Affix}(\mathcal{T}_n, s)| = m + m^2 + \cdots + m^n = \frac{m^{n+1} - m}{m-1} = B_{2k+1-n}$$

where $\mathcal{T}_n$ is as in Definition 7.5.

Proof. Let $s \in \{\triangleleft, \triangleright\}$, then from Definition 7.5, we see that

$$|\mathrm{Affix}(\mathcal{T}_0, s)| = 0 \quad \text{and} \quad |\mathrm{Affix}(\mathcal{T}_n, s)| = m^n + \mathrm{Affix}(\mathcal{T}_{n-1}, s)$$

for each $n \geqslant 1$. From this recurrence, it follows immediately that

$$|\mathrm{Affix}(\mathcal{T}_n, s)| = m + m^2 + \cdots + m^n = \frac{m^{n+1} - m}{m-1}$$

for each $n \in \{1, 2, 3, \dots, 2k\}$. □

From this, we then have the following result on the affixes of $\mathcal{W}$.

Corollary 7.12. *We have $|\mathrm{Affix}(\mathcal{W}, s)| = B_1 + m^{2k}$ for each $s \in \{\triangleleft, \triangleright\}$.*

Proof. From the definition of $\mathcal{W}$ in Definition 7.5, we see that

$$|\mathrm{Affix}(\mathcal{W}, s)| = |\mathrm{Affix}(\mathcal{T}_{2k}, s)| + m^{2k}$$

for each $s \in \{\triangleleft, \triangleright\}$. Thus, from Lemma 7.11, we see that

$$|\mathrm{Affix}(\mathcal{W}, s)| = B_1 + m^{2k}$$

for each $s \in \{\triangleleft, \triangleright\}$ as desired. □

We now define what it means for a word to be *heavy* or *light* as follows.

Definition 7.13. Let $n \in \{1, 2, 3, \dots, 2k\}$, we say that a word $u \in \Sigma^*$ is *n-heavy* if it contains a factor of the form $a^{2\Lambda_n}$, otherwise we say that u is *n-light*. Given a tuple $\vec{w} = (w_1, w_2, \dots, w_k)$, we define the set of indices $L_n(\vec{w}) \subseteq \{1, 2, \dots, k\}$ such that $i \in L_n(\vec{w})$ if and only if w_i is an n-light word.

We then define the following function on n-heavy words.

Definition 7.14. Let $u \in \Sigma^*$ be an n-heavy word, we then define

- $\mathrm{seg}_n(u, \triangleleft) = x$ to be the shortest prefix of u such that $u = xa^{2\Lambda_n}u'$; and
- $\mathrm{seg}_n(u, \triangleright) = x$ to be the shortest suffix of u such that $u = u'a^{2\Lambda_n}x$.

In the above, it should be understood that $u' \in \Sigma^*$.

We now define the functions $\mathrm{Rem}_n\colon \Sigma^* \to \Sigma^*$ as follows.

Definition 7.15. For $n \in \{1, 2, 3, \dots, 2k\}$, we define $\mathrm{Rem}_n\colon \Sigma^* \to \Sigma^*$ such that

- if $u \in \Sigma^*$ is n-light, then $\mathrm{Rem}_n(u) = u$; otherwise
- if $u \in \Sigma^*$ is n-heavy, then $\mathrm{Rem}_n(u) = \mathrm{seg}_n(u, \triangleleft)\,\mathrm{seg}(u, \triangleright)$.

Notice that in the second case, the word u is n-heavy, and thus it is well-defined.

We now define an additional function as follows.

Definition 7.16. We define the function $\mathrm{Strip}\colon \Sigma^* \to \Sigma^*$ such that

$$\mathrm{Strip}(u) \coloneqq \begin{cases} \varepsilon & \text{if } u = \mathrm{Affix}(u, \triangleleft) = \mathrm{Affix}(u, \triangleright), \\ u' & \text{where } u = \mathrm{Affix}(u, \triangleleft)\, u'\, \mathrm{Affix}(u, \triangleright) \text{ otherwise.} \end{cases}$$

That is, $\mathrm{Strip}(u)$ is the word obtained by removing all leading and trailing instances of a from u.

For example, $\mathrm{Strip}(aaababbabbba) = babbabbb$. We observe the following.

Lemma 7.17. *For each $n \in \{1, 2, 3, \ldots, 2k\}$, we have*

$$\Delta(\mathrm{Strip}(\mathcal{T}_n)) = -2m^n - 2\frac{m^{n+1} - m}{m - 1}.$$

Moreover, for $n \in \{1, 2, 3, \ldots, 2k\}$, let w_n be the word

$$w_n = \mathrm{Affix}(\mathcal{T}_n, \triangleright)\, \mathcal{T}_n\, \mathrm{Affix}(\mathcal{T}_n, \triangleleft),$$

then, for each factor u of w_n, we have $|\Delta(u)| \leqslant |\Delta(\mathrm{Strip}(\mathcal{T}_n))|$.

Proof. From the recursive formula given in Definition 7.5, we see that

$$\Delta(\mathrm{Strip}(\mathcal{T}_n)) = 2m\Delta(\mathcal{T}_{n-1}) - \Delta(\mathrm{Affix}(\mathcal{T}_{n-1}, \triangleleft)) - \Delta(\mathrm{Affix}(\mathcal{T}_{n-1}, \triangleright))$$

for each $n \geqslant 1$. Thus, from Lemmas 7.7 and 7.11, we see that

$$\begin{aligned} \Delta(\mathrm{Strip}(\mathcal{T}_n)) &= 2m(-2m^{n-1}) - 2\left(\frac{m^n - m}{m - 1}\right) \\ &= -4m^n - 2\frac{m^n - m}{m - 1} \\ &= -2m^n - 2\frac{m^{n+1} - m}{m - 1}, \end{aligned}$$

as desired.

Considering the factors u of $w_1 = a^{2m}b^{4m}a^{2m} = \mathrm{Affix}(\mathcal{T}_1, \triangleright)\, \mathcal{T}_1\, \mathrm{Affix}(\mathcal{T}_1, \triangleleft)$, we see that

$$-4m = \Delta(b^{4m}) \leqslant \Delta(u) \leqslant \Delta(a^{2m}) = 2m.$$

Thus, $|\Delta(u)| \leqslant 4m = |\Delta(\mathrm{Strip}(\mathcal{T}_1))|$.

Suppose, for induction, that $|\Delta(v)| \leqslant |\Delta(\mathrm{Strip}(\mathcal{T}_{n-1}))|$ for every factor v of the word

$$w_{n-1} = \mathrm{Affix}(\mathcal{T}_{n-1}, \triangleright)\, \mathcal{T}_{n-1}\, \mathrm{Affix}(\mathcal{T}_{n-1}, \triangleleft)$$

for some value $n \in \{2, 3, \ldots, 2k\}$. Let u be a factor of

$$w_n = \mathrm{Affix}(\mathcal{T}_n, \triangleright)\, \mathcal{T}_n\, \mathrm{Affix}(\mathcal{T}_n, \triangleleft).$$

Notice from Lemmas 7.7 and 7.11 that

$$\begin{aligned} \Delta(w_n) &= \Delta(\mathcal{T}_n) + \Delta(\mathrm{Affix}(\mathcal{T}_n, \triangleleft)) + \Delta(\mathrm{Affix}(\mathcal{T}_n, \triangleright)) \\ &= -2m^n + 2\left(\frac{m^{n+1} - m}{m - 1}\right) \\ &= 2\frac{m^n - m}{m - 1}. \end{aligned}$$

From Definition 7.5 and Lemma 7.11, we see

$$w_n = a^{B_{2k+1-n}}\, \mathcal{T}_n\, a^{B_{2k+1-n}} = a^{m^n + B_{2k+1-n}}\, (\mathcal{T}_{n-1})^{2m}\, a^{m^n + B_{2k+1-n}} \tag{7}$$

and thus, the factor u falls into one of the following 4 cases:

Case 1: The factor u has the form

$$u = a^i (\mathcal{T}_{n-1})^{2m} a^j$$

where $i, j \in \{0, 1, 2, \ldots, m^n + B_{2k+1-n}\}$. In this case, we see that

$$-4m^n = 2m\Delta(\mathcal{T}_{n-1}) = \Delta((\mathcal{T}_{n-1})^{2m}) \leqslant \Delta(u) \leqslant \Delta(w_n) = 2\frac{m^n - m}{m - 1} \leqslant 2m^n.$$

From this, we then see that $|\Delta(u)| \leqslant 4m^n \leqslant |\Delta(\mathrm{Strip}(\mathcal{T}_n))|$ as desired.

Case 2: The factor u has the form

$$u = a^i (\mathcal{T}_{n-1})^j v$$

where $i \in \{0, 1, 2, \ldots, m^n + B_{2k+1-n}\}$, $j \in \{0, 1, 2, \ldots, 2m-1\}$ and v is a factor of $\mathcal{T}_{n-1}$.

From our inductive hypothesis we see that

$$(2m-1)\Delta(\mathcal{T}_{n-1}) - |\Delta(\text{Strip}(\mathcal{T}_{n-1}))| \leqslant \Delta(u) \leqslant (m^n + B_{2k+1-n}) + |\Delta(\text{Strip}(\mathcal{T}_{n-1}))|.$$

From Lemmas 7.7 and 7.11, we see that

$$\begin{aligned}
\Delta(u) &\geqslant (2m-1)\Delta(\mathcal{T}_{n-1}) - |\text{Strip}(\mathcal{T}_{n-1})| \\
&= -(2m-1)2m^{n-1} - 2m^{n-1} - 2\frac{m^n - m}{m-1} \\
&= -4m^n - 2\frac{m^n - m}{m-1} \\
&= -2m^n - 2\frac{m^{n+1} - m}{m-1} = -|\Delta(\text{Strip}(\mathcal{T}_n))|
\end{aligned}$$

and

$$\begin{aligned}
\Delta(u) &\leqslant (m^n + B_{2k+1-n}) + |\Delta(\text{Strip}(\mathcal{T}_{n-1}))| \\
&= \left(m^n + \frac{m^{n+1} - m}{m-1}\right) + \left(2m^{n-1} + 2\frac{m^n - m}{m-1}\right) \\
&= \left(2m^n + \frac{m^n - m}{m-1}\right) + \left(2m^{n-1} + 2\frac{m^n - m}{m-1}\right) \\
&\leqslant \left(2m^n + 2m^{n-1}\right) + \left(2m^{n-1} + 2\frac{m^n - m}{m-1}\right) \\
&\leqslant 2m^n + 4m^{n-1} + 2\frac{m^n - m}{m-1} \\
&\leqslant 2m^n + 2m^n + 2\frac{m^n - m}{m-1} = 2m^n + 2\frac{m^{n+1} - m}{m-1} = |\Delta(\text{Strip}(\mathcal{T}_n))|
\end{aligned}$$

where some of the above inequalities hold since $m \geqslant 2$.

From this, we then see that $|\Delta(u)| \leqslant |\Delta(\text{Strip}(\mathcal{T}_n))|$ as desired.

