Equations in groups, formal languages and complexity

Alexander Levine

Submitted for the degree of

Doctor of Philosophy

Heriot-Watt University

Department of Mathematics,

School of Mathematical and Computer Sciences.

May, 2022

Abstract

We study the use of EDT0L languages to describe solutions to systems of equations in various classes of groups. We show that solutions to systems of equations with rational constraints in virtually abelian groups can be expressed as EDT0L languages. We also study the growth series of these solutions. In addition, we show that the class of groups where solutions can be described using EDT0L languages is closed under direct products, wreath products with finite groups and passing to finite-index subgroups, using standard normal forms in each of the constructions. Using these operations together, we show that the solutions to systems of equations, when expressed as suitable quasi-geodesic normal forms, in virtually direct products of hyperbolic groups, including dihedral Artin groups, can be described using EDT0L languages. We conclude by showing that single equations in one variable in the Heisenberg group can also be expressed using EDT0L languages, with words expressed in Mal'cev normal form. Proving this requires us to first show that solutions to quadratic equations in the ring of integers are EDT0L.

Acknowledgements

First of all, I would like to thank my supervisor Laura Ciobanu for the incredible amount of mathematical guidance, feedback on work and help with applications she has given me during my time at Heriot-Watt. I would also like to thank her for introducing me to EDT0L languages. I've greatly enjoyed learning about them and applying them to group theory.

I would like Alex Evetts and Alan Logan for the various bits of advice they have given about various aspects of academic life, as well as the many mathematical discussions we've had.

I would also like to thank Alex Bishop, Gemma Crowe, Murray Elder, Albert Garreta, Nick Gilbert, Antoine Goldsborough, Mark Lawson, Alexandre Martin, Ross Paterson, Alessandro Sisto and Nicolas Vaskou for discussions and mathematical advice, and the sharing of helpful knowledge and insight.

Thanks are also due to my flatmate Joshua Dibble for very patiently listening to me talk about languages and groups for the last three years, and for being a very nice person to live with, particularly during lockdown.

Finally, I would like to thank my family for their love and support throughout this process. In particular, I wish to thank my fiancée Isaree for her love, and for all of her patience with many of the drawbacks of being with someone in academia.

Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name:	**Alexander Levine**		
School:	**Mathematical and Computer Sciences**		
Version: *(i.e. First, Resubmission, Final)*	**Final**	Degree Sought:	**PHD**

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:	A Levine	Date:	16/05/2022

Submission

Submitted By *(name in capitals)*:	ALEX LEVINE
Signature of Individual Submitting:	A Levine
Date Submitted:	16/05/2022

For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes		No		Approved	Yes		No	
*E-thesis Submitted (**mandatory for final theses**)*										
Received in the SSC by *(name in capitals)*:						Date:				

Inclusion of Published Works

Please note you are only required to complete the Inclusion of Published Works Form if your thesis contains published works under Regulation 6 (9.1.2)

Declaration

This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

Citation details	Alex Evetts and Alex Levine, Equations in virtually abelian groups: Languages and growth, *Internat. J. Algebra Comput.* 32 (2022), no. 3, 411–442.
Author 1	Equal effort; all work was jointly discussed and worked on together
Author 2	Equal effort; all work was jointly discussed and worked on together
Signature:	A Levine
Date:	16/05/2022

Contents

Chapter 1

Introduction

Formal languages have been used in a variety of settings with finitely generated groups over the last few decades. This is, in part, because a finite generating set for a group can be thought of as an alphabet, with any set of words over the generating set being a language that is easily associated with a subset of the group itself. In 1971, Anisimov proved that the set of all words over a generating set for a group G that represent the identity element, called the *word problem* of G, forms a regular language if and only if G is finite [5]. This deep connection between formal languages and group theory has evolved over the subsequent decades with a number of striking results. Perhaps one of the most celebrated is the result of Muller and Schupp, finished later by Dunwoody, which states that the word problem of a group is a context-free language if and only if the group is virtually free ([74], [37]).

Languages can also be used to study the growth series of a group: a group with a regular geodesic normal form has a rational growth series. Regular languages can be used to describe $(\lambda,\ \mu)$-quasi-geodesics in hyperbolic groups if λ and μ are rational [54]. In addition, languages have been used to study conjugacy representatives in various classes of groups ([21], [55]). The complement of the word problem has also been studied for a wide variety of groups, including Thompson's groups, the Grigorchuk group and Baumslag-Solitar groups ([10], [20], [57], [12], [62] [48]).

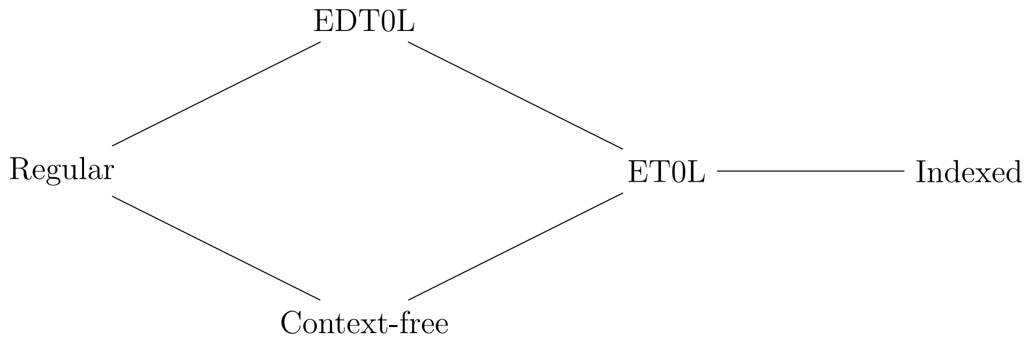
The primary focus of this thesis will be the use of languages to describe solutions to equations in groups. Equations are a generalisation of two of Dehn's decision problems for groups: the word problem and the conjugacy problem (checking whether any given pair of elements g and h in a group G are conjugate). An *equation* in a group G is an identity $w = 1$, where w is a word over G together with a finite set of *variables* and their inverses. A *solution* to an equation is an assignment of an element of G to each variable, such that plugging this into w yields a word equivalent to 1. Thus the conjugacy problem in a group G can be thought of as solving the equation $X^{-1}gXh^{-1} = 1$ for all $g,\ h \in G$.

Since the 1960s, many papers have discussed algorithms to decide whether or not equations in a variety of different classes of groups admit solutions. A first major positive result in this area is due to Makanin, during the 1980s, when he proved that it is decidable whether a finite system of equations in a free group is satisfiable ([67], [68], [69]). Since then, Makanin's work has been extended to show the decidability of the satisfiability of equations in hyperbolic groups ([82], [27]), solvable Baumslag-Solitar groups [60], right-angled Artin groups [33] and more.

Whilst Makanin's work can determine if an equation admits a solution, it does not describe the set of solutions. Razborov later created a method that allows one to construct the solutions to systems of equations in a free group ([79], [80]). Since sets of solutions are often infinite, there are multiple ways they can be represented, if at all. One method in which solutions can be described is by expressing the set of solutions as a language, and then defining a grammar for the language.

In 2016, Ciobanu, Diekert and Elder successfully employed languages to describe the set of solutions to systems of equations in free groups [17]. The class of languages used was the class of EDT0L languages. Diekert and Elder generalised this to virtually free groups [31], and Diekert, Jeż and Kufleitner extended it to right-angled Artin groups [32]. Hyperbolic groups ([18], [19]), virtually abelian groups [47] and virtually direct products of hyperbolic groups [65] followed later. Context-free languages do not in general work for describing equations. Even in $\mathbb{Z}$, the system $X = Y$ and $Y = Z$ would have the solution language $\{a^x\#a^x\#a^x \mid x \in \mathbb{Z}\}$ with respect to the standard normal form, and this is not a context-free language.

Figure 1.1: Reading left to right gives the strict containments of the classes of languages



Of course, it is possible to represent solutions in different language-theoretic ways, such as by writing solutions as tuples of words. This yields some results in the virtually abelian case, where solutions are shown to be accepted by multivariable finite-state automata, which we explore in Chapter 4. However, in most other classes of groups studies, this has not been shown to work. In the Heisenberg group, which is considered in Chapter 6, single equations with one variable will not in general have a context-free solution language, so EDT0L does appear to be more fitting.

In the 1960s, Lindenmayer introduced a collection of classes of languages called *L-systems*, which were originally used for the study of growth of organisms. EDT0L languages are one of the L-systems, and were introduced by Rozenberg in 1937 [85]. L-systems, including EDT0L languages, were the focus of a number of computer science articles in the 1970s and early 1980s. Since Ciobanu, Diekert and Elder's use of EDT0L languages to study equations in free groups, a number of other works have used EDT0L languages (or a similar class called ET0L languages) ([11], [14], [20]). Figure 1.1 shows how EDT0L and ET0L languages fit into the Chomsky hierarchy of languages.

The structure of this thesis is as follows. Chapter 2 covers the preliminary information used in later chapters, including formal languages, space complexity, regular, context-free and indexed languages, rational and recognisable subsets of groups, and group equations. Chapter 3 includes the definition of EDT0L and ET0L languages, and aims to provide a comprehensive introduction to these classes. It contains proofs of a number of results from the 1970s.

Chapter 4 is based on the joint work of the author with Alex Evetts [47] and the author's work [65], and covers the proof that solutions to systems of equations in virtually abelian groups can be represented as EDT0L languages. We prove the following:

Theorem 4.3.16. *The solution language to any system of equations with rational constraints in a virtually abelian group is accepted by a multivariable finite-state automaton.*

Corollary 4.3.17. *The solution language to any system of equations with rational constraints in a virtually abelian group is EDT0L.*

We also consider the growth series of solutions to equations in virtually abelian groups.

Theorem 4.4.3. *Let G be a virtually abelian group. Then every algebraic set of G has rational weighted growth series with respect to any finite generating set.*

Corollary 4.4.21. *Every algebraic set of a virtually abelian group has holonomic weighted multivariate growth series.*

Chapter 5 is based on the author's work [65], and proves that the solution languages to systems of equations in various extensions of groups are EDT0L. This is used to show that solutions to systems of equations in virtually direct products of hyperbolic groups, including dihedral Artin groups, can be expressed as EDT0L languages. We show:

Theorem 5.1.1. *Let G and H be groups where solution languages to systems of equations are EDT0L, with respect to normal forms η_G and η_H, respectively, and EDT0L systems are constructible in $\mathsf{NSPACE}(f)$, for some function $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then in the following groups, solutions to systems of equations are EDT0L, and an EDT0L system can be constructed in $\mathsf{NSPACE}(f)$:*

1. $G \wr F$, for any finite group F (Proposition 5.4.5);
2. $G \times H$ (Proposition 5.4.6);
3. Any finite index subgroup of G (Proposition 5.5.3);

In the following groups, solutions to systems of equations are EDT0L, and an EDT0L system can be constructed in $\mathsf{NSPACE}(n^4 \log n)$:

4. Any group that is virtually a direct product of hyperbolic groups (Corollary 5.6.9);
5. Dihedral Artin groups (Corollary 5.6.10).

If η_G and η_H are both quasi-geodesic or regular, then the same will be true for the normal forms used in (1), (2) and (3). It is possible to choose normal forms for the groups that are virtually direct products of hyperbolic groups in (4), and dihedral Artin groups in (5) that are regular and quasi-geodesic.

Chapter 6 is based on the author's work [66], and covers single equations in one variable in the Heisenberg group. This is done by first expressing solutions to quadratic Diophantine equations in the ring of integers as EDT0L languages, as in Theorem 6.5.15. We show the following:

Theorem 6.6.5. *Let L be the solution language to a single equation with one variable in the Heisenberg group, with respect to the Mal'cev generating set and normal form. Then*

1. The language L is EDT0L;
2. An EDT0L system for L is constructible in $\mathsf{NSPACE}(n^8 (\log n)^2)$, where the input size is the length of the equation as an element of $H(\mathbb{Z}) * F(X)$.

Theorem 6.5.15. *Let*

$$\alpha X^2 + \beta XY + \gamma Y^2 + \delta X + \epsilon Y + \zeta = 0$$

be a two-variable quadratic equation in the ring of integers, with a set S of solutions. Then

1. *The language* $L = \{a^x \# b^y \mid (x,\ y) \in S\}$ *is EDT0L over* $\{a,\ a^{-1},\ b,\ b^{-1},\ \#\}$*;*
2. *Taking the input size to be* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|)$*, an EDT0L system for* L *is constructible in* $\mathsf{NSPACE}(n^4 \log n)$*.*

In Chapter 7 we explicitly construct the systems of equations in the ring of integers that single equations in class 2 nilpotent groups are 'equivalent' to. We then use this to show that the satisfiability of single equations in virtually the Heisenberg group is decidable.

Theorem 7.3.6. *The single equation problem in a virtually Heisenberg group is decidable.*

Further work includes using some of the explicit constructions in Chapter 7 to show equations in more than one variable in the Heisenberg group are or are not EDT0L with respect to the Mal'cev normal form. This could also be extended to show that the satisfiability of single equations in virtually a class 2 nilpotent groups with a virtually cyclic commutator subgroup is decidable.

Chapter 2

Preliminaries

In this chapter we cover the preliminaries which we will need for later chapters. We start with the basic definitions of formal languages and space complexity. We then cover the definitions of three standard classes of languages: regular, context-free and indexed. Using the definition of regular languages, we can define rational and recognisable subsets of groups. We conclude with definitions and examples of group equations, as well as ways in which we can represent their solutions as languages.

Notation 2.0.1. Please note the following notation conventions:

1. Functions will be written to the right of their arguments; that is $(x)f$ or xf will be used instead of $f(x)$;
2. If S is a subset of a group, we define $S^{\pm} = S \cup S^{-1}$, where $S = \{s^{-1} \mid s \in S\}$.

2.1 Formal languages

Formal languages have been used across mathematics, computer science and linguistics for a variety of purposes. Many different classes of languages exist, often defined by a type of grammar. Using a language to represent a decision problem can measure a type of 'complexity' for the problem. This type of complexity is often related to space complexity. In Chapters 4 to 6 we will assign a 'language complexity' to the solutions to systems of equations in certain classes of groups.

We start with the definition of an alphabet, a word and a language.

Definition 2.1.1. An *alphabet* is a finite set. Elements of an alphabet are called *letters*.

A finite sequence $(a_1,\ \dots,\ a_n)$ of letters in an alphabet Σ is called a *word* over Σ, and denoted $a_1 \cdots a_n$. A *language* over Σ is a set of words over Σ.

The *empty word* is the word obtained from an empty sequence (over any alphabet), and is denoted by ε.

We gives some examples of languages.

Example 2.1.2.

1. The set of words in the English dictionary is a finite language over the alphabet $\{a,\ \dots,\ z\}$ with a few extra symbols;
2. The set $\{\varepsilon,\ a,\ a^2,\ \dots\}$ of all words over the single-letter alphabet $\{a\}$ is a language;
3. The set of all binary strings with the same number of 0s as 1s is a language over the alphabet $\{0,\ 1\}$.

We introduce two standard operations on languages.

Definition 2.1.3. Let L and M be languages. The *Kleene star closure* of L, denoted L^*, is defined by

$$L^* = \{w_1 \cdots w_n \mid n \in \mathbb{Z}_{\geq 0},\ w_1,\ \dots,\ w_n \in L\}.$$

The *concatenation* of L with M, denoted LM, is defined by

$$LM = \{uv \mid u \in L,\ v \in M\}.$$

Remark 2.1.4. If Σ is an alphabet then Σ^* is the set of all words over Σ.

We give some examples of the use of concatenation and Kleene star closure.

Example 2.1.5. Let $L = \{a\}^* = \{\varepsilon,\ a,\ a^2,\ \ldots\}$, and let $M = \{b\}^* = \{\varepsilon,\ b,\ b^2,\ \ldots\}$.

1. The concatenation LM is the language $\{a^m b^n \mid m,\ n \in \mathbb{Z}_{\geq 0}\}$;
2. The Kleene star closure of L is L;
3. The Kleene star closure of LM is $\{a,\ b\}^*$.

2.2 Free monoids

Free monoids and homomorphisms between free monoids appear throughout language theory and are central to the definition of an EDT0L language. We give a brief definition of a free monoid. Note that within this thesis we will only be considering finitely generated free monoids; however, the definition extends easily to the infinite case.

Definition 2.2.1. Let Σ be a finite set. The *(finitely generated) free monoid* on Σ, denoted Σ^*, is the monoid of all words over Σ, with the operation of concatenation.

Remark 2.2.2. We are using the same notation Σ^* to denote both the free monoid on Σ and the underlying set of the free monoid (the set of all words over Σ).

Remark 2.2.3. We will frequently refer to a *free monoid homomorphism*. This is a homomorphism $\phi\colon \Sigma^* \to \Delta^*$ for some free monoids Σ^* and Δ^*. Note that to fully define ϕ, we only need to know where every element of Σ goes, as the fact that ϕ is a homomorphism determines the action of ϕ from its action on Σ. We will frequently define free monoid homomorphisms by their action on their domains.

2.3 Space complexity

We give a brief definition of space complexity. We refer the reader to [77] for a comprehensive introduction to space complexity. We will be considering EDT0L systems in detail in later chapters, and we refer to [17] for the consideration of space complexity when constructing EDT0L systems.

Definition 2.3.1. Let $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ be a function. We say that an algorithm runs in $\mathsf{NSPACE}(f)$ if it can be performed by a non-deterministic Turing machine with the following:

1. A read-only input tape;
2. A write-only output tape;
3. A read-write work tape such that now computation path in the Turing machine uses more than $\mathcal{O}(nf)$ units of the work tape, for an input of length n.

An algorithm is said to run in *non-deterministic linear (resp. quadratic, resp. polynomial) space* if it runs in $\mathsf{NSPACE}(f)$, for some linear (resp. quadratic, resp. polynomial) function $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$.

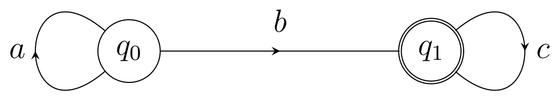
Remark 2.3.2. We will often say that grammars or automata that define languages are *constructible* in $\mathsf{NSPACE}(f)$. This means that there is an algorithm that runs in $\mathsf{NSPACE}(f)$, which takes an input that will be specified (which is sometimes other grammars or automata), and outputs the desired grammar or automaton.

2.4 Regular languages

The Chomsky hierarchy of languages is the ascending sequence (with respect to containment) which comprises the classes of regular, context-free, context-sensitive, recursive and recursively enumerable languages. Recursively enumerable languages are those accepted by a Turing machine, and recursive languages are those that are both recursively enumerable, and have a recursively enumerable complement. We define context-free languages in the appendix (A.1), and refer the reader to [58] for a comprehensive introduction to the languages in the Chomsky hierarchy.

The most restrictive class of languages in Chomsky's hierarchy is the class of regular languages. These have been widely studied and used in a variety of fields across mathematics and computer science. We begin with the definition of a finite-state automaton. We refer the reader to [56] for a more thorough introduction to regular languages.

Figure 2.1: Finite state automaton for for $\{a^m bc^n \mid m,\ n \in \mathbb{Z}_{\geq 0}\}$, with start state q_0 and accept state q_0.



Definition 2.4.1. A *finite-state automaton* is a tuple $\mathcal{A} = (\Sigma,\ \Gamma,\ q_0,\ F)$, where

1. Σ is an alphabet;
2. Γ is a finite edge-labelled directed graph with labels from $\Sigma \cup \{\varepsilon\}$;
3. $q_0 \in V(\Gamma)$ is called the *start state*;
4. $F \subseteq V(\Gamma)$ is called the set of *accept states.*

We call vertices in Γ *states.*

A word $w \in \Sigma^*$ is *accepted* by $\mathcal{A}$ if there is a path in Γ from q_0 to a state in F, where w is the word obtained by concatenating the labels of the edges in the path. The *language accepted* by $\mathcal{A}$ is the set of all words accepted by $\mathcal{A}$.

A language is called *regular* if it accepted by a finite-state automaton.

Example 2.4.2. We will show that the language $L = \{a^m bc^n \mid m,\ n \in \mathbb{Z}_{\geq 0}\}$ is regular over $\{a,\ b,\ c\}$. The finite-state automaton defined in Figure 2.1 accepts a language that is contained in L, as reading any word in the automaton results in reading any number of as, followed by one b, followed by any number of cs. Moreover, if $w = a^m bc^n \in L$, then we can use this automaton to accept w by traversing the edge labelled by a at q_0 m times, then reading one b to transfer to q_1, then traversing the c edge n times, before being accepted. Thus this automaton accepts L, and L is a regular language.

We consider the closure properties that regular languages satisfy. A *full abstract family of languages* is a class of languages that is closed under finite unions, finite intersections, Kleene star closure, images under free monoid homomorphisms and pre-images under free monoid homomorphisms. Regular languages form the smallest full abstract family of languages. For the space complexity claims, we refer the reader to Remark 2.3.2 for the definition of constructible.

Lemma 2.4.3. *Let L and M be regular languages over alphabets Σ_L and Σ_M, that are constructible in* $\mathsf{NSPACE}(f)$ *for some* $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$*. Let* $\phi\colon \Sigma_L^* \to \Sigma_M^*$ *be a free monoid homomorphism that is constructible in constant space. Then the following languages are regular, and constructible in* $\mathsf{NSPACE}(f)$*:*

1. *$L \cup M$ (union);*
2. *$L \cap M$ (intersection);*
3. *LM (concatenation);*
4. *L^* (Kleene star closure);*
5. *$L\phi$ (homomorphism);*
6. *$L\phi^{-1}$ (inverse homomorphism).*

Proof Let $\mathcal{A}_L = (\Sigma,\ \Gamma_L,\ q_L,\ F_L)$ and $\mathcal{A}_M = (\Sigma,\ \Gamma_M,\ q_M,\ F_M)$ be finite state automata accepting L and M, respectively, that are both constructible in $\mathsf{NSPACE}(f)$.

1. The finite state automaton that accepts $L \cup M$ is obtained by taking the union $\Gamma_L \cup \Gamma_M$ and adding an additional state, q_0. We attach an ε-labelled edge from q_0 to q_L and q_M, and then set q_0 to be the start state. The accept states will be $F_L \cup F_M$. Printing this can be done using the memory required to print both of $\mathcal{A}_L$ and $\mathcal{A}_M$, plus a constant, and thus it is constructible in $\mathsf{NSPACE}(f)$.
2. We can take $\mathcal{A}_L \times \mathcal{A}_M$, to be our finite state automaton for $L \cap M$, where the start state is $(q_L,\ q_M)$, and the set of accept states is $F_L \times F_M$. To write this down, we proceed with the construction of $\mathcal{A}_L$, but whenever we would normally output a state q, we instead output $\{q\} \times \mathcal{A}_M$, and whenever we would add an edge between states q_1 and q_2, we instead add all edges between $\{q_1\} \times \mathcal{A}_M$ and $\{q_2\} \times \mathcal{A}_M$, by going through the construction of $\mathcal{A}_M$. To do this, we never need to store more than the information required to write down both $\mathcal{A}_M$ and $\mathcal{A}_L$ plus a constant, and thus this can be completed in $\mathsf{NSPACE}(f)$.
3. We construct a new automaton whose directed graph is $\Gamma_L \cup \Gamma_M$, and with an ε-labelled edge added from each accept state of $\mathcal{A}_L$ to q_M. We then set the

start state to be q_L and the set of accept states to be F_M. By construction, the set of words accepted by this finite-state automaton will trace an accepted path in Γ_L, and then an accepted path in Γ_M, and thus will be LM. We can use the same argument from (1) to show that this is constructible in $\mathsf{NSPACE}(f)$.

4. To create a finite-state automaton accepting L^*, we modify $\mathcal{A}_L$ by adding an ε-labelled edge from every accept state to the start state. We only need the information to construct $\mathcal{A}_L$ to construct this, and so it can be done in $\mathsf{NSPACE}(f)$.

5. We can do this by constructing $\mathcal{A}_L$, except whenever we would output an edge labelled with a, we instead output a path labelled with $a\phi$. As we only need the information necessary to construct $\mathcal{A}$ and ϕ, this is constructible in $\mathsf{NSPACE}(f)$.

6. We adapt the construction in [19], Proposition 3.3. First let $\bar{\Sigma}_L = \{\bar{a} \mid a \in \Sigma_L\}$ be a copy of Σ_L, disjoint with Σ_M, and let $\bar{L}$ be the language obtained from L by replacing every occurrence of $a \in \Sigma_L$ with $\bar{a}$. Now let

$$K = \{y_0 x_1 y_1 \cdots x_n y_n \mid n \in \mathbb{Z}_{>0},\ x_1 \cdots x_n \in \bar{L},\ y_1,\ \ldots,\ y_n \in \Sigma_M^*\}.$$

We can construct a finite state automaton accepting K in $\mathsf{NSPACE}(f)$, by constructing $\mathcal{A}_L$, but replacing each occurrence of $a \in \Sigma_L$ with $\bar{a}$, and then for each $a \in \Sigma_M$, adding a loop in each vertex labelled with $a \in \Sigma$. Now consider the regular language

$$S = \{(y_1\phi)\bar{y}_1(y_2\phi)\bar{y}_2 \cdots (y_n\phi)\bar{y}_n \mid n \in \mathbb{Z}_{>0},\ y_1,\ \ldots,\ y_n \in \Sigma_L^*\}.$$

Note that the size of S is constant; it depends only on ϕ. Let $\tau \colon (\Sigma_M \cup \bar{\Sigma}_L)^* \to (\Sigma_M)^*$ be the free monoid homomorphism defined by $a\tau = a$ if $a \in \Sigma_M$, and $\bar{a}\tau = \varepsilon$, if $\bar{a} \in \bar{\Sigma}_L$. By construction, $L\phi^{-1} = (K \cap S)\tau$. Using (2) and (3), it follows that $L\phi^{-1}$ is constructible in $\mathsf{NSPACE}(f)$.

□

An alternative method of defining regular languages is through the use of rational expressions. These are discussed in detail in [58], Chapter 2. They are built by writing regular languages using finite languages and the following operations: finite union, concatenation and Kleene star closure. Lemma 2.4.3 shows that any language defined this way will indeed be regular. It still remains to show that they define all regular languages. We begin with the definition.

Definition 2.4.4. We define a *rational expression* (sometimes called a *regular expression*) inductively as follows.

1. Finite languages are rational expressions, which we denote $\{w_1, \ldots, w_n\}$;
2. If R and S are rational expressions, then RS is. This defines the concatenation of the languages that R and S define;
3. If R and S are rational expressions, then $R \cup S$ is a regular expression. This defines the union of the languages that R and S define;
4. If R is a rational expression, then R^* is. This defines the Kleene star closure of the language that R defines;
5. If R is a rational expression, then (R) is. This defines the same language as R. We use parentheses to give the order in which operations are to be performed.

When no parentheses are used, Kleene star operators are applied first, then concatenation, then union.

We give some examples of rational expressions and the languages they define.

Example 2.4.5.

1. As finite languages $\{a, b\}$, $\{b, cd\}$ and $\{d\}$ are rational expressions;
2. Since $\{a, b\}$ is a rational expression, $\{a, b\}^*$ is, which defined the Kleene star closure of $\{a, b\}$; that is, the set of all words over $\{a, b\}$;
3. As the concatenation of two rational expressions $\{a, b\}^*\{d\}$ is a rational expression. It defines the language

$$\{wd \mid w \in \{a, b\}^*\};$$

4. As the union of two rational expressions $(\{a, b\}^*\{d\}) \cup \{b, cd\}^*$ is a rational

expression. It defines the language

$$\{wd \mid w \in \{a,\ b\}^*\} \cup \{b,\ cd\}^*.$$

The class of languages accepted by rational expressions is the class of regular languages. We refer the reader to [58] for the proof.

Theorem 2.4.6 (Kleene's Theorem, [58], Theorem 2.3 and Theorem 2.4)**.** *A language is regular if and only if it is defined by a rational expression.*

2.5 Rational and recognisable subsets of monoids

We cover here the basic definitions of rational and recognisable subsets of monoids. Both types are used as constraints for variables in equations in groups, and we will use recognisable constraints to show that the class of groups where solutions to systems of equations form EDT0L languages is closed under passing to finite index subgroups. Rational subsets of monoids are required in the definition of an EDT0L language.

Definition 2.5.1. Let S be a monoid, and Σ be a (monoid) generating set for S. Define $\pi \colon \Sigma^* \to S$ to be the natural homomorphism. We say a subset $A \subseteq S$ is

1. *recognisable* if $A\pi^{-1}$ is a regular language over Σ;
2. *rational* if there is a regular language L over Σ, such that $A = L\pi$.

Remark 2.5.2. Recognisable sets are rational.

In a free monoid, rational sets are recognisable as well; this is part of Kleene's theorem that we omitted earlier. However, this is not true in general for arbitrary monoids.

We give a few examples of recognisable and rational sets.

Example 2.5.3. Finite subsets of any monoid are rational. Finite subsets of a group G are recognisable if and only if G is finite [53]. Finite index subgroups of any group are recognisable, and hence rational.

The following result of Grunschlag relates the rational subsets of a finite index subgroup of a group G to the rational subsets of G itself.

Lemma 2.5.4 ([52], Corollary 2.3.8)**.** *Let G be a group with finite generating set Σ, and H be a finite index subgroup of G. Let Δ be a finite generating set for H, and T be a right transversal for H in G. For each rational subset $R \subseteq G$, such that $R \subseteq Ht$ for some $t \in T$, there exists a (computable) rational subset $S \subseteq H$ (with respect to Δ), such that $R = St$.*

Herbst and Thomas proved that recognisable sets in a group G are always finite unions of cosets of a finite index normal subgroup of G [53]. This can be used to prove many facts about recognisable sets, including the following lemma.

Lemma 2.5.5. *Let G be a finitely generated group with a finite index subgroup H, and let $S \subseteq H$. Then S is recognisable in G if and only if S is recognisable in H.*

2.6 Group equations and languages

2.6.1 Group equations

We define here a system of equations within a group, and certain generalisations including twisting and constraints. Twisted equations prove useful in showing that systems of equations with rational constraints in finite extensions of a group G have EDT0L solutions, if systems of equations in G have EDT0L solutions.

Definition 2.6.1. Let G be a group, and $\mathcal{X}$ be a finite set of variables. A *finite system of equations* in G with *variables* $\mathcal{X}$ is a finite subset $\mathcal{E}$ of $G * F_{\mathcal{X}}$, where $F_{\mathcal{X}}$ is the free group on a finite set $\mathcal{X}$. If $\mathcal{E} = \{w_1, \ldots, w_n\}$, we view $\mathcal{E}$ as a system by writing $w_1 = w_2 = \cdots = w_n = 1$. A *solution* to a system $w_1 = \cdots = w_n = 1$ is a homomorphism $\phi \colon F_{\mathcal{X}} \to G$, and such that $w_1\bar{\phi} = \cdots = w_n\bar{\phi} = 1_G$, where $\bar{\phi}$ is the extension of ϕ to a homomorphism from $G * F_{\mathcal{X}} \to G$, defined by $g\bar{\phi} = g$ for all $g \in G$.

Let $\Omega \leq \mathrm{Aut}(G)$. A *finite system of* Ω*-twisted equations* in G with *variables* $\mathcal{X}$ is a finite subset $\mathcal{E}$ of the monoid $(G \cup F_{\mathcal{X}} \times \Omega)^*$, and is again denoted $w_1 = \cdots = w_n = 1$. Define the function

$$p \colon G \times \mathrm{Aut}(G) \to G$$

$$(g,\ \psi) \mapsto g\psi.$$

If $\phi \colon F_{\mathcal{X}} \to G$ is a homomorphism, let $\bar{\phi}$ denote the (monoid) homomorphism from $(G \cup F_{\mathcal{X}} \times \Omega)^*$ to $(G \times \Omega)^*$, defined by $(h,\ \psi)\bar{\phi} = (h\phi,\ \psi)$ for $(h,\ \psi) \in F_{\mathcal{X}} \times \Omega$ and $g\bar{\phi} = g$ for all $g \in G$. A *solution* is a homomorphism $\phi \colon F_{\mathcal{X}} \to G$, such that $w_1\bar{\phi}p = \cdots = w_n\bar{\phi}p = 1_G$. When $\Omega = \mathrm{Aut}(G)$, we omit the reference to Ω, and call such a system a *finite system of twisted equations*.

For the purposes of decidability in finitely generated groups, the elements of G will be represented as words over a finite generating set, and in twisted equations, automorphisms will be represented by their action on the generators.

A *finite system of (twisted) equations with rational (recognisable) constraints* $\mathcal{E}$ in a group G is a finite system of (twisted) equations $\mathcal{F}$ with variables $X_1,\ \ldots,\ X_n$, together with a tuple of rational (recognisable) subsets $R_1,\ \ldots,\ R_n$ of G. A *solution* to $\mathcal{E}$ is a solution ϕ to $\mathcal{F}$, such that $X_i\phi \in R_i$ for all i.

Remark 2.6.2. A solution to an equation with variables $X_1,\ \ldots,\ X_n$ will usually be represented as a tuple $(x_1,\ \ldots,\ x_n)$ of group elements, rather than a homomorphism. We can obtain the homomorphism from the tuple by defining $X_i \mapsto x_i$ for each i.

Example 2.6.3. Equations in $\mathbb{Z}$ are linear equations in integers, and elementary linear algebra can be used to determine satisfiability, and also describe solutions.

For example, if we write a as the free generator for $\mathbb{Z}$, then

$$a^3Xa^{-5}Y^{-1}a^7Ya^{-1}X = 1 \tag{2.1}$$

is an equation in $\mathbb{Z}$. We can rewrite (2.1) using additive notation as

$$3 + X - 5 - Y + 7 + Y - 1 + X = 0. \tag{2.2}$$

Using the fact that $\mathbb{Z}$ is abelian, we can manipulate (2.2) to obtain the following equation with the same set of solutions:

$$2X + 4 = 0.$$

Thus our set of solutions (when written as tuples) will be

$$\{(2,\ y) \mid y \in \mathbb{Z}\}.$$

Example 2.6.4. The conjugacy problem in any group can be viewed as an equation $X^{-1}gX = h$, where g and h are group elements, and X is a variable. For example, in the free group $F(a,\ b)$, one could consider the equation $X^{-1}abX = ba$. The set of solutions is $\{(ab)^n b^{-1} \mid n \in \mathbb{Z}\}$.

The twisted conjugacy problem can similarly be viewed as the equation $X^{-1}gX = h\Phi$, for some automorphism Φ.

Example 2.6.5. Let $\Phi = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \in \mathrm{GL}_2(\mathbb{Z})$. Consider the twisted equation in $\mathbb{Z}^2$, with the variables $\mathbf{X}$ and $\mathbf{Y}$:

$$(\mathbf{X})\Phi = \mathbf{Y}.$$

This is just the automorphism problem in $\mathbb{Z}^2$, which can be solved using elementary linear algebra. In the free group $F(a,\ b)$, an example of a twisted equation would be $X(Y\psi)aY = bX^{-1}$, for some $\psi \in \mathrm{Aut}(F(a,\ b))$ although computing solutions to this is more difficult.

2.6.2 Solution languages

We now explain how we represent solution sets as languages. We start by defining a normal form.

Definition 2.6.6. Let G be a group, and Σ be a finite generating set for G. A *normal form* for G, with respect to Σ, is a function $\eta \colon G \to (\Sigma^{\pm})^*$ that fixes $\Sigma^{\pm}$, and such that $g\eta$ represents g for all $g \in G$.

A normal form η is called

1. *regular* if $\operatorname{im} \eta$ is a regular language over $\Sigma^{\pm}$;
2. *geodesic* if $\operatorname{im} \eta$ comprises only geodesic words in G, with respect to Σ; that is for all $g \in G$, $|g\eta| = |g|_{(G,\Sigma)}$;
3. *quasi-geodesic* if there exists $\lambda > 0$ such that $|g\eta| \leq \lambda |g|_{(G,\Sigma)} + \lambda$ for all $g \in G$.

Note that we are insisting our normal forms produce a unique representative for each element, since functions can only map elements to one image.

We are now in a position to represent solutions as languages, with respect to a given normal form.

Definition 2.6.7. Let G be a group with a finite inverse closed generating set Σ, and let $\eta \colon G \to (\Sigma^{\pm})^*$ be a normal form for G with respect to Σ. Let $\mathcal{E}$ be a system of equations in G with a set $\mathcal{S}$ of solutions. The *solution language* to $\mathcal{E}$ is the language

$$\{(g_1\eta)\# \cdots \#(g_n\eta) \mid (g_1, \ \ldots, \ g_n) \in \mathcal{S}\}$$

over $\Sigma^{\pm} \sqcup \{\#\}$.

Remark 2.6.8. We now introduce space complexity to solution languages. We first need to define the 'size' of a system of equations, in order to measure our input. The definition of size can vary, as specific groups can have different ways of writing equations. For example, in [47], equations in virtually abelian groups were stored as tuples of integers, as this compressed the size of the equations, whilst storing all of the necessary information. This approach has not always been used in other cases when compression was possible. When we deal with virtually abelian groups on their own, we will use this definition when referring to equations.

When discussing equations in constructions based on other groups (such as direct products, finite index subgroups, wreath products), we will 'inherit' the input definition from the groups they are defined from. If these vary, we will use the general definition, which is less efficient than the specific virtually abelian case, and as a result, will still yield (at least) the same results. The general definition of equation size will also be used when talking about groups that are virtually direct products of hyperbolic groups.

We start with the general definition of equation length.

Definition 2.6.9 (General case)**.** Let G be a group, and $w = 1$ be an equation in G. Recall that $w \in F_V * G$, for some finite set V. Fix a generating set Σ for G. We define the *length* of $w = 1$ to be the length of the group element $w \in F_V * G$, with respect to the generating set $G \cup V$.

Let $\mathcal{E}$ be a finite system of equations in G. The *length* of $\mathcal{E}$ is the sum of the lengths of all equations in $\mathcal{E}$.

Before we define virtually abelian equation length, we must first consider the free abelian case. The compression is possible because we can store an integer n with $\log n + c$ bits, for some constant c. This is covered in greater detail in [47], Remark 3.6 and Remark 3.10.

Definition 2.6.10 (Virtually abelian case)**.** Let $a_1, \ldots, a_k$ denote the standard generators of $\mathbb{Z}^k$.

1. Let $w = 1$ be an equation in $\mathbb{Z}^k$ with a set $\{X_1, \ldots, X_n\}$ of variables. By reordering a given expression for w, we can assume $w = 1$ is in the form

$$X_1^{b_1} \cdots X_n^{b_n} a_1^{c_1} \cdots a_k^{c_k} = 1,$$

where $b_1, \ldots, b_n, c_1, \ldots, c_k \in \mathbb{Z}$. We can then define the *free abelian length* of $w = 1$ to be

$$\sum_{i=1}^{n} \log |b_i| + \sum_{j=1}^{k} \log |c_j| + Ckn.$$

2. Suppose now $u = 1$ is a twisted equation in $\mathbb{Z}^k$. By rearranging u, we can assume it is of the form

$$(X_1 B_1) \cdots (X_n B_n) a_1^{c_1} \cdots a_k^{c_k} = 1,$$

where each $B_r = [b_{rij}]$ is a $k \times k$ integer-valued matrix (not-necessarily invertible). These are described in more detail in the proof of Lemma 3.3 in [47]. The *free abelian length* of $w = 1$ is defined to be

$$\sum_{i,j=1}^{k} \log |b_{rij}| + C'k^2.$$

where C' is a constant.

From [47], any equation $u = 1$ in a virtually abelian group induces a twisted equation $\bar{u} = 1$ in a free abelian group, which is unique up to the choice of transversal. We fix a choice of transversal, then define the *virtually abelian length* of $u = 1$ to be the free abelian length of $\bar{u} = 1$.

Let $\mathcal{E}$ be a finite system of equations in a virtually abelian group. The *virtually abelian length* of $\mathcal{E}$ is the sum of the virtually abelian lengths of all equations in $\mathcal{E}$. *Free abelian length* of a system of equations is defined analogously.

We now use these lengths as our input size. Unless we explicitly state that we are using virtually or free abelian equation length, we will assume we are using the general version of equation length.

Definition 2.6.11. Let $\mathcal{C}$ be a class of languages, and fix a type of machine or grammar that constructs languages in $\mathcal{C}$. Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Let G be a group with a finite generating set Σ, and let η be a normal form for (G, Σ). We say that solutions to systems of equations in G, with respect to η, are $\mathcal{C}$ *in* $\mathsf{NSPACE}(f)$ if

1. The solution languages to systems of equations in G are $\mathcal{C}$ with respect to η;
2. Given a system of equations $\mathcal{E}$ in $\mathcal{G}$, a machine or grammar accepting the

solution language can be constructed in $\mathsf{NSPACE}(f)$, with $\mathcal{E}$ as the input.

Remark 2.6.12. Since the main class of languages we will be using to describe solutions is the class of EDT0L languages, we will usually say *EDT0L in* $\mathsf{NSPACE}(f)$, and the type of grammar we refer to when we say this is the EDT0L system. We use EDT0L, as it is (in most studied cases) the smallest class that works. We have yet to find an example where solution languages can be shown not to be EDT0L (at least in a case when the satisfiability of equations is decidable).

The next well-studied classes of languages containing indexed is context-sensitive. This is too general for results to be particularly interesting, as any group with a context-sensitive word problem will have a context-sensitive solution language to any system of equations. A language is context-sensitive if it is accepted by a Turing machine with a linearly bounded work tape (in terms of the input). Thus any system of equations could be copied onto the tape (leaving space for the variables), which uses a linear amount of space. We can then input our tuple into the Turing machine by writing each word into the gaps we left for tuples, and then solve the word problem. After this we intersect with our (context-sensitive) normal form, to obtain the desired solution language. The class of groups with a context-sensitive word problem is fairly wide; Shapiro showed that it contains all subgroups of automatic groups [90].

Chapter 3

EDT0L languages

3.1 Introduction

In the 1960s, Lindenmayer introduced a collection of classes of languages called *L-systems*. Originally used to study growth of organisms, L-systems saw significant interest in the 1970s and early 1980s, and Lindenmayer's original classes inspired the definitions of many other L-systems, including Rozenberg's EDT0L and ET0L languages [85]. These classes have recently had a wide variety of applications in and around group theory ([20], [14], [17], [19], [47], [32], [65], [11]).

Figure 1.1 gives the relationship between EDT0L, ET0L, regular, context-free and indexed languages (the definitions of context-free, indexed and ET0L languages are not used, but are included in the appendix for the sake of completeness). The facts that regular languages are EDT0L and EDT0L languages are ET0L are immediate from the definitions. Example 3.2.3 gives an example of an EDT0L language that is not context-free. The existence of context-free languages that are not EDT0L was first shown by Ehrenfeucht and Rozenberg [41]. The fact that ET0L languages are indexed is considered in Section B.8, and Ehrenfeucht, Rozenberg and Skyum first showed the existence of an indexed language that is not ET0L [42].

We give an introduction to the class of EDT0L languages and include proofs of some results. Whilst this section is far from an exhaustive survey of these classes of languages, we have included a variety of results, both in this chapter or Appendix B.

We begin with the definition of EDT0L languages in Section 3.2. Section 3.3 covers common alternative definitions and basic closure properties of these classes of languages. Appendix B contains more information on EDT0L languages and the definition of ET0L languages.

3.2 EDT0L languages

We start with the definition of an EDT0L language. Whilst the definition is fairly technical, EDT0L languages are very natural to work with, and a greater understanding of EDT0L languages can be gleaned from an example, such as Example 3.2.3. The original definition is due to Rozenberg [85], however, the use of the rational control, which often makes working with EDT0L languages much easier, is due to Asveld [6].

Definition 3.2.1. An *EDT0L system* is a tuple $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$, where

1. Σ is an alphabet, called the *(terminal) alphabet*;
2. C is a finite superset of Σ, called the *extended alphabet* of $\mathcal{H}$;
3. $\omega \in C^*$ is called the *start word*;
4. $\mathcal{R}$ is a regular (as a language) subset of $\mathrm{End}(C^*)$, called the *rational control* of $\mathcal{H}$.

The language *accepted* by $\mathcal{H}$ is

$$L(\mathcal{H}) = \{\omega\phi \mid \phi \in \mathcal{R}\} \cap \Sigma^*.$$

A language accepted by an EDT0L system is called an *EDT0L language*.

The *alphabet of the rational control* is a minimal set B of labels on a finite-state automaton accepting $\mathcal{R}$ (with respect to cardinality). We say that $\mathcal{H}$ is an *EPDT0L system* if $c\phi \neq \varepsilon$ for all $c \in C$ and $\phi \in B$.

There are a number of different definitions of an EDT0L system, that all generate the same class of languages. In [17] and [20], the definition is the same as given here, except for the insistence that the start word is a single letter. In [86], and many earlier publications, the definition is what is given above, except they only allow rational controls of the form Δ^*, for some finite set of endomorphisms Δ. This definition is again equivalent to the definition we have given [7], but proves to be cumbersome when proving languages are EDT0L. We show the equivalence of these definitions in Section 3.3.

To streamline the definition of specific EDT0L systems, we introduce the following notation convention for specifying endomorphisms of a given free monoid.

Notation 3.2.2. When defining endomorphisms of C^* for some extended alphabet C, within the definition of an EDT0L system, we will usually define each endomorphism by where it maps each letter in C. If any letter is not assigned an image within the definition of an endomorphism, we will say that it is fixed by that endomorphism.

The following is a standard example of an EDT0L language that is not context-free.

Example 3.2.3. The language $L = \{a^{n^2} \mid n \in \mathbb{Z}_{>0}\}$ is an EDT0L language over the alphabet $\{a\}$. This can be seen by considering $C = \{\perp,\ s,\ t,\ u,\ a\}$ as the extended alphabet of an EDT0L system accepting L, with $\perp$ as the start word, and using the finite state automaton from Figure 3.1 to define the rational control. Note that the rational control can also be written as $\varphi_\perp(\varphi_1\varphi_2)^*\varphi_3$.

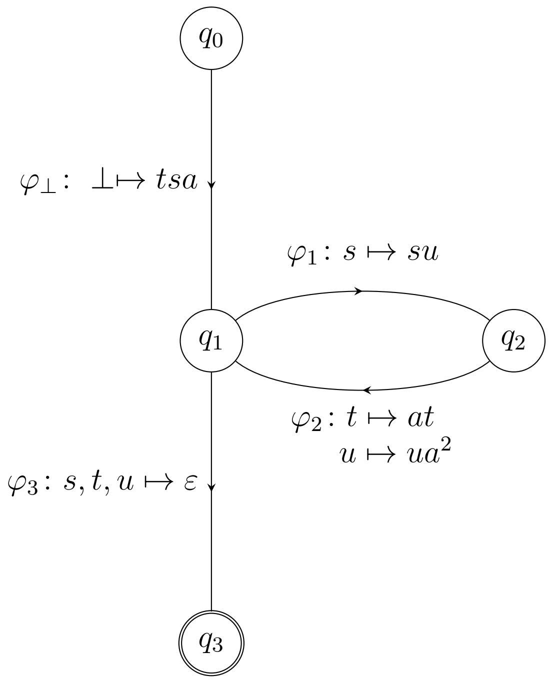
Since we devote a lot of time to EDT0L solution languages in future chapters, we give an example of an equation in a group with an EDT0L language of solutions.

Example 3.2.4. Consider the equation $XY^{-1} = 1$ in $\mathbb{Z}$ with the presentation $\langle a \mid \rangle$. The solution language with respect to the standard normal form is

$$L = \{a^n \# a^n \mid n \in \mathbb{Z}\},$$

over the alphabet $\{a,\ a^{-1},\ \#\}$. The language L is EDT0L; our system will have the extended alphabet $\{\perp,\ \#,\ a,\ a^{-1}\}$ and rational control defined by Figure 3.2.

Figure 3.1: Rational control for $L = \{a^{n^2} \mid n \in \mathbb{Z}_{>0}\}$, with start state q_0 and accept state q_3.



Note that the rational control can also be expressed using the rational expression $\{\varphi_-^*,\ \varphi_+^*\}\phi$.

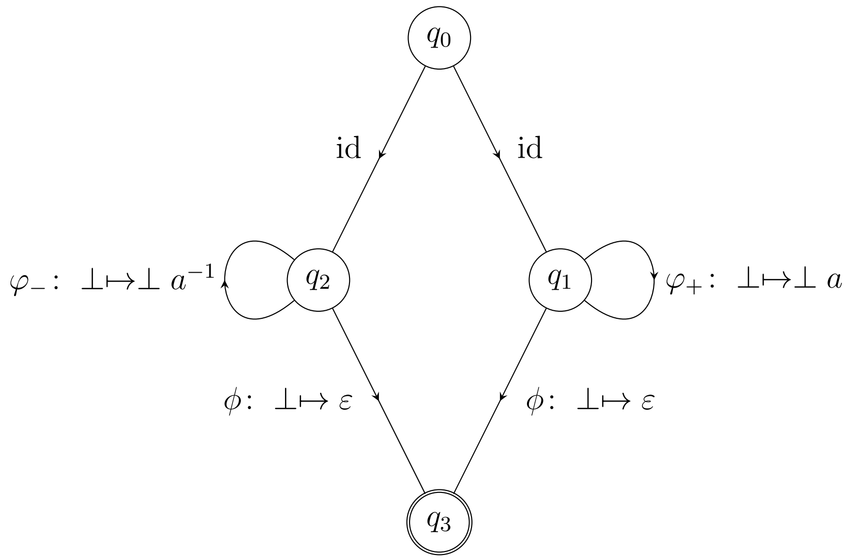
This language is not regular, which can be shown using the pumping lemma ([56], Theorem 2.5.17)

3.3 Alternative definitions and closure properties

We now consider some of the basic properties of EDT0L languages. We start with a theorem that gives a number of restrictions that can be put on EDT0L (and ET0L) systems, without affecting the class of languages they define. Frequently, EDT0L and ET0L systems are defined with these restrictions in place. In much of the literature during the 1970s, the restriction (3) is used as part of the definition. We refer the reader to Section B.1 for the definition of E(P)T0L languages.

Theorem 3.3.1. *The class of languages accepted by E(P)(D)T0L systems is unchanged if one assumes:*

Figure 3.2: Rational control for $L = \{a^n \# a^n \mid n \in \mathbb{Z}\}$ with start state q_0, and accept state q_3.



1. *The start word is a single letter;*
2. *Every table in the alphabet over which the rational control is a regular language fixes every letter in the terminal alphabet.*
3. *The rational control is of the form B^* for some finite set B of endomorphisms;*
4. *All of the above.*

In addition, one can switch between definitions, without affecting the space complexity of the systems.

Proof We have that if all E(P)(D)T0L languages are accepted by E(P)(D)T0L systems satisfying (4), it follows that all E(P)(D)T0L languages are accepted by E(P)(D)T0L systems satisfying (1), (2) and (3). Moreover, as they are still E(D)T0L systems, the languages accepted by systems satisfying (1), (2) and (3) will still be E(D)T0L. Thus it suffices to show that every E(P)(D)T0L language is accepted by a system satisfying (4).

Let L be a language accepted by an E(P)(D)T0L system $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$. Let $\perp \notin C$, and extend each table ϕ over C to a table $\bar{\phi}$ over $C \cup \{\perp\}$ by setting $\perp \bar{\phi} = \{\perp\}$. Define a table (which induces an endomorphism) ψ over $C \cup \{\perp\}$ by $\perp \psi = \{\omega\}$, and $c\psi = \{c\}$ for all $c \in C$. We have that $(\Sigma,\ C \cup \{\perp\},\ \perp,\ \psi\mathcal{R})$

accepts L.

For the space complexity, first note that $\perp$ can be written down using constant space. As C is constructible in $\mathsf{NSPACE}(f)$, we have that $C \cup \{\perp\}$ is constructible in constant space. To record ψ we need only the information required to record $C \cup \{\perp\}$ and then ω, all of which are constructible in $\mathsf{NSPACE}(f)$. It follows that $\psi\mathcal{R}$ is constructible in $\mathsf{NSPACE}(f)$, and thus the E(P)(D)T0L system is.

By redefining C to be $C \cup \{\perp\}$ and $\mathcal{R}$ to be $\mathcal{R}\psi$, we can now assume that $\mathcal{H}$ is of the form $(\Sigma,\ C,\ \perp,\ \mathcal{R})$.

Let B be the finite set of tables that is the alphabet of $\mathcal{R}$. We will create a new E(P)(D)T0L system from $\mathcal{H}$ so that every table in B fixes every letter in the target alphabet. Let $\bar{C} = \{\bar{c} \mid c \in C\}$ be a disjoint copy of C. For each $\phi \in B$ define the table $\bar{\phi}$ over $\bar{C} \cup \Sigma$ by

$$c\bar{\phi} = \begin{cases} \overline{d\phi} & c = \bar{d} \in \bar{C} \\ \{c\} & c \in \Sigma. \end{cases}$$

Also define θ by

$$c\theta = \begin{cases} \{c\} & c \notin \bar{\Sigma} \\ \{b\} & c = \bar{b} \in \bar{\Sigma}. \end{cases}$$

Let $\bar{\mathcal{R}}$ be the rational subset of $(\mathrm{End}(\bar{C} \cup \Sigma)^*)$ obtained by replacing each ϕ in B within a finite-state automaton for $\mathcal{R}$ with its barred version. By construction, the E(P)(D)T0L system $(\Sigma,\ \bar{C} \cup \Sigma,\ \bar{\omega},\ \bar{\mathcal{R}}\theta)$ accepts L.

We now show that this E(P)(D)T0L system is constructible in $\mathsf{NSPACE}(f)$. As C and Σ are constructible in $\mathsf{NSPACE}(f)$, it follows that $\bar{C} \cup \Sigma$ is also. Constructing a barred version of ω can be done in the same space required to construct ω. Since each barred version of a table in $\mathcal{R}$ requires the same space to construct, it follows that $\bar{\mathcal{R}}$, and hence $\bar{\mathcal{R}}\theta$ can be constructed in $\mathsf{NSPACE}(f)$.

We can now assume that $\mathcal{H}$ satisfies (1) and (2), and we will use this to define an E(P)(D)T0L system that satisfies (1), (2) and (3). Let $\mathcal{A} = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$ be a finite state automaton for $\mathcal{R}$. Let $C_{\text{ind}} = \{c^q \mid c \in C,\ q \in Q\} \cup \Sigma \sqcup \{\kappa\}$, where κ is a symbol not already used, and for each $c \in C$, c^q is a distinct new letter for all

$q \in Q$. We will use κ as a 'fail symbol'. For each transition in $\mathcal{A}$ from a state p_1 to a state p_2, labelled with $\phi \in B$, define a table $\hat{\phi}$ over C_{ind} by

$$c^q\hat{\phi} = \begin{cases} \{d^{p_2} \mid d \in c\phi\} & p_1 = q \\ \{c^q\} & p_1 \neq q \\ \{\kappa\} & c^q = \kappa. \end{cases}$$

Let $B_{\text{ind}} = \{\hat{\phi} \mid \phi \in B\}$, and $\Sigma_{\text{ind}} = \{a^q \mid a \in \Sigma,\ q \in F\}$. By construction, the E(P)(D)T0L system $(\Sigma_{\text{ind}},\ C_{\text{ind}},\ \perp^{q_0},\ B^*_{\text{ind}})$ accepts the language

$$M = \{a_1^q \cdots a_n^q \mid a_i \in \Sigma,\ q \in F,\ a_1 \cdots a_n \in L\}.$$

Now define $\theta \in \text{End}(C^*_{\text{ind}})$ by

$$c^q\theta = \begin{cases} c & \{c\} \in \Sigma,\ q \in F \\ \{\kappa\} & \text{otherwise.} \end{cases}$$

Thus L is accepted by the E(P)(D)T0L system $(\Sigma,\ C_{\text{ind}},\ \perp^{q_0},\ (B_{\text{ind}} \cup \{\theta\})^*)$. $\square$

The following theorem shows that even if EDT0L languages do not form a full abstract family of languages like regular, context-free and ET0L languages, they are closed under most of the standard operations that are frequently used to manipulate languages.

Theorem 3.3.2 ([86], Theorem V.1.7 and Exercise IV.3.2; [19], Proposition 3.3)**.** *The class of EDT0L languages is closed under the following operations:*

1. *Finite unions;*
2. *Intersection with regular languages;*
3. *Concatenation;*
4. *Kleene star closure;*
5. *Image under free monoid homomorphisms.*

The class of ET0L languages is closed under the above operations, together with:

6 *Pre-image under free monoid homomorphisms.*

Applying any of the operations will not affect the space complexity that systems for the languages involved can be constructed in, assuming that the regular language in (3), and the homomorphisms in (5) and (6) are constructible in constant space.

Proof Let $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$.

1. Let L and M be languages accepted by the E(D)T0L systems $(\Sigma_L,\ C_L,\ \omega_L,\ \mathcal{R}_L)$ and $(\Sigma_M,\ C_M,\ \omega_M,\ \mathcal{R}_M)$, which are constructible in $\mathsf{NSPACE}(f)$. Fix a symbol $\perp \notin C_L \cup C_M$. Let $\Sigma = \Sigma_L \cup \Sigma_M$ (this union need not be disjoint) and $C = C_L \cup C_M \cup \{\perp\}$. Let ϕ_L and ϕ_M be tables over C defined by $\perp \phi_L = \{\omega_L\}$ and $\perp \phi_M = \{\omega_M\}$, which fix all other letters in C. Extend the tables in the alphabets of $\mathcal{R}_L$ and $\mathcal{R}_M$ to C^* by fixing all letters in $C \backslash C_L$ and $C \backslash C_M$, respectively. Let $\mathcal{R} = (\phi_L \mathcal{R}_L) \cup (\phi_M \mathcal{R}_M)$. Note $\mathcal{R}$ is rational. By construction, $(\Sigma,\ C,\ \perp,\ \mathcal{R})$ accepts $L \cup M$, and so $L \cup M$ is E(D)T0L.
   As we can write down C_L and C_M in $\mathsf{NSPACE}(f)$, we can also write down $C = C_L \cup C_M \cup \{\perp\}$ in $\mathsf{NSPACE}(f)$. The start word is a single letter, and thus can be constructed in constant space. To show the rational control can be constructed in $\mathsf{NSPACE}(f)$, it is sufficient to show that the tables ϕ_L and ϕ_M can be, as $\mathcal{R}_L$ and $\mathcal{R}_M$ are constructible in $\mathsf{NSPACE}(f)$. However, to record these tables, we need only the information to construct ω_L and ω_M, both of which are constructible in $\mathsf{NSPACE}(f)$.
2. Let $\mathcal{A}$ be a finite-state automaton with a single accept state q_t, accepting a regular language L over Σ. Let Q be the set of states of $\mathcal{A}$, $\delta \subseteq (Q \times \Sigma) \times Q$ be the transition function and q_0 be the start state. Let $(\Sigma,\ C,\ \perp,\ \mathcal{R})$ be an E(D)T0L system accepting a language M with a single-letter start word, that is constructible in $\mathsf{NSPACE}(f)$. We can assume the start word is a single letter by Theorem 3.3.1. For each $c \in C$ and each $p,\ q \in Q$, define a distinct symbol $c^{p,q} \notin C$. Let κ be a symbol not already used, which we will use as a 'fail symbol' Let $D = \{c^{p,q} \mid p,\ q \in Q,\ c \in C\} \cup \{\kappa\} \cup \Sigma$. For each letter ϕ in the alphabet of the rational control $\mathcal{R}$, define the finite set Φ_ϕ of tables over

D to be the set of all $\theta_{\phi, p_1, \ldots, p_{k-1}}$ defined as follows:

$$\theta_{\phi, p_1, \ldots, p_{k-1}} \colon c^{p,q} \mapsto \{d_1^{p,p_1} d_2^{p_1,p_2} \cdots d_k^{p_{k-1},q} \mid d_1 \cdots d_k \in c\phi\}$$

$$\text{for all } c \in C,\ p,\ q,\ p_1,\ \ldots,\ p_{k-1} \in Q.$$

Define a new rational subset $\bar{\mathcal{R}} \subseteq \mathrm{End}(D^*)$ by replacing each occurrence of each letter ϕ in a rational expression for $\mathcal{R}$ with Φ_ϕ. Let

$$\psi \colon a^{p,q} \mapsto \begin{cases} \{a\} & ((p,\ a),\ q) \in \delta \\ \{\kappa\} & ((p,\ a),\ q) \notin \delta \end{cases} \quad \text{for all } a \in \Sigma.$$

By construction, the E(D)T0L system $\mathcal{H} = (\Sigma,\ D,\ \perp^{q_0,q_t},\ \bar{\mathcal{R}}\psi)$ accepts $L \cap M$. It remains to show that $\mathcal{H}$ is constructible in $\mathsf{NSPACE}(f)$. As we can choose $\mathcal{A}$ such that it is constructible in constant space, doing so allows us to construct C_{ind} in $\mathsf{NSPACE}(f)$. Note that we can construct each Φ_ϕ using the same information required to construct ϕ, and thus $\bar{\mathcal{R}}$ is constructible in $\mathsf{NSPACE}(f)$. Finally, note that ψ can be written down in constant space. We can conclude that $\mathcal{H}$ is constructible in $\mathsf{NSPACE}(f)$.

3. Let L and M be languages accepted by E(D)T0L systems $(\Sigma_L,\ C_L,\ \omega_L,\ \mathcal{R}_L)$ and $(\Sigma_M,\ C_M,\ \omega_M,\ \mathcal{R}_M)$ that are constructible in $\mathsf{NSPACE}(f)$. Let $K = \{uv \mid u \in L,\ v \in M\}$ be the concatenation. By Theorem 3.3.1, we can assume that the tables in the alphabets of $\mathcal{R}_L$ and $\mathcal{R}_M$ fix Σ_L and Σ_M, respectively. Let $\Sigma = \Sigma_L \cup \Sigma_M$, and $C = (C_L \backslash \Sigma_L) \sqcup (C_M \backslash \Sigma_M) \sqcup \Sigma$. As in (1), extend tables in the alphabets of $\mathcal{R}_L$ and $\mathcal{R}_M$ to become tables over C by fixing letters in $C \backslash C_M$ and $C \backslash C_L$, respectively. We can now define a new E(D)T0L system $\mathcal{H} = (\Sigma,\ C,\ \omega_L\omega_M,\ \mathcal{R}_L \cup \mathcal{R}_M)$ which accepts K.
   For the space complexity, as ω_L and ω_M can both be written down in $\mathsf{NSPACE}(f)$, it follows that $\omega_L\omega_M$ can. Similarly, as C_L, C_M, Σ, $\mathcal{R}_L$ and $\mathcal{R}_M$ are all constructible in $\mathsf{NSPACE}(f)$, it follows that the same holds for $C = (C_L \backslash \Sigma_L) \sqcup (C_M \backslash \Sigma_M) \sqcup \Sigma$ and $\mathcal{R}_L \cup \mathcal{R}_M$.
4. Let L be a language accepted by an E(D)T0L system $(\Sigma,\ C,\ \omega,\ \mathcal{R})$ that is constructible in $\mathsf{NSPACE}(f)$. We will use Theorem 3.3.1 to assume tables in $\mathcal{R}$ fix elements of Σ. Let $\perp \notin C$. Define tables ϕ and ψ over $C \cup \{\perp\}$ by

$\perp \phi = \{\omega \perp\}$ and $\perp \psi = \{\varepsilon\}$. For each take $\theta \in \mathcal{R}$, define $\bar{\theta}$ by $c\bar{\theta} = c\theta$ if $c \in C$, and $\perp \theta = \{\perp\}$. Let $\bar{\mathcal{R}} = \{\bar{\theta} \mid \theta \in \mathcal{R}\}$. It follows that the E(D)T0L system $(\Sigma,\ C \cup \{\perp\},\ \perp,\ (\phi\bar{\mathcal{R}})^*\psi)$ accepts L^*.

Note that we can write down $\perp$ and ψ in constant space. Since C and $\mathcal{R}$ are constructible in $\mathsf{NSPACE}(f)$, so are $C \cup \{\perp\}$ and $(\phi\bar{\mathcal{R}})^*\psi$.

5. Let L be a language accepted by an E(D)T0L system $(\Sigma,\ C,\ \omega,\ \mathcal{R})$ which is constructible in $\mathsf{NSPACE}(f)$, and let $\phi\colon \Sigma^* \to \Delta^*$ be a monoid homomorphism, where Δ is an alphabet. Let $\bar{\phi}$ be the table over $C \cup \Delta$ induced by the action of ϕ on Σ (and assumed to fix all letters in $(C \cup \Delta)\backslash\Sigma$). We have that $(\Delta,\ C \cup \Delta,\ \omega,\ \mathcal{R}\bar{\phi})$ accepts $L\phi$.

   Since ϕ (and hence Δ) are constructible in constant space, and C and $\mathcal{R}$ are constructible in $\mathsf{NSPACE}(f)$, we have that $C \cup \Delta$ and $\mathcal{R}\bar{\phi}$ are constructible in $\mathsf{NSPACE}(f)$.

6. Let L be a language accepted by an ET0L system $(\Sigma,\ C,\ \omega,\ \mathcal{R})$, constructible in $\mathsf{NSPACE}(f)$, and let $\phi\colon \Delta^* \to \Sigma^*$ be a homomorphism, where Δ is an alphabet. Let $\hat{\Delta} = \{\hat{a} \mid a \in \Delta\}$ be a copy of Δ disjoint from C. Consider the following language over $\Sigma \cup \hat{\Delta}$:

$$K = \{y_0 a_1 y_1 \cdots a_k y_k \mid a_1,\ \ldots,\ a_k \in \Sigma,\ a_1 \cdots a_k \in L,\ y_1,\ \ldots,\ y_k \in \hat{\Delta}^*\}.$$

   We will define an ET0L system for L that is constructible in $\mathsf{NSPACE}(f)$. Our extended alphabet will be $C \cup \bar{\Delta}$. Define the table θ over $C \cup \hat{\Delta}$ by

$$c\phi = \begin{cases} \{ycz \mid y,\ z \in \hat{\Delta} \cup \{\varepsilon\}\} & c \in \Sigma \\ \{c\} & c \notin \Sigma. \end{cases}$$

   For each $\psi \in \mathcal{R}$, let $\bar{\psi}$ be the table over $C \cup \hat{\Delta}$ obtained by extending ψ to fix elements of $\hat{\Delta}$, and let $\bar{\mathcal{R}} = \{\bar{\psi} \mid \psi \in \mathcal{R}\}$. Then $(\Sigma \cup \hat{\Delta},\ C \cup \hat{\Delta},\ \omega,\ \bar{\mathcal{R}}\theta^*)$ is an ET0L system for K. Moreover, as θ is only dependent on Δ and C, θ is constructible in $\mathsf{NSPACE}(f)$. Thus $\bar{\mathcal{R}}\theta^*$ is also constructible in $\mathsf{NSPACE}(f)$, and so the ET0L system for K is.

   Now consider the regular language $S = \{(u\phi)\hat{u} \mid u \in \Delta\}^*$. Note that we can construct a finite-state automaton accepting $\{(u\phi)\hat{u} \mid u \in \Delta\}$ is constant space, as we only need to remember the information required to construct ϕ. Thus a finite-state automaton for S is constructible in constant space. We can therefore use (2) to show that $K \cap S$ is accepted by an ET0L system that is constructible in $\mathsf{NSPACE}(f)$.

Consider the monoid homomorphism $\xi \colon (\Sigma \cup \hat{\Delta})^* \to \Sigma$ defined by

$$a\xi = \begin{cases} \varepsilon & a \in \Sigma \\ b & a = \hat{b} \in \hat{\Delta} \end{cases}$$

Then $(K \cap S)\xi = L\phi^{-1}$. Moreover, as it is only dependent on Σ and Δ, ξ is constructible in constant space. Thus by (5), an ET0L system for $L\phi^{-1}$ is constructible in $\mathsf{NSPACE}(f)$.

□

Chapter 4

Equations in virtually abelian groups

4.1 Introduction

This chapter is based on joint work with Alex Evetts [47], and the author's paper [65].

It has long been regarded as 'folklore' that it is decidable whether systems of equations in virtually abelian groups admit solutions, however it is unclear when this was first proved. In [44] the stronger result that virtually abelian groups have decidable first order theory is shown. A more direct proof of the solubility of equations in virtually abelian groups can be found in Lemma 5.4 of [26]. In this chapter we study the properties of solution sets of systems of equations in finitely generated virtually abelian groups. Such sets are also known as *algebraic sets*.

Given a choice of finite generating set, and a corresponding normal form, we study the language of representatives for algebraic sets. These will be called solution languages (see Definition 4.2.15). In Section 4.3 we show that the solution languages (with respect to a suitable generating set and normal form) are EDT0L. This will be a consequence of the stronger result that they are accepted by *multivariable finite-state automata* (see Definition 4.2.9):

Theorem 4.3.16. *The solution language to any system of equations with rational constraints in a virtually abelian group is accepted by a multivariable finite-state automaton.*

Corollary 4.3.17. *The solution language to any system of equations with rational constraints in a virtually abelian group is EDT0L.*

We also show that both the multivariable finite-state automata and the EDT0L systems that accept these solution languages can be constructed in non-deterministic quadratic space.

It is a standard fact that every regular language has rational growth series. That is, the generating function which counts the number of words in the language with increasing length lies in the ring of rational functions $\mathbb{Q}(z)$. A result of Chomsky and Schützenberger [15] asserts that the growth series of every unambiguous context-free language is algebraic over $\mathbb{Q}(z)$. In contrast, there is no reason to expect that those EDT0L languages which do not fall under these two cases have well-behaved growth series. Indeed, Corollary 8 of [20] implies that there are EDT0L languages with transcendental (i.e. non-algebraic) growth series. A priori, the language obtained in Corollary 4.3.17 is neither regular nor context-free. Nevertheless, we prove that its growth series is rational.

Proposition 4.3.20. *The solution language to any system of equations in a virtually abelian group has rational growth series.*

Algebraic sets in groups can be seen as an analogue of the fundamental notion of algebraic varieties – the zero-loci of systems of equations. Meuser [73], and later Denef [29], proved the rationality of the Poincaré series of varieties over the p-adic integers, which can be thought of as a form of growth series. In Section 4.4 we prove an analogous result for algebraic sets of virtually abelian groups, using a notion of growth appropriate to the setting of finitely generated groups, namely word growth. We will use the notion of a *polyhedral set*, which has its roots in the model theory of Presburger (see Section 4.2 for definitions).

Word growth in finitely generated groups is a much-studied topic. The growth function counts the number of group elements of length n, with respect to the metric arising from a choice of finite generating set. The asymptotics of this function are well understood, but many questions remain about the properties of the corresponding formal power series. For an introduction to the topic, the reader is directed to Mann's book [70].

Any subset of a group has a growth function, inherited from the group itself. This *relative growth* has been studied in many papers, including [28]. The relative growth series of any subgroup of a virtually abelian group was shown to be rational in [45]. In Section 4.4 we consider the relative growth of the algebraic sets of a virtually abelian group, as sets of tuples of group elements (with an appropriate metric). We show that the growth series of an algebraic set is always a rational function, regardless of the choice of finite weighted generating set.

Theorem 4.4.3. *Let G be a virtually abelian group. Then every algebraic set of G has rational weighted growth series with respect to any finite generating set.*

Moreover, we consider the natural *multivariate* growth series of the algebraic set, and demonstrate how recent results of Bishop imply that this series is holonomic (a class which includes algebraic functions and some transcendental functions).

Corollary 4.4.21. *Every algebraic set of a virtually abelian group has holonomic weighted multivariate growth series.*

We note that it may be useful for other purposes to have an explicit description of the algebraic sets of the groups in question, since this does not appear cleanly in the proofs. For such a statement, the interested reader is directed to Corollary 4.4.16, where the general structure of algebraic sets is noted, using the terminology of polyhedral sets.

4.2 Preliminaries

In this section we lay out the key definitions and basic results that will be required for the rest of the chapter.

Notation 4.2.1. If $w \in S^*$ is a word in the generators of some group G, we write $\overline{w} \in G$ for the group element that the word w represents.

4.2.1 Polyhedral sets

Our fundamental tool for proving that languages of representatives have rational growth series in Proposition 4.3.20 and Section 4.4 will be the theory of *polyhedral sets*. These ideas appear in model theory as early as Presburger [78]. Results regarding rationality can be found in [29], and the ideas also appear in the theory of Igusa local zeta functions (see [24]). The following definitions and results follow Benson's work [8], where it is proved that virtually abelian groups have rational (standard) growth series. More recently, polyhedral sets have again been used to prove rationality of various growth series of groups ([35], [45]).

Definition 4.2.2. Let $r \in \mathbb{Z}_{>0}$, and let $\cdot$ denote the Euclidean scalar product. Then we define the following.

1. Any subset of $\mathbb{Z}^r$ of the form $\{\mathbf{z} \in \mathbb{Z}^r \mid \mathbf{u} \cdot \mathbf{z} = a\}$, $\{\mathbf{z} \in \mathbb{Z}^r \mid \mathbf{u} \cdot \mathbf{z} > a\}$, or $\{\mathbf{z} \in \mathbb{Z}^r \mid \mathbf{u} \cdot \mathbf{z} \equiv a \mod b\}$, for any $\mathbf{u} \in \mathbb{Z}^r$, $a \in \mathbb{Z}$, $b \in \mathbb{Z}_{>0}$, will be called an *elementary set*;
2. any finite intersection of elementary sets will be called a *basic polyhedral set*;
3. any finite union of basic polyhedral sets will be called a *polyhedral set*.

If $\mathcal{P} \subset \mathbb{Z}^r$ is polyhedral and additionally no element contains negative coordinate entries, we call $\mathcal{P}$ a *positive polyhedral set*.

It is not hard to prove the following closure properties.

Proposition 4.2.3 (Proposition 13.1 and Remark 13.2 of [8]). *Let $\mathcal{P}$, $\mathcal{Q} \subseteq \mathbb{Z}^r$ and $\mathcal{R} \subseteq \mathbb{Z}^s$ be polyhedral sets for some positive integers r and s. Then the following are* *also polyhedral:* $\mathcal{P} \cup \mathcal{Q} \subseteq \mathbb{Z}^r$, $\mathcal{P} \cap \mathcal{Q} \subseteq \mathbb{Z}^r$, $\mathbb{Z}^r \setminus \mathcal{P}$, $\mathcal{P} \times \mathcal{R} \subseteq \mathbb{Z}^{r+s}$.

Benson also shows that polyhedral sets behave well under affine transformations, as follows.

Definition 4.2.4. We call a map $\mathcal{A} \colon \mathbb{Z}^r \to \mathbb{Z}^s$ an *integral affine transformation* if there exists an $r \times s$ matrix M with integer entries and some $\mathbf{q} \in \mathbb{Z}^s$ such that $\mathbf{p}\mathcal{A} = \mathbf{p}M + \mathbf{q}$ for $\mathbf{p} \in \mathbb{Z}^r$.

Proposition 4.2.5 (Propositions 13.7 and 13.8 of [8]). *Let* $\mathcal{A}$ *be an integral affine transformation. If* $\mathcal{P} \subseteq \mathbb{Z}^r$ *is a polyhedral set then* $\mathcal{P}\mathcal{A} \subseteq \mathbb{Z}^s$ *is a polyhedral set. If* $\mathcal{Q} \subseteq \mathbb{Z}^s$ *is a polyhedral set then the preimage* $\mathcal{Q}\mathcal{A}^{-1} \subseteq \mathbb{Z}^r$ *is a polyhedral set.*

We note that projection onto any subset of the coordinates of $\mathbb{Z}^r$ is an integral affine transformation.

Notation 4.2.6. We will now introduce weight functions. When talking about weighted lengths of elements of free abelian or virtually abelian groups, we will use $\| \cdot \|$ instead of $| \cdot |$, which will be used for 'standard' length of elements.

Let $\mathcal{P} \subseteq \mathbb{Z}^r$ be a polyhedral set. Given some choice of weight function $\|\mathbf{e}_i\| \in \mathbb{Z}_{>0}$ for the standard basis vectors $\{\mathbf{e}_i\}_{i=1}^r$ of $\mathbb{Z}^r$, we assign the weight $\sum_{i=1}^r a_i \|\mathbf{e}_i\|$ to the element $(a_1, \ldots, a_r) \in \mathcal{P}$. Define the spherical growth function

$$\sigma_{\mathcal{P}}(n) = \#\{\mathbf{p} \in \mathcal{P} \mid \|\mathbf{p}\| = n\},$$

and the resulting weighted growth series

$$\mathbb{S}_{\mathcal{P}}(z) = \sum_{n=0}^{\infty} \sigma_{\mathcal{P}}(n) z^n.$$

Our argument will rely on the following crucial proposition.

Proposition 4.2.7 (Proposition 14.1 of [8], and Lemma 7.5 of [29]). *If* $\mathcal{P}$ *is a positive polyhedral set, then the weighted growth series* $\mathbb{S}_{\mathcal{P}}(z)$ *is a rational function of* z.

We will need the following more general result.