Case 3: The factor u has the form

$$u = v(\mathcal{T}_{n-1})^j a^i$$

where $i \in \{0, 1, 2, \ldots, m^n + B_{2k+1-n}\}$, $j \in \{0, 1, 2, \ldots, 2m-1\}$ and v is a factor of $\mathcal{T}_{n-1}$.

This case is symmetric to the previous case. In particular, we see that $|\Delta(u)| \leqslant |\Delta(\text{Strip}(\mathcal{T}_n))|$ as desired.

Case 4: The factor u has the form

$$u = \alpha(\mathcal{T}_{n-1})^i \beta$$

where $i \in \{0, 1, 2, \ldots, 2m-2\}$, and α and β are factors of $\mathcal{T}_{n-1}$.

From our inductive hypothesis, we then see that

$$-2|\Delta(\text{Strip}(\mathcal{T}_{n-1}))| + (2m-2)\Delta(\mathcal{T}_n) \leqslant \Delta(u) \leqslant 2|\Delta(\text{Strip}(\mathcal{T}_{n-1}))|.$$

Thus, from Lemma 7.7 and the fact that $m \geqslant 2$, we then see that

$$\begin{aligned}
|\Delta(u)| &\leqslant 2\left(2m^{n-1} + 2\frac{m^n - m}{m-1}\right) + (2m-2)m^{n-1} \\
&= 2m^n + 2m^{n-1} + 4\frac{m^n - m}{m-1} \\
&\leqslant 2m^n + 2\frac{m^{n+1} - m}{m-1} = |\Delta(\text{Strip}(\mathcal{T}_n))|
\end{aligned}$$

where the last inequality holds since $m > 3$, in particular,

$$\begin{aligned}
2m^n + 2m^{n-1} + 4\frac{m^n - m}{m-1} \leqslant 2m^n + 2\frac{m^{n+1} - m}{m-1} &\Longleftarrow m^{n-1} + 2\frac{m^n - m}{m-1} \leqslant \frac{m^{n+1} - m}{m-1} \\
&\Longleftarrow m^{n-1}(m-1) + 2(m^n - m) \leqslant m^{n+1} - m \\
&\Longleftarrow m^{n+1} - 3m^n + m^{n-1} + m \geqslant 0 \\
&\Longleftarrow m^{n+1} - 3m^n \geqslant 0 \\
&\Longleftarrow m^n(m-3) \geqslant 0 \Longleftarrow m \geqslant 3.
\end{aligned}$$

Thus, $|\Delta(u)| \leqslant |\Delta(\mathrm{Strip}(\mathcal{T}_n))|$ as desired.
Conclusion:
In all cases, we have our desired bound on $|\Delta(u)|$. □

We then have the following lemma which we use to characterise the n-light factors of $\mathcal{W}$.

Lemma 7.18. *If u is a factor of $\mathcal{W}$ that does not contain a^{2B_n} as a factor, with $n \in \{1, 2, 3, \ldots, 2k\}$, then $|\Delta(u)| \leqslant \sigma_n$.*

Proof. For each $n \in \{1, 2, 3, \ldots, 2k\}$, notice from the definition of $\mathcal{W}$ that

$$\mathcal{W} \in a^*\mathcal{T}_{2k+1-n}a^*\mathcal{T}_{2k+1-n}a^*\mathcal{T}_{2k+1-n}a^* \cdots a^*\mathcal{T}_{2k+1-n}a^*.$$

From Lemma 7.11, we see that

$$\mathrm{Affix}(\mathcal{T}_{2k+1-n}, \triangleleft) = \mathrm{Affix}(\mathcal{T}_{2k+1-n}, \triangleright) = a^{B_n}.$$

From this we see that if u does not contain a^{2B_n} as a factor, then it must also be a factor of

$$w = \mathrm{Affix}(\mathcal{T}_{2k+1-n}, \triangleright)\, \mathcal{T}_{2k+1-n}\, \mathrm{Affix}(\mathcal{T}_{2k+1-n}, \triangleleft).$$

Thus, from Lemma 7.17, we see that

$$|\Delta(u)| \leqslant 2m^{2k+1-n} + 2\frac{m^{2k+2-n} - m}{m-1} = \sigma_n,$$

as desired. □

Corollary 7.19. *If u is a factor of $\mathcal{W}$, then*

$$|\Delta(\mathrm{Rem}_n(u))| \leqslant 2\sigma_n$$

for all $n \in \{1, 2, 3, \ldots, 2k\}$. If additionally u is n-light, then $|\Delta(\mathrm{Rem}_n(u))| \leqslant \sigma_n$.

Proof. From the definition of $\mathrm{Rem}_n \colon \Sigma^* \to \Sigma^*$, we have two cases as follows.

(1) The word u is n-light and thus $\mathrm{Rem}_n(u) = u$ does not contain $a^{2\Lambda_n}$ as a factor. From Lemmas 7.9 and 7.18, we see that $|\Delta(\mathrm{Rem}_n(u))| \leqslant \sigma_n$.
(2) The word u is n-heavy and thus $\mathrm{Rem}_n(u) = \mathrm{seg}_n(u, \triangleleft)\, \mathrm{seg}_n(u, \triangleright)$ where both $\mathrm{seg}_n(u, \triangleleft)$ and $\mathrm{seg}_n(u, \triangleright)$ are n-light and thus do not contain $a^{2\Lambda_n}$ as a factor. From Lemmas 7.9 and 7.18, we then see that $|\Delta(\mathrm{Rem}_n(u))| \leqslant 2\sigma_n$.

We thus have our desired bounds. □

7.4. **Decompositions.** In the following we define what it means for a k-tuple of words in Σ^* to have a decomposition. Later in this proof, we show that the property of having a decomposition is preserved, in some way, by the grammar M for L.

Definition 7.20. Let $n \in \{2, 3, \ldots, 2k\}$, then an *$n$-decomposition* of $\vec{w} = (w_1, w_2, \ldots, w_k) \in (\Sigma^*)^k$ is a nonempty subset

$$E \subseteq \{1, 2, \ldots, k\} \times \{\triangleleft, \triangleright\}$$

with $|E| = n$ such that, for each $(i, s) \in E$, the word w_i is n-heavy and

$$|\mathrm{Affix}(w_i, s)| \geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)). \tag{8}$$

Notice that $|L_n(\vec{w})|$ is the number of n-light components in $\vec{w}$. We say that $\vec{w}$ *has a decomposition* if it has an n-decomposition for some $n \in \{2, 3, \ldots, 2k\}$.

Remark 7.21. Notice that if $\vec{w} = (\mathcal{W}, \varepsilon, \varepsilon, \ldots, \varepsilon)$, then $E = \{(1, \triangleleft), (1, \triangleright)\}$ is a 2-decomposition of $\vec{w}$. In particular, we see that $w_1 = \mathcal{W}$ is 2-heavy as it contains $a^{B_1 + m^{2k}}$ as a prefix (see Corollary 7.12) where $B_1 + m^{2k} \geqslant 2\Lambda_1$ follows since we have $2\Lambda_1 = 2m^{2k}$ and $B_1 \geqslant m^{2k}$ from Definition 7.8; and moreover, for each $(i, s) \in E$, we see that the left-hand side of (8) evaluates to $B_1 + m^{2k}$ (from Corollary 7.12) while the right-hand side of (8) evaluates to

$$\Lambda_1 - C - (2k - 2 - (k-1))\sigma_2 - 0 = \Lambda_1 - C - (k-1)\sigma_2 = \Lambda_1 - C - (k-1)(2\Lambda_2 + 2B_2)$$

where the last equality follows from the definition of σ_n in Definition 7.8, thus (8) is equivalent to

$$B_1 + m^{2k} \geqslant \Lambda_1 - C - (k-1)(2\Lambda_2 + 2B_2)$$

which follows immediately as we have $B_1 > \Lambda_1$ from Lemma 7.9.

To simplify the proofs of later lemmas in this section, we often assume without loss of generality that a given n-decomposition is maximal as follows.

Definition 7.22. We say that an n-decomposition E of $\vec{w} = (w_1, w_2, \dots, w_k)$ is *maximal* if there is no n'-decomposition E' of $\vec{w}$ with $|E'| = n' > n = |E|$.

To simplify notation, we define a set of tuples which we call $\mathcal{W}$-*sentential tuples* as follows.

Definition 7.23. A $\mathcal{W}$-*sentential tuple* is a k-tuple of the form $\vec{w} = (w_1, w_2, \dots, w_k) \in (\Sigma^*)^k$ where

$$\mathcal{W} = u_0 w_1 u_1 w_2 u_2 \cdots w_k u_k$$

for some words $u_0, u_1, u_2, \dots, u_k \in \Sigma^*$, and $|\Delta(\vec{w})| \leqslant C$. Thus, each word vector which appears in the derivation of $\mathcal{W}$, in the grammar M, is an example of a $\mathcal{W}$-sentential tuple.

We have the following property of n-decompositions.

Lemma 7.24. *Let $\vec{w} = (w_1, w_2, \dots, w_k)$ be a $\mathcal{W}$-sentential tuple, and let E be an n-decomposition of $\vec{w}$. Then, $|\mathrm{Affix}(w_i, s)| \geqslant 5\Lambda_n$ for every $(i, s) \in E$.*

Proof. Since $\vec{w} = (w_1, w_2, \dots, w_k)$ is a $\mathcal{W}$-sentential tuple, we see that

$$|\mathrm{Affix}(w_i, s)| \geqslant \Lambda_{n-1} - C - 2k\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)).$$

From Corollary 7.19, we then see that

$$|\mathrm{Affix}(w_i, s)| \geqslant \Lambda_{n-1} - C - 4k\sigma_n.$$

From the definition of the constant Λ_n, we see that

$$\begin{aligned} |\mathrm{Affix}(w_i, s)| \geqslant \Lambda_{n-1} - C - 4k\sigma_n &= m\Lambda_n - C - 4k\sigma_n \\ &= 5\Lambda_n + (m-5)\Lambda_n - C - 4k\sigma_n. \end{aligned}$$

From Lemma 7.9, we then see that

$$|\mathrm{Affix}(w_i, s)| \geqslant 5\Lambda_n + (m-5)\Lambda_n - C - 24k\Lambda_n.$$

Thus,

$$|\mathrm{Affix}(w_i, s)| \geqslant 5\Lambda_n + (m - 5 - C - 24k)\Lambda_n.$$

From the value of m given in Definition 7.5, we see that $|\mathrm{Affix}(w_i, s)| \geqslant 5\Lambda_n$. □

We now have the following lemma which we use to construct decompositions.

Lemma 7.25. *Let $\vec{w} = (w_1, w_2, \dots, w_k)$ be a $\mathcal{W}$-sentential tuple, and suppose that*

$$E \subseteq \{1, 2, \dots, k\} \times \{\triangleleft, \triangleright\}$$

is a set of size $|E| = n + 1 \geqslant 2$ such that $|\mathrm{Affix}(w_i, s)| \geqslant \Lambda_n$ for each $(i, s) \in E$. Then, the set E is an $(n+1)$-decomposition of $\vec{w}$.

Proof. From the definition of m and Λ_n in Definitions 7.5 and 7.8, we see that $\Lambda_n \geqslant 2\Lambda_{n+1}$ and thus, for each $(i, s) \in E$, the word w_i is $(n+1)$-heavy. From Corollary 7.19, we see that

$$|\Delta(\mathrm{Rem}_{n+1}(w_i))| \leqslant \sigma_{n+1}$$

for each $(n+1)$-light word w_i. From the definition of $\mathrm{Rem}_n \colon \Sigma^* \to \Sigma^*$, if w_i is $(n+1)$-heavy, then

$$|\Delta(\mathrm{Rem}_{n+1}(w_i))| \leqslant |\Delta(\mathrm{seg}_{n+1}(w_i, \triangleleft))| + |\Delta(\mathrm{seg}_{n+1}(w_i, \triangleright))|.$$

Notice also that each $\mathrm{seg}_{n+1}(w_i, \triangleright))$, as above, is $(n+1)$-light and thus

$$|\Delta(\mathrm{seg}_{n+1}(w_i, \triangleleft))| \leqslant \sigma_{n+1}$$

from Corollary 7.19. Moreover, if $(i, s) \in E$, then $\mathrm{seg}_{n+1}(w_i, s) = \varepsilon$ and thus $|\Delta(\mathrm{seg}_{n+1}(w_i, s))| = 0$.

Let $H = \{1, 2, \dots, k\} \setminus L_{n+1}(\vec{w})$ be the indices of all the $(n+1)$-heavy words in $\vec{w}$. From the previous paragraph and the triangle inequality, we then see that

$$\left| \sum_{i \in H} \Delta(\mathrm{Rem}_{n+1}(w_i)) \right| \leqslant (2|H| - |E|)\sigma_{n+1}.$$

Notice that $2|H| - |E|$ counts the sides of the $(n+1)$-heavy words which are not in E.

From the above, we see that

$$\begin{aligned}\left|\sum_{i=1}^{k}\Delta(\mathrm{Rem}_{n+1}(w_i))\right| &\leqslant (|L_{n+1}(\vec{w})| + 2|H| - |E|)\sigma_{n+1}\\ &= (|L_{n+1}(\vec{w})| + 2(k - |L_{n+1}(\vec{w})|) - (n+1))\sigma_{n+1}.\\ &= (2k - (n+1) - |L_{n+1}(\vec{w})|)\sigma_{n+1}.\end{aligned}$$

From this bound, we then see that

$$(2k - (n+1) - |L_{n+1}(\vec{w})|)\sigma_{n+1} + \sum_{j=1}^{k}\Delta(\mathrm{Rem}_{n+1}(w_j)) \geqslant 0.$$

Thus, we see that

$$\Lambda_n - (C - \Delta(\vec{w})) - (2k - (n+1) - |L_{n+1}(\vec{w})|)\sigma_{n+1} - \sum_{j=1}^{k}\Delta(\mathrm{Rem}_{n+1}(w_j)) \leqslant \Lambda_n - (C - \Delta(\vec{w})).$$

Since $\vec{w}$ is a $\mathcal{W}$-sentential tuple, by definition, we have $|\Delta(\vec{w})| \leqslant C$, and thus

$$\Lambda_n - (C - \Delta(\vec{w})) - (2k - (n+1) - |L_{n+1}(\vec{w})|)\sigma_{n+1} - \sum_{j=1}^{k}\Delta(\mathrm{Rem}_{n+1}(w_j)) \leqslant \Lambda_n.$$

From the assumptions in our lemma statement, we then see that

$$|\mathrm{Affix}_{n+1}(w_i, s)| \geqslant \Lambda_n \geqslant \Lambda_n - (C - \Delta(\vec{w})) - (2k - (n+1) - |L_{n+1}(\vec{w})|)\sigma_{n+1} - \sum_{j=1}^{k}\Delta(\mathrm{Rem}_{n+1}(w_j))$$

for each $(i, s) \in E$. Hence, E is an $(n+1)$-decomposition as required. □

We have the following property of maximal decompositions.

Lemma 7.26. *Let E be a maximal n-decomposition of a $\mathcal{W}$-sentential tuple $\vec{w}$. Then, for each n-heavy component $i \notin L_n(\vec{w})$ and each $s \in \{\triangleleft, \triangleright\}$ we have $\mathrm{seg}_n(w_i, s) = \varepsilon$ if and only if $(i, s) \in E$.*

Proof. Let E be a maximal n-decomposition as in the lemma statement.

If $(i, s) \in E$, then from Lemma 7.24 we see that $|\mathrm{Affix}(w_i, s)| \geqslant 5\Lambda_n$ and thus $\mathrm{seg}_n(w_i, s) = \varepsilon$. Thus, if $(i, s) \in E$, then $\mathrm{seg}_n(w_i, s) = \varepsilon$. All that remains is to show the converse of this statement as follows.

Suppose for contradiction that w_i is an n-heavy word with $\mathrm{seg}_n(w_i, s) = \varepsilon$ and $(i, s) \notin E$. Then, from the definition of $\mathrm{seg}_n \colon \Sigma^* \times \{\triangleleft, \triangleright\} \to \Sigma^*$ in Definition 7.14, we see that $|\mathrm{Affix}(w_i, s)| \geqslant 2\Lambda_n$. Thus, from Lemmas 7.24 and 7.25, we see that $E' = E \cup \{(i, s)\}$ is an $(n+1)$-decomposition of $\vec{w}$ which contradicts the maximality of E. Thus, if $\mathrm{seg}_n(w_i, s) = \varepsilon$, then $(i, s) \in E$. □

We have the following lemma which is used in the proof of Proposition 7.30.

Lemma 7.27. *Let $w \in \Sigma^*$ be an n-heavy factor of $\mathcal{W}$, and suppose that $w = uv$ with $u, v \in \Sigma^*$. Then, at least one of the following five conditions holds.*

(1) $|\mathrm{Affix}(u, \triangleright)| \geqslant \Lambda_n$;
(2) $|\mathrm{Affix}(v, \triangleleft)| \geqslant \Lambda_n$;
(3) u and v are both n-heavy, and $\Delta(\mathrm{Rem}_n(u)) + \Delta(\mathrm{Rem}_n(v)) \geqslant \Delta(\mathrm{Rem}_n(w)) - \sigma_n$;
(4) u is n-light, v is n-heavy, and $\Delta(\mathrm{Rem}_n(u)) + \Delta(\mathrm{Rem}_n(v)) = \Delta(\mathrm{Rem}_n(w))$;
(5) u is n-heavy, v is n-light, and $\Delta(\mathrm{Rem}_n(u)) + \Delta(\mathrm{Rem}_n(v)) = \Delta(\mathrm{Rem}_n(w))$.

Proof. Suppose that w is an n-heavy factor of $\mathcal{W}$ with $w = uv$ as in the lemma statement. We separate our proof into two parts as follows.

Case 1: both u and v are n-light.

Here we see that u and v must be of the form

$$u = \mathrm{seg}_n(w, \triangleleft)a^p \qquad \text{and} \qquad v = a^q\mathrm{seg}(w, \triangleright),$$

respectively, where $p + q \geqslant 2\Lambda_n$. Thus, we find that either $p \geqslant \Lambda_n$ or $q \geqslant \Lambda_n$, and thus we are either in case (1) or (2), respectively.

Case 2: at least one of u or v is n-heavy.

Notice that if

$$|\mathrm{Affix}(u, \triangleright)| \geqslant \Lambda_n \qquad \text{or} \qquad |\mathrm{Affix}(v, \triangleleft)| \geqslant \Lambda_n,$$

then we are in case (1) or (2), respectively. Thus, in the remainder of this proof, we assume that both

$$|\mathrm{Affix}(u, \triangleright)| < \Lambda_n \qquad \text{and} \qquad |\mathrm{Affix}(v, \triangleleft)| < \Lambda_n,$$

and thus the factor

$$w' = \mathrm{seg}_n(u, \triangleright)\mathrm{seg}_n(v, \triangleleft)$$

of w does not contain $a^{2\Lambda_n}$ as a factor, that is, w' is n-light.

Case 2.1: both u and v are n-heavy.

From Lemmas 7.9 and 7.18, we then see that

$$\Delta(\mathrm{seg}_n(u, \triangleright)) + \Delta(\mathrm{seg}_n(u, \triangleleft)) = \Delta(w') \geqslant -\sigma_n.$$

From the definition of $\mathrm{Rem}_n \colon \Sigma^* \to \Sigma^*$, we see that

$$\begin{aligned}\Delta(\mathrm{Rem}_n(u)) + \Delta(\mathrm{Rem}_n(v)) &= \Delta(\mathrm{seg}_n(u, \triangleleft)) + \Delta(\mathrm{seg}_n(v, \triangleright)) + \Delta(\mathrm{seg}_n(u, \triangleright)) + \Delta(\mathrm{seg}_n(v, \triangleleft)) \\ &= \Delta(\mathrm{seg}_n(w, \triangleleft)) + \Delta(\mathrm{seg}_n(w, \triangleright)) + \Delta(\mathrm{seg}_n(u, \triangleright)) + \Delta(\mathrm{seg}_n(v, \triangleleft)) \\ &= \Delta(\mathrm{Rem}_n(w)) + \Delta(w') \geqslant \Delta(\mathrm{Rem}_n(w)) - \sigma_n.\end{aligned}$$

Thus, we are in case (3) of the lemma.

Case 2.2: u is n-light, and v is n-heavy.

The only way for u to be n-light is if it is either a factor of $\mathrm{seg}_n(w, \triangleleft)$ or a word of the form $\mathrm{seg}_n(w, \triangleleft)a^p$ where $p < 2\Lambda_n$. We consider these cases as follows.

- If u is a factor of $\mathrm{seg}_n(w, \triangleleft)$, then we see that

$$\begin{aligned}\mathrm{Rem}_n(w) &= \mathrm{seg}_n(w, \triangleleft)\,\mathrm{seg}_n(w, \triangleright) \\ &= u\,\mathrm{seg}_n(v, \triangleleft)\,\mathrm{seg}_n(v, \triangleright) = \mathrm{Rem}_n(u)\,\mathrm{Rem}_n(v).\end{aligned}$$

  Thus,

$$\Delta(\mathrm{Rem}_n(u)) + \Delta(\mathrm{Rem}_n(v)) = \Delta(\mathrm{Rem}_n(w))$$

  and we are in case (4) of the lemma.
- If $u = \mathrm{seg}_n(w, \triangleleft)a^p$ where $p < 2\Lambda_n$, then $v = a^q\, v'$ with $p + q \geqslant 2\Lambda_n$. So either $p \geqslant \Lambda_n$ or $q \geqslant \Lambda_n$, which correspond to cases (1) and (2) of the lemma.

Case 2.3: u is n-heavy, and v is n-light.

The proof of this case is symmetric to Case 2.2.

Conclusion: We see that in all cases, our words fall into one of the five cases listed in the lemma. Moreover, we see that these cases cover all possible situations. □

7.5. **Maintaining a decomposition.** In this subsection we show that, for $\mathcal{W}$-sentential tuples, the property of having a decomposition is maintained by the operations known as left letter deletion and left split (both of which are the reverse of some of the operations presented in Definition 6.7).