Corollary 4.2.8. *Let $\mathcal{P} \subset \mathbb{Z}^r$ be any polyhedral set (not necessarily positive). Then the weighted growth series $\mathbb{S}_{\mathcal{P}}(z)$ is a rational function of z.*

Proof We show that $\mathcal{P}$ may be expressed as a disjoint union of polyhedral sets, each in weight-preserving bijection with a positive polyhedral set. Let $\mathcal{Q}_1 = \{\mathbf{z} \in \mathbb{Z}^r \mid \mathbf{z} \cdot \mathbf{e}_i \geq 0,\ 1 \leq i \leq r\} = \bigcap_{i=1}^r \{\mathbf{z} \in \mathbb{Z}^r \mid \mathbf{z} \cdot \mathbf{e}_i \geq 0\}$ denote the non-negative orthant of $\mathbb{Z}^r$, and note that it is polyhedral. Let $\mathcal{Q}_2, \ \ldots, \ \mathcal{Q}_{2^r}$ denote the remaining orthants (in any order) obtained from $\mathcal{Q}_1$ by (compositions of) reflections along hyperplanes perpendicular to the axes and passing through the origin. By Proposition 4.2.5 these are also polyhedral sets. Let $\mathcal{P}_1 = \mathcal{P} \cap \mathcal{Q}_1$ and for each $2 \leq j \leq 2^r$, inductively define

$$\mathcal{P}_j = \left(\mathcal{P} \setminus \bigcup_{k<j} \mathcal{P}_k \right) \cap \mathcal{Q}_j.$$

Each $\mathcal{P}_j$ is a polyhedral set by Proposition 4.2.3, and we have a disjoint union $\mathcal{P} = \bigcup_{j=1}^{2^r} \mathcal{P}_j$. Each $\mathcal{P}_j$ is in weight-preserving bijection with a *positive* polyhedral set (by compositions of reflections along hyperplanes) and so $\mathbb{S}_{\mathcal{P}_j}(z)$ is rational. The result follows since $\mathbb{S}_{\mathcal{P}}(z) = \sum_{j=1}^{2^r} \mathbb{S}_{\mathcal{P}_j}(z)$. □

4.2.2 Multivariable finite-state automata

Since solutions to equations can be thought of as tuples, one method that can be used to study the language complexity of sets of solutions is using multivariable languages, which are sets of tuples of words over an alphabet. We start with the formal definition.

Definition 4.2.9. Let Σ be an alphabet, and $n \in \mathbb{Z}_{>0}$. An *n-variable word* over Σ is an element of the Cartesian product $(\Sigma^*)^n$, and an *n-variable language* over Σ is any subset of $(\Sigma^*)^n$.

We continue with a generalisation of a finite-state automaton to accept n-variable languages, for some positive integer n: the (asynchronous, non-deterministic) n-variable finite-state automaton.

Definition 4.2.10. Let $n \in \mathbb{Z}_{>0}$. An *n-variable finite-state automaton* is a tuple $\mathcal{A} = (\Sigma,\ \Gamma,\ q_0,\ F)$, where

1. Σ is an alphabet;
2. Γ is a finite edge-labelled graph, where labels are n-variable words in $(\Sigma^*)^n$, with at most one non-empty word entry. The vertices of Γ are called *states*;
3. $q_0 \in V(\Gamma)$ is called the *start state*;
4. $F \subseteq V(\Gamma)$ is called the set of *accept states.*

When tracing a path in Γ, we trace an n-variable word to be the concatenation of the labels of each edge traversed. Since each edge has at most one non-empty entry, the word will only get longer in one coordinate at a time. An n-variable word $\mathbf{w} \in (\Sigma^*)^n$ is *accepted* by $\mathcal{A}$ if there is a path γ in Γ from q_0 to a state in F, such that the n-variable word obtained by reading the labels in γ is $\mathbf{w}$. The *language accepted* by $\mathcal{A}$ is the set of all n-variable words accepted by $\mathcal{A}$.

Remark 4.2.11. Languages accepted by n-variable finite-state automata can equivalently be defined as rational subsets of $(\Sigma^*)^n$: the direct product of the free monoid with itself n times.

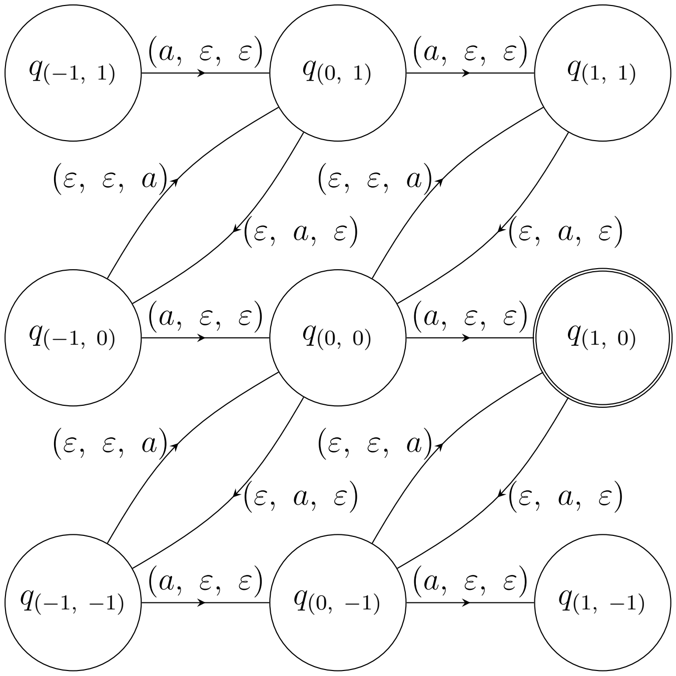
We give an example of a language accepted by a 3-variable finite-state automaton. In this case, the language represents the set of solutions to a system of equations in $\mathbb{Z}$.

Example 4.2.12. Let $\mathcal{E}$ be the following system of equations in $\mathbb{Z}$ (using additive notation):

$$X - Y + Z = 1 \qquad\qquad -Y + Z = 0.$$

Note that by subtracting the second equation from the first, it is not difficult to show that the set of solutions to this system is $\{(1,\ y,\ y) \mid y \in \mathbb{Z}\}$. To demonstrate a more general method we will use later on, we will construct the set of solutions, and show that $L = \{(a^x,\ a^y,\ a^z) \mid (x,\ y,\ z) \text{ is a solution to } \mathcal{E}\}$ is accepted by a 3-variable finite-state automaton over the alphabet $\{a,\ a^{-1}\}$, using a different method. We will show

Figure 4.1: The start state is $q_{(0,\ 0)}$, and $q_{(1,\ 0)}$ is the unique accept state.



1. The language $\{(a^x,\ a^y,\ a^z) \mid (x,\ y,\ z)$ is a solution to $\mathcal{E}$ and $x,\ y,\ z \geq 0\}$ is accepted by a 3-variable finite-state automaton;
2. To show L is accepted by a 3-variable finite-state automaton, we take the finite union across the possible configurations of signs of X, Y and Z and use the fact that finite unions of languages accepted by 3-variable finite-state automata are also accepted by 3-variable finite-state automata.

To show (1), consider the 3-variable finite-state automaton in Figure 1. This finite-state automaton works as follows:

1. Traversing an edge labelled by $(a,\ \varepsilon,\ \varepsilon)$, $(\varepsilon,\ a,\ \varepsilon)$ or $(\varepsilon,\ \varepsilon,\ a)$ corresponds to increasing x, y or z by 1, respectively. The states $q_{(i,\ j)}$ correspond to the value of $(x - y + z,\ -y + z)$, with the current values of x, y and z.
2. Once we have increased x, y and z to the desired values, if this is a solution to $\mathcal{E}$, then we must be in the accept state $q_{(1,\ 0)}$.
3. Note that we cannot increase the xs, ys and zs in any order, otherwise we would need an unbounded size of FSA. For example, the element $(a,\ a^l,\ a^l)$ of L, where $l \in \mathbb{Z}$ and $l > 1$ cannot be reached in the above system by traversing

one $(a,\ \varepsilon,\ \varepsilon)$ edge, then l $(\varepsilon,\ a,\ \varepsilon)$ edges, and then l $(\varepsilon,\ \varepsilon,\ a)$ edges, as after the l $(\varepsilon,\ a,\ \varepsilon)$ edges we would need a state $q_{(-l+1,\ -l)}$, which does not lie in the finite-state automaton. Moreover, we cannot add them to the finite-state automaton, as there are infinitely many such states. We prove the existence of an ordering of the edges (up to considering two edges with the same label equivalent) that works in Lemma 4.3.2. In this specific case, it is not hard to show that the ordering that starts with $(a,\ \varepsilon,\ \varepsilon)$, followed by l traversals of a path comprising one $(\varepsilon,\ a,\ \varepsilon)$ edge and one $(\varepsilon,\ \varepsilon,\ a)$ edge, for all $l > 0$, and a similar 'reversed' order would work if $l < 0$.

4. Note that not all states may be necessary, but it is simpler to construct them all.

We now show that the class of languages accepted by multivariable finite state automata is closed under intersecting in one coordinate with a regular language. The proof is analogous to the proof that the intersection of two regular languages is regular.

Lemma 4.2.13. *Let $n \in \mathbb{Z}_{>0}$ and Σ be an alphabet. Let L be a language accepted by an n-variable finite-state automaton that is constructible in* $\mathsf{NSPACE}(f)$*, for some function $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$, and let M be a regular language (that is constructible in constant space). Fix $i \in \{1,\ \ldots,\ n\}$. Then*

$$\{(w_1,\ \ldots,\ w_n) \in L \mid w_i \in M\}$$

is accepted by an n-variable finite-state automaton that is constructible in $\mathsf{NSPACE}(f)$.

Proof Let $\mathcal{M} = (\Sigma,\ \Lambda,\ p_0,\ E)$ be a finite-state automaton accepting M that is constructible in constant space, and $\mathcal{A} = (\Sigma,\ \Gamma,\ q_0,\ F)$ be a multivariable finite-state automaton accepting L that is constructible in $\mathsf{NSPACE}(f)$. Let $\Lambda \times \Gamma$ denote the edge-labelled graph obtained as follows:

1. The vertex set is $V(\Lambda) \times V(\Gamma)$;
2. Edges are labelled using n-variable words over Σ;
3. There is an edge from $(x_1,\ y_1)$ to $(x_2,\ y_2)$ if and only if there is an edge in Λ

from x_1 to x_2, labelled a and an edge in Γ from y_1 to y_2, labelled with $\mathbf{b}$, such that the ith coordinate of $\mathbf{b}$ is a.

Consider the n-variable finite-state automaton $\mathcal{B} = (\Sigma,\ \Lambda \times \Gamma, (p_0,\ q_0),\ E \times F)$. By viewing the path in the second coordinate of the graph $\Lambda \times \Gamma$, we can conclude that every word accepted by $\mathcal{B}$ is accepted by $\mathcal{A}$. By looking at the first coordinate, we have that the ith coordinate of every word accepted by $\mathcal{B}$ is a word accepted by $\mathcal{M}$. Conversely, any such word must always be accepted by $\mathcal{B}$, as it traces these paths in Γ and Λ.

It remains to show that $\mathcal{B}$ is constructible in $\mathsf{NSPACE}(f)$. Writing down $V(\Gamma)$ can be done in $\mathsf{NSPACE}(f)$. Since $\mathcal{M}$ is constructible in constant space, we have that $V(\Lambda) \times V(\Gamma)$ can be constructed in $\mathsf{NSPACE}(f)$. Writing each of edges in Γ can be done in $\mathsf{NSPACE}(f)$. We can thus follow this algorithm, but each time we attempt to write an edge, we check it against every edge in Λ, which we have stored. As Λ is constructible in constant space, this can also be done in $\mathsf{NSPACE}(f)$. The vertex $(p_0,\ q_0)$ can just be written down, and constructing $E \times F$ can also be done in $\mathsf{NSPACE}(f)$, as E is constructible in constant space. □

We will later need the following lemma that allows us to take finite unions of languages that are accepted by multivariable finite-state automata without changing the space complexity.

Lemma 4.2.14. *Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ be a function. A finite union of languages accepted by multivariable finite-state automata that are all constructible in $\mathsf{NSPACE}(f)$ is also accepted by a multivariable finite-state automaton that is constructible in $\mathsf{NSPACE}(f)$.*

Proof The automaton $\mathcal{M}$ we use is the automaton obtained by taking the union of all of the automata of the languages in the union, and collapsing the start states to a single state, which will be the start state. All accept states will remain accept states. We can construct $\mathcal{M}$ by constructing each of the automata in the union one at a time, which can be done in $\mathsf{NSPACE}(f)$. □

4.2.3 Multivariable solution languages

We now define an alternative language of solutions to study. We have so far considered the language of words that comprise the solutions concatenated with one another, delimited by an additional letter #. We now look at the language of n-variable words representing solutions.

Definition 4.2.15. Let G be a finitely generated group, with a finite monoid generating set Σ, and a normal form $\eta \colon G \to \Sigma^*$. Let $\mathcal{E}$ be a system of (twisted) equations in G, and let n be the number of variables in $\mathcal{E}$. Let $V = \{X_1, \ldots, X_n\}$ be the set of variables, and let $\mathcal{S}$ be the set of solutions, which are homomorphisms from $F_V * G$ to G.

The *multivariable solution language* to $\mathcal{E}$ with respect to Σ and η, is defined to be

$$\{(X_1\psi\eta, \ X_2\psi\eta, \ \ldots, \ X_n\psi\eta) \mid \psi \in \mathcal{S}\} \subset \Sigma^* \times \Sigma^* \times \cdots \times \Sigma^*.$$

We now show that a multivariable solution language being accepted by an n-variable finite-state automaton is sufficient for the corresponding solution language to be EDT0L.

Lemma 4.2.16. *Let L be an n-variable language over an alphabet Σ (where $n \in \mathbb{Z}_{>0}$), that is accepted by an n-variable finite-state automaton, constructible in* $\mathsf{NSPACE}(f)$*, for some $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then*

1. *The language $M = \{w_1\# \cdots \#w_n \mid (w_1, \ldots, w_n) \in L\}$ is an EDT0L language over $\Sigma \sqcup \{\#\}$;*
2. *An EDT0L system for M can be constructed in* $\mathsf{NSPACE}(f)$.

Proof We will construct an EDT0L system $\mathcal{H}$ for M as follows. The terminal alphabet will be $\Sigma \cup \{\#\}$, the extended alphabet will be $C = \Sigma \cup \{\#, \perp_1, \ldots, \perp_n\}$, and the start word will be $\perp_1 \# \cdots \# \perp_n$.

Let $\mathcal{A} = (\Sigma, \ \Gamma, \ q_0, \ F)$ be an n-variable finite-state automaton that accepts L. We will use $\mathcal{A}$ to define the rational control of $\mathcal{H}$. Let W be the set of all n-variable words that appear as edge labels within Γ. For each $\mathbf{w} = (w_1, \ldots, w_n) \in W$, define $\varphi_{\mathbf{w}} \in \mathrm{End}(C^*)$ by

$$\begin{aligned} \bot_1 \, \varphi_{\mathbf{w}} &= w_1 \, \bot_1 \\ &\vdots \\ \bot_n \, \varphi_{\mathbf{w}} &= w_n \, \bot_n \, . \end{aligned}$$

Also define $\psi \in \mathrm{End}(C^*)$

$$\bot_i \, \psi = \varepsilon,$$

for all $i \in \{1, \ldots, n\}$. Our rational control $\mathcal{R}$ will be a regular language over the set $\{\varphi_{\mathbf{w}} \mid \mathbf{w} \in W\}$. Let Γ' be the edge-labelled graph obtained from Γ, by replacing the label $\mathbf{w}$ on each edge with $\varphi_{\mathbf{w}}$.

Consider the (1-variable) finite-state automaton $\mathcal{B} = (\Sigma, \Gamma', q_0, F)$. Let K be the language accepted by $\mathcal{B}$. We have that K is precisely the set of all endomorphisms θ of C^* that can be written as products of endomorphisms $\varphi_{\mathbf{w}}$, for $\mathbf{w} \in W$, such that $(\bot_1 \# \cdots \# \bot_n)\theta = u_1 \bot_1 \# \cdots \# u_n \bot_n$, for some $(u_1, \ldots, u_n) \in L$. Therefore, the regular language $K\psi$ is the set of all endomorphisms that map $\bot_1 \# \cdots \# \bot_n$ to an element of M, and so taking $\mathcal{R} = K\psi$ gives the desired EDT0L system.

For (2), since a multivariable finite-state automaton contains an alphabet Σ, this can be obtained and output in $\mathsf{NSPACE}(f)$, and thus the alphabet for the EDT0L language, $\Sigma \cup \{\#\}$, and the extended alphabet $C = \Sigma \cup \{\#, \bot_1, \ldots, \bot_n\}$ can also be constructed and written to the output tape in $\mathsf{NSPACE}(f)$. The start word will always be $\bot_1 \# \cdots \# \bot_n$, regardless of the input, and we can just output this.

It remains to construct the rational control. As in the construction of $\mathcal{H}$, we use the same set of vertices and edges, but whenever the rational control in $\mathcal{H}$ labels an edge using $\varphi_{\mathbf{w}}$, we instead label it using $\mathbf{w}$, and note that $\varphi_{\mathbf{w}}$ can be effectively computed from $\mathbf{w}$. To record $\varphi_{\mathbf{w}}$, we only need to know where each $\bot_i$ maps (as they always fix everything else), and that is precisely the information that $\mathbf{w}$ contains. $\square$

4.3 Solution languages in virtually abelian groups

The purpose of this section is to prove that the multivariable solution languages to systems of equations in virtually abelian groups are accepted by multivariable finite-state automata, and so solution languages are EDT0L, all with respect to a natural generating set and normal form. We do this by first showing that the multivariable solution languages for systems of twisted equations in free abelian groups are recognised by finite-state automata, and then prove that equations in virtually abelian groups reduce to twisted equations in free abelian groups. Throughout this section, when referring to free abelian groups, we will use additive notation. This means that equations in free abelian groups will be expressed as sums rather than 'products'. When representing solution languages, we will express them using multiplicative notation, as this is more natural with languages, using $a_1, \ldots, a_k$ to be the standard generators of $\mathbb{Z}^k$.

The next lemmas are used to prove that systems of equations, and therefore twisted equations, in free abelian groups have multivariable solution languages accepted by n-variable finite-state automata, where n is the number of variables. The fact that free abelian groups have EDT0L solution languages is already known; Diekert, Jeż and Kufleitner [32] show that right-angled Artin groups have EDT0L solution languages, and Diekert [30] has a more direct method for systems of equations in $\mathbb{Z}$, which can easily be generalised to all free abelian groups. For the sake of completeness, we give our own argument here.

We begin with the following technical definition.

Definition 4.3.1. Let $B = [b_{ij}]$ be an $n \times m$ integer matrix. Then define a function $|\cdot|_B \colon \mathbb{R}^n \to \mathbb{R}$ via

$$|(y_1, \ldots, y_n)|_B = \max\left(\left|\sum_{i=1}^{n} y_i b_{i1}\right|, \left|\sum_{i=1}^{n} y_i b_{i2}\right|, \ldots, \left|\sum_{i=1}^{n} y_i b_{im}\right|\right).$$

In other words, $|\mathbf{y}|_B$ is the maximal absolute value of the coordinates of the vector $\mathbf{y}B$.

Note that if $\mathbf{y} \in \mathbb{Z}^n$ then $|\mathbf{y}|_B \in \mathbb{Z}$, and that $|\cdot|_B$ satisfies the triangle inequality.

We now show that we can construct any solution to a system of equations while controlling the value of $|\cdot|_B$ at each intermediate point.

Lemma 4.3.2. *Let B be an $n \times m$ integer matrix, X be a vector of n variables, $\mathbf{c} \in \mathbb{Z}^m$, and consider the system of n equations over $\mathbb{Z}$ given by $\mathbf{X}B = \mathbf{c}$. Write $b_{\max} = \max_{i,j} |b_{ij}|$ and let $K = \max(|c_1|, \ \ldots, \ |c_m|) + n^{3/2} \cdot b_{\max}$.*

Then, for each $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathbf{x}B = \mathbf{c}$, there is a sequence

$$\{0 = \mathbf{x}^{(0)}, \ \mathbf{x}^{(1)}, \ \ldots, \ \mathbf{x}^{(k)} = \mathbf{x}\} \subset \mathbb{Z}^n$$

with each $\mathbf{x}^{(j)} = \mathbf{x}^{(j-1)} + \mathbf{e}_j$ for some positive or negative standard basis vector $\mathbf{e}_j$, such that $|\mathbf{x}^{(j)}|_B \leq K$ for each $j \in \{1, \ \ldots, \ k\}$.

Proof First, consider the straight line segment $L \subset \mathbb{R}^n$ from 0 to $\mathbf{x}$. Since B defines a linear transformation $\mathbb{R}^n \to \mathbb{R}^m$, the function $|\cdot|_B \colon \mathbb{R}^n \to \mathbb{R}$ is monotone non-decreasing as we move along L from 0 to $\mathbf{x}$. Therefore, for each $\mathbf{y} \in L$, we have $|\mathbf{y}|_B \leq |\mathbf{x}|_B = \max(|c_1|, \ \ldots, \ |c_m|)$. To obtain the required sequence, we approximate L with a piecewise linear path comprised of (positive and negative) standard basis vectors.

Consider the set of unit n-cubes with integer-valued corners, which intersect L. From among the corners of these cubes, we can find a sequence $\{\mathbf{x}^{(j)}\} \subset \mathbb{Z}^n$ of integer-valued points, where subsequent terms share a cube edge (and so each $\mathbf{x}^{(j)} = \mathbf{x}^{(j-1)} + \mathbf{e}_j$ for some $\mathbf{e}_j$), such that $\mathbf{x}^{(0)} = 0$ and $\mathbf{x}^{(k)} = \mathbf{x}$, for some k. We will show that each point in this sequence satisfies the required bound.

Since the diameter of a unit n-cube is $\sqrt{n}$, each point $\mathbf{x}^{(j)}$ is a Euclidean distance of at most $\sqrt{n}$ from the line L. In other words, for each j we have $\mathbf{x}^{(j)} = \mathbf{y} + \mathbf{d}$ for some $\mathbf{y} \in L$ and $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{R}^n$ with $|d_i| \leq \sqrt{n}$. Then note that for any such

$\mathbf{d}$ we have

$$|\mathbf{d}|_B = \max\left(\left|\sum_{i=1}^{n} d_i b_{i1}\right|, \ \ldots, \ \left|\sum_{i=1}^{n} d_i b_{im}\right|\right)$$
$$\leq \max\left(\sum_{i=1}^{n} |d_i||b_{i1}|, \ \ldots, \ \sum_{i=1}^{n} |d_i||b_{im}|\right)$$
$$\leq \max\left(\sum_{i=1}^{n} \sqrt{n} \cdot b_{\max}, \ \ldots, \ \sum_{i=1}^{n} \sqrt{n} \cdot b_{\max}\right) = (n\sqrt{n})b_{\max}.$$

We can then bound each element of the sequence as follows:

$$|\mathbf{x}^{(j)}|_B = |\mathbf{y} + \mathbf{d}|_B \leq |\mathbf{y}|_B + |\mathbf{d}|_B \leq \max(|c_1|, \ \ldots, \ |c_m|) + (n\sqrt{n})b_{\max} = K.$$

Thus the sequence $\{\mathbf{x}^{(j)}\}$ satisfies the requirements of the Lemma. □

We now show that a system of twisted equations in $\mathbb{Z}^k$ can be reduced to a system of (non-twisted) equations in $\mathbb{Z}$.

Lemma 4.3.3. *Let $\mathcal{S}_\mathcal{E}$ be the solution set of a finite system $\mathcal{E}$ of twisted equations in $\mathbb{Z}^k$ in n variables. Then there is a finite system of equations $\mathcal{F}$ in $\mathbb{Z}$ with kn variables and solution set $\mathcal{S}_\mathcal{F}$ such that*

$$\mathcal{S}_\mathcal{E} = \{((x_1, \ \ldots, \ x_k), \ (x_{k+1}, \ \ldots, \ x_{2k}), \ \ldots, \ (x_{(k-1)n}, \ \ldots, \ x_{kn})) \mid (x_1, \ \ldots, \ x_{kn}) \in \mathcal{S}_\mathcal{F}\}.$$

Proof Consider a twisted equation in $\mathbb{Z}^k$

$$\mathbf{c}_0 + \mathbf{Y}_{i_1}\Phi_1 + \mathbf{c}_1 \cdots + \mathbf{Y}_{i_n}\Phi_n + \mathbf{c}_n = 0, \tag{4.1}$$

where $\mathbf{Y}_1, \ \ldots, \ \mathbf{Y}_n$ are variables, $\mathbf{c}_0, \ \ldots, \ \mathbf{c}_n \in \mathbb{Z}^n$ are constants, and $\Phi_1, \ \ldots, \ \Phi_n \in \mathrm{GL}_k(\mathbb{Z})$. Set $\mathbf{c} = \mathbf{c}_0 + \cdots + \mathbf{c}_n$. By grouping the occurrences of each $\mathbf{Y}_i$, we have that (4.1) is equivalent to the following identity

$$\mathbf{Y}_1 B_1 + \cdots \mathbf{Y}_n B_n + \mathbf{c} = 0, \tag{4.2}$$

where $B_1 = [b_{1ij}], \ \ldots, \ B_n = [b_{nij}]$ are $k \times k$ integer-valued matrices, although not necessarily in $\mathrm{GL}_k(\mathbb{Z})$. We will first show that the solution set of (4.2) is equal to the solution set of a system of k equations in $\mathbb{Z}$. Write $\mathbf{Y}_i = (Y_{i1}, \ \ldots, \ Y_{ik})$ and $\mathbf{c} = (c_1, \ \ldots, \ c_n)$ for variables Y_{ij} over $\mathbb{Z}$ and constants $c_i \in \mathbb{Z}$, for each i. Then $\mathbf{Y}_i B_i = \left(\sum_{j=1}^k b_{ij1} Y_{ij}, \ \ldots, \ \sum_{j=1}^k b_{ijk} Y_{ij}\right)$, for each i. It follows that the solution set of (4.2) is equal to the solution set of the following system of equations in $\mathbb{Z}$:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} b_{ij1} Y_{ij} + c_i = 0$$

$$\vdots$$

$$\sum_{i=1}^{n} \sum_{j=1}^{k} b_{ijk} Y_{ij} + c_i = 0.$$

We can conclude that the lemma holds for single twisted equations in $\mathbb{Z}^k$. It follows that the solution set to a system of m twisted equations in $\mathbb{Z}^k$ will be constructible as stated in the lemma, from the solution set to a system of m of the above systems; that is a system of km equations in $\mathbb{Z}$. □

Before we can prove Lemma 4.3.5, we need a slightly altered version of modular arithmetic, where we replace 0 with the quotient.

Notation 4.3.4. For each $n, \ r \in \mathbb{Z}_{\geq 0}$ with $r > 0$, define

$$n \operatorname{mod}^+ r = \begin{cases} n \operatorname{mod} r & n \operatorname{mod} r \neq 0 \\ r & n \operatorname{mod} r = 0. \end{cases}$$

We are now in a position to prove that multivariable solution languages to twisted equations in free abelian groups are accepted by multivariable finite state automata. We do this by expressing our equation as an identity of matrices, where the coefficients of the matrix determine the equation. This allows us to use the bound from Lemma 4.3.2 to construct our automaton.

Lemma 4.3.5. *The multivariable solution language to a system of twisted equations in a free abelian group, with respect to the standard generating set and normal form, is accepted by a multivariable finite-state automaton.*

Proof Let $\mathcal{E}$ be a system of m twisted equations in $\mathbb{Z}^k$ in n variables. Let $\{a_1, \ldots, a_k\}$ denote the standard generating set for $\mathbb{Z}^k$. By Lemma 4.3.3, there is a system of km equations $\mathcal{F}$ in $\mathbb{Z}$ with kn variables, such that the solution language to $\mathcal{E}$ is equal to

$$\mathcal{S}_{\mathcal{E}} = \left\{ \left(a_1^{t_1} \cdots a_k^{t_k},\ a_1^{t_{k+1}} \cdots a_k^{t_{2k}},\ \ldots,\ a_1^{t_{k(n-1)+1}} \cdots a_k^{t_{kn}} \right) \middle| (t_1, \ldots, t_{kn}) \text{ is a solution to } \mathcal{F} \right\}.$$

We represent this new system $\mathcal{F}$ via the identity $\mathbf{X}B = \mathbf{c}$ where

- $\mathbf{X} = (X_1, \ldots, X_{kn})$ is a row vector of kn variables,
- $B = [b_{ij}]$ is a $kn \times km$ matrix of coefficients, and
- $\mathbf{c} \in \mathbb{Z}^{km}$ is a row vector of constants.

The constant of Lemma 4.3.2 is then $K = \max(|c_1|, \ldots, |c_{km}|) + (kn)^{3/2} b_{\max}$.

We can now show that the multivariable solution language is accepted by a kn-variable finite-state automaton, using the method described in Example 4.2.12. We define our automaton $\mathcal{A}$ to have the set of states

$$\{q_{\mathbf{x}} \mid \mathbf{x} = (x_1, \ldots, x_{kn}) \in \mathbb{Z}^{kn},\ |x_i| \leq K\},$$

Our start state will be $q_{\mathbf{0}}$, and $q_{\mathbf{c}}$ will be our only accept state. Let $\mathbf{w}_i$ be the kn-variable word with $a_{i \bmod^+ k}$ in the ith position, and ε elsewhere. We have an edge from $q_{\mathbf{x}}$ to $q_{\mathbf{y}}$ labelled with $\mathbf{w}_j$ for all j such that $\mathbf{x} + (b_{j1}, \ldots, b_{j(kn)}) = \mathbf{y}$. By construction, the language accepted by $\mathcal{A}$ is contained within $\mathcal{S}_{\mathcal{E}}$. On the other hand, any word in $\mathcal{S}_{\mathcal{E}}$ is accepted by $\mathcal{A}$, by following an appropriate sequence as given by Lemma 4.3.2. □

We now consider the space complexity that is needed to construct the multivariable finite-state automaton defined in the proof of Lemma 4.3.5.

Recall the virtually abelian length of a system of equations (see Definition 2.6.10). We will be using this as the length of our input.

Lemma 4.3.6. *The multivariable finite-state automaton defined in Lemma 4.3.5 can be constructed in non-deterministic quadratic space, with the input taken as the* *virtually abelian length of the system.*

Proof Let k be the rank of the free abelian group, $\mathcal{E}$ be the system of equations, n be the number of variables, and m be the number of equations. We start by converting $\mathcal{E}$ into the form $\mathbf{X}B = \mathbf{c}$ (all we need to store is B and $\mathbf{c}$). Let $I \in \mathbb{Z}_{\geq 0}$ be the length of the input.

Index the equations $w_1, \ldots, w_m$. We copy each equation in $\mathcal{E}$ into the work tape, so our work tape will now have the same size as our input. We have assumed our equations are already in the form stated in Definition 2.6.10, and converting them to additive notation means they will be in the form

$$\mathbf{Y}_1 B_1 + \cdots \mathbf{Y}_n B_n = \mathbf{d},$$

where each B_i is a $k \times k$ matrix, each $\mathbf{Y}_i$ is a variable, and $\mathbf{d} \in \mathbb{Z}^k$. We will now construct the matrix B and the vector $\mathbf{c}$. We write $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{ik})$, and $B_1 = [b_{1ij}], \ldots, B_n = [b_{nij}]$. For each equation

$$\mathbf{Y}_1 B_1 + \cdots \mathbf{Y}_n B_n = \mathbf{d},$$

add the following vectors as columns to B, and store them in the work tape:

$$(b_{111}, \ldots, b_{nk1}), \ldots (b_{11k}, \ldots, b_{nkk}).$$

The matrix B will at this point be a $kn \times km$ matrix. For each equation, we also append the entries of $\mathbf{d}$ to the vector $\mathbf{c}$.

We now construct the states. Since our set of states is the set of all $q_{\mathbf{x}}$ such that $\mathbf{x} \in \mathbb{Z}^{kn}$ with each coordinate having absolute value at most K, where K is from Lemma 4.3.2, we can construct the set of states by remembering the last state constructed, together with the bound K, and proceeding in any 'sensible' systematic manner, such as starting in one 'corner', and running down each line in the 'cube'. To do this, we need a memory that can store a vector of length kn at any time, with entries within $[-K, K]$.

As in the proof of Lemma 4.3.5, $K = \max(|c_1|, \ \ldots, \ |c_{km}|) + (kn)^{3/2} \cdot b_{\max}$, where $b_{\max} = \max_{i,j} |b_{ij}|$. Recall $I = \sum_i \log |c_i| + \sum_{i,j} \log |b_{ij}| + Ckn$, where C is a constant, as mentioned in Definition 2.6.10. Then

$$\log K \leq \log |c_1| + \cdots + \log |c_{km}| + \frac{3}{2} \log(kn) + \log |b_{\max}| \leq \frac{3}{2} I$$

So storing an integer within $[-K, \ K]$ requires $\frac{3}{2} I$ bits, ignoring constants. Since $kn \leq I$, storing a vector of length kn with entries in $[-K, \ K]$ requires at most $\frac{3}{2} I^2$ bits.

We can simply assign $\mathbf{0}$ and $\mathbf{c}$ as the start and accept states.

We now need to compute the edges. Recall that we have an edge from $q_{\mathbf{x}}$ to $q_{\mathbf{y}}$ labelled with $\mathbf{w}_j$ for all j such that $\mathbf{x} + (b_{j1}, \ \ldots, \ b_{j(kn)}) = \mathbf{y}$, where $\mathbf{w}_j$ is the kn-variable word with $a_{i \bmod^+ k}$ in the ith position and ε elsewhere. Therefore, we can go through the states systematically and add all of the outgoing edges, and we only need to remember the state we are on in order to compute and output its outgoing edges and their labels. To do this, we only need to record a vector of length kn, the entries of which will lie in $[-K, \ K]$. As discussed before, this requires at most $\frac{3}{2} I^2$ bits to store. □

In the next lemma, we show that the solution set to a system of equations in an arbitrary group can be expressed in terms of the solution set to a system of *twisted* equations in a finite-index subgroup.

Lemma 4.3.7. *Let G be a group, and T be a finite transversal of a normal subgroup H of finite index. Let Ω be the group of automorphisms of H induced by conjugating H by elements of G. Let $\mathcal{S}$ be the solution set to a finite system $\mathcal{E}_G$ of equations with rational constraints in n variables in G. Then there is a finite set $B \subseteq T^n$, and for each $\mathbf{t} = (t_1, \ \ldots, \ t_n) \in B$, there is a solution set $A_{\mathbf{t}}$ to a system $\mathcal{E}_{H,\mathbf{t}}$ of Ω-twisted equations with rational constraints in H, such that*

$$\mathcal{S} = \bigcup_{(t_1, \ \ldots, \ t_n) \in B} \left\{ (h_1 t_1, \ \ldots, \ h_n t_n) \mid (h_1, \ \ldots, \ h_n) \in A_{(t_1, \ \ldots, \ t_n)} \right\}.$$

Proof Let

$$X_{i_{1j}}^{\epsilon_{1j}} g_{1j} \cdots X_{i_{pj}}^{\epsilon_{pj}} g_{pj} = 1 \tag{4.3}$$

be a system $\mathcal{E}_G$ of equations in G, with a set $\{R_{X_1}, \ldots, R_{X_n}\}$ of rational constraints, where $X_1, \ldots, X_n$ are the variables, and $j \in \{1, \ldots, k\}$. Let $\mathcal{S}$ be the solution set. Note that we can assume that these equations start with variables by conjugating leading constants to the right. For each X_i, define new variables Y_i over H, and Z_i over T, such that $X_i = Y_i Z_i$. For each constant g_i, we have $g_i = h_i t_i$, for some $h_i \in H$ and $t_i \in T$, and so substituting these into (4.3) gives that $\mathcal{E}_G$ is equivalent to

$$(Y_{i_{1j}} Z_{i_{1j}})^{\epsilon_{i_{1j}}} h_{1j} t_{1j} \cdots (Y_{i_{pj}} Z_{i_{pj}})^{\epsilon_{i_{pj}}} h_{pj} t_{pj} = 1. \tag{4.4}$$

For all $g \in G$, define $\psi_g \colon G \to G$ by $h\psi_g = ghg^{-1}$. Note that $\psi_g \restriction_H \in \Omega$ for all $g \in G$, by definition. By abusing notation, we can define ψ_{Z_i} for each i. For all $i \in \{1, \ldots, n\}$, and $j \in \{1, \ldots, k\}$ define

$$\delta_{ij} = \begin{cases} 0 & \epsilon_{ij} = 1 \\ 1 & \epsilon_{ij} = -1. \end{cases}$$

We can use this notation to rearrange (4.4) into

$$(Y_{i_{1j}}^{\epsilon_{1j}} \psi_{Z_{i_{1j}}}^{\delta_{1j}}) Z_{i_{1j}}^{\epsilon_{1j}} h_{1j} t_{1j} \cdots (Y_{i_{pj}}^{\epsilon_{pj}} \psi_{Z_{i_{pj}}}^{\delta_{pj}}) Z_{i_{pj}}^{\epsilon_{pj}} h_{pj} t_{pj} = 1. \tag{4.5}$$

For $l \in \{1, \ldots, p\}$, define

$$W_l = (Y_{i_{lj}}^{\epsilon_{lj}}) \psi_{Z_{i_{lj}}}^{\delta_{lj}} \psi_{t_{(l-1)j}} \psi_{Z_{i_{(l-1)j}}}^{\epsilon_{(l-1)j}} \cdots \psi_{t_{1j}} \psi_{Z_{i_{1j}}}^{\epsilon_{1j}},$$

$$f_l = (h_{lj}) \psi_{Z_{i_{lj}}}^{\epsilon_{lj}} \psi_{t_{(l-1)j}} \cdots \psi_{t_{1j}} \psi_{Z_{i_{1j}}}^{\epsilon_{1j}}.$$

By pushing all Y_is and h_is to the left within (4.5), we obtain

$$W_1 f_1 \cdots W_p f_p Z_{i_{1j}}^{\epsilon_{1j}} t_{1j} \cdots Z_{i_{pj}}^{\epsilon_{pj}} t_{pj} = 1. \tag{4.6}$$

As H is a finite index subgroup of G, Ht is a recognisable subset of G, for all $t \in T$. For each coset Ht of H, and each variable X_i let $R_{ti} = R_{X_i} \cap (Ht)$. Note that each set R_{ti} is rational, since each R_{ti} is the intersection of a rational set with a recognisable set.