Proposition 7.28. *Let $\vec{w} = (w_1, w_2, \ldots, w_k)$ and $\vec{v} = (v_1, v_2, \ldots, v_k)$ be $\mathcal{W}$-sentential tuples as in Definition 7.23, and suppose that $x \in \{1, 2, \ldots, k\}$ with*

$$(w_1, w_2, \ldots, w_k) = (v_1, v_2, \ldots, v_{x-1}, av_x, v_{x+1}, \ldots, v_k).$$

If $\vec{w}$ has an n-decomposition, then $\vec{v}$ has an n'-decomposition where $n' \geqslant n$. That is, given a $\mathcal{W}$-sentential tuple with an n-decomposition, if we still have a $\mathcal{W}$-sentential tuple after deleting an instance of the letter a from the left-hand side of some component, then there exists an n'-decomposition, with $n' \geqslant n$, for the new tuple. (Recall that if $\vec{w}$ is a $\mathcal{W}$-sentential tuple, then $|\Delta(\vec{w})| \leqslant C$. Thus, we cannot directly use this process to remove sequences of a's of arbitrarily large length and still obtain a decomposition.)

Proof. Without loss of generality, we assume that E is a maximal n-decomposition for the sentential tuple $\vec{w}$. We note here that $\Delta(\vec{v}) = \Delta(\vec{w}) - 1$, that $\mathrm{Rem}_n(v_i) = \mathrm{Rem}_n(w_i)$ for each $i \neq x$; and that $\mathrm{Affix}(v_i, s) = \mathrm{Affix}(w_i, s)$ for each $i \neq x$ and $s \in \{\triangleleft, \triangleright\}$. The remainder of this proof is separated into three cases as follows.

Case 1: $(x, \triangleleft) \in E$.

From Lemma 7.24, we see that

$$|\mathrm{Affix}(v_x, \triangleleft)| = |\mathrm{Affix}(w_x, \triangleleft)| - 1 \geqslant 5\Lambda_n - 1 \geqslant 2\Lambda_n.$$

Further, we see that

$$|\mathrm{Affix}(v_x, \triangleright)| \in \{|\mathrm{Affix}(w_x, \triangleright)|,\ |\mathrm{Affix}(w_x, \triangleright)| - 1\},$$

where the $|\mathrm{Affix}(w_x, \triangleright)| - 1$ corresponds to the case where $w_x = a^{|w_x|}$.

Thus,

$$|\mathrm{Affix}(v_i, s)| \geqslant |\mathrm{Affix}(w_i, s)| - 1$$

for each $i \in \{1, 2, \ldots, k\}$ and $s \in \{\triangleleft, \triangleright\}$.

We see that the words v_x and w_x are both n-heavy, and we see that

$$L_n(\vec{v}) = L_n(\vec{w}) \qquad \text{and} \qquad \mathrm{Rem}_n(v_x) = \mathrm{Rem}_n(w_x).$$

We are now ready to prove this case as follows.

From the above observations, we see that

$$\begin{aligned}
|\mathrm{Affix}(v_i, s)| &\geqslant |\mathrm{Affix}(w_i, s)| - 1 \\
&\geqslant \Lambda_{n-1} - (C - (\Delta(\vec{w}) - 1)) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\
&= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j))
\end{aligned}$$

for each $(i, s) \in E$. Hence, we find that E is an n-decomposition for $\vec{v}$.

Case 2: $(x, \triangleleft) \notin E$ and w_x is n-heavy.

From Lemma 7.26, we see that $\mathrm{seg}_n(w_x, \triangleleft) \neq \varepsilon$ and thus removing an a from the left-hand side of w_x does not affect its status as being n-heavy, and furthermore does not affect $\mathrm{Affix}(w_x, \triangleright)$. That is, we have

$$\mathrm{Affix}(v_x, \triangleright) = \mathrm{Affix}(w_x, \triangleright).$$

Moreover, from the definition of $\mathrm{Rem}_n \colon \Sigma^* \to \Sigma^*$ on n-heavy words, we see that

$$\Delta(\mathrm{Rem}_n(v_x)) = \Delta(\mathrm{Rem}_n(w_x)) - 1.$$

From the above observations, we see that

$$\begin{aligned}
|\mathrm{Affix}(v_i, s)| &\geqslant |\mathrm{Affix}(w_i, s)| \\
&\geqslant \Lambda_{n-1} - (C - (\Delta(\vec{w}) - 1)) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{x-1} \Delta(\mathrm{Rem}_n(w_j)) \\
&\qquad - (\Delta(\mathrm{Rem}_n(w_x)) - 1) - \sum_{j=x+1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\
&= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j))
\end{aligned}$$

for each $(i, s) \in E$. Hence, we find that E is an n-decomposition for $\vec{v}$.

Case 3: w_x is n-light.

Notice here that $(x, \triangleright) \notin E$ since w_x and thus v_x are n-light.

We then see that $a\,\mathrm{Rem}_n(v_x) = \mathrm{Rem}_n(w_x)$, and hence,

$$\Delta(\mathrm{Rem}_n(v_x)) = \Delta(\mathrm{Rem}_n(w_x)) - 1.$$

From the above observations, we see that

$$\begin{aligned}|\mathrm{Affix}(v_i,s)| &\geqslant |\mathrm{Affix}(w_i,s)|\\ &\geqslant \Lambda_{n-1}-(C-(\Delta(\vec{w})-1))-(2k-n-|L_n(\vec{w})|)\sigma_n-\sum_{j=1}^{x-1}\Delta(\mathrm{Rem}_n(w_j))\\ &\qquad -(\Delta(\mathrm{Rem}_n(w_x))-1)-\sum_{j=x+1}^{k}\Delta(\mathrm{Rem}_n(w_j))\\ &= \Lambda_{n-1}-(C-\Delta(\vec{v}))-(2k-n-|L_n(\vec{v})|)\sigma_n-\sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$

for each $(i,s)\in E$. Hence, we find that E is an n-decomposition for $\vec{v}$.

Conclusion:
We see that in all cases E is an n-decomposition for $\vec{v}$. □

Proposition 7.29. *Let $\vec{w}=(w_1,w_2,\ldots,w_k)$ and $\vec{v}=(v_1,v_2,\ldots,v_k)$ be $\mathcal{W}$-sentential tuples as in Definition 7.23, and suppose that $x\in\{1,2,\ldots,k\}$ with*

$$(w_1,w_2,\ldots,w_k)=(v_1,v_2,\ldots,v_{x-1},bv_x,v_{x+1},\ldots,v_k).$$

If $\vec{w}$ has an n-decomposition, then $\vec{v}$ has an n-decomposition. That is, given a $\mathcal{W}$-sentential tuple with an n-decomposition, if we still have a $\mathcal{W}$-sentential tuple after deleting an instance of the letter b from the left-hand side of some component, then there exists an n-decomposition for the new tuple.

Proof. Let E be an n-decomposition for $\vec{w}$. In this proof, we show that E is also an n-decomposition for $\vec{v}$. Notice that since the leftmost letter of w_x is b, we have $(x,\triangleleft)\notin E$. In addition, removing an occurrence of b from the left of w_x does not change its n-heavy/n-light status, and does not affect $\mathrm{Affix}(w_x,\triangleright)$. Thus,

$$L_n(\vec{v})=L_n(\vec{w})\qquad\text{and}\qquad \mathrm{Affix}(v_x,\triangleright)=\mathrm{Affix}(w_x,\triangleright).$$

Notice that $\Delta(\vec{v})=\Delta(\vec{w})+1$, that $\mathrm{Rem}_n(v_i)=\mathrm{Rem}_n(w_i)$ for each $i\neq x$, and $\mathrm{Affix}(v_i,s)=\mathrm{Affix}(w_i,s)$ for each $i\neq x$ and $s\in\{\triangleleft,\triangleright\}$. We also see that $b\,\mathrm{Rem}_n(v_x)=\mathrm{Rem}_n(w_x)$ and thus

$$\Delta(\mathrm{Rem}_n(v_x))=\Delta(\mathrm{Rem}_n(w_x))+1.$$

From our observations as above, we see that

$$\begin{aligned}|\mathrm{Affix}(v_i,s)| &\geqslant |\mathrm{Affix}(w_i,s)|\\ &\geqslant \Lambda_{n-1}-(C-(\Delta(\vec{w})+1))-(2k-n-|L_n(\vec{w})|)\sigma_n-\sum_{j=1}^{x-1}\Delta(\mathrm{Rem}_n(w_j))\\ &\qquad -(\Delta(\mathrm{Rem}_n(w_x))+1)-\sum_{j=x+1}^{k}\Delta(\mathrm{Rem}_n(w_j))\\ &= \Lambda_{n-1}-(C-\Delta(\vec{v}))-(2k-n-|L_n(\vec{v})|)\sigma_n-\sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$

for each $(i,s)\in E$. Hence, we see that E is an n-decomposition for $\vec{v}$. □

Proposition 7.30. *Let $\vec{w}=(w_1,w_2,\ldots,w_k)$ and $\vec{v}=(v_1,v_2,\ldots,v_k)$ be $\mathcal{W}$-sentential tuples as in Definition 7.23, and suppose that $x\in\{1,2,\ldots,k-1\}$ with $w_x=\varepsilon$, $w_{x+1}=v_xv_{x+1}$, and*

$$\vec{w}=(w_1,w_2,\ldots,w_k)=(v_1,v_2,\ldots,v_{x-1},\varepsilon,v_xv_{x+1},v_{x+2},\ldots,v_k).$$

If $\vec{w}$ has an n-decomposition, then $\vec{v}$ has an n'-decomposition where $n'\geqslant n$. That is, given a $\mathcal{W}$-sentential tuple $\vec{w}$ with an n-decomposition, if we split one of its components into two adjacent components to obtain another $\mathcal{W}$-sentential tuple, then the resulting tuple will also have an n'-decomposition with $n'\geqslant n$.

(Notice that if $\vec{w}$ and $\vec{v}$ are different vectors, then $\vec{v}$ will have one fewer component equal to ε then $\vec{w}$. Thus, this proposition cannot be directly repeated arbitrarily many times.)

Proof. Without loss of generality, we assume that E is a maximal n-decomposition for the sentential tuple $\vec{w}$. We note here that $\Delta(\vec{v}) = \Delta(\vec{w})$, and $v_i = w_i$ for each $i \notin \{x, x+1\}$.

The remainder of this proof is separated into cases based on whether the word w_{x+1} is n-heavy, and the memberships of its sides to the set E.

Case 1: The word w_{x+1} is n-light.