By Lemma 2.5.4, we have that for each $t \in T$, $R_{ti} = S_{ti}t$, for some rational subset S_{ti} of H. For every $(u_1, \ \ldots, \ u_n) \in T^n$ that forms a solution to the Z_is within a solution to (4.6), we have $u_{i_1}^{\epsilon_1} t_1 \cdots u_{i_p}^{\epsilon_p} t_p \in H$. Let $A \subseteq T^n$ be the set of all such n-tuples. If we plug a fixed choice of some $(u_1, \ \ldots, \ u_n) \in T^n$ into (4.6), we obtain the following system of Ω-twisted equations in H:

$$\bar{W}_1 f_1 \cdots \bar{W}_p f_p u_{i_{1j}}^{\epsilon_{1j}} t_{1j} \cdots u_{i_{pj}}^{\epsilon_{pj}} t_{pj} = 1,$$

where

$$\bar{W}_l = (Y_{i_{lj}}^{\epsilon_{lj}}) \psi_{u_{i_{lj}}}^{\delta_{lj}} \psi_{t_{(l-1)j}} \psi_{u_{i_{(l-1)j}}}^{\epsilon_{(l-1)j}} \cdots \psi_{t_{1j}} \psi_{u_{i_{1j}}}^{\epsilon_{1j}},$$

is W_l, with each Z_i being replaced by u_i. We can now apply the rational constraint S_{ti} to the variable Y_i, and we have a system of equations $\mathcal{E}_{H,(u_1,\ldots,u_n)}$ with rational constraints in H. Let $B_{(u_1, \ \ldots, \ u_n)}$ be the solution set to $\mathcal{E}_{H,(u_1,\ldots,u_n)}$. It follows that

$$\mathcal{S} = \bigcup_{(u_1, \ \ldots, \ u_n) \in A} \{(f_1 u_1, \ \ldots, \ f_n u_n) \mid (f_1, \ \ldots, \ f_n) \in B_{(u_1, \ \ldots, \ u_n)}\}.$$

□

Remark 4.3.8. Let G be a finite index overgroup of a group H. We will define a normal form for G, induced by an existing normal form on H. Let

- Σ_H be a finite generating set for H;
- η_H be a normal form for H, with respect to Σ_H;
- T be a (finite) right transversal for H in G.

We will use $\Sigma = \Sigma_H \sqcup T$ as our generating set for G. Each $g \in G$ can be written uniquely in the form $g = h_g t_g$ for some $h_g \in H$ and $t_g \in T$. Define $\eta \colon G \to (\Sigma^{\pm})^*$ by

$$g\eta = (h_g \eta_H) t_g.$$

Note that if η_H is regular, then η is regular, as the concatenation of $\operatorname{im}\eta_H$ with a finite language.

As the following lemma shows, this construction also preserves the property of being quasi-geodesic.

The following proposition reflects a well-known fact about decidability of systems of equations in groups: if a group G has a finite index normal subgroup H, such that there is an algorithm that determines if any system of twisted equations in H admits a solution, then there is an algorithm that determines if any system of (untwisted) equations G admits a solution. This fact turns out to be true regarding EDT0L solutions, and a variant of it is used in [31].

Proposition 4.3.9. *Let G be a group with a finite index normal subgroup H, such that the multivariable solution language to systems of Ω-twisted equations in H with rational constraints are accepted by an n-variable finite-state automaton, for some $n \in \mathbb{Z}_{>0}$, with respect to a generating set Σ, and normal form η.*

Then the multivariable solution language to any system of equations in G is accepted by an n-variable finite-state automaton, for some $n \in \mathbb{Z}_{>0}$, with respect to the generating set $\Sigma \cup T$, for any right transversal T of H, and the normal form ζ, where $g\zeta = (h\eta)t$, where $h \in H$ and $t \in T$ are (unique) such that $g = ht$.

Proof We have from Lemma 4.3.7, that the solution language is a finite union across valid choices of transversal vectors $(t_1, \ldots, t_n)$ of

$$\{((h_1\eta)t_1, \ \ldots, \ (h_n\eta)t_n) \mid (h_1, \ \ldots, \ h_n) \in A_{(t_1, \ldots, t_n)}\}, \tag{4.7}$$

where $A_{(t_1, \ldots, t_n)}$ is the solution set to a system of twisted equations in H. Since the class of languages accepted by n-variable finite-state automata is closed under finite unions, it suffices to show that (4.7) is accepted by an n-variable finite-state automaton.

By our assumptions on H, the language

$$\{(h_1\eta,\ \ldots,\ h_n\eta) \mid (h_1,\ \ldots,\ h_n) \in A_{(t_1,\ \ldots,\ t_n)}\}$$

is accepted by an n-variable finite-state automaton $\mathcal{M}$, for any valid choice of transversal vector $(t_1,\ \ldots,\ t_n)$. We can therefore modify this automaton to accept

$$\{(h_1t_1,\ \ldots,\ h_nt_n) \mid (h_1,\ \ldots,\ h_n) \in A_{(t_1,\ \ldots,\ t_n)}\}$$

We do this by adding a new state q, and with an edge labelled $(t_1,\ \ldots,\ t_n)$ from every accept state of $\mathcal{M}$, and making q the only accept state. By construction, this accepts the stated language. □

Lemma 4.3.10. *Let G, H, n and Σ be defined as in Proposition 4.3.9, and suppose the multivariable finite-state automaton that accepts a system of twisted equations in H, in the statement of Proposition 4.3.9, is constructible in* $\mathsf{NSPACE}(f)$*, where $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is a function. Then the automaton that accepts a system of equations in G is also constructible in* $\mathsf{NSPACE}(f)$*.*

Proof By Lemma 4.2.14, it suffices to show that each automaton that accepts a language

$$\{(h_1t_1,\ \ldots,\ h_nt_n) \mid (h_1,\ \ldots,\ h_n) \in A_{(t_1,\ \ldots,\ t_n)}\},$$

where $A_{(t_1,\ldots,t_n)}$ is as defined in the proof of Proposition 4.3.9. Recall that this is constructed from the automaton $\mathcal{M}$ that accepts a system of twisted equations in H by adding one additional state q, and edges from each accept state to q, all labelled $(t_1,\ \ldots,\ t_n)$, and then by making q the only accept state. We do this by modifying the algorithm that constructs $\mathcal{M}$ to add the state q at the beginning, then perform the algorithm that constructs $\mathcal{M}$, except whenever we would label a state p as an accept state, we instead add an edge from p to q, labelled by $(t_1,\ \ldots,\ t_n)$. This does not use a longer work tape than the algorithm that constructs $\mathcal{M}$. □

Lemma 4.3.11. *Let G, H, Σ, Σ_H, T, η and η_H be defined as in Remark 4.3.8. Then η_H is quasi-geodesic if and only if η is quasi-geodesic.*

Proof (⇒): Suppose η_H is quasi-geodesic. Then there exists $\lambda > 0$, such that $|h\eta_H| \leq \lambda |h|_{(H,\Sigma_H)} + \lambda$ for all $h \in H$.

For each $t \in T$ and $a \in \Sigma^{\pm}$, $ta = \nu_{t,a}$, for some $\nu_{t,a} \in \operatorname{im} \eta$. For all $t,\ t' \in T$, we have $tt' = \rho_{t,t'}$, for some $\rho_{t,t'} \in \operatorname{im} \eta$. For each $t^{-1} \in T^{-1}$, we have that $t^{-1} =_G x_{t^{-1}}$, where $x_{t^{-1}} \in (\Sigma_H^{\pm} \cup T)^*$. Let

$$\mu = \max_{t^{-1} \in T^{-1}} |x_{t^{-1}}| + \max_{t,t' \in T} |\rho_{t,t'}| + \max_{\substack{a \in \Sigma_H \\ t \in T}} |\nu_{t,a}|$$

Let $w \in (\Sigma^{\pm})^*$ be a geodesic. We will convert $w \in (\Sigma^{\pm})^*$ into a word u, such that $u =_G w$ and $u \in \operatorname{im} \eta$, and we will show that $|u| \leq \mu^2 \lambda |w| + \mu^2 \lambda$.

We first replace each occurrence of $t^{-1} \in T^{-1}$ with the word $x_{t^{-1}}$ within w. Since $|x_{t^{-1}}| \leq \mu$ for all $t^{-1} \in T$, doing this will result in a new word $w_1 \in (\Sigma_H^{\pm} \cup T)^*$, such that $w_1 =_G w$, and $|w_1| \leq \mu |w|$.

We now modify w_1 into a word w_2 such that $w_1 =_G w_2$, and w_2 contains no subword of the form ta or tt', where $t,\ t' \in T$ and $a \in \Sigma_H^{\pm}$. For each subword ta of w, we can replace ta with $\nu_{t,a}$, and for every occurrence of tt', we can replace this with $\rho_{t,t'}$. Each time we do this, we increase the length of the word by at most μ. Repeating this process until no subwords of the form ta remain, will yield w_2.

To ensure we don't need to do too many of these replacements to satisfy linear bound of the length of w_2 in terms of w_1, we will always apply the leftmost substitution possible. As every replacement involves a letter $t \in T$ at the beginning of a two-letter word, and results in a word with exactly one two-letter in T at the end, one 'sweep' along w_1 will be sufficient to reach a word where no substitutions are possible. It follows that we can make at most $|w_1|$ replacements, and since each substitution increases the length by at most μ, we have that $|w_2| \leq \mu |w_1|$.

We have that $w_2 = vt$, for some $v \in (\Sigma_H^{\pm})^*$, and some $t \in T$. To convert w_3 into u, it remains to replace v with an equivalent word $q \in \operatorname{im} \eta_H$. As η_H is quasi-geodesic with the constant λ, $|q| \leq \lambda |v| + \lambda$. If we take $u = qt$, then u is equivalent in G to

w, and $u \in \operatorname{im} \eta$. Note also that $|u| \leq \lambda |w_2| + \lambda$. Therefore

$$|u| = \lambda |w_2| + \lambda \leq \mu\lambda|w_1| + \mu\lambda \leq \mu^2\lambda|w| + \mu^2\lambda.$$

It follows that η_H is quasi-geodesic, with respect to the constant $\lambda\mu^2$.

($\Leftarrow$): Suppose η is quasi-geodesic, with respect to a constant $\lambda > 0$. Let $w \in (\Sigma_H^{\pm})^*$ be a geodesic, $u \in \operatorname{im} \eta_H$ be such that $u =_H w$, and $v \in (\Sigma^{\pm})^*$ be a geodesic in G, such that $v =_G w$. Note that $u \in \operatorname{im} \eta$. As η is quasi-geodesic, $|u| \leq \lambda|v| + \lambda$. Moreover, since $|w|$ and $|v|$ are both geodesic words representing elements that lie in H, but v is over the generating set Σ_G that contains the generating set Σ_H for w, $|v| \leq |w|$. Thus $|u| \leq \lambda|w| + \lambda$, as required. ☐

We now have enough to prove our first main result. Our generating set is the union of the standard generating set Σ of the finite index free abelian subgroup together with a right transversal T. We use the quasi-geodesic normal form

$$\{at \mid a \in \Sigma, \ t \in T\}.$$

Definition 4.3.12. Let $k \in \mathbb{Z}_{>0}$. A subset of $\mathbb{Z}^k$ that can be written in the form

$$\{\mathbf{c}_1 n_1 + \cdots + \mathbf{c}_r n_r + \mathbf{d} \mid n_1, \ \ldots, \ n_r \in \mathbb{Z}_{\geq 0}\},$$

where $\mathbf{c}_i, \ \mathbf{d} \in \mathbb{Z}^k$ for all i, is called *linear*. A finite union of linear sets is called *semilinear.*

Showing that semilinear sets are rational is immediate from the definition. The converse is also true, thus giving a full classification of rational sets in free abelian groups.

Lemma 4.3.13 ([43])**.** *A subset of a free abelian group is rational if and only if it is semilinear.*

Since semilinear sets are defined in terms of equations and inequalities, we can use this to describe sets of solutions to systems of twisted equations with rational constraints in free abelian groups.

Lemma 4.3.14. *Let $\mathcal{S}_\mathcal{E}$ be the solution set of a finite system $\mathcal{E}$ of twisted equations in $\mathbb{Z}^k$ in n variables with rational constraints. Then there is a finite disjunction $\mathcal{F}$ of finite systems of arbitrary equations, and inequalities of the form $X \geq 0$, for some variable X, in $\mathbb{Z}$ with kn variables and solution set $\mathcal{S}_\mathcal{F}$ such that*

$$\mathcal{S}_\mathcal{E} = \{((x_1, \ldots, x_k), \ldots, (x_{(k-1)n+1}, \ldots, x_{kn})) \mid (x_1, \ldots, x_{kn}, y_1, \ldots, y_r) \in \mathcal{S}_\mathcal{F}\}.$$

Proof Converting the twisted system into a system over $\mathbb{Z}$ can be done by replacing each variable $\mathbf{X}$ over $\mathbb{Z}^k$ with k variables $X_1, \ldots, X_k$ over $\mathbb{Z}$, and considering the system that results from looking at each coordinate individually, as we did in the proof of Lemma 4.3.7. Now consider the membership problem of a variable $\mathbf{X}$ into a linear set $R = \{\mathbf{c}_1 n_1 + \cdots + \mathbf{c}_r n_r + \mathbf{d} \mid n_1, \ldots, n_r \in \mathbb{Z}_{\geq 0}\}$ (we will then generalise to semilinear).

Write $\mathbf{c}_i = (c_{i1}, \ldots, c_{ik})$ and $\mathbf{d} = (d_1, \ldots, d_k)$. Consider the following system of equations and inequalities over $\mathbb{Z}$.

$$Y_i \geq 0, \quad X_j = c_{1j} Y_1 + \cdots c_{rj} Y_r + d_j$$

for all $i \in \{1, \ldots, r\}$, and $j \in \{1, \ldots, k\}$, where $Y_1, \ldots, Y_r$ are new variables over $\mathbb{Z}$. We have that $(x_1, \ldots, x_k) \in \mathbb{Z}^k$ occurs within a solution $(x_1, \ldots, x_k, y_1, \ldots, y_k)$ to the above system, if and only if $(x_1, \ldots, x_k) \in R$.

The result follows from the fact that the solution set to a disjunction of systems is just the union of the solution sets, so if we take the disjunction of the systems obtained from each linear set used in the finite union of a semilinear set, we obtain the desired disjunction. □

We are now in a position to describe the solution language to a system of twisted equations with constraints in a free abelian group, using an EDT0L system.

Lemma 4.3.15. *The multivariable solution language to a system of twisted equations with rational constraints in a free abelian group, with respect to the virtually* *abelian equation length, and the standard generating set and normal form, is accepted by a multivariable finite-state automaton, which is constructible in non-deterministic quadratic space.*

Proof We will use $\Sigma = \{a_1, \ldots, a_k\}$ to denote the standard generating set for $\mathbb{Z}^k$. Let $\mathcal{E}$ be a system of equations in $\mathbb{Z}^k$ with a Multivariable solution language L. By Lemma 4.3.14, there is a disjunction $\mathcal{F}$ of systems of equations, and inequalities of the form $X \geq 0$, in $\mathbb{Z}$, with set of solutions $\mathcal{S}_\mathcal{F}$, such that

$$L = \{(a_1^{x_1} \cdots a_k^{x_k}, \ \ldots, \ a_1^{x_{(k-1)n+1}} \cdots a_k^{x_{kn}}) \mid (x_1, \ \ldots, \ x_{kn}, \ y_1, \ \ldots, \ y_r) \in \mathcal{S}_\mathcal{F}\}.$$

Consider the following language

$$M = \{(a_1^{x_1}, \ldots, a_k^{x_k}, \ldots, a_1^{x_{(k-1)n+1}}, \ldots, a_k^{x_{kn}}, b_1^{y_1}, \ldots, b_r^{y_r}) \mid (x_1, \ldots, x_{kn}, y_1, \ldots, y_r) \in \mathcal{S}_\mathcal{F}\}.$$

We will start by showing that M is accepted by a kn-variable finite-state automaton, constructible in $\mathsf{NSPACE}(n^2)$. First note that as this class is closed under finite unions (Lemma 2.4.3), we can assume $\mathcal{F}$ is a single system of equations and inequalities, rather than a disjunction of systems. Let m be the number of inequalities of the form $X \geq 0$ within $\mathcal{F}$.

We will proceed by induction on m. If $m = 0$, then $\mathcal{F}$ is a system of equations in $\mathbb{Z}$, and thus the solutions are accepted by a kn-variable finite-state automaton that is constructible in $\mathsf{NSPACE}(n^2)$, by Lemma 4.3.5 and Lemma 4.3.6. Inductively suppose M is accepted by such a kn-variable finite-state automaton, constructible in $\mathsf{NSPACE}(n^2)$, when $m = r$, where $r \in \mathbb{Z}_{\geq 0}$. If $m = r + 1$, then $\mathcal{F}$ can be obtained from a system of equations and inequalities $\mathcal{G}$, with the addition of a single inequality $X \geq 0$. By our inductive hypothesis, the solution language of $\mathcal{G}$ is EDT0L, and an EDT0L system can be constructed in $\mathsf{NSPACE}(n^2)$.

The addition of the inequality $X \geq 0$, can be achieved by intersecting the coordinate of the solution language of $\mathcal{G}$ corresponding to the variable X with the regular language $\{a_i\}^*$, where a_i is the free abelian generator corresponding to X. The fact that this intersection is still accepted by a kn-variable finite-state automaton constructible in $\mathsf{NSPACE}(n^2)$ follows from Lemma 4.2.13. Since this intersection equals M, we have that M is accepted by a kn-variable finite-state automaton, constructible in $\mathsf{NSPACE}(f)$.

To obtain L from M, all we have to do is ignore the last r coordinates, so we take an automaton for M, and we define one for L by removing the last r coordinates from each label. Since this can be done using the same space, the result follows. □

We now have everything needed to show the following.

Theorem 4.3.16. *Solutions to a system of equations with rational constraints in a virtually abelian group are accepted by a multivariable finite-state automaton, with respect to the regular quasi-geodesic normal form from Remark 4.3.8, induced by the standard normal form on free abelian groups. Moreover, this automaton is constructible in non-deterministic quadratic space, with respect to the virtually abelian equation length.*

Proof This fact that the solutions are accepted by a multivariable finite-state automaton, constructible in $\mathsf{NSPACE}(n^2)$ follows from Lemma 4.3.15 and Proposition 4.3.9. The fact that the normal form is regular and quasi-geodesic follows from Remark 4.3.8, and Lemma 4.3.11, respectively, together with the fact that the standard normal form on a free abelian group is regular and quasi-geodesic. □

Corollary 4.3.17. *Solutions to a system of equations with rational constraints in a virtually abelian group are EDT0L in non-deterministic quadratic space, with respect to the virtually abelian equation length, and with respect to the regular quasi-geodesic normal form from Remark 4.3.8, induced by the standard normal form on free abelian groups.*

Proof This follows from Theorem 4.3.16 and Lemma 4.2.16. □

Remark 4.3.18. Corollary 4.3.17 uses the normal form defined by writing an element of a virtually abelian group as a product of a word in the finite-index free abelian normal subgroup, written in standard normal form, with an element of the (finite) transversal for that subgroup.

We can change our generating set to any other generating set, and there will exist a normal form such that solution languages are still EDT0L. Adding a new generator does not change the language at all, as we can keep the normal form the same, and so our new generator will not appear in any normal form word. To remove a redundant generator c, we can fix a word w_c over the remaining generators and their inverses that represents the same element as c, and apply the free monoid homomorphism that maps c to w_c. This corresponds to changing the normal form used by replacing every occurrence of c with w_c.

Changing the normal form is more difficult. In [19], Section 5, Ciobanu and Elder show that changing between quasi-geodesic normal forms will not affect whether or not the solution language to a given system is EDT0L. This relies on the fact that in a hyperbolic group G, the set of all pairs $(u,\ v)$ of $(\lambda,\ \mu)$-quasi-geodesics such that $u =_G v$ is accepted by a 2-variable finite-state automaton. Unfortunately, this doesn't work in $\mathbb{Z}^2$, so a different approach would be required to preserve the EDT0L status of the language when changing between normal forms in virtually abelian groups.

We now study the growth series of the solution language to a system of equations in virtually abelian groups. For this, we need the following result on polyhedral sets.

Lemma 4.3.19. *Let $A \subseteq (\mathbb{Z}^k)^n$ be the solution set to a system of twisted equations in $\mathbb{Z}^k$ (with n variables). Then A is a polyhedral subset of $\mathbb{Z}^{kn}$.*

Proof By Lemma 4.3.3, A may be viewed as the set of solutions to a system of (non-twisted) equations in $\mathbb{Z}$, with kn variables, with each element of A given as a vector in $\mathbb{Z}^{kn}$, with respect to the standard basis of $\mathbb{Z}^{kn}$. Now a single such equation in $\mathbb{Z}$ may be expressed as

$$\sum_{i=1}^{kn} a_i x_i = b$$

for variables x_i and constants $a_i,\ b \in \mathbb{Z}$. Therefore the solution set to such an equation has the form

$$\left\{(x_1,\ \ldots,\ x_{kn}) \in \mathbb{Z}^{kn} \;\middle|\; \sum_{i=1}^{kn} a_i x_i = b\right\} = \left\{\mathbf{x} \in \mathbb{Z}^{kn} \,\middle|\, \mathbf{a}\cdot\mathbf{x} = b\right\}$$

and is thus an elementary set (see Definition 4.2.2). The solution set to a system of equations is then the intersection of finitely many elementary sets, and is therefore a polyhedral set by the definition.

□

We can now use the polyhedral structure of solution sets in $\mathbb{Z}^k$ to prove the following Proposition about the growth of solution languages in virtually abelian groups.

Proposition 4.3.20. *The solution language of any system of equations in a virtually abelian group has rational growth series.*

Proof As before, let G be a virtually abelian group and let $\mathbb{Z}^k$ denote a free abelian normal subgroup of finite index, and T a choice of transversal. The normal form on $\mathbb{Z}^k$ given by the standard basis vectors is denoted η. By Lemma 4.3.7, the solution language is given by a finite union of sets of the form

$$\{(h_1\eta)t_1\#(h_2\eta)t_2\#\cdots\#(h_n\eta)t_n \mid (h_1,\ \ldots,\ h_n) \in A_{\mathbf{t}}\} \tag{4.8}$$

where n is the number of variables, $\mathbf{t} = (t_1,\ \ldots,\ t_n)$ is some subset of T^n, and each $A_{\mathbf{t}}$ is the solution set to some system of twisted equations in $\mathbb{Z}^k$.

Now, the word $(h_1\eta)t_1\#\cdots\#(h_n\eta)t_n \in (T \cup \{\#\} \cup \{\pm e_i \mid 1 \leq i \leq kn\})^*$ has length $2n - 1 + |(h_1,\ \ldots,\ h_n)|$. So the growth series of the set (4.8) is equal to the growth series of $A_{\mathbf{t}}$ multiplied by z^{2n-1}. That is,

$$z^{2n-1}\sum_{m=0}^{\infty} \#\{(h_1,\ \ldots,\ h_n) \in A_{\mathbf{t}} \mid |(h_1,\ \ldots,\ h_t)| = m\}z^m.$$

Since each $A_{\mathbf{t}}$ is polyhedral by Lemma 4.3.19, Corollary 4.2.8 implies that their growth series (with the weight of each generator equal to 1 in this case) is rational, and hence the growth series of (4.8) is also rational. So the growth series of the solution language is a finite sum of rational functions, and is therefore rational. □

Remark 4.3.21. We note that the language above will not be context-free in general. For example, suppose the underlying group is $\mathbb{Z} = \langle x \rangle$, and consider the equation $X = Y = Z$ (more formally the system of equations $XY^{-1} = YZ^{-1} = 1$). In the notation of this thesis, the set of solutions is $\{a^m \# a^m \# a^m \mid m \in \mathbb{Z}\}$, which is not context-free over the alphabet $\{a,\ a^{-1},\ \#\}$ by standard techniques.

Thus we have a large class of EDT0L languages, with rational growth series, which are not, in general, context-free.

4.4 Relative growth of algebraic sets

We now study the nature of algebraic sets from a different point of view. Expanding on the theme of Proposition 4.3.20, we consider the *growth* of algebraic sets, this time as sets of tuples of group elements, with respect to a metric inherited from the word metric on the group.

The usual notion of the growth function of a group can be altered by restricting to a subset. This is known as *relative growth*. The study of relative growth of subgroups in particular has attracted significant interest, for example Davis-Olshanskii [28], and recently Cordes-Russell-Spriano-Zalloum [25]. Here, we define and study the relative growth of algebraic sets. Since such a set is a subset of G^n, rather than G itself, we must decide how to assign lengths to tuples. We do this in perhaps the most obvious way, by taking the sum of the lengths of the components (see Definition 4.4.2).

Since the growth of virtually abelian groups is always polynomial (that is, the number of elements of length n is at most polynomial in n), it is clear that the same will be true of algebraic sets. Instead, we study the growth series, the formal power series associated to the relative growth function of an algebraic set, and show that this is always a rational function (see Theorem 4.4.3). This means that there exists a set of unique geodesic representatives for each algebraic set, which has rational growth series as a language.

An alternative approach which avoids the need to define the length of n-tuples of group elements is to study the *multivariate* growth series, the formal power series in n variables, which correspond to the n variables of the system of equations in question (see Definition 4.4.2). In this case, we have the weaker result that the series is always holonomic (Corollary 4.4.21).

From now on we will assume that G is virtually abelian with a normal, finite index subgroup isomorphic to $\mathbb{Z}^k$ for some positive integer k.

Definition 4.4.1. Let G be generated by a finite set S and suppose S is equipped with a weight function $\|\cdot\|\colon S \to \mathbb{Z}_{>0}$. This naturally extends to S^* so that $\|s_1 s_2 \cdots s_k\| = \sum_{i=1}^{k} \|s_i\|$.

1. Define the weight of a group element as

$$\|g\| = \min\{\|w\| \mid w \in S^*,\ w =_G g\}.$$

   Any word representing g whose weight is equal to $\|g\|$ will be called *geodesic*. This coincides with the usual notion of word length when the weight of each non-trivial generator is equal to 1.
2. Let $V \subseteq G$ be any subset. Then the *relative weighted growth function* of V relative to G, with respect to S, is defined as

$$\sigma_{V\subseteq G,S}(m) = \#\{g \in V \mid \|g\| = m\}.$$

   For simplicity of notation, we will write $\sigma_V(m)$ when the other information is clear from context.
3. The corresponding *weighted growth series* is the formal power series

$$\mathbb{S}_{V\subseteq G,S}(z) = \sum_{m=0}^{\infty} \sigma_{V\subseteq G,S}(m) z^m.$$

Benson proved in [8] that the series $\mathbb{S}_{G\subseteq G}(z)$ is always rational (that is, the standard growth series of G), and Evetts proved in [45] that for any subgroup H of G, the series $\mathbb{S}_{H\subseteq G}(z)$ is always rational. Both of these results hold regardless of the choice of finite weighted generating set. As discussed, we wish to apply these ideas to algebraic sets, which are subsets of G^n in general, for some positive integer n. Therefore, we extend Definition 4.4.1 as follows.

Definition 4.4.2. Let G be generated by a finite inverse closed set S, equipped with a weight function $\|\cdot\|$.

1. We use $\|\cdot\|$ on S to define a function $\|\cdot\|\colon G \to \mathbb{Z}_{\geq 0}$ by

$$\|x\| = \min\{\|v_1\| + \cdots + \|v_k\| \mid v_i \in S,\ v_1 \cdots v_k =_G x\}.$$

2. Let $\mathbf{x} = (x_1,\ \ldots,\ x_n) \in G^n$ be any n-tuple of elements of G. Define the weight of $\mathbf{x}$ as follows:

$$\|\mathbf{x}\| = \sum_{i=1}^{n} \|x_i\|.$$

3. Let $V \subseteq G^n$ be any set of n-tuples of elements. Then the *relative weighted growth function* of V is defined as the function

$$\sigma_{V \subseteq G^n, S}(m) = \#\{\mathbf{x} \in V \mid \|\mathbf{x}\| = m\}.$$

4. The corresponding (univariate) *weighted growth series* is

$$\mathbb{S}_{V \subseteq G^n, S}(z) = \sum_{m=0}^{\infty} \sigma_{V \subseteq G^n, S}(m) z^m \in \mathbb{Q}[[z]].$$

5. The *multivariate growth series* is the formal power series

$$\mathbb{M}_{V \subseteq G^n, S}(z_1, \ldots, z_n) = \sum_{(x_1, \ldots, x_n) \in V} z_1^{\|x_1\|} \cdots z_n^{\|x_n\|} \in \mathbb{Q}[[z_1, z_2, \ldots, z_n]],$$

We will suppress some or all of the subscripts when it is clear what the notation refers to.

With these definitions, we can state the main result of this section.

Theorem 4.4.3. *Let G be a virtually abelian group. Then every algebraic set of G has rational weighted growth series with respect to any finite generating set.*

4.4.1 Structure of virtually abelian groups

To prove the Theorem, we will extend the framework used in [8] and [45] to apply to our setting. We give the necessary definitions and results below, and refer the reader to the above mentioned articles for full details.

Definition 4.4.4. As above, fix a finite inverse closed generating set S for G.

1. We define $A = S \cap \mathbb{Z}^k$ and $B = S \setminus A$. Any word in B^* will be called a *pattern*.
2. Let $A = \{x_1, \ldots, x_r\}$, and $\pi = y_1 y_2 \cdots y_l$ be some pattern (with each $y_i \in B$). Then a word in S^* of the form

$$w = x_1^{i_1} x_2^{i_2} \cdots x_r^{i_r} y_1 x_1^{i_{r+1}} x_2^{i_{r+2}} \cdots x_r^{i_{2r}} y_2 \cdots y_l x_1^{i_{lr+1}} x_2^{i_{lr+2}} \cdots x_r^{i_{lr+r}} \tag{4.9}$$

   for non-negative integers i_j is called a *π-patterned word*. For a fixed $\pi \in B^*$, denote the set of all such words by W^π.

This definition allows us to identify patterned words with vectors of non-negative integers, by focussing on just the powers of the generators in A as follows.

Definition 4.4.5. Fix a pattern π of length l, and write $m_\pi = lr + r$. Define a bijection $\phi_\pi \colon W^\pi \to \mathbb{Z}_{\geq 0}^{m_\pi}$ via

$$\phi_\pi \colon x_1^{i_1} x_2^{i_2} \cdots x_r^{i_r} y_1 x_1^{i_{r+1}} x_2^{i_{r+2}} \cdots x_r^{i_{2r}} y_2 \cdots y_l x_1^{i_{lr+1}} x_2^{i_{lr+2}} \cdots x_r^{i_{lr+r}} \mapsto (i_1, i_2, \ldots, i_{lr+r}).$$

This bijection will allow us to count subsets of $\mathbb{Z}^{m_\pi}$ in place of sets of words. We apply the weight function $\|\cdot\|$ to $\mathbb{Z}^{m_\pi}$ in the natural way, weighting each coordinate with the weight of the corresponding $x \in A$. More formally, we have

$$\|(i_1, \ldots, i_{m_\pi})\| := \sum_{j=1}^{m_\pi} i_j \|x_{j \bmod^+ r}\|.$$

Then ϕ_π preserves the weight of words in W^π, up to a constant:

$$\|w\phi_\pi\| = \|w\| - \|\pi\|.$$

Fix a transversal T for the cosets of $\mathbb{Z}^k$ in G. Note that, since $\mathbb{Z}^k$ is a normal subgroup, we can move each y_i in the word (4.9) to the right, modifying only the generators from A, and we have $\overline{w} \in \mathbb{Z}^k\overline{\pi}$. Thus $\overline{W^\pi} \subset \mathbb{Z}^k t_\pi$ for some $t_\pi \in T$ where $\overline{\pi} \in \mathbb{Z}^k t_\pi$.

It turns out that we can pass from a word $w \in W^\pi$ to the normal form (with respect to T and the standard basis for $\mathbb{Z}^k$) of the element $\overline{w}$ using an integral affine transformation.

Proposition 4.4.6 (Section 12 of [8])**.** *For each pattern $\pi \in B^*$, there exists an integral affine transformation $\mathcal{A}^\pi \colon \mathbb{Z}^{m_\pi}_{\geq 0} \to \mathbb{Z}^k$ such that $\overline{w} = (w\phi_\pi \mathcal{A}^\pi)\, t_\pi$ for each $w \in W^\pi$.*

Observe that the union $\bigcup W^\pi$ of patterned sets taken over all patterns π contains a geodesic representative for every group element (since any geodesic can be arranged into a patterned word without changing its image in the group). However, this is an infinite union, since patterns are simply elements of B^*.

Consider the extended generating set $\widetilde{S}$ defined as follows:

$$\widetilde{S} = \{s_1 s_2 \cdots s_c \mid s_i \in S,\ 1 \leq c \leq [G \colon \mathbb{Z}^k]\}.$$

Define a weight function $\|\cdot\|_\sim \colon \widetilde{S} \to \mathbb{Z}_{>0}$ via $\|s_1 s_2 \cdots s_c\|_\sim = \sum_{i=1}^{c} \|s_i\|$. Notice that although group elements will have different lengths with respect to this new generating set, we have $\|g\|_\sim = \|g\|$ for any $g \in G$. Thus the weighted growth functions, and hence series, of any subset $V \subseteq G$ with respect to S and $\widetilde{S}$ are equal. The following fact shows that passing to this extended generating set means we only need consider finitely many patterns.

Proposition 4.4.7 (11.3 of [8])**.** *Every element of G has a geodesic representative with a pattern whose length (with respect to $\widetilde{S}$) does not exceed $[G \colon \mathbb{Z}^k]$.*

Definition 4.4.8. Let P denote the set of patterns of length at most $[G \colon \mathbb{Z}^k]$ (with respect to $\widetilde{S}$).

From now on we will implicitly work with the extended generating set, allowing us to restrict ourselves to the finite set of patterns P.

We now reduce each W^π so that we have only a single geodesic representative for each element of G.

Theorem 4.4.9 (Section 12 of [8]). *For each $\pi \in P$, there exists a set $U^\pi \subset W^\pi$ such that every word in U^π is geodesic, every element in G is represented by some word in $\bigcup_{\pi \in P} U^\pi$, and no two words in $\bigcup_{\pi \in P} U^\pi$ represent the same element. Furthermore, each $U^\pi \phi_\pi$ is a polyhedral set in $\mathbb{Z}^{m_\pi}$.*

Corollary 4.4.10. *The weighted growth series $\mathbb{S}_{G \subseteq G}(z)$ of G is rational, with respect to all generating sets.*

Proof The growth series $\mathbb{S}_{G \subseteq G}$ is precisely the growth series of $\bigcup_{\pi \in P} U^\pi$ as a set. From Definition 4.4.5 we have

$$\mathbb{S}_{U^\pi \subseteq G}(z) = z^{\|\pi\|} \mathbb{S}_{U^\pi \phi_\pi}(z)$$

and thus

$$\mathbb{S}_{G \subseteq G}(z) = \sum_{\pi \in P} z^{\|\pi\|} \mathbb{S}_{U^\pi \phi_\pi}(z)$$

is rational, since each $\mathbb{S}_{U^\pi \phi_\pi}(z)$ is a positive polyhedral set and hence rational by Proposition 4.2.7

□

4.4.2 Univariate growth series of algebraic sets

We can now demonstrate our main result. This will be a consequence of a more general rationality criterion. First, we make the following definitions, extending the framework explained above to n-tuples of group elements.

Definition 4.4.11. Let $\underline{\pi} = (\pi_1, \ldots, \pi_n) \in P^n$ be a tuple of patterns, with respect to $\widetilde{S}$.

1. Let $W^{\underline{\pi}} = W^{\pi_1} \times \cdots \times W^{\pi_n}$ and $U^{\underline{\pi}} = U^{\pi_1} \times \cdots \times U^{\pi_n} \subset (S^*)^n$. Note that $U^{\underline{\pi}}$ is a polyhedral set by Proposition 4.2.3.

2. Let $m_{\underline{\pi}} = \sum_{i=1}^{n} m_{\pi_i}$, and $\|\underline{\pi}\| = \sum_{i=1}^{n} \|\pi_i\|$.
3. Define a map $\phi_{\underline{\pi}} \colon W^{\underline{\pi}} \to \mathbb{Z}_{\geq 0}^{m_{\underline{\pi}}}$ in the natural way via

$$(w_1, \ \ldots, \ u_n) \mapsto (w_1\phi_{\pi_1}, \ \ldots, \ u_n\phi_{\pi_n}).$$

   As in the above discussion, $\phi_{\underline{\pi}}$ preserves the weight of words, up to a constant, i.e.

$$\|(w_1, \ \ldots, \ u_n)\phi_{\underline{\pi}}\| = \sum_{i=1}^{n} \|w_i\| - \|\underline{\pi}\|.$$

4. Given $\mathcal{A}^{\pi_i}$ as in Proposition 4.4.6, define an integral affine transformation $\mathcal{A}^{\underline{\pi}} \colon \mathbb{Z}_{\geq 0}^{\underline{\pi}} \to \mathbb{Z}^{kn}$ in the natural way via

$$(x_1, \ \ldots, \ x_n) \mapsto (x_1\mathcal{A}^{\pi_1}, \ \ldots, \ x_n\mathcal{A}^{\pi_n}) \in \mathbb{Z}^k \times \cdots \times \mathbb{Z}^k.$$

Now we define a class of subsets of finitely generated virtually abelian groups which is particularly amenable to study using the tools we have described.

Definition 4.4.12. Let T be a choice of transversal for the finite index normal subgroup $\mathbb{Z}^k$. A subset $V \subseteq G^n$ will be called *coset-wise polyhedral* if, for each $\mathbf{t} = (t_1, \ \ldots, \ t_n) \in T^n$, the set

$$V_{\mathbf{t}} = \left\{ \left(g_1 t_1^{-1}, \ g_2 t_2^{-1}, \ \ldots, \ g_n t_n^{-1}\right) \mid (g_1, \ \ldots, \ g_n) \in V, \ g_i \in \mathbb{Z}^k t_i \right\} \subseteq \mathbb{Z}^{kn}$$

is polyhedral.

Remark 4.4.13. Note that the definition is independent of the choice of T. Indeed, suppose that we chose a different transversal T' so that for each $t_j \in T$ we have $t'_j \in T'$ with $\mathbb{Z}^k t_j = \mathbb{Z}^k t'_j$. Then there exists $y_j \in \mathbb{Z}^k$ with $t_j = y_j t'_j$ for each j, and so $g{t'_j}^{-1} = g_j t_j^{-1} y_i$ for any $g \in \mathbb{Z}^k t_j = \mathbb{Z}^k t'_j$. So changing the transversal changes the set $V_{\mathbf{t}}$ by adding a constant vector $(y_1, \ \ldots, \ y_n)$, and so it remains polyhedral by Proposition 4.2.5.

As an example of Definition 4.4.12, we provide a brief proof that subgroups are coset-wise polyhedral.

Proposition 4.4.14. *Let G be a virtually abelian group, with normal free abelian subgroup $\mathbb{Z}^k$, and let H be any subgroup. Then H is coset-wise polyhedral.*

Proof By the Second Isomorphism Theorem, H is itself virtually abelian, with finite-index (free) abelian subgroup $H \cap \mathbb{Z}^k$. Furthermore, $c := [H \colon H \cap \mathbb{Z}^k] \leq [G \colon \mathbb{Z}^k] =: d$. Choose a set of representatives $\{t_1, \ldots, t_c\}$ for the cosets of $H \cap \mathbb{Z}^k$ in H, and extend this to a set of representatives $\{t_1, \ldots, t_c, t_{c+1}, \ldots, t_d\}$ for the cosets of $\mathbb{Z}^k$ in G. For each t_i with $i \leq c$, the set

$$H_{t_i} = \left\{ ht_i^{-1} \,\middle|\, h \in H,\ h \in \mathbb{Z}^k t_i \right\} = \left\{ ht_i^{-1} \,\middle|\, h \in \left(H \cap \mathbb{Z}^k \right) t_i \right\} = H \cap \mathbb{Z}^k.$$

For $i > c$, H_{t_i} is empty. Now since $H \cap \mathbb{Z}^k$ is free abelian, it is a polyhedral set when viewed as a subset of $\mathbb{Z}^k$. The empty set is also polyhedral (as, say, the intersection of a pair of disjoint hyperplanes). Hence H is coset-wise polyhedral. □

In light of Proposition 4.4.14, the following Theorem is in some sense a generalisation of Theorem 3.3 of [45], namely that every subgroup has rational relative growth series.

Theorem 4.4.15. *Let G be virtually abelian, with normal free abelian subgroup $\mathbb{Z}^k$, and let S be any finite weighted generating set. If $V \subseteq G^n$ is coset-wise polyhedral, then the weighted growth series $\mathbb{S}_{V \subseteq G^n, S}(z)$ is a rational function.*

Proof Fix a transversal T. For each $\mathbf{t} \in T^n$, let $P_{\mathbf{t}} \subset P^n$ denote the set of n-tuples of patterns of the form $\underline{\pi} = (\pi_1, \ldots, \pi_n)$ where each $\pi_i \in \mathbb{Z}^k t_i$. Let $U^{\underline{\pi}} = U^{\pi_1} \times \cdots \times U^{\pi_n} \subset (S^*)^n$. Then by Theorem 4.4.9, the disjoint union $\bigcup_{\underline{\pi} \in P_{\mathbf{t}}} U^{\underline{\pi}}$ consists of exactly one n-tuple of geodesic representatives for each n-tuple in $\mathbb{Z}^k t_1 \times \cdots \times \mathbb{Z}^k t_n$. We are only interested in n-tuples of elements which lie in the set V. Each element of V lies in a unique product of cosets, so we partition V into such products:

$$V = \bigcup_{\mathbf{t} \in T^n} \left\{ (g_1, \ldots, g_n) \in V \mid g_i \in \mathbb{Z}^k t_i \right\} = \bigcup_{\mathbf{t} \in T^n} \left\{ (g_1, \ldots, g_n) \in G^n \mid (g_1 t_1^{-1}, \ldots, g_n t_n^{-1}) \in V_{\mathbf{t}} \right\} \tag{4.10}$$

Now, for a fixed $\mathbf{t}$, $(g_1, \ldots, g_n)$ has a unique geodesic representative in the set $U^{\underline{\pi}}$, for some $\underline{\pi} \in P_{\mathbf{t}}$ determined by $\mathbf{t}$. So the growth series of each component in the union (4.10) is equal to the growth series of the set

$$\bigcup_{\underline{\pi} \in P_{\mathbf{t}}} \{(u_1, \ldots, u_n) \in U^{\underline{\pi}} \mid (u_1\phi_{\pi_1}\mathcal{A}^{\pi_1}, \ldots, u_n\phi_{\pi_n}\mathcal{A}^{\pi_n}) \in V_{\mathbf{t}}\} = \bigcup_{\underline{\pi} \in P_{\mathbf{t}}} V_{\mathbf{t}}(\phi_{\underline{\pi}}\mathcal{A}^{\underline{\pi}})^{-1} \cap U^{\underline{\pi}}.$$

Applying the map $\phi_{\underline{\pi}}$ to a component of the union yields the set

$$\{(u_1\phi_1, \ldots, u_n\phi_n) \in U^{\underline{\pi}}\phi_{\underline{\pi}} \mid (u_1\phi_{\pi_1}\mathcal{A}^{\pi_1}, \ldots, u_n\phi_{\pi_n}\mathcal{A}^{\pi_n}) \in V_{\mathbf{t}}\} = V_{\mathbf{t}}\,(\mathcal{A}^{\underline{\pi}})^{-1} \cap U^{\underline{\pi}}\phi_{\underline{\pi}}.$$

Now by Propositions 4.2.3 and 4.2.5, this last set is polyhedral, and so has rational growth. Since both T^n and $P_{\mathbf{t}}$ are finite, the growth series of V is a finite sum of growth series of sets of the form $V_{\mathbf{t}}\,(\mathcal{A}^{\underline{\pi}})^{-1} \cap U^{\underline{\pi}}\phi_{\underline{\pi}}$ (each multiplied by $z^{\|\underline{\pi}\|}$ for the appropriate $\underline{\pi}$) and is therefore rational, finishing the proof. □

We can now prove the main result of this section.

Proof (of Theorem 4.4.3.) Let $\mathcal{S}$ denote an algebraic set. By Theorem 4.4.15, it suffices to show that $\mathcal{S}$ is coset-wise polyhedral. By Lemma 4.3.7 we have

$$\begin{aligned} \mathcal{S} &= \bigcup_{(t_1,\ldots,t_n)\in B} \left\{(h_1t_1, \ldots, h_nt_n) \mid (h_1, \ldots, h_n) \in \mathcal{S}_{(t_1,\ldots,t_n)}\right\} \\ &= \bigcup_{(t_1,\ldots,t_n)\in T^n} \left\{(h_1t_1, \ldots, h_nt_n) \mid (h_1, \ldots, h_n) \in \mathcal{S}_{(t_1,\ldots,t_n)}\right\} \end{aligned}$$

where each $\mathcal{S}_{(t_1,\ldots,t_n)}$ is the solution set to some system of twisted equations in $\mathbb{Z}^k$ (and is empty for $(t_1, \ldots, t_n) \notin B$. By Lemma 4.3.19, each $\mathcal{S}_{(t_1,\ldots,t_n)}$ is a polyhedral subset of $\mathbb{Z}^{kn}$, and thus $\mathcal{S}$ is coset-wise polyhedral as required. □

For clarity, we explicitly state the description of algebraic sets in terms of polyhedral sets, which is a consequence of the proof of Theorem 4.4.3.

Corollary 4.4.16. *Let G be a finitely generated virtually abelian group (with a finite-index free abelian normal subgroup $\mathbb{Z}^k$ for some k). Choose a transversal T.*

Suppose $\mathcal{S} \subset G^n$ is an algebraic set. Then for each $\mathbf{t} = (t_1, \ldots, t_n) \in T^n$, there exists a polyhedral set $\mathcal{S}_{\mathbf{t}} \subseteq \mathbb{Z}^{kn}$ such that $\mathcal{S}$ decomposes as a finite disjoint union:

$$\mathcal{S} = \bigcup_{\mathbf{t} \in T^n} \left\{ (g_1, \ldots, g_n) \in \mathbb{Z}^k t_1 \times \cdots \times \mathbb{Z}^k t_n \,\middle|\, (g_1 t_1^{-1}, \ldots, g_n t_n^{-1}) \in \mathcal{S}_{\mathbf{t}} \right\}.$$

4.4.3 Multivariate Growth Series

We now turn to the *multivariate* growth series (see Definition 4.4.2) and demonstrate that for an algebraic set V, the multivariate growth series $\mathbb{M}_{V \subseteq G^n, S}(z)$ is a *holonomic* function.

Definition 4.4.17. For clarity, we also define the multivariate growth series of a language. Let L be a language over some finite weighted alphabet $A = \{a_1, \ldots, a_r\}$ (with weights denoted $\|a_i\|$) and let $|w|_i$ denote the number of occurrences of a_i in a word $w \in L$. The *weighted multivariate growth series* of L is the formal power series

$$\sum_{w \in L} z_1^{\|a_1\| \cdot |w|_1} z_2^{\|a_2\| \cdot |w|_2} \cdots z_r^{\|a_r\| \cdot |w|_r} \in \mathbb{Q}[[z_1, z_2, \ldots, z_r]].$$

Let $\mathbf{z} = (z_1, \ldots, z_n)$ and ∂_{z_i} denote the partial derivative with respect to z_i.

Definition 4.4.18. A multivariate function $f(\mathbf{z})$ is *holonomic* if the span of the set of partial derivatives

$$\{\partial_{z_1}^{j_1} \partial_{z_2}^{j_2} \cdots \partial_{z_n}^{j_n} f(\mathbf{z}) \mid j_i \in \mathbb{Z}_{\geq 0}\}$$

over the ring of rational functions $\mathbb{C}(\mathbf{z})$ is finite-dimensional.

From this definition, we see that a function is holonomic if and only if it satisfies a linear differential equation involving partial derivatives of finite order, and rational coefficients, for each variable z_i. Holonomic functions thus generalise the class of algebraic functions. For a more complete introduction to this topic, see [49].

In recent work [9], Bishop extends results of Massazza [71] to show that a certain class of formal languages has holonomic multivariate growth series. The following Lemma follows easily from Proposition 4.3 of [9], and the fact that holonomic functions are closed under algebraic substitution (Theorem B.3 of [49]).

Lemma 4.4.19. *The weighted multivariate growth series of a polyhedral set (viewed as a formal language over the alphabet consisting of standard basis vectors) is holonomic.*

As in the univariate case, we prove a more general statement about coset-wise polyhedral subsets.

Theorem 4.4.20. *Let $V \subset G^n$ be a coset-wise polyhedral set of tuples of elements of a virtually abelian group G. Then the weighted multivariate growth series $\mathbb{M}_{V \subseteq G^n, S}$ is holonomic, with respect to any generating set S.*

Proof Following the proof of Theorem 4.4.15, the coset-wise polyhedral set V is represented by a finite disjoint union of polyhedral sets in $\mathbb{Z}^{kn}$, where k is the dimension of the finite-index free abelian normal subgroup of G.

Lemma 4.4.19 implies that the weighted multivariate growth series of each of these polyhedral sets (in the sense of Definition 4.4.17) is holonomic. These series will involve kn variables, say

$$(z_{11}, \ldots, z_{1k},\ z_{21}, \ldots, z_{2k},\ \ldots,\ z_{n1}, \ldots, z_{nk}).$$

To obtain the weighted multivariate growth series of V (in the sense of Definition 4.4.2), we need only set each $z_{ij} = z_i$ and multiply each of the finitely many growth series by an appropriate constant to account for the contribution from each pattern $\underline{\pi}$. The closure properties of holonomic functions (Theorem B.3 of [49]) ensure that the resulting growth series is still holonomic (with variables $z_1, \ldots, z_n$ corresponding to the variables in the system of equations). □

We currently do not have an example of an algebraic set in a virtually abelian group where the multivariate growth series is not rational. Given how many well-studied subsets of virtually abelian groups seem to have rational growth, there is reason to believe this might be rational as well. However, using Bishop's result [9], we are able to show that they are (at least) holonomic.

Corollary 4.4.21. *An algebraic set in a virtually abelian group has holonomic weighted multivariate growth series.*

Proof The proof of Theorem 4.4.3 shows that any algebraic set is coset-wise polyhedral. □

Chapter 5

Equations in extensions

5.1 Introduction

This chapter is based on the work of the author [65].

The focus of this chapter will be the stability of the class of groups where solutions to systems of equations can be expressed as EDT0L languages under various constructions. We use these facts together to prove that groups that are virtually direct products of hyperbolic groups belong to this class. As a corollary to this, we also obtain the solutions to systems of equations in dihedral Artin groups can be expressed as EDT0L languages.

Theorem 5.1.1 collects the main results in this chapter. The format used to express solutions as words is explained in the preliminaries (Section 5.2).

Theorem 5.1.1. *Let G and H be groups where solution languages to systems of equations are EDT0L, with respect to normal forms η_G and η_H, respectively, and EDT0L systems are constructible in $\mathsf{NSPACE}(f)$, for some f. Then in the following groups, solutions to systems of equations are EDT0L, and an EDT0L system can be constructed in non-deterministic f-space:*

1. *$G \wr F$, for any finite group F (Proposition 5.4.5);*
2. *$G \times H$ (Proposition 5.4.6);*

3. *Any finite index subgroup of G (Proposition 5.5.3);*

In the following groups, solutions to systems of equations are EDT0L, and an EDT0L system can be constructed in $\mathsf{NSPACE}(n^4 \log n)$*:*

4. *Any group that is virtually a direct product of hyperbolic groups (Corollary 5.6.9);*
5. *Dihedral Artin groups (Corollary 5.6.10).*

If η_G *and* η_H *are both quasi-geodesic or regular, then the same will be true for the normal forms used in (1), (2) and (3). It is possible to choose normal forms for the groups that are virtually direct products of hyperbolic groups in (4), and dihedral Artin groups in (5) that are regular and quasi-geodesic.*

Whilst an understanding of the set of solutions to a system of equations in a direct product follows immediately from understanding the solutions to the projection onto each of the groups in the direct product, showing that the language can be expressed in the correct format requires more work, which we explore in Section 5.3. This format is also required to prove Theorem 5.1.1(1).

The proof of Theorem 5.1.1(4) is based on Ciobanu, Holt and Rees' proof of the fact the satisfiability of systems of equations in these groups is decidable [22], in a work that also looks at recognisable constraints. We show that the addition of recognisable constraints to any system of equations preserves the property of having an EDT0L solution language, and use this to show that the class of groups where systems of equations have EDT0L solutions is closed under passing to finite index subgroups.

Section 5.2 covers the preliminaries of the topics used. In Section 5.3, we prove Proposition 5.3.7 on the parallel concatenation of words, which is an important part of the proofs of the stability of groups where systems of equations have EDT0L solution languages under direct products (Proposition 5.4.6), and wreath products with finite groups (Proposition 5.4.5). Section 5.4 covers the proofs of those propositions.

Section 5.5 includes the addition of recognisable constraints to equations with EDT0L solutions, and is used to prove that the property of systems of equations having EDT0L solution languages passes to finite index subgroups, with respect to the Schreier normal form, based on the normal form used in the finite index overgroup (Proposition 5.5.3). Section 5.6 concludes with the proof that systems of equations in groups that are virtually direct products of hyperbolic groups have EDT0L solution languages.

Notation 5.1.2. We introduce some notation to be used throughout the chapter.

- Let G be a group. We use $\mathrm{FIN}(G)$ to denote the class of groups that contain G as a finite index subgroup;
- If L is a language over an alphabet Σ, we use L^c to denote the complement of L within Σ^*.

5.2 Preliminaries

5.2.1 Dihedral Artin groups

We briefly define dihedral Artin groups. An application of Corollary 5.6.9 is that solution sets to systems of equations in these groups form EDT0L languages.

Definition 5.2.1. A *dihedral Artin group* DA_m, where $m \geq 2$, is defined by the presentation

$$\langle a,\ b \mid \underbrace{aba\cdots}_{m} = \underbrace{bab\cdots}_{m} \rangle.$$

The following lemma is widely known. A brief sketch of the proof can be found in [22], Section 2.

Lemma 5.2.2. *A dihedral Artin group is virtually a direct product of free groups.*

5.2.2 Schreier generators

We use Schreier generators, along with the normal form they induce, in order to show that the class of groups where systems of equations have EDT0L languages of solutions is stable under passing to finite index subgroups. This subsection is based on Section 1.4 of [56].

We start with the definition of Schreier generators.

Definition 5.2.3. Let G be a group, generated by a finite set Σ, H be a finite index subgroup of G, and T be a right transversal of H in G. For each $g \in G$, let $\bar{g}$ be the (unique) element of T that lies in the coset Hg. The *Schreier generating set* for H, with respect to T and Σ, is defined to be

$$Z = \{tx\overline{tx}^{-1} \mid t \in T,\ x \in \Sigma\}.$$

Whilst the fact that the Schreier generating set is a finite generating set for H is widely known, we include a proof, as we later use ideas from the proof.

Lemma 5.2.4. *Let G be a group, generated by a finite set Σ, H be a finite index subgroup of G, and T be a right transversal of H in G. Let Z be the Schreier generating set for H. Then Z is finite, and*

$$H = \langle Z \rangle.$$

Proof We first show that

$$Z^{-1} = \{tx^{-1}\overline{tx^{-1}}^{-1} \mid t \in T,\ x \in \Sigma\}.$$

Let $S = \{tx^{-1}\overline{tx^{-1}}^{-1} \mid t \in T,\ x \in \Sigma\}$. Let $g = \overline{tx}x^{-1}t^{-1} = (tx\overline{tx}^{-1})^{-1} \in Z^{-1}$. Let $v = \overline{tx}$. Note that $\overline{vx^{-1}} = \overline{\overline{tx}x^{-1}} = t$. Then $g = vx^{-1}\overline{vx^{-1}}^{-1} \in S$, and so $Z^{-1} \subseteq S$.

Let $g = tx^{-1}\overline{tx^{-1}}^{-1} \in S$. Then $g^{-1} = \overline{tx^{-1}}xt^{-1}$. Let $v = \overline{tx^{-1}}$. Then $\overline{vx} = t$, and so $g^{-1} = vx\overline{vx}^{-1} \in Z$. We can conclude that $S \subseteq Z^{-1}$.

The fact that Z is finite follows from the fact that T and Σ are finite. Let t_0 be the unique element of $T \cap H$. Let $h \in H$ (this will usually be 1, but does not need to be). Then $t_0^{-1}ht_0 = a_1 \cdots a_n$, for some $a_1,\ \ldots,\ a_n \in \Sigma^{\pm}$. Let $t_i = \overline{a_1 \cdots a_i}$ for each

$i \in \{1, \ldots, n\}$, and note $t_n = t_0$. We have

$$h = (t_0 a_1 t_1^{-1})(t_1 a_2 t_2^{-1}) \cdots (t_{n-1} a_n t_n^{-1}).$$

Note that $\overline{t_i a_{i+1}} = \overline{\overline{a_1 \cdots a_i} a_{i+1}} = t_{i+1}$, and so

$$h = (t_0 a_1 \overline{t_0 a_1}^{-1})(t_1 a_2 \overline{t_1 a_2}^{-1}) \cdots (t_{n-1} a_n \overline{t_{n-1} a_n}^{-1}).$$

Each of the parenthesised terms lie in Z if $a_i \in \Sigma$, or S if $a_i \in \Sigma^{-1}$. Since $S = Z^{-1}$, we have $h \in \langle Z \rangle$. □

The proof of Lemma 5.2.4 induced a normal form for the finite index subgroup, with respect to the Schreier generating set. We now give a formal definition of this normal form.

Definition 5.2.5. Let G be a group, generated by a finite set Σ, H be a finite index subgroup of G, and T be a right transversal of H in G. Let Z be the Schreier generating set for H. Fix a normal form η for (G, Σ).

We define the *Schreier normal form* ζ for (H, Z), with respect to η, as follows. Let $h \in H$, and suppose $h\eta = a_1 \cdots a_n$, where $a_1, \ldots, a_n \in \Sigma^{\pm}$. Let t_0 be the unique element of $T \cap H$, and define $t_i = \overline{a_1 \cdots a_i}$. Define $h\zeta$ by

$$h\zeta = (t_0 a_1 \overline{t_0 a_1}^{-1})(t_1 a_2 \overline{t_1 a_2}^{-1}) \cdots (t_{n-1} a_n \overline{t_{n-1} a_n}^{-1}). \quad (5.1)$$

The fact that this indeed defines an element of H, and equals h is contained in the proof of Lemma 5.2.4.

If the normal form from the finite index overgroup is regular or quasi-geodesic, then the Schreier normal form is regular or quasi-geodesic, respectively. The latter requires an additional lemma that we prove later, however we can show that regularity is preserved without additional results.

Lemma 5.2.6. *Let G be a group, generated by a finite set Σ, H be a finite index subgroup of G, and T be a right transversal of H in G. Let Z be the Schreier generating set for H. Fix a normal form η for (G, Σ).*

Let ζ be the Schreier normal form with respect to η, as in (5.1)*. If η is regular with respect to Σ, then ζ is regular with respect to Z.*

Proof We will extend ζ to the whole of G, with respect to the generating set $Z \cup \{txu^{-1} \mid u,\ t \in T,\ x \in \Sigma\}$. Let $g \in G$, and suppose $t_0 g t_0^{-1}\eta = a_1 \cdots a_n$ where each $a_i \in \Sigma^{\pm}$. Define $\tilde{\zeta}\colon G \to ((Z \cup \{txu^{-1} \mid u,\ t \in T,\ x \in \Sigma\})^{\pm})^*$ by

$$g\tilde{\zeta} = (t_0 a_1 \overline{t_0 a_1}^{-1})(t_1 a_2 \overline{t_1 a_2}^{-1}) \cdots (t_{n-1} a_n t_0^{-1}).$$

Note that $\tilde{\zeta}$ is an extension of ζ. We will first show that $\tilde{\zeta}$ is regular, then use an intersection to show ζ is regular.

Consider a finite state automaton $\mathcal{A}$ that accepts $\operatorname{im}\eta$, with set of states Q, start state q_0, and set F of accept states. We will construct a new finite state automaton $\mathcal{B}$ to accept $\operatorname{im}\tilde{\zeta}$. Our set of states will be $(Q \times T \times \{0,\ 1\}) \cup \{\lambda\}$, where λ is a new state, our start state will be $(q_0,\ t_0,\ 0)$, and λ will be our only accept state. For each transition $(p,\ a) \to q$ in $\mathcal{A}$, and each $t \in T$, define the following transitions in $\mathcal{B}$:

$$((p,\ t,\ 0),\ a) \to (q,\ \overline{ta},\ 1),$$

$$((q,\ \overline{ta},\ 1),\ \overline{ta}^{-1}) \to (q,\ \overline{ta},\ 0).$$

For each $q \in Q$ and $t \in T$, we also have a transition

$$((q,\ t,\ 1),\ t_0^{-1}) \to \lambda.$$

By construction, whenever we read ta, we must follow with $\overline{ta}^{-1}$, unless we are going to the accept state (at the end of the word), in which case we follow with t_0^{-1}. As a result, $\mathcal{B}$ only accepts words in $\operatorname{im}\tilde{\zeta}$. Conversely, $\mathcal{B}$ accepts any word in $\operatorname{im}\eta$ after its conversion into a word in $\operatorname{im}\tilde{\zeta}$, and we can therefore conclude that $\mathcal{B}$ accepts $\operatorname{im}\tilde{\zeta}$.

We have that $\operatorname{im}\zeta = \operatorname{im}\tilde{\zeta} \cap (Z^{\pm})^*$. As an intersection of regular languages, this is regular. $\square$

5.3 EDT0L languages about a distinguished letter

Recall that we denote a solution $(g_1, \ldots, g_n)$ to a system of equations in a group G using the word $(g_1\eta)\#\cdots\#(g_n\eta)$. In order to show that groups where systems of equations have EDT0L solution languages are closed under certain types of extension (such as direct products), we are required to prove Proposition 5.3.7, which allows us to concatenate in parallel two EDT0L languages where every word is of the form $u_0\#\cdots\#u_n$.

The following lemma allows us to use different symbols for each # that delimits the group elements, rather than the same one each time. The proof is joint work with Alex Evetts.

Lemma 5.3.1. *Let $n \in \mathbb{Z}_{>0}$, $\{\#, \#_1, \ldots, \#_n\}$ be a set of formal symbols, and Δ be an alphabet, such that $\#, \#_1, \ldots, \#_n \notin \Delta$. Let A be a set of n-tuples of words over Δ. Define languages L and M over $\Delta \cup \{\#\}$ and $\Delta \cup \{\#_1, \ldots, \#_n\}$, respectively, by*

$$L = \{w_1\#w_2\#\cdots\#w_n \mid (w_1, \ldots, w_n) \in A\}$$

$$M = \{w_1\#_1w_2\#_2\cdots\#_{n-1}w_n\#_n \mid (w_1, \ldots, w_n) \in A\}.$$

Let $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then

1. *The language L is EDT0L if and only if M is;*
2. *There exists an EDT0L system for L that is constructible in $\mathsf{NSPACE}(f)$ if and only such an EDT0L system for M exists.*

Proof Applying the monoid homomorphism $\#_1, \ldots, \#_{n-1} \mapsto \#$, $\#_n \mapsto \varepsilon$ maps M to L, so the backward directions of (1) and (2) follow by Theorem 3.3.2.

Suppose L is EDT0L. We will first show that

$$N := \{w_1\#_1w_2\#_2\cdots\#_{n-1}w_n \mid (w_1, \ldots, w_n) \in A\}$$

is EDT0L. Consider an EDT0L system $\mathcal{H}_L = (\Sigma \sqcup \{\#\}, C, \perp, \mathcal{R})$ that accepts

L, and that is constructible in $\mathsf{NSPACE}(f)$. Note that we can assume our start word is a single letter, instead of a word ω by adding an additional letter $\perp$, and preconcatenating the rational control with an endomorphism $\perp \mapsto \omega$. Let $B \subseteq \mathrm{End}(C^*)$ be an alphabet of $\mathcal{R}$.

We will construct a new EDT0L system from $\mathcal{H}_L$ which will accept M. Let $C_{\text{ind}} = \{c^{i,i+1,\ldots,j} \mid c \in C,\ i,\ j \in \{1,\ \ldots,\ n\}\}$ be the set of symbols obtained by indexing elements of C with a section of the sequence $(1,\ \ldots,\ n)$, including the empty sequence (if $i > j$). By convention, we will consider a letter $c \in C$ indexed by the empty sequence to be equal to c, and so $C \subseteq C_{\text{ind}}$. Our extended alphabet will be C_{ind}. Let $\phi \in B$. Define $\Phi_\phi \subseteq \mathrm{End}(C_{\text{ind}})$ to be the set of all endomorphisms ψ defined to by

$$c^{i,\ldots,j}\psi = x_1^{i_{11},\ldots,i_{1k_1}} x_2^{i_{21},\ldots i_{2k_2}} \cdots x_r^{i_{r1},\ldots,i_{rk_r}},$$

where $x_1 \cdots x_r = c\phi$, and $(i_{11},\ \ldots,\ i_{rk_r}) = (i,\ \ldots,\ j)$. Note that some (or all) of the sequences may be empty. Let $\bar{\mathcal{R}}$ be the rational subsets of endomorphisms of C_{ind}^* obtained from $\mathcal{R}$ by replacing each $\phi \in B$ with Φ_ϕ. The EDT0L system $\mathcal{H}_M = (\Sigma \cup \{\#_1,\ \ldots,\ \#_n,\ C_{\text{ind}},\ \perp_{1,\ldots,n}, \bar{\mathcal{R}})$ will only accept words of the form $a_1^{i_{11},\ldots,i_{1k_1}} \cdots a_r^{i_{r1},\ldots,i_{rk_r}}$, where $(i_{11},\ \ldots,\ i_{rk_r}) = (1,\ \ldots,\ n)$, and $a_1 \cdots a_r \in L$. However, since our alphabet is $\Sigma \cup \{\#_1,\ \ldots,\ \#_n\}$, it can only accept words over that alphabet, which are precisely words of the form $w_0\#_1 \cdots \#_n w_n$, where $w_1\# \cdots \# w_n \in L$, and thus will accept M.

It now remains to show $\mathcal{H}_M$ is constructible in $\mathsf{NSPACE}(f)$. It doesn't require extra memory beyond a constant to add $\perp$ as the start symbol. To write down the new extended alphabet C_{ind}, we just proceed as we would when constructing $\mathcal{H}_L$, but whenever we write a symbol c, we also write all of the indexed versions. To do this we just need to record the letter c we are on, along with the previous index written, so this is still possible in $\mathsf{NSPACE}(f)$.

To output $\bar{R}$, we simply proceed with writing down the finite state automaton that accepts $\mathcal{R}$, and replace each edge labelled by $\phi \in B$ with a set of edges between the same states, labelled with each $\psi \in \Phi_\phi$. To do this, we can compute Φ_ϕ, store it, and remove each $\psi \in \Phi_\phi$ from the memory as we write it. This will require n times as much memory as writing down $\mathcal{R}$, but since n is a constant, it is constructible in $\mathsf{NSPACE}(f)$. □

We introduce the concept of a $(\#_1, \ldots, \#_n)$-separated EDT0L system, which is key in the proof of Proposition 5.3.7.

Definition 5.3.2. Let Σ be an alphabet, and $\#_1, \ldots, \#_n \in \Sigma$. A $(\#_1, \ldots, \#_n)$-*separated EDT0L system* is an EDT0L system $\mathcal{H}$, with a start word of the form $\omega_0 \#_1 \omega_1 \#_2 \cdots \#_n \omega_n$, where $\omega_i \in (\Sigma \backslash \{\#_1, \ldots, \#_n\})^*$ for all i, and such that $\#_i \phi^{-1} = \{\#_i\}$, for all i, and every ϕ in the rational control.

We now show that the class of $(\#_1, \ldots, \#_n)$-separated EDT0L languages is stable under finite unions.

Lemma 5.3.3. *Let L and M be languages over an alphabet Σ that are accepted by $(\#_1, \ldots, \#_n)$-separated EDT0L systems. Then*

1. *The language $L \cup M$ is accepted by a $(\#_1, \ldots, \#_n)$-separated EDT0L system $\mathcal{M}$;*
2. *If L and M are accepted by EDT0L systems constructible in $\mathsf{NSPACE}(f)$, for some $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$, then $\mathcal{M}$ is also constructible in $\mathsf{NSPACE}(f)$.*

Proof Let $\mathcal{H}_L = (\Sigma, C_L, \omega_0 \#_1 \cdots \#_n \omega_n, \mathcal{R}_L)$ and $\mathcal{H}_M = (\Sigma, C_M, \nu_0 \#_1 \cdots \#_n \nu_n, \mathcal{R}_M)$ be $(\#_1, \ldots, \#_n)$-separated EDT0L systems accepting L and M, respectively, that are both constructible in $\mathsf{NSPACE}(f)$. We will assume without loss of generality that $C_L \backslash \Sigma$ and $C_M \backslash \Sigma$ are disjoint. Let $C = C_L \cup C_M \cup \{\perp_0, \ldots, \perp_n\}$, where each $\perp_i$ is a symbol not already used. For each $\phi \in \mathcal{R}_L$, define $\bar{\phi} \in \mathrm{End}(C^*)$ by

$$c\bar{\phi} = \begin{cases} c\phi & c \in C_L \\ c & c \notin C_L. \end{cases}$$

Define $\bar{\phi}$ for each $\phi \in \mathcal{R}_M$ analogously. Let $\mathcal{R} = \{\bar{\phi} \mid \phi \in \mathcal{R}_L \cup \mathcal{R}_M\}$, and note that $\mathcal{R}$ is rational. Define $\psi_L, \psi_M \in \mathrm{End}(C^*)$ by

$$\perp_i \psi_L = \omega_i, \quad \perp_i \psi_M = \nu_i,$$

for all i. We can conclude that $L \cup M$ is accepted by $(\Sigma,\ C,\ \bot_0\ \#_1 \cdots \#_n\ \bot_n$ $,\ \{\psi_L,\ \psi_M\}\mathcal{R})$.

Since $\mathcal{R}_L$ and $\mathcal{R}_M$ can be constructed in $\mathsf{NSPACE}(f)$, it follows that $\mathcal{R}_L \cup R_M$, and hence $\mathcal{R}$ can also be constructed in $\mathsf{NSPACE}(f)$. The same follows for $C = C_L \cup C_M$, and we can conclude that the EDT0L system is constructible in $\mathsf{NSPACE}(f)$. $\square$

Before we can start the proofs of Lemma 5.3.5 and Proposition 5.3.7, we need the concept of a derivation within an EDT0L system.

Definition 5.3.4. Let $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$ be an EDT0L system accepting a language L. Let $B \subseteq \text{End}(C^*)$ be an alphabet of $\mathcal{R}$. A *derivation* of a word $u \in L$ is a finite sequence $(\omega = \nu_0,\ \ldots,\ \nu_n = u)$ of words in C^*, such that there is a finite sequence $(\phi_1,\ \ldots,\ \phi_n)$ of elements of B, with $\phi_1 \cdots \phi_n \in \mathcal{R}$, and $\nu_i = \omega\phi_1 \cdots \phi_i$. We say the *length* of the derivation is $n + 1$ (the length of the sequence).

Lemma 5.3.5. *Let L be a language accepted by an EDT0L system $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$, such that every word in L contains precisely one occurrence of the letter $\# \in \Sigma$. Then*

1. *There is a $(\#)$-separated EDT0L system $\mathcal{M}$ that accepts L;*
2. *If $\mathcal{H}$ is $(\$_1,\ \ldots,\ \$_n)$-separated, for some $\$_1,\ \ldots,\ \$_n \in \Sigma$, then so is $\mathcal{M}$;*
3. *If $\mathcal{H}$ is constructible in $\mathsf{NSPACE}(f)$ for some $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$, then so is $\mathcal{M}$.*

The proof of Lemma 5.3.5 is an induction on the length of derivations of words over C that contain the symbol $\#$. Let $B \subseteq \text{End}(C^*)$ an alphabet of $\mathcal{R}$. We replace symbols $c \in C$ that are mapped to $\#$ by endomorphisms in B, and move the $\#$ left within the derivation until $\#$ is in the start word. Our strategy is to split $\mathcal{H}$ into finitely many EDT0L systems, whose languages union to make L. Each of these languages is constructed to have the property that there is a unique element $c \in C$ such that $c\phi = \sigma\#\tau$, for some $\sigma,\ \tau \in C^*$ and ϕ in the rational control.

Proof Step 1: Preprocessing.

Let $B \subseteq \text{End}(C^*)$ be an alphabet of $\mathcal{R}$. We will first show that we can assume that all elements of Σ are fixed by elements of B. Let Δ be the set of all letters in Σ

not fixed by elements of B. Make a copy c_a of each $a \in \Delta$, and add each of these copies to C. We will initially assume these copies are fixed by elements of B. Define $\theta \in \text{End}(C^*)$ by $a\theta = c_a$ for all $a \in \Delta$. Replace each occurrence of each $a \in \Delta$ within the start word with c_a. Replace each $\phi \in B$ by $\phi\theta$. Finally, let $\psi \in \text{End}(C^*)$ be defined by $c_a\psi = a$ for all $a \in \Delta$, and redefine the rational control to be $\mathcal{R}\psi$. Now all letters in Σ are fixed by elements of B. Note that this preserves the fact that $\mathcal{H}$ is $(\$_1, \ \ldots, \ \$_n)$-separated.

We now add a new symbol F to C, which all elements of B (and therefore $\mathcal{R}$) will fix. Initially, F will be unused; $c\phi \neq F$ for all $c \in C$, however, we will later modify $\mathcal{H}$, or other EDT0L systems obtained from $\mathcal{H}$ to use F. This letter will be used as a 'fail symbol'. That is, if $\phi \in \text{End}(C^*)$ is such that $\omega\phi$ contains F, for some $\omega \in C^*$, then for all $\psi \in \text{End}(C^*)$ such that $\phi\psi \in \mathcal{R}$, we will have that $\omega\phi\psi$ contains F, and so $\omega\phi\psi \notin \Sigma^*$, and will therefore not be accepted by $\mathcal{H}$.

Step 2: Splitting $\mathcal{H}$ into finitely many EDT0L systems.
Suppose $\phi \in B$ is such that there exists $c \in C$ and $\theta, \ \psi \in \text{End}(C^*)$, satisfying $\theta\phi\psi \in \mathcal{R}$, $c\phi = \sigma\#\tau$ for some $\sigma, \ \tau \in C^*$, and $c\phi\psi \in \Sigma^*$. Let X be the set of all such $\phi \in B$, and for each $\phi \in X$, let D_ϕ be the set of letters in $C\backslash\{\#\}$ which ϕ maps to $\#$. For each $\phi \in X$, we will define a new EDT0L system $\mathcal{H}_\phi$ as follows. For each $\phi \in X$, define $\bar{\phi} \in \text{End}(C^*)$ by

$$d\bar{\phi} = \begin{cases} F & d \in D_\phi \\ d\phi & d \notin D_\phi. \end{cases}$$

Let $\mathcal{H}_\phi$ be the EDT0L system obtained from $\mathcal{H}$ by replacing each $\psi \in X\backslash\{\phi\}$ with $\bar{\psi}$. Note that each system $\mathcal{H}_\phi$ is $(\$_1, \ \ldots, \ \$_n)$-separated.

By construction, we have that for all $\phi \in X$, $L(\mathcal{H}_\phi) \subseteq L$. Let $w \in L$. Since endomorphisms in B never map $\#$ to ε (as Σ is fixed pointwise by every endomorphism in the rational control), it follows that w can only be derived using a derivation involving one $\phi \in X$ mapping a letter in $C\backslash\{\#\}$ to $\#$ (as w contains precisely one occurrence of $\#$). Note that the ϕ may occur again within the derivation, but only once will it map a letter (other than #) to #. Thus $w \in \mathcal{H}_\phi$, and we have shown

$$L = \bigcup_{\phi \in X} L(\mathcal{H}_\phi).$$

Step 3: Induction.

Let $\mathcal{A}$ be the collection of the EDT0L systems $\mathcal{H}_\phi$. For each $\mathcal{G} \in \mathcal{A}$, let $C_\mathcal{G}$ be the extended alphabet, $B_\mathcal{G}$ be a the alphabet of $\mathcal{R}_\mathcal{G}$ of $\mathcal{G}$, and let $n_\mathcal{G}$ be the minimal length of a derivation of a word over the extended alphabet containing # in $\mathcal{G}$. Let $n = \max_{\mathcal{G} \in \mathcal{A}} n_\mathcal{G}$.

Note that if we redefine $\mathcal{A}$, we will assume n has been updated accordingly. We will proceed by induction on n. If $n = 0$, then # appears in the start words of every $\mathcal{G} \in \mathcal{A}$, and so there is nothing to prove.

Let $k > 0$, and inductively assume that the result holds whenever $n < k$. Suppose $n = k$. Let $\mathcal{G} \in \mathcal{A}$ be such that $n_\mathcal{G} = k$. Let ϕ be the unique element of $B_\mathcal{G}$, that $\mathcal{G} = \mathcal{H}_\phi$. Let Ψ be the set of all $\psi \in B$, such that $d\psi$ contains a letter in D_ϕ, for some $d \in C$.

We will redefine the rational control $\mathcal{R}_\mathcal{G}$ of $\mathcal{H}_\mathcal{G}$ as follows. Firstly, enlarge $B_\mathcal{G}$ by adding $\theta_{\psi,\phi}$ for each $\psi \in \Psi$, where $\theta_{\psi,\phi} = \psi\phi$. Let $\mathcal{R}_0$ be the set obtained from $\mathcal{R}_\mathcal{G}$ by replacing each occurrence of $\psi\phi$ with $\theta_{\psi,\phi}$, for all $\psi \in \Psi$ and $\phi \in \Phi$. Note that this corresponds to finitely many preimages of free monoid endomorphisms, and so the set $\mathcal{R}_0$ is indeed rational. We now redefine $\mathcal{R}_\mathcal{G}$ to be $\mathcal{R}_\mathcal{G} \cup \mathcal{R}_0$.

Note one $d \in C$ that was mapped by some $\psi \in \Psi$ to $\sigma c \tau$, for some $\sigma,\ \tau \in C^*$ and $c \in D_\phi$, will now mapped by some $\theta_{\psi,\phi}$ directly to $\rho \# \mu$, for some $\rho,\ \mu \in C^*$.