Since $w_{x+1} = v_x v_{x+1}$, we then see that both v_x and v_{x+1} are also n-light. Hence,

$$L_n(\vec{v}) = L_n(\vec{w}), \qquad \Delta(w_{x+1}) = \Delta(v_x) + \Delta(v_{x+1})$$

and thus

$$\begin{aligned}\Delta(\mathrm{Rem}_n(w_x)) + \Delta(\mathrm{Rem}_n(w_{x+1})) &= \Delta(\mathrm{Rem}_n(w_{x+1})) \\ &= \Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})).\end{aligned}$$

Since w_{x+1} is n-light and $w_x = \varepsilon$, it then follows that

$$(x, \triangleleft), (x, \triangleright), (x+1, \triangleleft), (x+1, \triangleright) \notin E.$$

Thus, we have $v_i = w_i$ if $(i, s) \in E$ for some $s \in \{\triangleleft, \triangleright\}$. Hence, we find that

$$\begin{aligned}|\mathrm{Affix}(v_i, s)| &= |\mathrm{Affix}(w_i, s)| \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$

for each $(i, s) \in E$. Thus, E is an n-decomposition for $\vec{v}$.

Case 2: The word w_{x+1} is n-heavy with $(x+1, \triangleleft), (x+1, \triangleright) \notin E$.

Notice that since $w_x = \varepsilon$, we have that

$$(x, \triangleleft), (x, \triangleright), (x+1, \triangleleft), (x+1, \triangleright) \notin E.$$

Thus, we have $v_i = w_i$ if $(i, s) \in E$ for some $s \in \{\triangleleft, \triangleright\}$.

From Lemma 7.27, we are in one of the following five cases.

(1) We have $|\mathrm{Affix}(v_x, \triangleright)| \geqslant \Lambda_n$.
From Lemmas 7.24 and 7.25, we see that $E' = E \cup \{(x, \triangleright)\}$ is an $(n+1)$-decomposition for $\vec{v}$.

(2) We have $|\mathrm{Affix}(v_{x+1}, \triangleleft)| \geqslant \Lambda_n$.
From Lemmas 7.24 and 7.25, we see that $E' = E \cup \{(x+1, \triangleleft)\}$ is an $(n+1)$-decomposition for $\vec{v}$.

(3) We have that v_x and v_{x+1} are both n-heavy, and

$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) \geqslant \Delta(\mathrm{Rem}_n(w_{x+1})) - \sigma_n. \tag{9}$$

It then follows that $L_n(\vec{v}) = L_n(\vec{w}) \setminus \{x\}$, in particular, $|L_n(\vec{v})| = |L_n(\vec{w})| - 1$.
Recall that $w_i = u_i$ if $(i, s) \in E$ for some $s \in \{\triangleleft, \triangleright\}$. Hence, for each $(i, s) \in E$, we have

$$\begin{aligned}|\mathrm{Affix}(v_i, s)| &= |\mathrm{Affix}(w_i, s)| \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - (|L_n(\vec{w})| - 1))\sigma_n + \sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).\end{aligned}$$

Notice that the last inequality, as above, follows from our case assumption in (9).

We then see that E is an n-decomposition for $\vec{v}$.

(4) We have that v_x is n-light, v_{x+1} is n-heavy, and

$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})).$$

Notice that we have $L_n(\vec{v}) = L_n(\vec{w})$, in particular, $|L_n(\vec{v})| = |L_n(\vec{w})|$.
Recall that $w_i = u_i$ if $(i, s) \in E$ for some $s \in \{\triangleleft, \triangleright\}$. Hence, for each $(i, s) \in E$, we have

$$\begin{aligned}|\mathrm{Affix}(v_i, s)| &= |\mathrm{Affix}(w_i, s)| \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).\end{aligned}$$

Notice that this follows from our case assumption since $\Delta(\mathrm{Rem}_n(w_x)) = \Delta(\mathrm{Rem}_n(\varepsilon)) = 0$.
Thus, we see that E is an n-decomposition for $\vec{v}$.

(5) We have that v_x is n-heavy, v_{x+1} is n-light, and

$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})).$$

We have $L_n(\vec{v}) = (L_n(\vec{w}) \setminus \{x\}) \cup \{x + 1\}$, in particular, $|L_n(\vec{v})| = |L_n(\vec{w})|$.
Recall that $w_i = u_i$ if $(i, s) \in E$ for some $s \in \{\triangleleft, \triangleright\}$. Hence, for each $(i, s) \in E$, we have

$$\begin{aligned}|\mathrm{Affix}(v_i, s)| &= |\mathrm{Affix}(w_i, s)| \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).\end{aligned}$$

Notice that this follows from our case assumption since $\Delta(\mathrm{Rem}_n(w_x)) = \Delta(\mathrm{Rem}_n(\varepsilon)) = 0$.
Thus, we see that E is an n-decomposition for $\vec{v}$.

Case 3: The word w_{x+1} is n-heavy with $(x + 1, \triangleleft) \in E$ and $(x + 1, \triangleright) \notin E$.

Notice that either v_x is a prefix of $\mathrm{Affix}(w_{x+1}, \triangleleft)$, or $\mathrm{Affix}(w_{x+1}, \triangleleft)$ is a prefix of v_x. We consider these two cases separately as follows.

Case 3.1: The word v_x is a prefix of $\mathrm{Affix}(w_{x+1}, \triangleleft)$.

From Lemma 7.24, we see that v_x and v_{x+1} are words of the form

$$v_x = a^p \qquad \text{and} \qquad v_{x+1} = a^q u$$

with

$$p + q = |\mathrm{Affix}(w_{x+1}, \triangleleft)| \geqslant 5\Lambda_n \qquad \text{and} \qquad u \in \Sigma^*.$$

Notice that if $p \geqslant \Lambda_n$, then from Lemmas 7.24 and 7.25, we see that

$$E' = (E \setminus \{(x + 1, \triangleleft)\}) \cup \{(x, \triangleleft), (x, \triangleright)\}$$

is an $(n + 1)$-decomposition for $\vec{v}$. We consider the case where $p < \Lambda_n$ as follows.

Since $p < \Lambda_n$, we see that v_x is n-light, and that v_{x+1} is n-heavy. From this, we see that $L_n(\vec{v}) = L_n(\vec{w})$. Further, we then see that

$$|\mathrm{Affix}_n(v_{x+1}, \triangleleft)| = |\mathrm{Affix}_n(w_{x+1}, \triangleleft)| - p.$$

Moreover, since $q \geqslant 2\Lambda_n$, we have $\mathrm{seg}_n(v_{x+1}, \triangleleft) = \varepsilon$, and

$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) = p + \Delta(\mathrm{Rem}_n(w_{x+1})).$$

We then see that

$$\begin{aligned}|\mathrm{Affix}(v_i,s)| &\geqslant |\mathrm{Affix}(w_i,s)| - p\\ &\geqslant \Lambda_{n-1} - (C-\Delta(\vec{w}))\\ &\qquad - (2k-n-|L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{x}\Delta(\mathrm{Rem}_n(w_j))\\ &\qquad - (p+\Delta(\mathrm{Rem}_n(w_{x+1}))) - \sum_{j=x+2}^{k}\Delta(\mathrm{Rem}_n(w_j))\\ &= \Lambda_{n-1} - (C-\Delta(\vec{v})) - (2k-n-|L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$

for each $(i,s)\in E$. Thus, we see that E is an n-decomposition for $\vec{v}$.

Case 3.2: The word $\mathrm{Affix}(w_{x+1},\triangleleft)$ is a prefix of v_x.
In this case, the words v_x and v_{x+1} are of the form

$$v_x = \mathrm{Affix}(w_x,\triangleleft)\,u \qquad \text{and} \qquad v_{x+1}\in\Sigma^*$$

where $u\in\Sigma^*$. From Lemma 7.24, $|\mathrm{Affix}(w_x,\triangleleft)|\geqslant 5\Lambda_n$ and thus v_x is n-heavy.

We may apply Lemma 7.27 to the factorisation $w_{x+1}=v_xv_{x+1}$ to obtain the following five cases.

(1) We have $|\mathrm{Affix}(v_x,\triangleright)|\geqslant\Lambda_n$.
Then, from Lemmas 7.24 and 7.25, we see that

$$E' = (E\setminus\{(x+1,\triangleleft)\})\cup\{(x,\triangleleft),(x,\triangleright)\}$$

is an $(n+1)$-decomposition for $\vec{v}$.

(2) We have $|\mathrm{Affix}(v_{x+1},\triangleleft)|\geqslant\Lambda_n$.
Then, from Lemmas 7.24 and 7.25, we see that

$$E' = E\cup\{(x,\triangleleft)\}$$

is an $(n+1)$-decomposition for $\vec{v}$.

(3) We have that v_x and v_{x+1} are both n-heavy, and

$$\Delta(\mathrm{Rem}_n(v_x))+\Delta(\mathrm{Rem}_n(v_{x+1}))\geqslant\Delta(\mathrm{Rem}_n(w_{x+1}))-\sigma_n.$$

We then see that $L_n(\vec{v})=L_n(\vec{w})\setminus\{x\}$, in particular, $|L_n(\vec{v})|=|L_n(\vec{w})|-1$. Let

$$E' = (E\setminus\{(x+1,\triangleleft)\})\cup\{(x,\triangleleft)\}$$

and notice that

$$\begin{aligned}|\mathrm{Affix}(v_i,s)| &\geqslant \Lambda_{n-1}-(C-\Delta(\vec{w}))-(2k-n-|L_n(\vec{w})|)\sigma_n-\sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(w_j))\\ &= \Lambda_{n-1}-(C-\Delta(\vec{w}))-(2k-n-(|L_n(\vec{w})|-1))\sigma_n+\sigma_n-\sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(w_j))\\ &\geqslant \Lambda_{n-1}-(C-\Delta(\vec{v}))-(2k-n-|L_n(\vec{v})|)\sigma_n-\sum_{j=1}^{k}\Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$

for each $(i,s)\in E'$. Thus, we see that E' is an n-decomposition for $\vec{v}$.

(4) We have that v_x is n-light, v_{x+1} is n-heavy, and

$$\Delta(\mathrm{Rem}_n(v_x))+\Delta(\mathrm{Rem}_n(v_{x+1}))=\Delta(\mathrm{Rem}_n(w_{x+1})).$$

This case does not need to be considered as it contradicts our earlier case assumption that v_x is n-heavy.

(5) We have that v_x is n-heavy, v_{x+1} is n-light, and

$$\Delta(\mathrm{Rem}_n(v_x))+\Delta(\mathrm{Rem}_n(v_{x+1}))=\Delta(\mathrm{Rem}_n(w_{x+1})).$$

We have $L_n(\vec{v})=(L_n(\vec{w})\setminus\{x\})\cup\{x+1\}$, in particular, $|L_n(\vec{v})|=|L_n(\vec{w})|$. Let

$$E' = (E\setminus\{(x+1,\triangleleft)\})\cup\{(x,\triangleleft)\}$$

and notice that

$$|\mathrm{Affix}(v_i, s)| \geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j))$$

$$= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j))$$

for each $(i, s) \in E'$. Thus, we see that E' is an n-decomposition for $\vec{v}$.

Case 4: The word w_{x+1} is n-heavy with $(x+1, \triangleleft) \notin E$ and $(x+1, \triangleright) \in E$.
The proof of this case is symmetric to the proof of Case 3 as above.