We have now reduced $n_\mathcal{G}$ by 1, however we have potentially broken the hypothesis that there is a unique endomorphism in $B_\mathcal{G}$ that maps some $c \in C_\mathcal{G}$ to $\sigma \# \tau$ for any $\sigma,\ \tau \in C^*$. We can apply the same process we used to construct the EDT0L systems $\mathcal{H}_\phi$ from $\mathcal{H}$ to $\mathcal{G}$, to create a number of new EDT0L systems, the union of whose languages will be $L(\mathcal{G})$, but such that for each of these EDT0L systems $\mathcal{G}'$, we have that $n_{G'} = k - 1$. We can replace $\mathcal{G}$ in $\mathcal{A}$ with all of these new EDT0L systems.

Applying this method to all $\mathcal{G} \in \mathcal{A}$ with $n_{\mathcal{G}} = k$, will cause n to equal $k-1$, and so the result follows by induction.

Step 4: Space complexity.
First note that in the initial system, the length of the shortest derivation of a word involving # will be at most $|C|$. As splitting $\mathcal{H}$ into the finitely many $\mathcal{H}_\phi$ does not affect this number, we have that $n \leq |C|$ before any iterations of the induction hypothesis are applied.

It therefore remains to consider the space complexity of each iteration, along with the space complexity of splitting into finitely many systems $\mathcal{H}_\phi$.

Starting from $\mathcal{H}$ (or any $\mathcal{G} \in \mathcal{A}$), constructing the EDT0L systems $\mathcal{H}_\phi$ can be done as follows. We first compute the set X, which can be done by looking at each endomorphism in B. Following this, we choose a $\phi \in X$, and construct $\mathcal{H}_\phi$ the same way we can construct $\mathcal{H}$, except using $\bar{\phi}$ instead of ϕ for all $\phi \in X \backslash \{\phi\}$. This can be done in $\mathsf{NSPACE}(f)$, as we only need to remember X, and $|X| \leq |B|$. We then remove ϕ from X, and continue for each remaining $\phi \in X$. This can all therefore be done in $\mathsf{NSPACE}(f)$.

We now consider the complexity of the construction in the induction step: replacing $\mathcal{R}_{\mathcal{G}}$ with $\mathcal{R}_G \cup \mathcal{R}_0$. To do this, we simply need that show that regular languages that are unions of regular languages constructible in $\mathsf{NSPACE}(f)$, or preimages of regular languages under free monoid homomorphisms that constructible in $\mathsf{NSPACE}(f)$, are also definable in $\mathsf{NSPACE}(f)$, which follows by Lemma 2.4.3. □

Using Lemma 5.3.1, followed by Lemma 5.3.5 n times (once for each $\#_i$), we can prove the following.

Lemma 5.3.6. *Let L be an EDT0L language, such that every word in L contains precisely n occurrences of the letter #, where $n \in \mathbb{Z}_{\geq 0}$. Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then*

1. *There is a $(\#, \ldots, \#)$-separated EDT0L system $\mathcal{H}$ that accepts L.*
2. *If an EDT0L system for L is constructible in $\mathsf{NSPACE}(f)$, then $\mathcal{H}$ is constructible in $\mathsf{NSPACE}(f)$.*

Proof By Lemma 5.3.1, it suffices to show that

$$N = \{u_0\#_1 \cdots \#_n u_n \mid u_0\# \cdots \#u_n \in L\}$$

is EDT0L, and the system is constructible in $\mathsf{NSPACE}(f)$. The result now follows by Lemma 5.3.5 used n times (once for each $\#_i$). □

We are now able to prove the main result of this section.

Proposition 5.3.7. *Let L and M be EDT0L languages, such that every word in $L \cup M$ contains precisely n occurrences of the letter $\#$. Let $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then*

1. *The language*

$$N = \{u_0v_0\# \cdots \#u_nv_n \mid u_0\# \cdots \#u_n \in L,\ v_0\# \cdots \#v_n \in M\},$$

   is EDT0L;
2. *If EDT0L systems for L and M are constructible in $\mathsf{NSPACE}(f)$, then an EDT0L system for N is constructible in $\mathsf{NSPACE}(f)$.*

Proof By Lemma 5.3.6, we have that L and M are accepted by EDT0L systems $\mathcal{H}_L$ and $\mathcal{H}_M$, with start words $\omega_0\#_1 \cdots \#_n\omega_n$ and $\nu_0\#_1 \cdots \#_n\nu_n$, respectively, such that nothing other than $\#_i$ is mapped to $\#_i$ within both $\mathcal{H}_L$ and $\mathcal{H}_M$. Suppose also that these systems are constructible in $\mathsf{NSPACE}(f)$. Let C_L and C_M be the extended alphabets of $\mathcal{H}_L$ and $\mathcal{H}_M$, and let Σ_L and Σ_M be the terminal alphabets. Without loss of generality assume $C_L \backslash \Sigma_L$ and $C_M \backslash \Sigma_M$ are disjoint. Let $\mathcal{R}_L$ and $\mathcal{R}_M$ be the rational controls, and let B_L and B_M be alphabets of $\mathcal{R}_L$ and $\mathcal{R}_M$, respectively.

Let $\Sigma = \Sigma_L \cup \Sigma_M$, and let $C = C_L \cup C_M$. For each $\phi \in B_L$, define $\bar{\phi} \in \mathrm{End}(C^*)$ by

$$c\bar{\phi} = \begin{cases} c\phi & c \in C_L \\ c & c \notin C_L. \end{cases}$$

Define $\bar{\phi}$ for each ϕ in B_M analogously, and extend the bar notation to composition of functions, that is, $\overline{\phi\psi} = \bar{\phi}\bar{\psi}$. Let $\mathcal{R} = \{\bar{\phi} \mid \phi \in \mathcal{R}_L \cup \mathcal{R}_M\}$, and note that $\mathcal{R}$ is a rational set. Thus, N is accepted by the EDT0L system $(\Sigma,\ C,\ \omega_0\nu_0\#_1\cdots\#_n\omega_n\nu_n,\ \mathcal{R})$, as required.

Suppose there exist EDT0L systems for L and M, which are constructible in $\mathsf{NSPACE}(f)$. By Lemma 5.3.6, $\mathcal{H}_L$ and $\mathcal{H}_M$ are also constructible in $\mathsf{NSPACE}(f)$. We can construct C with the memory required to construct C_M and C_L. The set $\{\bar{\phi} \mid \phi \in \mathcal{B}_L\}$ is constructible in $\mathsf{NSPACE}(f)$, by following the construction of $\mathcal{R}_L$, but writing a $\bar{\phi}$ instead of a ϕ, for each occurrence of $\phi \in B_L$. By symmetry, $\{\bar{\phi} \mid \phi \in \mathcal{B}_M\}$ is constructible in $\mathsf{NSPACE}(f)$. Since $\mathcal{R}$ is the union of these sets, we can construct $\mathcal{R}$ in $\mathsf{NSPACE}(f)$ by Lemma 2.4.3. □

5.4 Equations in extensions

This section shows that the class of groups where systems of equations have EDT0L solution languages is closed under various extensions, including wreath products with finite groups and direct products. These facts are used in the proof of Theorem 5.6.8 on groups that are virtually a direct product of hyperbolic groups.

We can use Lemma 4.3.11 to show that passing to the Schreier normal form also preserves the property of being quasi-geodesic.

Lemma 5.4.1. *Let G be a group, generated by a finite set Σ, H be a finite index subgroup of G, and T be a right transversal of H in G, containing 1. Let Z be the Schreier generating set for H. Fix a normal form η for $(G,\ \Sigma)$. If η is quasi-geodesic with respect to Σ, then the Schreier normal form with respect to η is quasi-geodesic with respect to the Schreier generators.*

Proof Let ζ be the Schreier normal form for H, with respect to η. We will show that the normal form from Remark 4.3.8, inherited from ζ, is quasi-geodesic. The result will then follow by the backward direction of Lemma 4.3.11. Since η is quasi-geodesic, there exists $\lambda > 0$, such that $|g\eta| \leq \lambda |g|_{(G,\Sigma)} + \lambda$ for all $g \in G$.

Let ξ denote the normal form from Remark 4.3.8, inherited from ζ, with respect to the transversal T. Let $w \in (\Sigma^{\pm})^*$ be geodesic. We have that there exists $v \in \operatorname{im}\eta$, such that $v =_G w$, and $|v| \leq \lambda|w| + \lambda$. We also have that there exists $t_0 \in T$ such that vt_0 represents an element of H. We can then convert this into Schreier normal form to give a word u. Note that $|u| \leq |vt_0|$.

We also have that there exists $t_1 \in T$, such that $ut_1 =_G w$. Note that $ut_1 \in \operatorname{im} \xi$. Combining our inequalities that relate u, v and w, gives:

$$|ut_1| \leq |vt_0t_1| = |v| + 2 \leq \lambda|w| + 2\lambda.$$

So ξ is quasi-geodesic, with respect to a constant 2λ. The result now follows by Lemma 4.3.11. □

In order to prove our results about wreath products and direct products, we need some normal forms on groups made using these constructions.

Remark 5.4.2. Let $H_1, \ \ldots, \ H_k$ be groups, with finite generating sets $\Sigma_{H_1}, \ \ldots, \ \Sigma_{H_k}$, and normal forms $\eta_{H_1}, \ \ldots, \ \eta_{H_k}$, respectively. Let $G = \prod_{i=1}^{k} H_i$. We will use $\Sigma = \Sigma_{H_1} \sqcup \cdots \sqcup \Sigma_{H_k}$ as a generating set for G. Define the $\eta \colon G \to (\Sigma^{\pm})^*$ by

$$(h_1, \ \ldots, \ h_k)\eta = (h_1\eta_{H_1}) \cdots (h_k\eta_{H_k}).$$

Since concatenations of regular languages are regular, if every η_{H_i} is regular, then η is a regular normal form.

In addition, the length of any element $g \in G$ with respect to Σ is just the sum of the lengths of the projection of g to each H_i, and from this it follows that if every η_{H_i} is (quasi)geodesic, then so is η.

Remark 5.4.3. Let H be a group, and K be a finite group. Let Σ_H be a generating set for H, and η_H be a normal form with respect to Σ_H. We define a generating set and normal form for $H \wr K$, using Σ_H and η_H. Note that $H \wr K$ contains $\prod_{i=1}^{n} H_i$ as a finite index subgroup, where $n \in \mathbb{Z}_{>0}$, and $H_i \cong H$ for all i. We endow $\prod_{i=1}^{n} H_i$ with a generating set and normal form using Remark 5.4.2. After this, we can use the generating set and normal form from Remark 4.3.8 for $H \wr K$, with respect the generating set and normal form of $\prod_{i=1}^{n} H_i$.

Since the two constructions we have used to produce a normal form for $H \wr K$ preserve the properties of regular and quasi-geodesic, if η_H is regular or quasi-geodesic, then so is the normal form on $H \wr K$.

We show that the class of groups with EDT0L solutions to systems of equations is closed under direct products, and wreath products with finite groups. We start with the latter. We refer the reader to [56] for the definition of a wreath product.

We first consider the properties of the normal forms we will be using.

Lemma 5.4.4. *Let H and η_H be as in Remark 5.4.3. If η_H is regular or quasi-geodesic, then the normal form on $H \wr K$ from Remark 5.4.3 will be regular or quasi-geodesic, respectively.*

Proof Recall that the normal form in Remark 5.4.3 is created by using the normal form for direct products (Remark 5.4.2), followed by the normal form for finite extensions 4.3.8. Since both of these constructions preserve the properties regular and quasi-geodesic, the result follows. $\square$

We can now show that equations in wreath products have the desired properties.

Proposition 5.4.5. *Let H be a group such that solutions to systems of equations with respect to a normal form η_H are EDT0L in* $\mathsf{NSPACE}(f)$*, where $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Let K be a finite group. Then*

1. *Solutions to systems of equations in $H \wr K$ are EDT0L in* $\mathsf{NSPACE}(f)$*, with respect to the normal form from Remark 5.4.3;*
2. *If η_H is regular or quasi-geodesic, then the normal form on $H \wr K$ will be regular or quasi-geodesic, respectively.*

Proof First note that (2) follows from Lemma 5.4.4.

Let A be the finite set that K acts on, and define $H \wr K$ with respect to this action. Let $H_1, \ldots, H_{|A|}$ be the isomorphic copies of H. Using Proposition 4.3.9, it suffices to show that that solutions to systems of Ω-twisted equations in $G := \prod_{i=1}^{|A|} H_i$ are

EDT0L in $\mathsf{NSPACE}(f)$, with respect to the normal form from Remark 5.4.2, where Ω is the set of automorphisms defined by permuting the H_is.

Consider a system $\mathcal{E}$ of Ω-twisted equations in G in n variables. As every element of G can be written in the form $h_1 \cdots h_{|A|}$, where $h_i \in H_i$ for all i, for each variable X in $\mathcal{E}$, we can define new variables X_i over H_i for each i, by $X = X_1 \cdots X_{|A|}$. As the elements of H_i commute with the elements of H_j for each $i \neq j$, we can view any (untwisted) equation in G as a system of $|A|$ equations in H, each with disjoint set of variables, by projecting the original equation onto H_i. The fact that these sets are disjoint follows from the fact that the ith equation in the system will be the projection to H_i, whose variables will be of the form X_i, for some original variable X.

Let $\Phi \in \Omega$, and let $\sigma \in S_n$ be the permutation induced by the action of Φ. Then $X\Phi = (X_1 \cdots X_{|A|})\Phi = X_{1\sigma} \cdots X_{(|A|)\sigma}$. It follows that any twisted equation in G can be viewed as a system of $|A|$ equations in H, again using projections to each H_i. The variables of each of the equations will no longer be disjoint, however. It follows that a system of twisted equations in G projects to a system of equations in H. Thus, there exists a system $\mathcal{F}$ of equations in H with solution set $S_{\mathcal{F}}$, such that $\mathcal{F}$ has $|A|n$ variables, and each variable is assigned an index in $\{1, \ldots, |A|\}$, such that precisely n variables have each index, and such that the solution language of $\mathcal{E}$ is equal to

$$\{x_{11} \cdots x_{1|A|} \# \cdots \# x_{n1} \cdots x_{n|A|} \mid (x_{1i}, \ldots, x_{ni}) \in S_{\mathcal{F}} \text{ with each variable indexed by } i \text{ for all } i\}.$$

From our assumptions, we have that the solution language to $\mathcal{F}$ is EDT0L, and can be constructed in $\mathsf{NSPACE}(f)$. It follows that the language

$$L_i = \{x_{1i} \# \cdots \# x_{n_i} \mid (x_{1i}, \ldots, x_{ni}) \in S_{\mathcal{F}} \text{ with each variable indexed by } i\}$$

is EDT0L for each choice of i, and constructible in $\mathsf{NSPACE}(f)$, using Lemma 5.3.1, and then taking the image under an appropriate free monoid endomorphism with Theorem 3.3.2. Proposition 5.3.7 then shows that the solution language to $\mathcal{E}$ is EDT0L in $\mathsf{NSPACE}(f)$. $\square$

We conclude this section with the proof that direct products also preserve the property of having EDT0L solution languages.

Proposition 5.4.6. *Let* $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. *Let* G *and* H *be finitely generated groups where solutions to systems of equations are EDT0L in* $\mathsf{NSPACE}(f)$. *Then*

1. *The same holds in* $G \times H$, *with respect to the normal form from Remark 5.4.2;*
2. *If the normal forms on* G *and* H *are regular or quasi-geodesic, then the normal form on* $G \times H$ *will be regular or quasi-geodesic, respectively, with respect to the union of the generating sets for* G *and* H.

Proof Part (2) follows from Remark 5.4.2.

Let Σ_G be a finite generating set for G, and Σ_H be a finite generating set for H. We will use $\Sigma = \Sigma_G \sqcup \Sigma_H$ as our generating set for $G \times H$. Consider an equation $\omega = 1$ in $G \times H$. Let $\mathcal{X}$ be the set of variables in ω. We have that every element of $G \times H$ can be expressed in the form gh for some $g \in G$ and $h \in H$. We can reflect this in the variables as well, by defining new variables X_G over G and X_H over H, for each $X \in \mathcal{X}$, such that $X = X_G X_H$. Let $\mathcal{X}_G = \{X_G \mid X \in \mathcal{X}\}$, and $\mathcal{X}_H = \{X_H \mid X \in \mathcal{X}\}$.

As elements of G commute with elements of H, we can rearrange $\omega = 1$ into the form $\nu\zeta = 1$, where $\nu \in (\Sigma_G^{\pm} \cup X_G^{\pm})^*$ and $\zeta \in (\Sigma_H^{\pm} \cup X_H^{\pm})^*$. Consider a potential solution $(g_1 h_1, \ldots, g_n h_n)$ to $\omega = 1$, where each $g_i \in G$ and each $h_i \in H$. We have that this is a solution if and only if $(g_1, \ldots, g_n)$ is a solution to the equation $\nu = 1$, and $(h_1, \ldots, h_n)$ is a solution to the equation $\zeta = 1$. Note that these are equations in G and H, respectively.

Let $\mathcal{E}$ be a system of equations in $G \times H$. It follows that there exist systems of equations in G and H with solution sets $\mathcal{S}_G$ and $\mathcal{S}_H$, such that the solution set to $\mathcal{E}$ equals

$$\{(g_1 h_1, \ldots, g_n h_n) \mid (g_1, \ldots, g_n) \in \mathcal{S}_G, (h_1, \ldots, h_n) \in \mathcal{S}_H\}.$$

If $\mathcal{L}_G$ and $\mathcal{L}_H$ are EDT0L solution languages corresponding to these systems in G

and H, respectively, it follows that the solution language to $\mathcal{E}$ equals

$$\{\omega_0\nu_0\#\cdots\#\omega_n\nu_n \mid \omega_0\#\cdots\#\omega_n \in \mathcal{L}_G,\ \nu_0\#\cdots\#\nu_n \in \mathcal{L}_H\}.$$

The result now follows by Proposition 5.3.7. □

5.5 Recognisable constraints and finite index subgroups

This section is used to show Proposition 5.5.3, that is, that the class of groups where systems of equations have EDT0L solutions is closed under passing to finite index subgroups. We use recognisable constraints to show this fact, by first proving that the addition of recognisable constraints to a system of equations with an EDT0L solution set does not change the fact that the solution set is EDT0L with respect to the ambient normal form of the group. We can then use the fact that finite index subgroups are recognisable, however the resulting language will be expressed as words over the generators for the ambient group. Expressing solutions to the finite index subgroup as words over one of its own generating sets, such as the Schreier generators, requires additional arguments.

We start by showing that the addition of recognisable constraints to systems of equations in a group preserves the property that all such systems have EDT0L solution languages.

Proposition 5.5.1. *Let G be a finitely generated group such that solutions to systems of equations are EDT0L in* $\mathsf{NSPACE}(f)$ *with respect to some normal form η, where $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Then solutions to systems of equations in G with recognisable constraints are EDT0L in* $\mathsf{NSPACE}(f)$*, with respect to η.*

Proof Let Σ be a finite generating set for G, and fix a normal form η for $(G,\ \Sigma)$ such that solution languages to systems of equations are EDT0L. Consider a system of equations $\mathcal{E}$ with recognisable constraints in G with n variables. Let $R_1,\ \ldots,\ R_n$

denote the constraints. Let L be the solution language to $\mathcal{E}$ with the constraints removed. Let $\pi \colon \Sigma^* \to G$ be the natural homomorphism. Note that

$$S = (R_1\pi^{-1})\#(R_2\pi^{-1})\# \cdots \#(R_n\pi^{-1})$$

is a regular language. By Theorem 3.3.2, $L \cap S$ is EDT0L, and if an EDT0L system for L is constructible in $\mathsf{NSPACE}(f)$, then one for $L \cap S$ is also constructible in $\mathsf{NSPACE}(f)$. As $L \cap S$ is the solution language to $\mathcal{E}$, the results follow. □

Since finite index subgroups are examples of recognisable sets, we can show the following.

Lemma 5.5.2. *Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Let G be a finitely generated group where solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(f)$, with respect to some normal form η, and let H be a finite index subgroup of G. Let $\mathcal{E}$ be a system of equations in G. Then*

1. *the language of all solutions to $\mathcal{E}$ that lie in H forms an EDT0L language, with respect to the normal form η restricted to H;*
2. *The EDT0L system for this language is constructible in $\mathsf{NSPACE}(f)$.*

Proof In order to restrict our solutions to H, we add the constraint that every variable lies in H, which is a recognisable subset of G. The results now follow from Proposition 5.5.1. □

Proposition 5.5.3. *Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Let G be a group where solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(f)$, with respect to a normal form η. Then the same holds in any finite index subgroup of G with respect to the Schreier normal form, inherited from η.*

Proof Let X be a finite generating set for G, T be a right transversal for H, and Z be the Schreier generating set for H. Let ζ be the Schreier normal form for H.

Fix a system $\mathcal{E}$ of equations in H. This can be considered as a system of equations in G, with the restriction that the solutions must lie in H. Let L be the solution language to $\mathcal{E}$ when expressed as words over G using the normal form η; that is

$$L = \{(g_1\eta)\#\cdots\#(g_n\eta) \mid (g_1, \ \ldots, \ g_n) \text{ is a solution to } \mathcal{E}\}.$$

Note that we require that solutions lie in H, as $\mathcal{E}$ is a system over H. By Lemma 5.5.2, L is an EDT0L language over an alphabet Σ. Let $\mathcal{H} = (\Sigma, \ C, \ \omega, \ \mathcal{R})$ be an EDT0L system for L that is constructible in $\mathsf{NSPACE}(f)$. Let $B \subseteq \mathrm{End}(C^*)$ be the alphabet of $\mathcal{R}$.

By Lemma 5.3.6, we can assume our start word is of the form $\omega_1\#\cdots\#\omega_n$. By adding new letters $\bot_1 \cdots \bot_n$ to C, and preconcatenating the rational control by the endomorphism defined by $\bot_i \mapsto \omega_i$ for all i, we can assume our start word is of the form $\bot_1 \# \cdots \# \bot_n$. Note that as we can easily construct our new start word from our existing one, this will not affect space complexity.

We will construct a new EDT0L system from $\mathcal{H}$. Our extended alphabet will be letters in C transversal element $t \in T$. Define

$$C_{\text{ind}} = \{c^{t,a} \mid c \in C, \ t \in T, \ a \in X^{\pm}\} \cup \{\bot_1, \ \ldots, \ \bot_n\}.$$

Our alphabet will be $\Sigma_{\text{ind}} \cup \{\#\} = \{a^{t,a} \mid a \in X^{\pm}, \ t \in T\} \cup \{\#\}$. Our start word will be $\bot_1 \# \cdots \# \bot_n$. We define our rational control as follows. For each $\phi \in B$, define Φ_ϕ to be the set of all $\psi \in \mathrm{End}(C^*_{\text{ind}})$ such that

$$c^{t,a}\psi = x_1^{t_1,b_1} \cdots x_k^{t_k,b_k},$$

where $c\psi = x_1 \cdots x_k$, with every $x_i \in C$, each $t_i \in T$, $t_1 = t$, and $\overline{t_i b_i} = t_{i+1}$. Let $\mathcal{R}_1$ be the rational set obtained by replacing each occurrence of $\phi \in B$ with the finite set Φ_ϕ. Let t_0 be the unique element in $T \cap H$. Let $\Psi \subseteq \mathrm{End}(C^*_{\text{ind}})$ be the set of all ψ defined by

$$\bot_i \ \psi = \bot_i^{t_0, \ a_i},$$

for some $a_1, \ \ldots, \ a_n \in X^{\pm}$. Define $\mathcal{G} = (\Sigma_{\text{ind}}, \ C_{\text{ind}}, \ \bot_1 \# \cdots \# \bot_n, \ \Psi\mathcal{R}_1)$. By construction, $\mathcal{G}$ accepts words in L, where each letter, excluding $\#$, has an index $(t, a) \in T \times X^{\pm}$, and such that for each indexed word $w = a_1^{t_1,a_1} \cdots a_k^{t_k,a_k}$, the following hold:

1. $t_1 = t_0$;
2. $t_i a_i = t_{i+1}$ for all i.

To show that the solution language to $\mathcal{E}$ is EDT0L with respect to ζ, it remains to apply the free monoid homomorphism $\theta\colon \Sigma_{\text{ind}}^* \to (Z^{\pm} \cup \{\#\})^*$ to $L(\mathcal{G})$, defined by

$$a^{t,a} \mapsto at\overline{at}^{-1}.$$

It now remains to show that this EDT0L system can be constructed in $\mathsf{NSPACE}(f)$. By Theorem 3.3.2, applying the homomorphism θ does not affect the space complexity, so it is sufficient to show that $\mathcal{G}$ is constructible in $\mathsf{NSPACE}(f)$. The number of indices we use is $2|X||T|$, which is constant, as it is based only on the group H. It follows that we can write down C_{ind} and Σ_{ind} in $\mathsf{NSPACE}(f)$. The set Ψ is again only based on $|X|$, and so to show our rational control is constructible in $\mathsf{NSPACE}(f)$, it suffices to prove that $\mathcal{R}_1$ is.

Note that $|\Phi_\phi|$ is again only based on $|X||T|$, and so is constant. We construct $\mathcal{R}_1$ by proceeding with the procedure we used to construct $\mathcal{R}$, except whenever we would add an edge labelled $\phi \in B$ between two states, we add edges labelled with all of Φ_ϕ between the same states. We can compute Φ_ϕ each time we need it, so we need only record the information we used to construct $\mathcal{R}$. We can conclude that $\mathcal{G}$ is constructible in $\mathsf{NSPACE}(f)$, and so the language of solutions to $\mathcal{E}$ is EDT0L in $\mathsf{NSPACE}(f)$. $\square$

5.6 Virtually direct products of hyperbolic groups

In this section, we show that solution languages to systems of equations in groups that are virtually direct products of hyperbolic groups are EDT0L. We adapt the method that Ciobanu, Holt and Rees use to show that the satisfiability of systems of equations in these groups is decidable [22]. For an introduction to hyperbolic groups, we refer the reader to [56], Chapter 6.

We start with some lemmas needed to prove this result. The following lemma gives an embedding as a finite index subgroup of a group that is virtually a direct product of hyperbolic groups, into a direct product of groups where equations are better understood.

Lemma 5.6.1 ([22], Lemma 3.5)**.** *Let G be a group that contains a group of the form $K_1 \times \cdots \times K_n$ as a finite index normal subgroup, such that every conjugate of each of the subgroups K_i lies in the set $\{K_1, \ldots, K_n\}$. Then*

1. *If the groups K_i are all conjugate to each other, then G is isomorphic to a finite index subgroup of $J \wr P$, where $J \cong N_G(K_1)/(K_2 \times \cdots \times K_n)$ contains a finite index subgroup isomorphic to K_1, and P is finite;*
2. *Suppose $K_1, \ldots, K_k$ are representatives of the conjugacy classes of $K_1, \ldots, K_n$ within G. Then G is isomorphic to a finite index subgroup of a direct product $W_1 \times \cdots \times W_k$, where $W_i = J_i \wr P_i$, J_i contains K_i as a finite index subgroup, and P_i is finite, for all i.*

We define a normal form for groups that are virtually direct products.

Remark 5.6.2. Let G be a group that has a finite index subgroup of the form $K_1 \times \cdots \times K_n$. Fix a finite generating set Σ_{K_i}, and normal form η_{K_i} for each K_i. Using Lemma 5.6.1, G embeds as a finite index subgroup of $W_1 \times \cdots \times W_k$, where $W_i = J_i \wr P_i$, K_i embeds as a finite index subgroup of J_i, and P_i is finite.

- We start by defining a generating set and normal form for each J_i. Since J_i contains K_i as a finite index subgroup, we can use the generating set and normal form from Remark 4.3.8, induced by Σ_{K_i} and η_{K_i}. We will denote this generating set and normal form using Σ_{J_i} and η_{J_i}, respectively;
- Using Σ_{J_i} and η_{J_i}, we can use the generating set and normal form defined in Remark 5.4.3 to define a normal form for each $W_i = J_i \wr P_i$. Using these generating sets and normal forms, Remark 5.4.2 gives us a generating set Δ and a normal form μ for $W_1 \times \cdots \times W_k$;
- As G embeds as a finite index subgroup of $W_1 \times \cdots \times W_k$, we can use the Schreier generating set Z and normal form ζ on G, induced by Δ and μ.

Lemma 5.6.3. *Let G be a group that has a finite index subgroup of the form $K_1 \times \cdots \times K_n$, and let η_{K_i} be defined as in Remark 5.6.2. Let ζ be the normal form on G from Remark 5.6.2. If each η_{K_i} is regular or quasi-geodesic, then ζ is regular or quasi-geodesic, respectively.*

Proof Since each of the constructions we have used to create ζ preserve the properties of being regular and quasi-geodesic (Lemma 5.2.6, Remark 5.4.2, Lemma 5.4.4, Lemma 4.3.11, Lemma 5.4.1), if every η_{K_i} is regular or every η_{K_i} is quasi-geodesic, then ζ will be regular or quasi-geodesic, respectively. □

We now use Lemma 5.6.1 to show that the group that is virtually a direct product has an EDT0L solution language, subject to conditions on the groups it is virtually a direct product of.

Proposition 5.6.4. *Let G be a group that contains a group of the form $K_1 \times \cdots \times K_n$ as a finite index normal subgroup, such that every conjugate of each of the subgroups K_i lies in the set $\{K_1, \ \ldots, \ K_n\}$. Let $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$.*

1. *If solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(f)$ in each group in $\mathrm{FIN}(K_i)$, with respect to a normal form η_{K_i}, then solutions systems of equations in G are EDT0L $\mathsf{NSPACE}(f)$, with respect to the normal form ζ from Remark 5.6.2;*
2. *If all the normal forms used in the groups in $\mathrm{FIN}(K_i)$ are regular or quasi-geodesic, then ζ will be regular or quasi-geodesic, respectively.*

Proof By Lemma 5.6.1, we have that G embeds as a finite index subgroup into $W_1 \times \cdots \times W_k$, where $W_i = J_i \wr P_i$ for finite index overgroups J_i of K_i, and finite groups P_i. By Lemma 5.5.3, it suffices to show that solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(f)$ in $W_1 \times \cdots \times W_k$. The fact that solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(f)$ in each of the groups W_i follows by our assumptions, together with Proposition 5.4.5. We can then use Proposition 5.4.6 to show that the same holds in $W_1 \times \cdots \times W_k$.

Part (2) follows from Lemma 5.6.3. □

We now apply Proposition 5.6.4 to the specific case when the groups in the direct product comprise one virtually abelian group, and other non-elementary hyperbolic groups.

Lemma 5.6.5 ([22], Proposition 4.4)**.** *Let A be a virtually abelian group, and let $H_1, \ldots, H_n$ be non-elementary hyperbolic groups. Let G be a group with a finite index subgroup H that is isomorphic to $A \times H_1 \times \cdots \times H_n$. Then G has a finite index normal subgroup isomorphic to $B \times K_1 \times \cdots \times K_n$, where B is a finite index subgroup of A, and each K_i is a finite index subgroup of H_i, such that every conjugate of each of the subgroups K_i lies in the set $\{K_1, \ldots, K_n\}$.*

We finally need the fact that languages of solutions to systems of equations in hyperbolic groups are EDT0L.

Lemma 5.6.6 ([19])**.** *Solutions to a system of equations in any hyperbolic group are EDT0L in $\mathsf{NSPACE}(n^4 \log n)$, with respect to any finite generating set, and any quasi-geodesic normal form. If the hyperbolic group is torsion-free, the solutions are EDT0L in $\mathsf{NSPACE}(n^2 \log n)$.*

We are now in a position to show that groups that are virtually direct products of hyperbolic groups have EDT0L languages of solutions. Since every hyperbolic group admits a regular geodesic normal form, if these normal forms are used to induce the normal forms in the hyperbolic groups, then the normal form in the virtually direct product will be quasi-geodesic and regular.

Remark 5.6.7. In the following theorem, our groups are constructed from virtually abelian groups and other groups. As such, we are measuring our input size using equation length, not virtually abelian equation length. However, we will continue to use space complexity results from Chapter 4 that use virtually abelian length. This is okay, since virtually abelian equation length is approximately the log of equation length, and so the actual space complexity will be at least as small. It is possible that the space complexity will be a smaller than stated, but it will still be polynomial.

Theorem 5.6.8. *Let G be a group that is virtually $A \times H_1 \times \cdots \times H_n$, where A is virtually abelian, and $H_1, \ldots, H_n$ are non-elementary hyperbolic. Then*

1. *Solutions to systems of equations in G are EDT0L in $\mathsf{NSPACE}(n^4 \log n)$, with respect to the normal form ζ from Remark 5.6.2;*
2. *If, in addition, all of the groups H_i are torsion-free, then the solutions are EDT0L in $\mathsf{NSPACE}(n^2 \log n)$;*
3. *The normal form ζ can be chosen to be quasi-geodesic and regular.*

Proof We have from Lemma 5.6.5, that G has a finite index subgroup isomorphic to $B \times K_1 \times \cdots \times K_n$, where B is a finite index subgroup of A, and each K_i is a finite index subgroup of H_i, such that every conjugate of each of the subgroups K_i lies in the set $\{K_1, \ \ldots, \ K_n\}$. We have that B is virtually abelian and the groups K_i are non-elementary hyperbolic. Thus, all groups in $\text{FIN}(B)$ are virtually abelian, and all groups in $\text{FIN}(K_i)$ are hyperbolic for each i. We can equip each of these with a regular quasi-geodesic normal form. Theorem B and Lemma 5.6.6 imply that solutions to systems of equations are EDT0L in $\mathsf{NSPACE}(n^4 \log n)$ in all of these groups. The result then follows from Proposition 5.6.4. □

We can reformulate Theorem 5.6.8 in the following way.

Corollary 5.6.9. *Let G be a group that is virtually a direct product of hyperbolic groups (resp. torsion-free hyperbolic groups). Then the solutions to systems of equations in G are EDT0L in $\mathsf{NSPACE}(n^4 \log n)$ (resp. $\mathsf{NSPACE}(n^2 \log n)$), with respect to the normal form from Remark 5.6.2, which can be constructed to be quasi-geodesic and regular.*

As dihedral Artin groups are virtually a direct product of free groups, we have the following result. Note that the generating set and normal form will not be the standard Artin group ones; they are derived by taking the Schreier generators with respect to some finite index overgroup. As with Theorem 5.6.8, we can choose the regular geodesic normal forms for the free groups that dihedral Artin groups are virtually a direct product of, to give a regular quasi-geodesic normal form for these dihedral Artin groups.

Corollary 5.6.10. *The solutions to systems of equations in dihedral Artin groups are EDT0L in $\mathsf{NSPACE}(n^2 \log n)$, with respect to the normal form from Remark 5.6.2, which can be constructed to be quasi-geodesic and regular.*

Proof This follows from Corollary 5.6.9, together with the fact that dihedral Artin groups are virtually direct products of free groups (Lemma 5.2.2). □

Remark 5.6.11. The generating set and normal form from Remark 5.6.2 will be the Schreier generating set and normal form inherited from some finite index overgroup. This will not (necessarily) be a 'sensible' generating set and normal form for groups that are virtually a direct product of hyperbolic groups, or any of the standard normal forms used in dihedral Artin groups.

It is easy to change the generating set whilst preserving the property of EDT0L solutions. To add a (redundant) generator a, one can use the existing normal form, which never uses a, and so the solution language will be unchanged. To remove a redundant generator b, one can apply the free monoid homomorphism that maps b to some word w_b over the remaining generators and inverses, that represents b, to the solution language to remove all occurrences of b. Applying the free monoid homomorphism that maps b^{-1} to w_b^{-1} after this, will give a new solution language, with b removed from the generating set. As images of EDT0L languages under free monoid homomorphisms are EDT0L, this new solution language will also be EDT0L.

Changing the normal form is more difficult. Section 5 of [19] contains a successful attempt at this for hyperbolic groups, which uses Ehrenfeucht and Rozenberg's Copying Lemma [40]; a common tool used to show preimages of EDT0L languages under free monoid homomorphisms are EDT0L in certain cases, along with a result about languages of quasi-geodesics in hyperbolic groups. Languages of quasi-geodesics in virtually abelian groups are not so well behaved, and any attempt to show that alternative normal forms work in many of the groups considered here will need an alternative approach.

Chapter 6

Equations in the Heisenberg group

6.1 Introduction

This chapter is based on the author's work [66].

We consider single equations in the Heisenberg group in one variable. The fact that satisfiability of equations with one variable in the Heisenberg group is decidable was first shown by Repin [81]. We show that the solutions to these equations, when written as words in Mal'cev normal form are EDT0L, with an EDT0L system constructible in non-deterministic polynomial space.

Whilst this is only a 'partial result' towards understanding solution languages to equations in nilpotent groups, there are few cases in nilpotent groups in which the satisfiability of even single equations is decidable. Duchin, Liang and Shapiro [34] showed that the satisfiability of a single equation in a class 2 nilpotent group with a virtually cyclic commutator subgroup is decidable, however Roman'kov showed that this is not the case for general class 2 nilpotent groups [84]. Duncan, Evetts, Holt and Rees recently released some partial results about equations in solvable Baumslag-Solitar groups [36], although this remains the only other attempt at showing solution languages to equations are EDT0L in groups that are not non-positively curved. Duchin Liang and Shapiro also showed that the satisfiability of systems of equations in the Heisenberg group is undecidable. Thus the only ways of generalising this result within nilpotent groups is to increase the number of variables, or generalise to more class 2 nilpotent groups. Doing either of these would require understanding solutions to quadratic equations in the ring of integers in arbitrarily many variables.

Theorem 6.6.5. *Let L be the solution language to a single equation with one variable in the Heisenberg group, with respect to the Mal'cev generating set and normal form. Then*

1. *The language L is EDT0L;*
2. *An EDT0L system for L is constructible in $\mathsf{NSPACE}(n^8(\log n)^2)$, where the input size is the length of the equation as an element of $H(\mathbb{Z}) * F(X)$.*

Proving Theorem 6.6.5 involves reducing the problem of solving one-variable equations in the Heisenberg group to describing solutions to two-variable quadratic equations in the ring of integers. This uses a similar construction to the method of Duchin, Liang and Shapiro, which was used to show that the satisfiability of single equations in any class 2 nilpotent group with a virtually cyclic commutator subgroup is decidable [34]. The proofs that many nilpotent groups have an undecidable satisfiability of equations involve reducing the question to systems of quadratic equations in integers ([83], [50], [34]). Then Matijasevič's result that the satisfiability of systems of quadratic equations in integers is undecidable [72] can be applied. Duchin, Liang and Shapiro's positive result involves reducing the problem to single quadratic equations in integers, and then applying Siegel's result that the satisfiability of such equations is decidable [91].

Despite the extensive use EDT0L languages have had in describing solutions to group equations, there have been no attempts to describe solutions to equations in the ring of integers using EDT0L languages, other than linear equations, which are just equations in an abelian group. In order to make progress studying equations in the Heisenberg group, we will have to first learn to what extent EDT0L languages can be used to describe solutions to quadratic equations in the ring of integers. Our result for equations in the Heisenberg group involves reducing to the two-variable case of quadratic equations in integers.

Theorem 6.5.15. *Let*

$$\alpha X^2 + \beta XY + \gamma Y^2 + \delta X + \epsilon Y + \zeta = 0 \tag{6.1}$$

be a two-variable quadratic equation in the ring of integers, with a set S of solutions. Then

1. *The language $L = \{a^x \# b^y \mid (x,\ y) \in S\}$ is EDT0L over the alphabet $\{a,\ b,\ \#\}$;*
2. *Taking the input size to be* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|)$*, an EDT0L system for L is constructible in* $\mathsf{NSPACE}(n^4 \log n)$*.*

We prove this theorem using Lagrange's method. This involves reducing an arbitrary two-variable quadratic equation to a generalised Pell's equation $X^2 - DY^2 = N$. This again reduces to Pell's equation $X^2 - DY^2 = 1$, the set of solutions of which is well-understood. The reduction involves writing solutions to the two variable quadratic equation (6.1) in the form $\frac{\lambda x + \mu y + \xi}{\eta}$, where $(x,\ y)$ is a solution to some computable Pell's equation, and $\lambda,\ \mu,\ \xi,\ \eta \in \mathbb{Z}$ with $\eta \neq 0$ are all computable.

Showing that the set of solutions to Pell's equation can be expressed as an EDT0L language is not too difficult. However, studying $\frac{\lambda x + \mu y + \xi}{\eta}$ requires more work, particularly when the signs of λ, μ and ξ are not all the same, or when $|\eta| \geq 2$. To deal with the division, we use the concept of #-separated EDT0L systems, first introduced in [65], and work in the world of EDT0L languages.

Understanding $\lambda x + \mu y + \xi$, when λ, μ and ξ are not all the same sign is more difficult to resolve by manipulating EDT0L systems. This is because we represent the integer n by a^n; that is a word of length n comprising n occurrences of the letter a (when $n \geq 0$) or n occurrences of the letter a^{-1} (when $n \leq 0$). Adding 4 to -2 corresponds to concatenating a^4 with a^{-2}, resulting in $a^4 a^{-2}$, which is not equal as a word to a^2. We cannot simply 'cancel' as and a^{-1}s either; in general the language obtained by freely reducing all words in an EDT0L language is not EDT0L (it need not even be recursive). Therefore, we work with facts about the solutions themselves to show that for fixed integers λ, μ and ξ, the set

$$\{\lambda x + \mu y + \xi \mid (x,\ y) \text{ is a solution to } X^2 - DY^2 = 1\}$$

is sufficiently well-behaved that we can describe it using an EDT0L language. We can then apply our method for the 'division' to obtain the desired language.

We cover the preliminaries of the considered topics in Section 6.2. In Section 6.3, we prove our result about 'division' of EDT0L languages by a constant that is a key part of the proof that the solutions to two-variable quadratic equations in the ring of integers are EDT0L, which appears in Section 6.5. In Section 6.4, we study the solutions to Pell's equation, and their images under linear functions. The proof of the fact that solutions to two-variable quadratic equations are EDT0L involves reducing to the case of Pell's equation. This reduction is contained in Section 6.5. Section 6.6 includes the reduction from equations in the Heisenberg group to quadratic equations in the ring of the integers, and the proof that single equations in one variable in the Heisenberg group are expressible as EDT0L languages.

6.2 Preliminaries

6.2.1 Nilpotent groups

We start with the definitions of a nilpotent group and the Heisenberg group. For a comprehensive introduction to nilpotent groups we refer the reader to [23].

Definition 6.2.1. Let G be a group. Define $\gamma_i(G)$ for all $i \in \mathbb{Z}_{>0}$ inductively as follows:

$$\gamma_1(G) = G$$

$$\gamma_i(G) = [G,\ \gamma_{i-1}(G)] \text{ for } i > 1.$$

The subnormal series $(\gamma_i(G))$ is called the *lower central series* of G. We call G

nilpotent of *class* c if $\gamma_c(G)$ is trivial.

Definition 6.2.2. The *Heisenberg group* $H(\mathbb{Z})$ is the class 2 nilpotent group defined by the presentation

$$H(\mathbb{Z}) = \langle a,\ b,\ c \mid c = [a,\ b],\ [a,\ c] = [b,\ c] = 1\rangle.$$

Note that whilst the generator c is redundant, it is often easier to work with the generating set $\{a,\ b,\ c\}$ than $\{a,\ b\}$.

The *Mal'cev generating set* for the Heisenberg group is the set $\{a,\ b,\ c\}$.

6.2.2 Mal'cev normal form

We now define the normal form that we will be using to represent our solutions. This is used in [34], and we include the proof of uniqueness and existence for completeness.

The following facts about commutators in class 2 nilpotent groups will be used to induce the methods for 'pushing' bs past as in the Heisenberg group.

Lemma 6.2.3. *Let G be a class 2 nilpotent group, and $g,\ h \in G$. Then*

1. $[g^{-1},\ h^{-1}] = [g,\ h]$,
2. $[g^{-1},\ h] = [g,\ h]^{-1}$.

Proof For (1), since commutators are central,

$$[g^{-1},\ h^{-1}] = ghg^{-1}h^{-1} = ghg^{-1}h^{-1}ghh^{-1}g^{-1} = gh[g,\ h]h^{-1}g^{-1} = [g,\ h]ghh^{-1}g^{-1} = [g,\ h].$$

Similarly, for (2), we have

$$[g^{-1},\ h] = gh^{-1}g^{-1}h = gh^{-1}g^{-1}hgg^{-1} = g[g,\ h]^{-1}g^{-1} = gg^{-1}[g,\ h]^{-1} = [g,\ h]^{-1}.$$

□

Using Lemma 6.2.3, we now have a number of useful identities for 'pushing' as past

bs in expressions over the Mal'cev generating set.

Lemma 6.2.4. *The following identities hold for the Mal'cev generators of the Heisenberg group:*

$$ba = abc$$

$$ba^{-1} = a^{-1}bc^{-1}$$

$$b^{-1}a = ab^{-1}c^{-1}$$

$$b^{-1}a^{-1} = a^{-1}b^{-1}c.$$

Proof We have

$$ba = abb^{-1}a^{-1}ba = abc$$

$$ba^{-1} = a^{-1}bb^{-1}aba^{-1} = a^{-1}b[b,\ a^{-1}] = a^{-1}bc^{-1}$$

$$b^{-1}a = ab^{-1}ba^{-1}b^{-1}a = ab^{-1}[b^{-1},\ a] = ab^{-1}c^{-1}$$

$$b^{-1}a^{-1} = a^{-1}b^{-1}bab^{-1}a^{-1} = a^{-1}b^{-1}[b^{-1},\ a^{-1}] = a^{-1}b^{-1}c.$$

□

The following lemma allows us to define the Mal'cev normal form for the Heisenberg group.

Lemma 6.2.5. *For each $g \in H(\mathbb{Z})$ there exists a unique word of the form $a^i b^j c^k$ that represents g, where $i,\ j,\ k \in \mathbb{Z}$.*

Proof Existence: Let $w \in \{a,\ b,\ c,\ a^{-1},\ b^{-1},\ c^{-1}\}^*$. To transform w into an equivalent word in the form $a^i b^j c^k$, first note that c is central, so w is equal to uc^k, where $u \in \{a,\ b,\ a^{-1},\ b^{-1}\}^*$, and $k \in \mathbb{Z}$, which is obtained by pushing all cs and c^{-1}s in w to the right, then freely reducing. We can then look for any bs or b^{-1}s before as or a^{-1}s, and use the rules of Lemma 6.2.4 to 'swap' them, by adding a commutator.

After doing these swaps, we can push the 'new' cs and c^{-1}s to the back, to assume our word remains within $\{a,\ b,\ a^{-1},\ b^{-1}\}^*(\{c\}^* \cup \{c^{-1}\}^*)$. By repeating this process, we will eventually have no more as or a^{-1}s occurring after any b or b^{-1}, and so will be in the form $a^i b^j c^k$, where $i,\ j,\ k \in \mathbb{Z}$.

Uniqueness: Suppose $i_1,\ i_2,\ j_1,\ j_2,\ k_1,\ k_2 \in \mathbb{Z}$ are such that $a^{i_1} b j_1 c^{k_1} =_{H(\mathbb{Z})} a^{i_2} b^{j_2} c^{k_2}$. Then

$$\begin{aligned} 1 &= a^{i_1} b j_1 c^{k_1} (a^{i_2} b^{j_2} c^{k_2})^{-1} \\ &= a^{i_1} b^{j_1} c^{k_1} c^{-k_2} b^{-j_2} a^{-i_2} \\ &= a^{i_1} b^{j_1 - j_2} a^{-i_2} c^{k_1 - k_2} \\ &= a^{i_1} a^{-i_2} b^{j_1 - j_2} c^{-i_2(j_1 - j_2)} c^{k_1 - k_2} \\ &= a^{i_1 - i_2} b^{j_1 - j_2} c^{-i_2(j_1 - j_2) + k_1 - k_2}. \end{aligned}$$

As 1 lies in the commutator subgroup, we have that the above word lies in $\langle c \rangle$. But since c commutes with a and b, $a^i b^j \in \langle c \rangle$ if and only if $i = j = 0$. Thus $i_1 - i_2 = j_1 - j_2 = 0$. It follows that the above word equals $c^{k_1 - k_2}$. Since this is a freely reduced word in $\langle c \rangle$ as a power of c, this represents the identity if and only if $k_1 - k_2 = 0$. Thus $k_1 - k_2 = 0$, and the two words represent the same element of $H(\mathbb{Z})$. □

Definition 6.2.6. The *Mal'cev normal form* for the Heisenberg group is the normal form that maps an element $g \in H(\mathbb{Z})$ to the unique word of the form $a^i b^j c^k$, where $i,\ j,\ k \in \mathbb{Z}$, that represents g.

6.2.3 Equations in the ring of integers

We briefly define an equation in integers.

Definition 6.2.7. An *equation* in the ring of integers is an identity $(X_1,\ \ldots,\ X_n)f = 0$, where $(X_1,\ \ldots,\ X_n)f \in \mathbb{Z}[X_1,\ \ldots,\ X_n]$ is a polynomial. The indeterminates

$X_1, \ldots, X_n$ are called *variables*. An equation is called *quadratic* if the degree of $(X_1, \ldots, X_n)f$ is at most 2.

A *solution* to an equation $(X_1, \ldots, X_n)f = 0$ in the ring of integers is a ring homomorphism $\phi\colon \mathbb{Z}[X_1, \ldots, X_n] \to \mathbb{Z}$ that fixes $\mathbb{Z}$ pointwise, and such that $(X_1, \ldots, X_n)f\phi = 0$.

A *system* of equations in integers is a finite set of equations. A *solution* to the system is any ring homomorphism that is a solution to every equation in the system.

When we create algorithms that take equations in integers as input, we will explicitly state the size of the input.

Remark 6.2.8. As with group equations, we will usually use a tuple $(x_1, \ldots, x_n)$ rather than a ring homomorphism $\phi\colon \mathbb{Z}[X_1, \ldots, X_n] \to \mathbb{Z}$. The homomorphism ϕ can be obtained from the tuple by defining $X_1\phi = x_i$ for all i, and $n\phi = n$ for all $n \in \mathbb{Z}$. Since ϕ is a ring homomorphism, the action of ϕ on the remainder of $\mathbb{Z}[X_1, \ldots, X_n]$ is now determined.

6.2.4 Solution languages

We now define an analogous notion for systems of equations in the ring of integers. We pick a letter as a generator, and write the non-negative integer n as this letter to the power of n. For negative integers, we introduce an 'inverse' of this letter, and express each $n < 0$ as the inverse letter to the power of $|n|$.

Definition 6.2.9. Define $\mu\colon \mathbb{Z} \to \{a\}^* \cup \{a^{-1}\}^*$ by $n\mu = a^n$.

Let $\mathcal{E}$ be a system of equations in the ring of integers, with variables $X_1, \ldots, X_n$. The *solution language* to $\mathcal{E}$ is the language

$$\{(X_1)\phi\mu\# \cdots (X_n)\phi\mu \mid \phi \text{ is a solution to } \mathcal{E}\}$$

over $\{a,\ a^{-1},\ \#\}$.

6.3 'Dividing EDT0L' languages by a constant

The purpose of this section is to show that given an EDT0L language where all words are of the form $a^i\#b^j$, 'dividing' the number of as and the number of bs in a given word by constant values, and removing all words that are not divisible yields an EDT0L language. We proceed in a similar fashion to the arguments used in [65], Section 3, using #-separated EDT0L systems, however the argument for 'dividing' is new.

The concept of #-separated EDT0L systems was used in [65] to show that solution languages to systems of equations in direct products of groups where systems of equations have EDT0L solution languages are also EDT0L. We use a slightly different definition here: we only need a single # rather than arbitrarily many, so our definition is less general, and we also insist that the start word is of a specified form. The latter assumption does not affect the expressive power of these systems; preconcatenating the rational control with an appropriate endomorphism can convert a #-separated system with an arbitrary start word into one with a start word of the form we use.

Definition 6.3.1. Let Σ be an alphabet, and $\# \in \Sigma$. A *#-separated EDT0L system* is an EDT0L system $\mathcal{H}$, with an extended alphabet C, a terminal alphabet Σ and a start word of the form $\bot_1 \# \bot_2$, where $\bot_1,\ \bot_2 \in C \backslash \{\#\}$, and $c\phi = \#$ if and only if $c = \#$, for every $c \in C$, and ϕ in any fixed choice of alphabet of the rational control.

For space complexity purposes, we will need bounds on the size of extended alphabets, and the size of images of letters under endomorphisms in the rational control in many of the EDT0L systems we use. We define the term g-bounded to capture this.

Definition 6.3.2. Let $\mathcal{H} = (\Sigma,\ C,\ \bot_1 \# \bot_2,\ \mathcal{R})$ be a #-separated EDT0L system, and let $g \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ be a function in terms of a given input size I. Let B be an alphabet of $\mathcal{R}$. We say that $\mathcal{H}$ is *g-bounded* if

1. $|C| \leq (I)g$;
2. $\max\{|c\phi| \mid c \in C,\ \phi \in B\} \leq (I)g$.

We will need the fact that the class of languages accepted by #-separated EDT0L systems is closed under finite unions, with space complexity properties being preserved when taking these unions.

Lemma 6.3.3. *Let L and M be languages over an alphabet Σ, accepted by #-separated EDT0L systems $\mathcal{H}$ and $\mathcal{G}$, that are both g-bounded and constructible in* $\mathsf{NSPACE}(f)$*, for some* $f,\ g\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$*. Then*

1. *There is a #-separated EDT0L system $\mathcal{F}$ for $L \cup M$;*
2. *The system $\mathcal{F}$ is constructible in* $\mathsf{NSPACE}(f)$*;*
3. *The system $\mathcal{F}$ is $(2g+2)$-bounded.*

Proof Let $\mathcal{H} = (\Sigma,\ C,\ \perp_1 \#\$_1,\ \mathcal{R})$ and $\mathcal{G} = (\Sigma,\ D,\ \perp_2 \#\$_2,\ \mathcal{S})$. Let B_1 and B_2 be the alphabets of $\mathcal{R}$ and $\mathcal{S}$, respectively. We can assume without loss of generality that endomorphisms in $B_1 \cup B_2$ fix elements of Σ, and also that $C \backslash \Sigma$ and $D \backslash \Sigma$ are disjoint.

Let $\perp$ and \$ be symbols not already used, and let $E = C \cup D \cup \{\perp,\ \$\}$. For each $\phi \in B_1$, define $\bar{\phi}$ to be the extension of ϕ to E by $d\bar{\phi} = d$ for all $d \in E \backslash C$. Similarly extend each $\phi \in B_2$ to $\bar{\phi} \in \mathrm{End}(E^*)$ by $c\bar{\phi} = c$ for all $c \in E \backslash D$. Define $\theta_1,\ \theta_2 \in \mathrm{End}(E^*)$ by

$$c\theta_1 = \begin{cases} \perp_1 & c = \perp \\ \$_1 & c = \$ \\ c & \text{otherwise,} \end{cases} \qquad c\theta_2 = \begin{cases} \perp_2 & c = \perp \\ \$_2 & c = \$ \\ c & \text{otherwise.} \end{cases}$$

By construction, $L \cup M$ is accepted by the #-separated EDT0L system $\mathcal{F} = (\Sigma,\ E,\ \perp \#\$,\ \theta_1 \mathcal{R} \cup \theta_2 \mathcal{S})$.

Note that θ_1 and θ_2 can both be constructed in constant space, and thus the rational control of $\mathcal{F}$ is constructible in $\mathsf{NSPACE}(f)$. The start word is constructible in constant space. As a union of C and D with a constant number of additional symbols, E can be constructed using the same information required to construct C and D, and is thus constructible in $\mathsf{NSPACE}(f)$.

We have that $|E| = |C| + |D| + 2$, and so is bounded by $2g + 2$. In addition,

$B = B_1 \cup B_2 \cup \{\theta_1,\ \theta_2\}$, and so $\max\{|c\phi| \mid c \in E,\ \phi \in B\} = \max(g,\ 2) \leq 2g + 2$. □

We can now prove the central result of this section, about 'division' of certain EDT0L languages by a constant. To show the space complexity properties, we need the EDT0L system we start with to be exponentially bounded by the space complexity in which is can be constructed. We will use the following notation:

Notation 6.3.4. Let Σ be an alphabet, $a \in \Sigma$, and $w \in \Sigma^*$. Define $\#_a(w)$ to be the number of occurrences of the letter a within w.

Lemma 6.3.5. *Let $X \subseteq \mathbb{Z}^2_{\geq 0}$ be such that for each $x,\ y \in \mathbb{Z}_{\geq 0}$ there is at most one $x' \in \mathbb{Z}_{\geq 0}$ such that $(x,\ x') \in X$, and at most one $y' \in \mathbb{Z}_{\geq 0}$ with $(y',\ y) \in X$. Let $\gamma,\ \zeta \in \mathbb{Z}$ be non-zero. Let*

$$L = \{a^x \# b^y \mid (x,\ y) \in X\}.$$

1. *If L is EDT0L, then so is the language*

$$L_{\gamma,\zeta} = \{a^{\frac{x}{\gamma}} \# b^{\frac{y}{\zeta}} \mid (x,\ y) \in X,\ \gamma | x,\ \zeta | y\};$$

2. *If L is accepted by a $\#$-separated EDT0L system $\mathcal{H} = (\Sigma,\ C,\ \perp_1 \# \perp_2,\ \mathcal{R})$ that is $\exp(f)$-bounded and constructible in $\mathsf{NSPACE}(f)$, where $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is at least linear, then $L_{\gamma,\eta}$ is accepted by an EDT0L system that is constructible in $\mathsf{NSPACE}(fgh)$, where g is linear in $|\gamma|$, and h is linear in $|\zeta|$.*

Proof We will use $\mathcal{H}$ to define an EDT0L system for

$$M = \{a^{\frac{x}{|\gamma|}} \# b^y \mid (x,\ y) \in X,\ \gamma | x\}.$$

Firstly note that if $|\gamma| = 1$, then $M = L$, and thus M is accepted by $\mathcal{H}$, which satisfies the conditions in (2). So assume $|\gamma| \geq 2$. Let ¢ and \$ be symbols not already used. Let $\hat{c}^\nu$ be a distinct copy of c for each $c \in C$ and $\nu \in \{¢,\ \$\}^*$. Let $C^{\text{ind}} = \{\hat{c}^\nu \mid \nu \in \{¢,\ \$\}^*,\ |\nu| \leq |\gamma|\} \sqcup C \sqcup \{F\}$, where F is a new symbol. We will use F as a 'fail symbol'.

Let $B \subseteq \text{End}(C^*)$ be the alphabet of $\mathcal{R}$. For each $\phi \in B$, the finite set $\Phi_\phi \subseteq \text{End}((C^{\text{ind}})^*)$ of all $\psi \in \text{End}((C^{\text{ind}})^*)$ is defined as follows. If $\nu \in \{¢,\ \$\}^*$ satisfies $|\nu| \leq |\gamma|$, and $c \in C$ is such that $c\phi = d_1 \cdots d_n$, with $n \geq 1$, $d_1, \ \ldots, \ d_n \in C$ (in particular, $c\phi \neq \varepsilon$), then

$$\hat{c}^\nu \psi = \hat{d}_1^{\alpha_1} \cdots \hat{d}_n^{\alpha_n},$$

for some $\alpha_1, \ \ldots, \ \alpha_n \in \{¢,\ \$\}^*$ such that $|\alpha_i| \leq |\gamma|$ for all i, and one of the following holds:

1. $\#_\$(\alpha_1 \cdots \alpha_n) = \#_\$(\nu)$, and $\#_¢(\alpha_1 \cdots \alpha_n) = \#_¢(\nu)$;
2. $\#_\$(\alpha_1 \cdots \alpha_n) = \#_\$(\nu) + 1$, and $\#_¢(\alpha_1 \cdots \alpha_n) = \#_¢(\nu) + |\gamma| - 1$.

If $c\phi = \varepsilon$, then

$$\hat{c}^\nu \psi = \begin{cases} F & \nu \neq \varepsilon \\ \varepsilon & \nu = \varepsilon. \end{cases}$$

In addition, ψ fixes F, and acts the same way as ϕ on letters in C. We define Φ_ϕ to be the set of all endomorphisms ψ satisfying these conditions, and let $\bar{\mathcal{R}}$ be the rational set of endomorphisms defined by replacing each occurrence of ϕ within $\mathcal{R}$ with Φ_ϕ. Now define $\theta \in \text{End}((C^{\text{ind}})^*)$ by

$$\hat{c}^\nu \theta = \begin{cases} c & c \in \Sigma, \ \nu = \$ \\ \varepsilon & c \in \Sigma, \ \nu = ¢ \\ F & \text{otherwise}, \end{cases} \qquad F\theta = F, \qquad c\theta = c \text{ for all } c \in C.$$

Let $\mathcal{G} = (\Sigma, \ C^{\text{ind}}, \ \hat{\bot}_1 \# \bot_2, \ \bar{\mathcal{R}}\theta)$. By construction, any word in $\hat{\bot}_1 \bar{\mathcal{R}}$ either contains an F, or is a word in $\bot_1 \mathcal{R}$ with hats on letters and indices that concatenate to form a word $\nu \in \{¢,\ \$\}^*$ of length $n|\gamma|$ for some $n \in \mathbb{Z}_{\geq 0}$, with $\#_¢(\nu) = n(|\gamma| - 1)$, and $\#_\$(\nu) = n$. Thus the set of words in $\hat{\bot}_1 \bar{\mathcal{R}}\theta \cap \Sigma$ equals $\{a^{\frac{x}{|\gamma|}} \mid (x, \ y) \in S, \ \gamma | x\}$. It follows that $\mathcal{G}$ accepts M.

We now consider the space complexity in which $\mathcal{G}$ can be built. Firstly, note that to output C^{ind} we simply need to output $(2^{|\gamma|+1} - 1)$ (the number of words of length at most $|\gamma|$ over a two letter alphabet) additional copies of C, plus the letter F. Doing this simply requires us to track the copy we're on, and since $\log(2^{|\gamma|+1} - 1)$ is linear in $|\gamma|$, this can be done in $\mathsf{NSPACE}(f)$. The start word can be output in constant space.

We now consider the rational control. To construct $\bar{\mathcal{R}}$, we need to follow the process to construct $\mathcal{R}$, except we need to construct Φ_ϕ whenever the finite-state automaton for $\mathcal{R}$ constructs ϕ. Let $\psi \in \Phi_\phi$, and note that if $c \in C$ and $\nu \in \{¢,\ \$\}^*$ is such that $|\nu| \leq |\gamma|$, then there are at most

$$(\max\{|c\varphi| \mid c \in C,\ \varphi \in B\})^{|\nu|} \leq (\max\{|c\varphi| \mid c \in C,\ \varphi \in B\})^{|\gamma|}$$

possible values that $\hat{c}^\nu\psi$ can take. As a result,

$$|\Phi_\phi| \leq (\max\{|c\varphi| \mid c \in C,\ \varphi \in B\})^{|\gamma|} \cdot |C| \cdot (2^{|\gamma|+1} - 1).$$

Thus

$$\log|\Phi_\phi| \leq |\gamma| \log(\max\{|c\varphi| \mid c \in C,\ \varphi \in B\}) + \log|C| + (|\gamma| + 1)\log 2.$$

To construct Φ_ϕ, we simply need to store the information required to construct ϕ, together with a counter to tell us how many ψ in Φ_ϕ we have already constructed. Since $\log|\Phi_\phi|$ is bounded by fg for some linear function g in $|\gamma|$, we can construct Φ_ϕ, and hence $\bar{\mathcal{R}}$ in $\mathsf{NSPACE}(fg)$. As θ can be constructed in constant space, it follows that the rational control, and hence $\mathcal{G}$, can be constructed in $\mathsf{NSPACE}(fg)$.

To see that the language accepted by $\mathcal{G}$ is in fact M, first note that for any $\bar{\phi} \in \bar{\mathcal{R}}$, $\hat{\perp}_1\bar{\phi}$ will be obtained from a word $\perp_1 \phi$, for some $\phi \in \mathcal{R}$ by attaching $k(|\gamma| - 1)$ ¢ indices and k \$ indices, for some $k \in \mathbb{Z}_{\geq 0}$. This will only be accepted if $(\perp_1 \# \perp_2)\phi \in \Sigma^*$, and every letter in $\perp_1 \phi$ has precisely one index on it. In such a case, $|\perp_1 \phi| = k|\gamma|$ (in fact $|\perp_1 \phi| = a^{\pm k|\gamma|}$), and precisely k of these letters will be indexed by \$, the rest being indexed by ¢. Hitting such a word with θ will delete all letters indexed with a single ¢, and map the \$-indexed as to a and \$-indexed a^{-1}s to a^{-1}, leaving the word $a^{\pm x}\#b^y$ to be accepted. Thus M is accepted by $\mathcal{G}$.

We now show that

$$N = \{a^{\frac{x}{\gamma}}\#b^y \mid (x,\ y) \in X,\ \gamma | x\}$$

is accepted by an EDT0L system, constructible in $\mathsf{NSPACE}(fg)$. Note that if $\gamma \geq 0$, then $M = N$, and there is nothing to prove. Otherwise, $\gamma < 0$. Define $\pi \in \mathrm{End}((C^{\mathrm{ind}})^*)$ by $a\pi = a^{-1}$, $a^{-1}\pi = a$ and all other letters are fixed by π. Then $(\Sigma,\ C^{\mathrm{ind}},\ \perp_1 \# \perp_2,\ \bar{\mathcal{R}}\theta\pi)$ accepts N, as we have just flipped the sign of the as in M. Moreover, as $\mathcal{G}$ is constructible in $\mathsf{NSPACE}(fg)$, so is our system for N. In addition, the stated bounds on the size of the extended alphabet and the images of endomorphisms of $\mathcal{G}$ hold for our system for N as well.

To obtain an EDT0L system for $\{a^{\frac{x}{\gamma}} \# b^{\frac{y}{\zeta}} \mid (x,\ y) \in X,\ \gamma | x,\ \zeta | y\}$ from N, we simply apply the same method we used to obtain N from L, except modifying $\perp_2$ and b, rather than $\perp_1$ and a. $\square$

6.4 Pell's equation

The purpose of this section is to study solutions to Pell's equation, which eventually allows us to show that the solution language to a quadratic equation in the ring of integers is EDT0L.

We start with a lemma that shows languages that arise as part of recursively defined integer sequences with non-negative integer coefficients are EDT0L. We will later show that solutions to Pell's equation are of this form.

Lemma 6.4.1. *Let $(p_n)_{n \geq 0}$, $(q_n)_{n \geq 0}$ and $(r_n)_{n \geq 0}$ be integer sequences, defined recursively by a relation*

$$p_n = \alpha_1 p_{n-1} + \alpha_2 q_{n-1} + \alpha_3 r_{n-1},\ q_n = \beta_1 p_{n-1} + \beta_2 q_{n-1} + \beta_3 r_{n-1},\ r_n = \gamma_1 p_{n-1} + \gamma_2 q_{n-1} + \gamma_3 r_{n-1}$$

where $\alpha_1,\ \alpha_2,\ \alpha_3,\ \beta_1,\ \beta_2,\ \beta_3,\ \gamma_1,\ \gamma_2,\ \gamma_3 \in \mathbb{Z}_{\geq 0}$. Suppose also that $p_0,\ q_0,\ r_0 \in \mathbb{Z}_{\geq 0}$ or $p_0,\ q_0,\ r_0 \in \mathbb{Z}_{\leq 0}$. Then

1. *The language $L = \{a^{p_n} \mid n \in \mathbb{Z}_{\geq 0}\}$ is EDT0L;*
2. *Taking the input size to be $I = \max(\alpha_1,\ \alpha_2,\ \alpha_3,\ \beta_1,\ \beta_2,\ \beta_3,\ \gamma_1,\ \gamma_2,\ \gamma_3,\ p_0,\ q_0,\ r_0)$, an EDT0L system $\mathcal{H}$ for L is constructible in non-deterministic logarithmic space;*

3. *The system H is f-bounded for some linear function f;*
4. *The rational control of $\mathcal{H}$ is of the form $\theta\varphi^*\psi$, and $\perp \theta\varphi^n\psi = a^{p_n}$, where $\perp$ is the start word of $\mathcal{H}$.*

Proof We will define an EDT0L system to accept L. Let $\Sigma = \{a, \ a^{-1}\}$. Our extended alphabet will be $C = \Sigma \cup \{a_p, \ a_p^{-1}, \ a_q, \ a_q^{-1}, \ a_r, \ a_r^{-1}, \ \perp\}$, and our start word will be $\perp$. Define $\theta \in \mathrm{End}(C^*)$ by

$$c\theta = \begin{cases} a_p^{p_0} a_q^{q_0} a_r^{r_0} & c = \perp \\ c & \text{otherwise.} \end{cases}$$

Define $\varphi \in \mathrm{End}(C^*)$ by

$$a_p^{\pm 1}\varphi = a_p^{\pm\alpha_1} a_q^{\pm\beta_1} a_r^{\pm\gamma_1}$$

$$a_q^{\pm 1}\varphi = a_p^{\pm\alpha_2} a_q^{\pm\beta_2} a_r^{\pm\gamma_2}$$

$$a_r^{\pm 1}\varphi = a_p^{\pm\alpha_3} a_q^{\pm\beta_3} a_r^{\pm\gamma_3}$$

and fix all other letters. Finally, define $\psi \in \mathrm{End}(C^*)$ by

$$a_p^{\pm 1}\psi = a^{\pm 1}$$

$$a_q^{\pm 1}\psi = a_r^{\pm 1}\psi = \varepsilon,$$

and all other letters are fixed. Our rational control will be $\theta\varphi^*\psi$.

First note that $u = \perp \theta\varphi^n$ contains either a_p or a_p^{-1}, but not both, and the same holds for a_q and a_q^{-1}, and a_r and a_r^{-1}. So we can abuse notation and take the definition of $\#_{a_p}$ when applied to such a word to be $\#_{a_p}(u)$ if u contains an a_p, $-\#_{a_p^{-1}}(u)$ if u contains an a_p^{-1}, and 0 if it contains neither. We similarly abuse notation with $\#_{a_q}$ and $\#_{a_r}$.

We will show by induction that $u = \perp \theta\varphi^n$ satisfies $\#_{a_q}(u) = p_n$, $\#_{a_q}(u) = q_n$, and $\#_{a_r}(u) = r_n$. This holds by definition for $n = 0$. Inductively suppose it is true for some $k - 1$. Then $\perp \theta\varphi^k = u$, for some $u \in \{a_p, \ a_p^{-1}, \ a_q, \ a_q^{-1}, \ a_r, \ a_r^{-1}\}^*$, with

$\#_{a_q}(u) = p_{k-1}$, $\#_{a_q}(u) = q_{k-1}$, and $\#_{a_r}(u) = r_{k-1}$. Using the definition of φ, and our inductive hypothesis we have

$$\#_{a_p}(u\varphi) = \alpha_1 \#_{a_p}(u) + \alpha_2 \#_{a_q}(u) + \alpha_3 \#_{a_r}(u) = \alpha_1 p_{k-1} + \alpha_2 q_{k-1} + \alpha_3 r_{k-1} = p_k$$

$$\#_{a_q}(u\varphi) = \beta_1 \#_{a_p}(u) + \beta_2 \#_{a_q}(u) + \beta_3 \#_{a_r}(u) = \beta_1 p_{k-1} + \beta_2 q_{k-1} + \beta_3 r_{k-1} = q_k$$

$$\#_{a_r}(u\varphi) = \gamma_1 \#_{a_p}(u) + \gamma_2 \#_{a_q}(u) + \gamma_3 \#_{a_r}(u) = \gamma_1 p_{k-1} + \gamma_2 q_{k-1} + \gamma_3 r_{k-1} = r_k.$$

It now follows that $\perp \theta\varphi^n\psi = a^{p_n}$, and thus (1) and (4) are true.

We now show that the EDT0L system $(\Sigma,\ C,\ \perp,\ \theta\varphi^*\psi)$ is constructible in non-deterministic linear space. Writing down Σ, C, ψ and the start word can be done in constant space. Writing down θ can be done by remembering p_0, q_0 and r_0, and thus can be done in non-deterministic logarithmic space, since storing an integer r requires $\log(r)$ plus a constant bits. It remains to show that φ can be defined in non-deterministic logarithmic space. To write down φ, we simply need to know the coefficients α_i, β_i and γ_i for $i \in \{1,\ 2,\ 3\}$. Since these can all be stored using $\log \alpha_i$, $\log \beta_i$ and $\log \gamma_i$ bits, respectively plus constants, (2) follows.

Finally note that $|C| = 8$, which is constant. In addition, $|c\varphi|$, for $c \in C$, is bounded by a linear function of the values α_i, β_i and γ_i, $|c\theta| \leq p_0 + q_0 + r_0$, and $|c\psi| \leq 1$. We have now shown (3). □

To show that the solution language to a general quadratic equation in two variables is EDT0L, we follow Lagrange's method to reduce it to the generalised Pell's equation, and then to Pell's equation. This reduction is detailed in [89]. We start with the definition of Pell's equation.

Definition 6.4.2. *Pell's equation* is the equation $X^2 - DY^2 = 1$ in the ring of the integers, where X and Y are variables, and $D \in \mathbb{Z}_{>0}$ is not a perfect square. The *fundamental solution* to Pell's equation $X^2 - DY^2 = 1$ is the minimal (with respect to the ℓ^1 metric on $\mathbb{Z}^2$) non-negative integer solution that is not $(1,\ 0)$.

The solutions to Pell's equation have long been understood. The following lemma details one of several ways of constructing them.

Lemma 6.4.3 ([4], Theorem 3.2.1)**.** *There are infinitely many solutions to Pell's equation* $X^2 - DY^2 = 1$*, and these are* $\{(x_n,\ y_n) \mid n \in \mathbb{Z}_{\geq 0}\}$*, where* $(x_0,\ y_0) = (1,\ 0)$*, and* $(x_n,\ y_n)$ *is recursively defined by*

$$x_n = x_1 x_{n-1} + D y_1 y_{n-1}, \qquad y_n = y_1 x_{n-1} + x_1 y_{n-1},$$

where $(x_1,\ y_1)$ *is the fundamental solution.*

We give an explicit example of Pell's equation and its solutions.

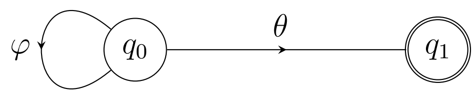
Example 6.4.4. Consider Pell's equation $X^2 - 2Y^2 = 1$. It is not hard to check using brute force that the fundamental solution is $(3,\ 2)$ (although there are more efficient methods of doing this: see for example [4]). Thus by Lemma 6.4.3, we can construct the set of all solutions using the sequence $(x_n,\ y_n) \subseteq \mathbb{Z}^2$, defined recursively by $(x_0,\ y_0) = (1,\ 0)$, and

$$x_n = 3x_{n-1} + 4y_{n-1}, \qquad y_n = 2x_{n-1} + 3y_{n-1}.$$

At this point, we could just apply Lemma 6.4.1 and Theorem 3.3.2 to show that the language $\{a^x \# a^y \mid (x,\ y) \in \mathbb{Z}^2_{\geq 0}$ is a solution to $X^2 - 2Y^2 = 1\}$ is EDT0L, however we will explicitly construct an EDT0L system. Our extended alphabet will be $C = \{a_x,\ \bar{a}_x,\ \ a_y,\ \bar{a}_y,\ a,\ \#\}$ and our start word will be $a_x \# \bar{a}_x$. Let $\varphi \in End(C^*)$ be defined by

$$\begin{aligned} a_x\varphi &= a_x^3 \bar{a}_y^2 & \qquad \bar{a}_x\varphi &= \bar{a}_x^3 a_y^2 \\ a_y\varphi &= \bar{a}_x^4 a_y^3 & \qquad \bar{a}_y\varphi &= a_x^4 \bar{a}_y^3, \\ a\varphi &= a & \qquad \#\varphi &= \#. \end{aligned}$$

Figure 6.1: Rational control for $L = \{a^x\#a^y \mid (x,\ y) \in \mathbb{Z}^2_{\geq 0}$ is a solution to $X^2 - 2Y^2 = 1\}$, with start state q_0 and accept state q_1.



Define $\theta \in \text{End}(C^*)$ by

$$a_x\theta = a_y\theta = a\theta = a$$

$$\bar{a}_x\theta = \bar{a}_y\theta = \varepsilon$$

$$\#\theta = \#.$$

Our rational control will be $\varphi^*\theta$ (alternatively, see Figure 6.1). Recall that for any word w and letter b, we use $\#_b(w)$ to denote the number of occurrences of b within w. By construction, $\#_{a_x}(a_x\#\bar{a}_x\varphi^n) = \#_{\bar{a}_x}(a_x\#\bar{a}_x\varphi^n) = x_n$ and $\#_{a_y}(a_x\#\bar{a}_x\varphi^n) = \#_{\bar{a}_y}(a_x\#\bar{a}_x\varphi^n) = y_n$, and thus $a_x\#\bar{a}_x\varphi^n\theta = a^{x_n}\#a^{y_n}$.

In addition to the recursive structure of all solutions, we need a bound on the size of the fundamental solution. This allows us to give a bound on the space complexity in which the EDT0L system can be constructed.