Case 5: The word w_{x+1} is n-heavy with $(x+1, \triangleleft), (x+1, \triangleright) \in E$.
This case is separated into two subcases:

(5.1) w_{x+1} does not contain the letter b, in particular,

$$w_{x+1} = \mathrm{Affix}(w_{x+1}, \triangleleft) = \mathrm{Affix}(w_{x+1}, \triangleright); \text{ and}$$

(5.2) w_{x+1} contains at least one b, in particular,

$$w_{x+1} = \mathrm{Affix}(w_{x+1}, \triangleleft)\, u\, \mathrm{Affix}(w_{x+1}, \triangleright)$$

where $u \in \Sigma^*$ with $|u| \geqslant 1$.

We consider these cases as follows.

Case 5.1: $w_{x+1} = \mathrm{Affix}(w_{x+1}, \triangleleft) = \mathrm{Affix}(w_{x+1}, \triangleright)$.
Here we see that

$$v_x = a^p \qquad \text{and} \qquad v_{x+1} = a^q$$

where $p + q = |w_{x+1}|$. We then separate this into three subcases based on the value of p as follows.

Case 5.1.1: $\Lambda_n \leqslant p \leqslant |w_{x+1}| - \Lambda_n$ and thus $\Lambda_n \leqslant q \leqslant |w_{x+1}| - \Lambda_n$.
From Lemmas 7.24 and 7.25, we see that

$$E' = E \cup \{(x, \triangleleft)\} \qquad \text{and} \qquad E'' = E \cup \{(x, \triangleright)\}$$

are both $(n+1)$-decompositions of $\vec{v}$.

Case 5.1.2: $p < \Lambda_n$ and thus $q > |w_{x+1}| - \Lambda_n$.
We then see that v_x is n-light. Moreover, from Lemma 7.24, we see that $|w_{x+1}| \geqslant 5\Lambda_n$ and thus v_{x+1} is n-heavy. We then notice that $L_n(\vec{v}) = L_n(\vec{w})$,

$$\Delta(\mathrm{Rem}_n(v_x)) = p \qquad \text{and} \qquad \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})) = 0$$

For each $(i, s) \in E$, we then see that

$$|\mathrm{Affix}(v_i, s)| \geqslant |\mathrm{Affix}(w_i, s)| - p$$

$$\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - p - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j))$$

$$= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).$$

Thus, E is an n-decomposition for $\vec{v}$.

Case 5.1.3: $p > |w_{x+1}| - \Lambda_n$ and thus $q < \Lambda_n$.
We see that v_{x+1} is n-light. Moreover, from Lemma 7.24 we see that $|w_{x+1}| \geqslant 5\Lambda_n$ and thus v_x is n-heavy. We notice that $L_n(\vec{v}) = (L_n(\vec{w}) \setminus \{x\}) \cup \{x+1\}$, and thus

$$|L_n(\vec{v})| = |L_n(\vec{w})|.$$

Moreover,

$$\Delta(\mathrm{Rem}_n(v_x)) = \Delta(\mathrm{Rem}_n(w_{x+1})) = 0 \qquad \text{and} \qquad \Delta(\mathrm{Rem}_n(v_{x+1})) = q$$

For each $(i,s) \in E \setminus \{(x+1,\triangleleft),(x+1,\triangleright)\}$, we then see that

$$\begin{aligned}
|\mathrm{Affix}(v_i,s)| &\geqslant |\mathrm{Affix}(w_i,s)| \\
&\geqslant |\mathrm{Affix}(w_i,s)| - q \\
&\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - q - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\
&= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).
\end{aligned}$$

Moreover, we see that for each $s \in \{\triangleleft,\triangleright\}$,

$$\begin{aligned}
|\mathrm{Affix}(v_x,s)| &= |\mathrm{Affix}(w_{x+1},s)| - q \\
&\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - q - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\
&r\Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).
\end{aligned}$$

Thus,

$$E' = (E \setminus \{(x+1,\triangleleft),(x+1,\triangleright)\}) \cup \{(x,\triangleleft),(x,\triangleright)\}$$

is an n-decomposition for $\vec{v}$.

Case 5.2: $w_{x+1} = \mathrm{Affix}(w_{x+1},\triangleleft)\, u\, \mathrm{Affix}(w_{x+1},\triangleright)$ where $u \in \Sigma^*$ with $|u| \geqslant 1$.
We separate this case into five subcases based on the length of v_x as follows.

Case 5.2.1: $|v_x| < 2\Lambda_n$.
We then see that v_x is n-light, v_{x+1} is n-heavy and thus $L_n(\vec{v}) = L_n(\vec{w})$.

From Lemma 7.24, we see that v_x and v_{x+1} are of the form

$$v_x = a^p \qquad \text{and} \qquad v_{x+1} = a^q\, u\, \mathrm{Affix}(w_{x+1},\triangleright)$$

where $p + q = |\mathrm{Affix}(w_{x+1},\triangleleft)| \geqslant 5\Lambda_n$, $p < 2\Lambda_n$ and $q \geqslant 3\Lambda_n$. We then see that

$$\Delta(\mathrm{Rem}_n(v_x))) = p \qquad \text{and} \qquad \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})) = 0.$$

For each $(i,s) \in E$, we then have

$$\begin{aligned}
|\mathrm{Affix}(v_i,s)| &\geqslant |\mathrm{Affix}(w_i,s)| - p \\
&\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - p - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\
&= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).
\end{aligned}$$

Notice that this follows since $\Delta(\mathrm{Rem}_n(w_x))) = \Delta(\mathrm{Rem}_n(w_{x+1})) = 0$.

Thus, we see that E is an n-decomposition for $\vec{v}$.

Case 5.2.2: $2\Lambda_n \leqslant |v_x| \leqslant |\mathrm{Affix}(w_{x+1},\triangleleft)|$.
In this case we see that

$$v_x = a^p \qquad \text{and} \qquad v_{x+1} = a^q\, u\, \mathrm{Affix}(w_{x+1},\triangleright)$$

for some $u \in \Sigma^*$ where $p + q = |\mathrm{Affix}(w_{x+1},\triangleleft)|$ and $p \geqslant 2\Lambda_n$.

From Lemmas 7.24 and 7.25, we see that

$$E' = (E \setminus \{(x+1,\triangleleft)\}) \cup \{(x,\triangleleft),(x,\triangleright)\}$$

is an $(n+1)$-decomposition of $\vec{v}$.

Case 5.2.3: $|\mathrm{Affix}(w_{x+1},\triangleleft)| \leqslant |v_x| \leqslant |w_{x+1}| - |\mathrm{Affix}(w_{x+1},\triangleright)|$.
We notice then that

$$v_x = \mathrm{Affix}(w_{x+1},\triangleleft)\, u \qquad \text{and} \qquad v_{x+1} = u'\, \mathrm{Affix}(w_{x+1},\triangleright)$$

for some $u, u' \in \Sigma^*$. Thus, both v_x and v_{x+1} are n-heavy.

From Lemma 7.27, we are in one of the following five cases.

(1) We have $|\mathrm{Affix}(v_x, \triangleright)| \geqslant \Lambda_n$.
Then, from Lemmas 7.24 and 7.25, we see that
$$E' = (E \setminus \{(x+1, \triangleleft)\}) \cup \{(x, \triangleleft), (x, \triangleright)\}$$
is an $(n+1)$-decomposition for $\vec{v}$.

(2) We have $|\mathrm{Affix}(v_{x+1}, \triangleleft)| \geqslant \Lambda_n$.
Then, from Lemmas 7.24 and 7.25, we see that
$$E' = E \cup \{(x, \triangleleft)\}$$
is an $(n+1)$-decomposition for $\vec{v}$.

(3) We have that v_x and v_{x+1} are both n-heavy, and
$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) \geqslant \Delta(\mathrm{Rem}_n(w_{x+1})) - \sigma_n.$$
We then see that $L_n(\vec{v}) = L_n(\vec{w}) \setminus \{x\}$, in particular, $|L_n(\vec{v})| = |L_n(\vec{w})| - 1$. Let
$$E' = (E \setminus \{(x+1, \triangleleft)\}) \cup \{(x, \triangleleft)\}$$
and notice that
$$\begin{aligned}|\mathrm{Affix}(v_i, s)| &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - (|L_n(\vec{w})| - 1))\sigma_n + \sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j))\end{aligned}$$
for each $(i, s) \in E'$. Thus, we see that E' is an n-decomposition for $\vec{v}$.

(4) We have that v_x is n-light, v_{x+1} is n-heavy, and
$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})).$$
This case does not need to be considered as it contradicts our case assumption v_x is n-heavy.

(5) We have that v_x is n-heavy, v_{x+1} is n-light, and
$$\Delta(\mathrm{Rem}_n(v_x)) + \Delta(\mathrm{Rem}_n(v_{x+1})) = \Delta(\mathrm{Rem}_n(w_{x+1})).$$
This case does not need to be considered as it contradicts our case assumption v_{x+1} is n-heavy.

Case 5.2.4: $|w_{x+1}| - |\mathrm{Affix}(w_{x+1}, \triangleright)| \leqslant |v_x| \leqslant |w_{x+1}| - 2\Lambda_n$.

In this case we see that
$$v_x = \mathrm{Affix}(w_{x+1}, \triangleleft)\, u\, a^p \qquad \text{and} \qquad v_{x+1} = a^q$$
for some $u \in \Sigma^*$ where $p + q = |\mathrm{Affix}(w_{x+1}, \triangleleft)|$ and $q \geqslant 2\Lambda_n$.

From Lemmas 7.24 and 7.25, we see that
$$E' = E \cup \{(x, \triangleleft)\}$$
is an $(n+1)$-decomposition of $\vec{v}$.

Case 5.2.5: $|v_x| > |w_{x+1}| - 2\Lambda_n$.

We then see that v_x is n-heavy, v_{x+1} is n-light and thus
$$L_n(\vec{v}) = (L_n(\vec{w}) \setminus \{x\}) \cup \{x+1\},$$
in particular, this means that $|L_n(\vec{v})| = |L_n(\vec{w})|$.

From Lemma 7.24, we see that v_x and v_{x+1} are of the form
$$v_x = \mathrm{Affix}(w_{x+1}, \triangleleft)\, u\, a^p \qquad \text{and} \qquad v_{x+1} = a^q$$
where $p + q = |\mathrm{Affix}(w_{x+1}, \triangleleft)| \geqslant 5\Lambda_n$, $p > 3\Lambda_n$ and $q < 2\Lambda_n$. We then see that
$$\Delta(\mathrm{Rem}_n(v_x))) = \Delta(\mathrm{Rem}_n(w_{x+1})) = 0 \qquad \text{and} \qquad \Delta(\mathrm{Rem}_n(v_{x+1})) = q$$

For each $(i,s) \in E \setminus \{(x+1,\triangleleft),(x+1,\triangleright)\}$, we then have

$$\begin{aligned}|\mathrm{Affix}(v_i,s)| &= |\mathrm{Affix}(w_i,s)| \\ &\geqslant |\mathrm{Affix}(w_i,s)| - q \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - q - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).\end{aligned}$$

Moreover, for each $s \in \{\triangleleft,\triangleright\}$, we see that

$$\begin{aligned}|\mathrm{Affix}(v_x,s)| &\geqslant |\mathrm{Affix}(w_{x+1},s)| - q \\ &\geqslant \Lambda_{n-1} - (C - \Delta(\vec{w})) - (2k - n - |L_n(\vec{w})|)\sigma_n - q - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(w_j)) \\ &= \Lambda_{n-1} - (C - \Delta(\vec{v})) - (2k - n - |L_n(\vec{v})|)\sigma_n - \sum_{j=1}^{k} \Delta(\mathrm{Rem}_n(v_j)).\end{aligned}$$

Thus,

$$E' = (E \setminus \{(x+1,\triangleleft),(x+1,\triangleright)\}) \cup \{(x,\triangleleft),(x,\triangleright)\}$$

is an n-decomposition for $\vec{v}$.

Conclusion:

In all cases, if $\vec{w}$ has a maximal n-decomposition, then we can either construct an n-decomposition or construct an $(n+1)$-decomposition for $\vec{v}$. □

7.6. **Mirroring decompositions.** We introduce $\mathrm{Mirror}\colon \Sigma^* \to \Sigma^*$ such that $\mathrm{Mirror}(w) = w_n \cdots w_2 w_1$ for each $w = w_1 w_2 \cdots w_n \in \Sigma^*$. We then extend this function to operate on tuples of words as follows. We define $\mathrm{Mirror}\colon (\Sigma^*)^k \to (\Sigma^*)^k$ such that for each $\vec{w} = (w_1, w_2, \ldots, w_k) \in (\Sigma^*)^k$,

$$\mathrm{Mirror}(\vec{w}) = (\mathrm{Mirror}(w_k), \ldots, \mathrm{Mirror}(w_2), \mathrm{Mirror}(w_1))$$

We define $\mathrm{Mirror}\colon \mathcal{P}(\{1,2,\ldots,k\} \times \{\triangleleft,\triangleright\}) \to \mathcal{P}(\{1,2,\ldots,k\} \times \{\triangleleft,\triangleright\})$ on decompositions as follows. Suppose that $E \subseteq \{1,2,\ldots,k\} \times \{\triangleleft,\triangleright\}$ is an n-decomposition of $\vec{w}$. We then define $\mathrm{Mirror}(E)$ such that

- $(i,\triangleleft) \in \mathrm{Mirror}(E)$ if and only if $(k+1-i,\triangleright) \in E$; and
- $(i,\triangleright) \in \mathrm{Mirror}(E)$ if and only if $(k+1-i,\triangleleft) \in E$.

We then see that $\mathrm{Mirror}(E)$ is an n-decomposition for the word vector $\mathrm{Mirror}(\vec{w})$.

Notice from the definition of the word $\mathcal{W}$ in Definition 7.5 that $\mathrm{Mirror}(\mathcal{W}) = \mathcal{W}$. Thus, from the definition of $\mathcal{W}$-sentential tuple in Definition 7.23, we see that if $\vec{w}$ is a $\mathcal{W}$-sentential tuple, then $\mathrm{Mirror}(\vec{w})$ is also a $\mathcal{W}$-sentential tuple. Finally, notice that in each of the above usages of Mirror, the function Mirror is an involution. Using these maps, we may now derive the following proposition which is a generalisation of Propositions 7.28, 7.29 and 7.30.

Proposition 7.31. *Let $\vec{w} = (w_1, w_2, \ldots, w_k)$ and $\vec{v} = (v_1, v_2, \ldots, v_k)$ be $\mathcal{W}$-sentential tuples as in Definition 7.23, and suppose that $\vec{w}$ has one of the four forms*

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, v_2, \ldots, v_{x-1}, av_x, v_{x+1}, \ldots, v_k), \tag{10}$$

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, v_2, \ldots, v_{x-1}, v_x a, v_{x+1}, \ldots, v_k), \tag{11}$$

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, v_2, \ldots, v_{x-1}, bv_x, v_{x+1}, \ldots, v_k), \tag{12}$$

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, v_2, \ldots, v_{x-1}, v_x b, v_{x+1}, \ldots, v_k) \tag{13}$$

for some $x \in \{1,2,\ldots,k\}$, that is, $\vec{v}$ is obtained from $\vec{w}$ by deleting an a or b from the left or right-hand side of some component of $\vec{w}$; or $\vec{w}$ has one of the following two forms

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, \ldots, v_{x-1}, \varepsilon, v_x v_{x+1}, v_{x+2}, \ldots, v_k), \tag{14}$$

$$\vec{w} = (w_1, w_2, \ldots, w_k) = (v_1, \ldots, v_{x-1}, v_x v_{x+1}, \varepsilon, v_{x+2}, \ldots, v_k) \tag{15}$$

for some $x \in \{1,2,\ldots,k-1\}$, that is, $\vec{v}$ is a vector for which $(w_x, w_{x+1}) \in \{(v_x v_{x+1}, \varepsilon), (\varepsilon, v_x v_{x+1})\}$ and $w_i = v_i$ for each $i \notin \{x, x+1\}$. If the tuple $\vec{w}$ has an n-decomposition, then $\vec{v}$ has an n'-decomposition for some $n' \geqslant n$. (Compare the above cases with the grammar rules in Definitions 6.4, 6.5 and 6.6.)

Proof. We see that cases (10), (12) and (14) follow from Propositions 7.28, 7.29 and 7.30, respectively. We consider the remaining cases as follows.

- Suppose our vectors $\vec{w}$ and $\vec{v}$ are in (11). Thus, the pair of vectors $\mathrm{Mirror}(\vec{w})$ and $\mathrm{Mirror}(\vec{v})$ are related as in (10), and $\mathrm{Mirror}(\vec{w})$ has an n-decomposition. From Proposition 7.28, we obtain an n'-decomposition E, with $n' \geqslant n$, for $\mathrm{Mirror}(\vec{v})$. Thus, $\mathrm{Mirror}(E)$ is an n'-decomposition for $\vec{v}$.
- Suppose our vectors $\vec{w}$ and $\vec{v}$ are in (13). Thus, the pair of vectors $\mathrm{Mirror}(\vec{w})$ and $\mathrm{Mirror}(\vec{v})$ are related as in (12), and $\mathrm{Mirror}(\vec{w})$ has an n-decomposition. From Proposition 7.29, we obtain an n'-decomposition E, with $n' \geqslant n$, for $\mathrm{Mirror}(\vec{v})$. Thus, $\mathrm{Mirror}(E)$ is an n'-decomposition for $\vec{v}$.
- Suppose our vectors $\vec{w}$ and $\vec{v}$ are in (15). Thus, the pair of vectors $\mathrm{Mirror}(\vec{w})$ and $\mathrm{Mirror}(\vec{v})$ are related as in (14), and $\mathrm{Mirror}(\vec{w})$ has an n-decomposition. From Proposition 7.30, we obtain an n'-decomposition E, with $n' \geqslant n$, for $\mathrm{Mirror}(\vec{v})$. Thus, $\mathrm{Mirror}(E)$ is an n'-decomposition for $\vec{v}$.

Hence, we have proven each of the cases of the proposition. □

7.7. **Main result.** We are now ready to prove our main result as follows.

Theorem C. *The word problem for the infinite cyclic group $\mathbb{Z}$ is not EDT0L.*

Proof. Earlier in this section, we assumed for contradiction that there exists some R-MCFG $M = (\Sigma, Q, S, P)$ which is in normal form (as in Lemma 6.8) that generates the language L as introduced in the beginning of Section 7. Moreover, we let C be the constant for the grammar M as in Corollary 7.4.

From Lemma 7.7, we see that $\mathcal{W} \in L$ and so there must be a derivation for $\mathcal{W}$ in the grammar M of the form

$$\begin{aligned} S(\mathcal{W}) \leftarrow H_1(\mathcal{W}, \varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow H_2(w_{1,1}, w_{1,2}, \ldots, w_{1,k}) \\ \leftarrow H_3(w_{2,1}, w_{2,2}, \ldots, w_{2,k}) \leftarrow \cdots \leftarrow H_{t+1}(w_{t,1}, w_{t,2}, \ldots, w_{t,k}) \\ \leftarrow H_{t+2}(\varepsilon, \varepsilon, \ldots, \varepsilon) \leftarrow \end{aligned}$$

for some $t \in \mathbb{N}$ where each $w_{i,j} \in \Sigma^*$, and each $H_i \in Q$.

We may then define a sequence of tuples $\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_{t+2}$ such that

$$\vec{w}_1 = (\mathcal{W}, \varepsilon, \varepsilon, \ldots, \varepsilon), \quad \vec{w}_{t+2} = (\varepsilon, \varepsilon, \ldots, \varepsilon) \quad \text{and} \quad \vec{w}_i = (w_{i-1,1}, w_{i-1,2}, \ldots, w_{i-1,k})$$

for each $i \in \{2, 3, \ldots, t+1\}$. From the definition of the constant C in Corollary 7.4, and the definition of an R-MCFG multiple context-free grammar, we see that each vector $\vec{w}_j$, as above, is a $\mathcal{W}$-sentential tuple as defined in Definition 7.23.

Notice here that the six cases of Proposition 7.31 correspond to the reverse of the insertion and merge replacement rules as in Definitions 6.4 and 6.5. In particular, given an application of such a rule as

$$H(w_1, w_2, \ldots, w_k) \leftarrow K(v_1, v_2, \ldots, v_k),$$

the vectors $\vec{w} = (w_1, w_2, \ldots, w_k)$ and $\vec{v} = (v_1, v_2, \ldots, v_k)$ satisfy one of the relations in Proposition 7.31. In particular, (10–13) correspond to left and right insertion rules (as in Definition 6.4); and (14) and (15) corresponds to left and right merge rules (as in Definition 6.5).

Hence, from Proposition 7.31, for each $i \in \{1, 2, \ldots, t+1\}$, we see that if $\vec{w}_i$ has an n-decomposition, then $\vec{w}_{i+1}$ has an n'-decomposition for some $n' \geqslant n$. Thus, by an induction on i, we find that if $\vec{w}_1$ has a decomposition, then so must $\vec{w}_{t+2}$.

From Corollary 7.12 and Lemmas 7.9 and 7.25, we see that $E = \{(1, \triangleleft), (1, \triangleright)\}$ is a 2-decomposition of the $\mathcal{W}$-sentential tuple $\vec{w}_1 = (\mathcal{W}, \varepsilon, \varepsilon, \ldots, \varepsilon)$ (as demonstrated in Remark 7.21). Thus, the sentential tuple $\vec{w}_{t+2} = (\varepsilon, \varepsilon, \ldots, \varepsilon)$ must have a decomposition. However, it is not possible for $\vec{w}_{t+2}$ to have a decomposition since none of its components are n-heavy for any $n \in \{2, 3, \ldots, 2k\}$. In particular, for a component to be n-heavy, it must contain a factor $a^{2\Lambda_n}$ where each $\Lambda_n > 0$ from Lemma 7.9. We thus conclude that no such grammar M for the language L can exist, and thus L cannot be an EDT0L language. □

7.8. **Corollary.** We can immediately generalise Theorem C to obtain Theorem D.

Theorem D. *If G has an EDT0L word problem, then G is a torsion group.*

Proof. Suppose for contradiction that the group G has an EDT0L word problem and an element $g \in G$ of infinite order, so $\langle g \rangle \cong \mathbb{Z}$. From Proposition A, it follows that the subgroup $\langle g \rangle \cong \mathbb{Z}$ has an EDT0L word problem with respect to the generating set $\{g, g^{-1}\}$. Thus, $\mathbb{Z}$ has an EDT0L word problem which contradicts Theorem C. □

Acknowledgements

The first author acknowledges support from Swiss NSF grant 200020-200400. The second author was supported by Australian Research Council grant DP210100271, and the London Mathematical Society Visiting Speakers to the UK–Scheme 2, 2024. The third author was supported by the Heilbronn Institute for Mathematical Research. The fifth author was supported by the Heilbronn Institute for Mathematical Research and the EPSRC Fellowship grant EP/V032003/1 'Algorithmic, topological and geometric aspects of infinite groups, monoids and inverse semigroups'.

References

[1] A. V. Anisimov. The group languages. *Kibernetika (Kiev)*, 7(4):18–24, 1971.
[2] Jean-Michel Autebert, Luc Boasson, and Géraud Sénizergues. Groups and NTS languages. *J. Comput. System Sci.*, 35(2):243–267, 1987.
[3] Laurent Bartholdi, Leon Pernak, and Emmanuel Rauzy. Groups with presentations in EDT0L, 2024. arXiv:2402.01601.
[4] Alex Bishop and Murray Elder. Bounded automata groups are co-ET0L. In *Language and automata theory and applications*, volume 11417 of *Lecture Notes in Comput. Sci.*, pages 82–94. Springer, Cham, 2019.
[5] Tara Brough, Laura Ciobanu, Murray Elder, and Georg Zetzsche. Permutations of context-free, ET0L and indexed languages. *Discrete Math. Theor. Comput. Sci.*, 17(3):167–178, 2016.
[6] Laura Ciobanu, Volker Diekert, and Murray Elder. Solution sets for equations over free groups are EDT0L languages. *Internat. J. Algebra Comput.*, 26(5):843–886, 2016.
[7] Laura Ciobanu and Murray Elder. The complexity of solution sets to equations in hyperbolic groups. *Israel J. Math.*, 245(2):869–920, 2021.
[8] Laura Ciobanu, Murray Elder, and Michal Ferov. Applications of L systems to group theory. *Internat. J. Algebra Comput.*, 28(2):309–329, 2018.
[9] Volker Diekert, Artur Jeż, Manfred Kufleitner, and Alexander Thumm. Solutions of word equations over partially commutative structures, 2025. arXiv:1603.02966.
[10] Andrew Duncan, Alex Evetts, Derek F. Holt, and Sarah Rees. Using EDT0L systems to solve some equations in the solvable B aumslag-Solitar groups. *J. Algebra*, 630:434–456, 2023.
[11] A. Ehrenfeucht and G. Rozenberg. Three useful results concerning L languages without interactions. In *L systems (Third Open House, Comput. Sci. Dept., Aarhus Univ., Aarhus, 1974)*, volume Vol. 15 of *Lecture Notes in Comput. Sci.*, pages 72–77, 327–338. Springer, Berlin-New York, 1974.
[12] A. Ehrenfeucht and G. Rozenberg. A pumping theorem for deterministic ET0L languages. *Revue française d'automatique informatique recherche opérationnelle. Informatique théorique*, 9(R2):13–23, 1975.
[13] A. Ehrenfeucht and G. Rozenberg. On inverse homomorphic images of deterministic ET0L languages. In *Automata, languages, development*, pages 179–189. North-Holland, Amsterdam-New York-Oxford, 1976.
[14] A. Ehrenfeucht and G. Rozenberg. On some context free languages that are not deterministic ETOL languages. *RAIRO Informat. Théor.*, 11(4):273–291, i, 1977.
[15] Murray Elder, Mark Kambites, and Gretchen Ostheimer. On groups and counter automata. *Internat. J. Algebra Comput.*, 18(8):1345–1364, 2008.
[16] Joost Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10(1):289–303, December 1976.
[17] Joost Engelfriet, Erik Meineche Schmidt, and Jan Van Leeuwen. Stack machines and classes of nonnested macro languages. *Journal of the ACM (JACM)*, 27(1):96–117, 1980.
[18] Paul Gallot. *Safety of data transformations*. PhD thesis, Université Lille Nord de France, 2021.
[19] Robert Gilman. *4 Groups with multiple context-free word problems*, pages 161–182. De Gruyter, Berlin, Boston, 2024.
[20] Rostislav Grigorchuk and Igor Pak. Groups of intermediate growth: an introduction for beginners, July 2006. arXiv:math/0607384.
[21] Thomas Herbst. On a subclass of context-free groups. *RAIRO Inform. Théor. Appl.*, 25(3):255–272, 1991.
[22] Derek F. Holt, Matthew D. Owens, and Richard M. Thomas. Groups and semigroups with a one-counter word problem. *J. Aust. Math. Soc.*, 85(2):197–209, 2008.
[23] Sanjay Jain, Alexei Miasnikov, and Frank Stephan. The complexity of verbal languages over groups. *Journal of Computer and System Sciences*, 101:68–85, 2019.
[24] M Latteux. EDT0L-systemes ultralineaires et operateurs associes. *Publication du laboratoire de Calcul de l'Université de Lille I*, 100, 1977.
[25] Michel Latteux. Sur les générateurs algébriques et linéaires. *Acta Inform.*, 13(4):347–363, 1980.
[26] Quang Loc Le and Mengda He. A decision procedure for string logic with quadratic equations, regular expressions and length constraints. In Sukyoung Ryu, editor, *Programming Languages and Systems*, pages 350–372, Cham, 2018. Springer International Publishing.
[27] Alex Levine. EDT0L solutions to equations in group extensions. *J. Algebra*, 619:860–899, 2023.
[28] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983.
[29] Gabriela AslıÑino Nesin and Richard M. Thomas. Groups whose word problem is a Petri net language. In *Descriptional complexity of formal systems*, volume 9118 of *Lecture Notes in Comput. Sci.*, pages 243–255. Springer, Cham, 2015.
[30] G. Rozenberg and A. Ehrenfeucht. Some ET0L languages which are not deterministic. Technical Report CU-CS-018-73, University of Colorado Boulder, December 1973.
[31] G. Rozenberg and D. Vermeir. On ET0L systems of finite index. *Information and Control*, 38(1):103–133, 1978.

[32] Grzegorz Rozenberg. Extension of tabled 0L-systems and languages. *International Journal of Computer & Information Sciences*, 2(4):311–336, December 1973.
[33] Brigitte Rozoy. The Dyck language $D_1'^*$ is not generated by any matrix grammar of finite index. *Inform. and Comput.*, 74(1):64–89, 1987.
[34] Takao Yuyama. Groups whose word problems are accepted by abelian G-automata. In *Developments in language theory*, volume 13911 of *Lecture Notes in Comput. Sci.*, pages 246–257. Springer, Cham, 2023.

Section de mathématiques, Université de Genève, rue du Conseil-Général 7-9, 1205 Genève, Switzerland
Email address: alexbishop1234@gmail.com

School of Mathematical and Physical Sciences, University of Technology Sydney, Broadway NSW 2007, Australia
Email address: murray.elder@uts.edu.au

Department of Mathematics, The University of Manchester, Manchester M13 9PL, UK
Email address: alex.evetts@manchester.ac.uk

Database Group, Universität Bremen - FB 03 Postfach 33 04 40, 28334 Bremen, Germany
Email address: pgallot@uni-bremen.de

School of Engineering, Mathematics and Physics, University of East Anglia, Norwich NR4 7TJ, UK
Email address: a.levine@uea.ac.uk

Appendix A. An alternative proof of Proposition A

We note here that this paper is almost completely self-contained. In particular, the only result which we do not prove is in Section 5 where we cite that the family of finite-index EDT0L languages is closed under inverse monoid homomorphism. This result is used in Proposition A to show that having an EDT0L word or co-word problem is invariant under change of generating set. Our reason for structuring the paper in this way is to further motivate the study of finite-index EDT0L languages.

Here we give a short alternative proof of Proposition A which only uses results proven in this paper. This alternative proof is a corollary to the following lemma which is itself a specialisation of a result of Gilman (see Theorem 4.26 and 4.29 in [19]).

Lemma A.1. *Let $\mathcal{F}$ be a family of formal languages which contains the regular languages and is closed under mapping by string transducer; let G be a group with finite monoid generating set X, and let Y be a finite set that generates a submonoid of G. If $\mathrm{WP}(G,X)$ (resp. $\mathrm{coWP}(G,X)$) lies in the family of languages $\mathcal{F}$, then $\mathrm{WP}(G,Y)$ (resp. $\mathrm{coWP}(G,Y)$) also lies in $\mathcal{F}$.*

Proof. Notice that if G is finite, then this result follows from the fact that the word problem of a finite group, and any submonoid thereof, is a regular language (see [1, Theorem 1]). Thus, in the remainder of this proof, we may assume without loss of generality that G is an infinite group.

In the remainder of this proof, suppose that $Y = \{y_1, y_2, \ldots, y_n\}$. Notice that, for each $y_i \in Y$, we can choose a word $u_i \in X^*$ for which $\overline{y_i} =_G \overline{u_i}$. From Lemma 5.1, there exists a choice of words $w_1, w_2, \ldots, w_n, w_{n+1} \in X^*$ such that $W = \{w_1u_1, w_2u_2, \ldots, w_nu_n\}$ is an antichain with respect to the prefix order where each $\overline{w_i} = 1$.

We define a map $f\colon \mathcal{P}(X^*) \to \mathcal{P}(Y^*)$ as

$$f(L) = \left\{ y_{i_1}y_{i_2}\cdots y_{i_k} \in Y^* \;\middle|\; \begin{array}{c}(w_{i_1}u_{i_1})(w_{i_2}u_{i_2})\cdots(w_{i_k}u_{i_k}) \in L \\ \text{where each } i_j \in \{1,2,\ldots,n\}\end{array} \right\}.$$

From Lemma 4.5, we see that f is a string transducer.

We that $f(\mathrm{WP}(G,X)) = \mathrm{WP}(G,Y)$ and $f(\mathrm{coWP}(G,X)) = \mathrm{coWP}(G,Y)$. Since the family $\mathcal{F}$ is closed under string transduction, we conclude that $\mathrm{WP}(G,Y)$ and $\mathrm{coWP}(G,Y)$ both belong to $\mathcal{F}$. □

We then obtain an alternative proof of Proposition A as follows.

Proposition A. *Let G be a group with finite monoid generating sets X and Y. If $\mathrm{WP}(G,X)$ (resp. $\mathrm{coWP}(G,X)$) is EDT0L, then $\mathrm{WP}(G,Y)$ (resp. $\mathrm{coWP}(G,Y)$) is EDT0L of finite index. These results also hold if Y instead generates a submonoid of G. In particular, this implies that, if a word problem is EDT0L, then it is EDT0L of finite index; and that having a word problem that is EDT0L of finite index is independent of choice of generating set.*

Proof. This follows by combining Lemma A.1 with Lemmas 3.3 and 4.4, and Proposition 5.2. □