Lemma 6.4.5 ([63], Section 3)**.** *Let $(x_1,\ y_1)$ be the fundamental solution to Pell's equation $X^2 - DY^2 = 1$. Then*

$$\log(x_1 + y_1\sqrt{D}) < \sqrt{D}(\log(4D) + 2).$$

Understanding solutions to arbitrary two-variable quadratic equations using Lagrange's method requires us to have an understanding of the images of the solutions to Pell's equation under linear functions: that is $\alpha x + \beta y + \gamma$ for constant $\alpha,\ \beta,\ \gamma \in \mathbb{Z}$, where $(x,\ y)$ is a solution. If α, β and γ are either all non-negative or all non-positive, this corresponds to concatenating EDT0L languages in parallel, which is not too difficult using standard EDT0L constructions.

On the other hand, if the signs of these three integers are not all the same, more work needs to be done. This occurs because we represent the integer $n \in \mathbb{Z}$ by a^n, where a is a letter. Thus if we want to 'add' -3 and 5, this corresponds in language terms to trying to concatenate a^{-3} and a^5, which results in $a^{-3}a^5$, which is not equal (as a word) to a^2. One cannot, in general, freely reduce all words in an EDT0L language to form an EDT0L language. There are in fact cases where such a reduction will result in a language that is not recursive; that is a language which is not accepted by a Turing machine, or whose complement is not accepted by a Turing machine.

To tackle the harder cases presented to us by 'subtraction', we instead study the integer sequences themselves, and show they satisfy recurrence relations that can be used to define EDT0L systems.

Lemma 6.4.6. *Let (x_n), $(y_n) \subseteq \mathbb{Z}_{\geq 0}$ be sequences of solutions to Pell's equation $X^2 - DY^2 = 1$. Let α, $\beta \in \mathbb{Z}_{\geq 0}$. Let $(z_n) \subseteq \mathbb{Z}$ be the sequence defined by $z_n = \alpha x_n - \beta y_n$. Then, for all $n \in \mathbb{Z}_{\geq 2}$*

1. $x_n = 2x_1 x_{n-1} - x_{n-2}$;
2. $y_n = 2x_1 y_{n-1} - y_{n-2}$;
3. $z_n = 2x_1 z_{n-1} - z_{n-2}$.

Proof We will proceed by induction on n to show (1) and (2). First note that

$$2x_1 y_1 - y_0 = x_1 y_1 + y_1 x_1 = y_2.$$

Additionally,

$$2x_1 x_1 - x_0 = x_1^2 + (x_1^2 - 1) = x_1^2 + Dy_1^2 = x_2.$$

Thus (1) and (2) hold when $n = 2$. Suppose the result holds when $n = k$. Then

$$x_{k+1} = x_1 x_k + D y_1 y_k$$

$$= x_1(2x_1 x_{k-1} - x_{k-2}) + D y_1(2x_1 y_{k-1} - y_{k-2})$$

$$= 2x_1(x_1 x_{k-1} + D y_1 y_{k-1}) - (x_1 x_{k-2} + D y_1 y_{k-2})$$

$$= 2x_1 x_k - x_{k-1}.$$

$$y_{k+1} = y_1 x_k + x_1 y_k$$

$$= y_1(2x_1 x_{k-1} - x_{k-2}) + x_1(2x_1 y_{k-1} - y_{k-2})$$

$$= 2x_1(y_1 x_{k-1} + x_1 y_{k-1}) - (y_1 x_{k-2} + x_1 y_{k-2})$$

$$= 2x_1 y_k - y_{k-1}.$$

It remains to show (3). We have, using (1) and (2),

$$z_n = \alpha x_n - \beta y_n$$

$$= \alpha(2x_1 x_{n-1} - x_{n-2}) - \beta(2x_1 y_{n-1} - y_{n-2})$$

$$= 2x_1(\alpha x_{n-1} - \beta y_{n-1}) - (\alpha x_{n-2} - \beta y_{n-2})$$

$$= 2x_1 z_{n-1} - z_{n-2}.$$

□

Using Lemma 6.4.6, we can now prove some results about the sequence (z_n) that show that it is indeed a type of sequence as mentioned by Lemma 6.4.1.

Lemma 6.4.7. *Let (x_n), $(y_n) \subseteq \mathbb{Z}_{\geq 0}$ be sequences of solutions to Pell's equation $X^2 - DY^2 = 1$. Let α, $\beta \in \mathbb{Z}_{\geq 0}$. Let $(z_n) \subseteq \mathbb{Z}$ be the sequence defined by $z_n = \alpha x_n - \beta y_n$. Then*

1. *If $N = \left\lceil \log_2 \frac{\alpha}{\beta} \right\rceil$, then $(z_n)_{n \geq N} \subseteq \mathbb{Z}_{\geq 0}$ or $(z_n)_{n \geq N} \subseteq \mathbb{Z}_{<0}$;*

2. *The sequence* $(w_n)_{n\geq 1} \subseteq \mathbb{Z}$ *defined by* $w_n = z_n - z_{n-1}$ *satisfies for all* $n \in \mathbb{Z}_{\geq 2}$*,*

$$z_n = (2x_1 - 1)z_{n-1} + w_{n-1}, \qquad w_n = (2x_1 - 2)z_{n-1} + w_{n-1};$$

3. *If* $(z_n)_{n\geq N}$ *is a sequence of non-negative integers then* $(w_n)_{n\geq N}$ *is, and if* $(z_n)_{n\geq N}$ *is a sequence of non-positive integers then* $(w_n)_{n\geq N}$ *is;*
4. *The sequence* $(w_n)_{n\geq N}$ *is monotone;*
5. *If* $\gamma \in \mathbb{Z}$ *and* $M = \left\lceil \log_2 \frac{(\gamma+3)\alpha}{\beta} \right\rceil$*, then* $(z_n + \gamma)_{n\geq M},\ (w_n + \gamma)_{n\geq M} \subseteq \mathbb{Z}_{\geq 0}$ *or* $(z_n + \gamma)_{n\geq M},\ (w_n + \gamma)_{n\geq M} \subseteq \mathbb{Z}_{\leq 0}$*.*

Proof We start by showing (1). Let $\gamma = \frac{\beta}{\sqrt{D}}$. Then, if $n \in \mathbb{Z}_{\geq 0}$,

$$z_n = \alpha x_n - \beta y_n = \alpha x_n - \gamma\sqrt{D}y_n.$$

We have that $z_n \geq 0$ if and only if $z_n(\gamma x_n + \alpha\sqrt{D}y_n) \geq 0$. Note that

$$\begin{aligned} z_n(\gamma x_n + \alpha\sqrt{D}y_n) &= (\alpha x_n - \gamma\sqrt{D}y_n)(\gamma x_n + \alpha\sqrt{D}y_n) \\ &= \alpha\gamma x_n^2 + \alpha^2\sqrt{D}x_n y_n - \gamma^2\sqrt{D}x_n y_n - \alpha\gamma D y_n^2 \\ &= \alpha\gamma(x_n^2 - Dy_n^2) + \sqrt{D}x_n y_n(\alpha^2 - \gamma^2) \\ &= \alpha\gamma + \sqrt{D}x_n y_n(\alpha^2 - \gamma^2). \end{aligned}$$

If $\alpha \geq \gamma$, the above expression must be at least 0, so $z_n \geq 0$ for all $n \in \mathbb{Z}_{\geq 0}$, and there is nothing to prove. Otherwise, suppose $\gamma > \alpha$, and write $\gamma = \alpha + \delta$ for some $\delta > 0$. Then

$$\begin{aligned} z_n(\gamma x_n + \alpha\sqrt{D}y_n) &= \alpha\gamma + \sqrt{D}x_n y_n(\alpha^2 - \gamma^2) \\ &= \alpha(\alpha + \delta) + \sqrt{D}x_n y_n(\alpha^2 - (\alpha + \delta)^2) \\ &= \alpha^2 + \alpha\delta - \sqrt{D}x_n y_n(\delta^2 + 2\alpha\delta). \end{aligned}$$

It follows that $z_n < 0$ if and only if $\alpha^2 + \alpha\delta - \sqrt{D}x_ny_n(\delta^2 + 2\alpha\delta) < 0$. That is,

$$x_ny_n > \frac{\alpha^2 + \alpha\delta}{\sqrt{D}(\delta^2 + 2\alpha\delta)}.$$

Noting that x_n and y_n are both strictly increasing, and if $n \geq 1$, $x_ny_n > 1$, it suffices to find $N \in \mathbb{Z}_{>0}$ such that if $n = N$ the above inequality holds. By Lemma 6.4.3, we have that $x_n \geq x_1x_{n-1}$ and $y_n \geq x_1y_{n-1}$, and so $x_ny_n \geq x_1^{2n-1}y_1$. Noting that $x_1 \geq 2$ and $y_1 \geq 1$, it follows that $x_ny_n \geq 2^n$. Note that $\frac{\alpha}{\beta} = \frac{\alpha\gamma}{\sqrt{D}} = \frac{\alpha^2+\alpha\delta}{\sqrt{D}} \geq \frac{\alpha^2+\alpha\delta}{\sqrt{D}(\delta^2+2\alpha\delta)}$, and so choosing $N = \left\lceil \log_2 \frac{\alpha}{\beta} \right\rceil$ will satisfy the stated conditions.

For (2), let $n \in \mathbb{Z}_{\geq 2}$. Then, using Lemma 6.4.6,

$$z_n = 2x_1z_{n-1} - z_{n-2} = (2x_1 - 1)z_{n-1} + w_{n-1}.$$

$$w_n = z_n - z_{n-1} = 2x_1z_{n-1} - z_{n-2} - z_{n-1} = (2x_1 - 2)z_{n-1} + w_{n-1}.$$

We now show (3). As with our proof of (1), let $\gamma = \frac{\beta}{\sqrt{D}}$. Then, for all $n \in \mathbb{Z}_{>0}$,

$$w_n = z_n - z_{n-1} = \alpha x_n - \gamma\sqrt{D}y_n - \alpha x_{n-1} + \gamma\sqrt{D}y_{n-1} = \alpha(x_n - x_{n-1}) - \gamma\sqrt{D}(y_n - y_{n-1}).$$

Since x_n and y_n are both strictly increasing, $w_n \geq 0$ if and only if $w_n(\gamma(x_n - x_{n-1}) + \alpha\sqrt{D}(y_n - y_{n-1})) \geq 0$. Let $u_n = w_n(\gamma(x_n - x_{n-1}) + \alpha\sqrt{D}(y_n - y_{n-1}))$. Write

$v_n = (\alpha^2 - \gamma^2)\sqrt{D}(x_n - x_{n-1})(y_n - y_{n-1})$. We have

$$\begin{aligned}
u_n &= (\alpha(x_n - x_{n-1}) - \gamma\sqrt{D}(y_n - y_{n-1}))(\gamma(x_n - x_{n-1}) + \alpha\sqrt{D}(y_n - y_{n-1})) \\
&= \alpha\gamma((x_n - x_{n-1})^2 - D(y_n - y_{n-1})^2) + (\alpha^2 - \gamma^2)\sqrt{D}(x_n - x_{n-1})(y_n - y_{n-1}) \\
&= \alpha\gamma(x_n^2 - 2x_nx_{n-1} + x_{n-1}^2 - Dy_n^2 + 2Dy_ny_{n-1} - Dy_{n-1}^2) + v_n \\
&= \alpha\gamma((x_n^2 - Dy_n^2) + (x_{n-1}^2 - Dy_{n-1}^2) + 2Dy_ny_{n-1} - 2x_nx_{n-1}) + v_n \\
&= 2\alpha\gamma(1 + Dy_ny_{n-1} - x_nx_{n-1}) + v_n \\
&= 2\alpha\gamma(1 + D(y_1x_{n-1} + x_1y_{n-1})y_{n-1} - (x_1x_{n-1} + Dy_1y_{n-1})x_{n-1}) + v_n \\
&= 2\alpha\gamma(1 + Dy_1x_{n-1}y_{n-1} + Dx_1y_{n-1}^2 - x_1x_{n-1}^2 - Dy_1y_{n-1}x_{n-1}) + v_n \\
&= 2\alpha\gamma(1 + Dx_1y_{n-1}^2 - x_1x_{n-1}^2) + v_n \\
&= 2\alpha\gamma(1 - x_1(x_{n-1}^2 - Dy_{n-1}^2)) + v_n \\
&= 2\alpha\gamma(1 - 1) + v_n \\
&= (\alpha^2 - \gamma^2)\sqrt{D}(x_n - x_{n-1})(y_n - y_{n-1}).
\end{aligned}$$

Note that $(\alpha^2 - \gamma^2)\sqrt{D}(x_n - x_{n-1})(y_n - y_{n-1}) \geq 0$ if and only if $\alpha \geq \gamma$; that is $\frac{\alpha}{\beta}\sqrt{D} \geq 1$. As we saw in the proof of part (1), $\frac{\alpha}{\beta}\sqrt{D} \geq 1$ implies $(z_n)_{n\geq N}$ is a sequence of non-negative integers, and $\frac{\alpha}{\beta}\sqrt{D} < 1$ implies $(z_n)_{n\geq N}$ is a sequence of non-positive integers, as required.

For (4), we show $(w_n)_{n\geq N}$ is monotone. First note that $(w_n)_{n\geq N}$ and $(z_n)_{n\geq N}$ are both sequences of non-negative integers or sequences of non-positive integers. In addition, $w_n = w_{n-1} + 2x_1z_{n-1}$ for all $n \in \mathbb{Z}_{>0}$. So if $n \in \mathbb{Z}_{\geq N}$, then $|w_n| = |w_{n-1}| + |2x_1z_{n-1}| \geq |w_{n-1}|$. As $(w_n)_{n\geq N}$ is a sequence of non-negative integers or a sequence of non-positive integers, it must be monotone.

We finally consider (5). It suffices to show that $|z_M| > |\gamma|$ and $|w_M| \geq |\gamma|$, then together with the fact that $M \geq N$, and using the fact that $(z_n)_{n\geq N}$ is monotone by (3), and w_n is monotone by (4), we have that $(z_n)_{n\geq M}$ and $(w_n)_{n\geq M}$ are both sequences of non-positive or non-negative integers. We know that $|z_{n+N}| > 2^{n-1}$ and $w_n \geq 2^{n-2}$ using (2), together with the fact that $x_1 > 1$, and so $2x_1 - 1 > 2$, so taking any $M \geq N + \log_2(|\gamma| + 2)$ suffices. As $N = \left\lceil \log_2 \frac{\alpha}{\beta} \right\rceil$, taking $M = \left\lceil \log_2 \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil$, as per the statement of the lemma, satisfies the desired condition. □

Before we apply Lemma 6.4.1 to show that some of these solution languages are EDT0L, we need to add constants to the differences of multiples of solutions.

Lemma 6.4.8. *Let (x_n), $(y_n) \subseteq \mathbb{Z}_{\geq 0}$ be sequences of solutions to Pell's equation $X^2 - DY^2 = 1$. Let α, $\beta \in \mathbb{Z}_{\geq 0}$ and $\gamma \in \mathbb{Z}$. Let (z_n), $(t_n) \subseteq \mathbb{Z}$ be sequences defined by $z_n = \alpha x_n - \beta y_n$ and $t_n = z_n + \gamma$. Then*

1. *The sequence $(s_n)_{n\geq 1} \subseteq \mathbb{Z}$ defined by $s_n = z_n - z_{n-1} + \gamma$ satisfies for all $n \in \mathbb{Z}_{\geq 2}$,*

$$t_n = (2x_1 - 1)z_{n-1} + s_{n-1}, \qquad s_n = (2x_1 - 2)z_{n-1} + s_{n-1};$$

2. *If $\gamma \in \mathbb{Z}$ and $M = \left\lceil \log_2 \frac{(\gamma+2)\alpha}{\beta} \right\rceil$, then $(t_n)_{n\geq M} \subseteq \mathbb{Z}_{\geq 0}$ or $(t_n)_{n\geq M} \subseteq \mathbb{Z}_{\leq 0}$;*
3. *If $(t_n)_{n\geq M}$ is a sequence of non-negative integers then $(s_n)_{n\geq M}$ and $(z_n)_{n\geq M}$ are, and if $(t_n)_{n\geq M}$ is a sequence of non-positive integers then $(s_n)_{n\geq M}$ and $(z_n)_{n\geq M}$ are;*

Proof We start with (1). Let $w_n = z_n - z_{n-1}$, for all $n \in \mathbb{Z}_{>0}$. If $n \in \mathbb{Z}_{>0}$, then using Lemma 6.4.7

$$\begin{aligned} t_n &= z_n + \gamma \\ &= (2x_1 - 1)z_{n-1} + w_{n-1} + \gamma \\ &= (2x_1 - 1)z_{n-1} + s_{n-1}, \\ s_n &= w_n + \gamma \\ &= (2x_1 - 2)z_{n-1} + w_{n-1} + \gamma \\ &= (2x_1 - 2)z_{n-1} + s_{n-1}. \end{aligned}$$

Parts (2) and (3) follow by Lemma 6.4.7 (5). □

To allow us to show space complexity properties, we need bounds of many of the integers we have introduced.

Lemma 6.4.9. *Let S be the set of all non-negative solutions (as ordered pairs) to Pell's equation $X^2 - DY^2 = 1$. Let α, $\beta \in \mathbb{Z}_{>0}$ and $\gamma \in \mathbb{Z}$, and $M = \max\left(2, \left\lceil \log_2 \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil\right)$. Let $z_n = \alpha x_n - \beta y_n$ and $t_n = z_n + \gamma$ for all $n \in \mathbb{Z}_{\geq 0}$. Let $w_n = z_n - z_{n-1}$ and $s_n = w_n + \gamma$ for all $n \in \mathbb{Z}_{>0}$.*

Then there is a function f that is logarithmic in α, β and $|\gamma|$, and a function g that is linear in D, such that $\log(x_M)$, $\log(y_M)$, $\log|z_M|$, $\log|w_M|$, $\log|t_M|$ and $\log|s_M|$ are all bounded by fg.

Proof Lemma 6.4.6, together with the fact that (x_n) is strictly increasing, implies that $x_n \leq (2x_1)^n$ for all $n \geq 1$. Thus $x_M \leq (2x_1)^M = (2x_1)^{\left\lceil \log_2 \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil} = \left\lceil \frac{(|\gamma|+2)\alpha}{\beta} \right\rceil x_1^{\left\lceil \log_2 \frac{(|\gamma|+2)\alpha}{\beta} \right\rceil}$. Using Lemma 6.4.5, we have that

$$\log(x_M) \leq \log \left\lceil \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil + \left\lceil \log_2 \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil \log(x_1)$$

$$\leq \log(\alpha+1) + \log(|\gamma|+4) + \log(\beta+1) + (\log(\alpha+1) + \log(|\gamma|+4) + \log(\beta+1))\sqrt{D}$$

$$\leq (\log(\alpha+1) + \log(|\gamma|+4) + \log(\beta+1))(2+3D).$$

Since $y_M < x_M$, we have that $\log(y_M)$ is also bounded by $(\log(\alpha+1) + \log(|\gamma| + 4) + \log(\beta+1))(2+3D)$.

Let $n \geq 1$. Then, using the fact that αx_n and $\beta(y_n+1)$ are both at least 1, we have

$$\log|z_n| = \log|\alpha x_n - \beta y_n|$$

$$= \log(\alpha) + \log(x_n) + \log(\beta) + \log(y_n+1)$$

$$\leq \log(\alpha) + \log(x_n) + \log(\beta) + \log(y_n) + 1.$$

Using the fact that x_M and y_M are bounded by $(\log(\alpha+1) + \log(|\gamma|+4) + \log(\beta +$

$1))(2+3D)$, we now have that $z_M \leq 2(\log(\alpha+1)+\log(|\gamma|+4)+\log(\beta+1))(2+3D)+\log(\alpha)+\log(\beta)+1$.

We have that $w_M = z_M - z_{M-1}$. Noting that $M-1 \geq 1$, $x_{M-1} \leq x_M$ and $y_{M-1} \leq y_M$, it follows that

$$\begin{aligned}\log|w_M| &= \log|z_M - z_{M-1}| \\ &\leq \log|z_M| + \log|z_{M-1}| \\ &\leq 2\log(\alpha) + 2\log(\beta) + 4(\log(\alpha+1)+\log(|\gamma|+4)+\log(\beta+1))(2+3D)(2+3D)+2\end{aligned}$$

Since $t_M = z_M + \gamma$ and $s_M = w_M + \gamma$, we have that t_M and s_M are bounded by the same expressions as z_M and w_M if $\gamma = 0$. Otherwise,

$$\begin{aligned}\log|t_M| &\leq \log|z_M| + \log|\gamma| \\ &\leq 2(\log(\alpha+1)+\log(|\gamma|+4)+\log(\beta+1))(2+3D)+\log(\alpha)+\log(\beta)+1+\log(|\gamma|), \\ \log|s_M| &\leq \log|w_M| + \log|\gamma| \\ &\leq \log(\alpha)+2\log(\beta)+4(\log(\alpha+1)+\log(|\gamma|+4)+\log(\beta+1))(2+3D)+2+\log|\gamma|.\end{aligned}$$

Taking $f = 4(\log(\alpha+1)+\log(|\gamma|+4)+\log(\beta+1))+2$ and $g = 2+3D$, the result follows. $\square$

We have now completed the set up to show that the solution language to Pell's equation is always EDT0L. More than that, we can show that applying linear functions to the variables will still give this outcome. We need the bounds on the size of our extended alphabet and images of endomorphisms so that we can apply Lemma 6.3.5 later on.

Lemma 6.4.10. *Let S be the set of all non-negative solutions (as tuples) to Pell's equation $X^2 - DY^2 = 1$, and $\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta} \mid (x,\ y) \in S\}$ is EDT0L;*

2. *A #-separated EDT0L system $\mathcal{H}$ for L is constructible in* $\mathsf{NSPACE}(fg)$*, where* f *is logarithmic in* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|)$*, and* g *is linear in* D*;*
3. *The system $\mathcal{H}$ is h_1h_2-bounded, where h_1 is linear in* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|)$*, and h_2 is exponential in D.*

Proof Let $z_n = \alpha x_n + \beta y_n$ and $t_n = z_n + \gamma$ for $n \in \mathbb{Z}_{\geq 0}$, and $s_n = z_n - z_{n-1} + \gamma$ for $n \in \mathbb{Z}_{>0}$. Let $M_\gamma = \max\left(2,\ \left\lceil \log_2 \frac{(|\gamma|+3)\alpha}{\beta} \right\rceil\right)$, $M_\zeta = \max\left(2,\ \left\lceil \log_2 \frac{(|\zeta|+3)\alpha}{\beta} \right\rceil\right)$, and $M = \max(M_\gamma,\ M_\zeta)$. We will first construct an EDT0L system for

$$K = \{a^{t_n} \mid n \in \mathbb{Z}_{\geq M)}\}.$$

If $\alpha \leq 0$ and $\beta \geq 0$, or $\alpha \geq 0$ and $\beta \leq 0$, Lemma 6.4.8 tells us that the sequences $(t_n)_{n\geq M}$, $(z_n)_{n\geq M}$ and $(s_n)_{n\geq M}$ satisfy the conditions of Lemma 6.4.1, and thus K is accepted by an EDT0L system $\mathcal{H} = (\{a,\ a^{-1}\},\ C,\ \perp,\ \theta\varphi^*\psi)$.

If α and β are both non-negative or non-positive, then

$$z_n = \alpha(x_1x_{n-1} + Dy_1y_{n-1}) + \beta(y_1x_{n-1} + x_1y_{n-1}) = (\alpha x_1 + \beta y_1)x_{n-1} + (\alpha Dy_1 + \beta x_1)y_{n-1}.$$

This, together with the recurrence relations in Lemma 6.4.3, gives that $(z_n)_{n\geq M}$, $(x_n)_{n\geq M}$ and $(y_n)_{n\geq M}$ satisfy the conditions of Lemma 6.4.1, and so in we also have in this case that L is accepted by an EDT0L system $\mathcal{H} = (\{a,\ a^{-1}\},\ C,\ \perp,\ \theta\varphi^*\psi)$.

We next consider the space complexity in which $\mathcal{H}$ can be constructed. By Lemma 6.4.5, $\log(x_1)$ and $\log(y_1)$ are both bounded by $2 + 3D$. By Lemma 6.4.9, $\log(x_M)$, $\log(y_M)$, $\log|z_M|$, $\log|t_M|$ and $\log|s_M|$ are all bounded by fg, where f is logarithmic in $|\alpha|$, $|\beta|$, $|\gamma|$ and $|\zeta|$, and g is linear in D. Thus we can use Lemma 6.4.1 (2) to say that $\mathcal{H}$ is constructible in $\mathsf{NSPACE}(fg)$. We also know from Lemma 6.4.1 (4), that $|C|$ and $\max\{|c\phi| \mid c \in C,\ \phi \in \{\psi,\ \theta,\ \varphi\}\}$ are both bounded in terms of an exponential function of fg. Thus $|C|$ and $\max\{c\phi \mid c \in C,\ \phi \in \{\psi,\ \theta,\ \varphi\}\}$ are both bounded by h_1h_2 where h_1 is linear in $|\alpha|$, $|\beta|$, $|\gamma|$ and $|\zeta|$, and h_2 is exponential in D.

Let $\hat{z}_n = \delta x + \epsilon y$ and $\hat{t}_n = \hat{z}_n + \zeta$ for $n \in \mathbb{Z}_{\geq 0}$, and $\hat{s}_n = \hat{z}_n - \hat{z}_{n-1} + \zeta$ for $n \in \mathbb{Z}_{>0}$.

With the same arguments we used to show K is accepted by $\mathcal{H}$, we have that

$$\{b^{\hat{t}_n} \mid n \in \mathbb{Z}_{\geq M}\},$$

is accepted by an EDT0L system $\hat{\mathcal{H}} = (\{b,\ b^{-1}\},\ D,\ \$,\ \sigma\rho^*\tau)$. In addition, $\hat{\mathcal{H}}$ is constructible in $\mathsf{NSPACE}(\hat{f}\hat{g})$, and $|D|$ and $\max\{|c\phi| \mid c \in D,\ \phi \in \{\sigma,\ \rho,\ \tau\}\}$ are both bounded by $\hat{h}_1\hat{h}_2$, where $\hat{f}$ and $\hat{h}_1$ are logarithmic and linear respectively in $|\delta|$, $|\epsilon|$, $|\gamma|$ and $|\zeta|$, and $\hat{g}$ and $\hat{h}_2$ are linear and exponential respectively in D. Redefining f, g, h_1 and h_2 to be the sum of themselves and their hatted versions, gives that both $\mathcal{H}$ and $\hat{\mathcal{H}}$ are constructible in $\mathsf{NSPACE}(fg)$. In addition, $|C|$, $|D|$, $\max\{|c\phi| \mid c \in C,\ \phi \in \{\psi,\ \theta,\ \varphi\}\}$ and $\max\{|c\phi| \mid c \in D,\ \phi \in \{\sigma,\ \rho,\ \tau\}\}$ are all bounded by h_1h_2.

Without loss of generality, we can assume that C and D are disjoint, and $\# \notin C \cup D$. For each endomorphism $\phi \in \{\psi,\ \theta,\ \varphi,\ \sigma,\ \rho,\ \tau\}$, let $\bar{\phi} \in \mathrm{End}\,(C \cup D \cup \{\#\})^*$ be defined to be the extension of ϕ to $C \cup D \cup \{\#\}$ which acts as the identity on wherever it was not previously defined on. It follows that

$$P = \{a^{t_n}\#b^{\hat{t}_n} \mid n \in \mathbb{Z}_{\geq M}\}$$

is accepted by the $\#$-separated EDT0L system $\mathcal{G} = (\{a,\ a^{-1},\ b,\ b^{-1},\ \#\},\ C \cup D \cup \{\#\},\ \perp \#\$,\ \theta\sigma(\varphi\rho)^*\psi\tau)$.

Since $\mathcal{H}$ and $\hat{\mathcal{H}}$ are constructible in $\mathsf{NSPACE}(fg)$, so is $\mathcal{G}$. In addition, $|C \cup D \cup \{\#\}|$ is bounded by $h_1h_2 + 1$, and $\max\{|c\bar{\phi}| \mid c \in C \cup D \cup \{\#\},\ \phi \in \{\psi,\ \theta,\ \varphi,\ \sigma,\ \rho,\ \tau\}\}$ is bounded by h_1h_2. Redefining h_2 to be $h_2 + 1$, gives that $\mathcal{G}$ satisfies all of the conditions of the lemma.

We now consider the language

$$Q = \{a^{t_n}\#b^{\hat{t}_n} \mid n \in \{0,\ \ldots,\ M-1\}\}$$

Note that using Lemma 6.3.3, it now suffices to show that Q is accepted by a $\#$-separated EDT0L system that is constructible in $\mathsf{NSPACE}(fg)$, and whose extended alphabet and images of letters under endomorphisms in the alphabet of the rational control are bounded by $h_1 h_2$.

Let $E = \{\perp_1,\ \perp_2,\ a,\ a^{-1},\ b,\ b^{-1},\ \#\}$. We will use E as our extended alphabet, and $\perp_1 \# \perp_2$ as our start symbol. For each $n \in \{0,\ \ldots,\ M-1\}$, define $\pi_n \in \mathrm{End}(E^*)$ by

$$c\pi_n = \begin{cases} a^{t_n} & c = \perp_1 \\ b^{\hat{t}_n} & c = \perp_2 \\ c & \text{otherwise.} \end{cases}$$

It follows that Q is accepted by the #-separated EDT0L system

$$\mathcal{F} = (\{a,\ a^{-1},\ b,\ b^{-1},\ \#\},\ E,\ \perp_1 \# \perp_2,\ \{\pi_0,\ \ldots,\ \pi_{M-1}\}).$$

Note that $t_0 = \alpha + \gamma$, $\hat{t_0} = \delta + \zeta$. Thus $\log|t_0|$ and $\log|\hat{t_0}|$ are both bounded by a logarithmic function f_1 in terms of $|\alpha|$, $|\beta|$, $|\gamma|$, $|\delta|$, $|\epsilon|$ and $|\zeta|$. By redefining f to be $f + f_1$, we have that $\log|t_0|$ and $\log|\hat{t}_0|$ are bounded by fg. In addition, $\log|t_M|$ and $\log|\hat{t}_M|$ are both bounded by fg. Since (t_n) and $(\hat{t}_n)$ are monotone, and terms are effectively computable by Lemma 6.4.8, each π_n can be constructed in $\mathsf{NSPACE}(fg)$. As E and $\perp_1 \# \perp_2$ are constructible in constant space, it follows that $\mathcal{F}$ is also constructible in $\mathsf{NSPACE}(fg)$.

We have that $|t_0|$ and $|\hat{t_0}|$ are bounded by a linear function f_3 in terms of $|\alpha|$, $|\beta|$, $|\gamma|$, $|\delta|$, $|\epsilon|$ and $|\zeta|$. By redefining h_1 to be $h_1 + f_3$, we have that $|t_0|$, $|\hat{t}_0|$, $|t_M|$ and $|\hat{t}_M|$ are all bounded by $h_1 h_2$, and thus $\max\{|c\pi_i| \mid c \in E,\ i \in \{0,\ \ldots,\ M-1\}\}$ and $|E|$ are both bounded by $h_1 h_2$. □

6.5 Quadratic equations in the ring of integers

Having completed the work on Pell's equation, we now consider more general quadratic equations in the ring of integers, working up to an arbitrary two-variable equation. Our main goal is to show that the solution language to an arbitrary two-variable quadratic equation is EDT0L, with an EDT0L system that is constructible in non-deterministic polynomial space. We start with the general Pell's equation.

Definition 6.5.1. A *general Pell's equation* is an equation $X^2 - DY^2 = N$ in the ring of integers, where X and Y are variables, $N \in \mathbb{Z}\backslash\{0\}$ and $D \in \mathbb{Z}_{>0}$ is not a perfect square.

A non-negative integer solution $(x,\ y)$ to the general Pell's equation $X^2 - DY^2 = N$ is called *primitive* if $\gcd(x,\ y) = 1$.

Before we can generalise Lemma 6.4.10 to a general Pell's equation, we first generalise it to the primitive solutions to a general Pell's equation. The following result allows us to construct primitive solutions to a general Pell's equation from the solutions to the corresponding Pell's equation, and a given primitive solution.

Lemma 6.5.2 ([4], Section 4.1)**.** *Let $(x_0,\ y_0)$ be a primitive solution to the general Pell's equation $X^2 - DY^2 = N$. Let $(u_n,\ v_n)$ be the sequence of solutions (as described in Lemma 6.4.3) to $U^2 - DV^2 = 1$. Define $((x_n,\ y_n))_n \subseteq \mathbb{Z}^2_{\geq 0}$ by*

$$x_n = x_0 u_n + D y_0 v_n, \qquad y_n = y_0 u_n + x_0 v_n.$$

Then $(x_n,\ y_n)$ is a primitive solution to $X^2 - DY^2 = N$ for all $n \in \mathbb{Z}_{\geq 0}$.

We will put an equivalence relation on the set of primitive solutions to a general Pell's equation. This will allow us to consider one class at a time, then use Lemma 6.3.3 to take the union.

Definition 6.5.3. Let $(x,\ y)$ and $(x',\ y')$ be primitive solutions to the general Pell's equation $X^2 - DY^2 = N$. If there exists a primitive solution $(x_0,\ y_0)$ such that $(x,\ y) = (x_m,\ y_m)$ and $(x',\ y') = (x_n,\ y_n)$, for some $m,\ n \in \mathbb{Z}_{\geq 0}$ (using the construction in Lemma 6.5.2), we say $(x,\ y)$ and $(x',\ y')$ are *associated* with each other.

Lemma 6.5.4. *Association of primitive solutions to a general Pell's equation is an equivalence relation.*

Equivalence classes of primitive solutions, which we will call classes, have a notion of a fundamental solution, similar to the fundamental solution to Pell's equation.

Definition 6.5.5. The *class* of a primitive solution $(x,\ y)$ of a general Pell's equation is the equivalence class of all primitive solutions associated with $(x,\ y)$.

The *fundamental solution* of a class of primitive solutions to a general Pell's equation is the minimal element of the class.

We will need the following bounds for the space complexity results.

Lemma 6.5.6 ([4], Theorem 4.1.1 and Theorem 4.12)**.** *Let $(x_0,\ y_0)$ be the fundamental solution of a class of primitive solutions to the general Pell's equation $X^2 - DY^2 = N$. Let $(u_1,\ v_1)$ be the fundamental solution to $X^2 - DY^2 = 1$. If $N > 0$, then*

$$0 \leq x_0 \leq \sqrt{\frac{N(u_1+1)}{2}}, \qquad 0 < y_0 \leq \frac{v_1\sqrt{N}}{\sqrt{2(u_1+1)}}.$$

If $N < 0$, then

$$0 \leq x_0 \leq \sqrt{\frac{|N|(u_1-1)}{2}}, \qquad 0 < y_0 \leq \frac{v_1\sqrt{|N|}}{\sqrt{2(u_1-1)}}.$$

Since the size of fundamental solutions to a general Pell's equation is bounded, there can only be finitely many, and hence only finitely many classes.

Lemma 6.5.7. *There are finitely many classes of primitive solutions to a general Pell's equation.*

We now show that the results stated in Lemma 6.4.10 hold for primitive solutions to a general Pell's equation. We use the characterisation in Lemma 6.5.2 to reduce the problem to Pell's equation, and then apply Lemma 6.4.10.

Lemma 6.5.8. *Let S be the set of primitive solutions to the general Pell's equation $X^2 - DY^2 = N$, and $\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta} \mid (x,\ y) \in S\}$ is EDT0L;*
2. *A #-separated EDT0L system $\mathcal{H}$ for L is constructible in $\mathsf{NSPACE}(fg)$, where f is logarithmic in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$, and g is linear in D;*

3. *The system* $\mathcal{H}$ *is* h_1h_2*-bounded, where* h_1 *is linear in* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|)$, *and* h_2 *is exponential in* D.

Proof Since finite unions of EDT0L languages are EDT0L, and the properties in (2) and (3) are preserved (Lemma 6.3.3), using Lemma 6.5.7 it is sufficient to show that for any class of primitive solutions K, the language

$$M = \{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta} \mid (x,\ y) \in K\}$$

is accepted by an EDT0L system that satisfies the conditions (2) and (3). Let $((u_n,\ v_n))_n$ be the sequence of non-negative integer solutions to $X^2 - DY^2 = 1$. Let $(x_0,\ y_0)$ be the fundamental solution in K. Then we can write elements of K as $(x_n,\ y_n)$, where

$$x_n = x_0u_n + Dy_0v_n, \qquad y_n = y_0u_n + x_0v_n,$$

for some $n \in \mathbb{Z}_{\geq 0}$. For any $n \in \mathbb{Z}_{\geq 0}$,

$$\begin{aligned} \alpha x_n + \beta y_n + \gamma &= \alpha(x_0u_n + Dy_0v_n) + \beta(y_0u_n + x_0v_n) + \gamma \\ &= (\alpha x_0 + \beta y_0)u_n + (\alpha Dy_0 + \beta x_0)v_n + \gamma, \\ \delta x_n + \epsilon y_n + \zeta &= \delta(x_0u_n + Dy_0v_n) + \epsilon(y_0u_n + x_0v_n) + \zeta \\ &= (\delta x_0 + \epsilon y_0)u_n + (\delta Dy_0 + \epsilon x_0)v_n + \zeta \end{aligned}$$

Note that, by Lemma 6.5.6 and Lemma 6.4.5

$$\log(x_0) = \frac{1}{2}\log\left(\frac{N(u_1+1)}{2}\right) \leq \log(N) + \log(u_1+1) \leq 2 + 3D + \log(N),$$

$$\log(y_0) \leq \log(v_1\sqrt{N}) \leq 2 + 3D + \log(N).$$

Thus

$$\begin{aligned}
\log|\delta x_0 + \epsilon y_0| &\leq \log(|\delta|(x_0+1)) + \log(|\epsilon|(y_0+1)) \\
&\leq \log|\delta| + \log|\epsilon| + \log(x_0) + \log(y_0) + 2 \\
&\leq \log|\delta| + \log|\epsilon| + 2 + 6D + 2\log(N), \\
\log|\delta D y_0 + \epsilon x_0| &\leq \log(|\delta| D(y_0+1)) + \log(|\epsilon|(x_0+1)) \\
&\leq \log|\delta| + \log|\epsilon| + \log(D) + \log(x_0) + \log(y_0) + 2.
\end{aligned}$$

Note that the above inequalities also hold with δ replaced by α, and ϵ replaced by β. The result now follows from Lemma 6.4.10. □

We now consider all solutions to a general Pell's equation. We start with a reduction from a non-primitive solution to a primitive solution.

Lemma 6.5.9. *Let $(x,\ y) \in \mathbb{Z}^2_{\geq 0}$, and let $k = \gcd(x,\ y)$. Then $(x,\ y)$ is a solution to the general Pell's equation $X^2 - DY^2 = N$ if and only if $k^2 | N$, and $\left(\frac{x}{k},\ \frac{y}{k}\right)$ is a primitive solution to the general Pell's equation $X^2 - DY^2 = \frac{N}{k^2}$.*

Proof We have $x^2 - Dy^2 = N$ if and only if

$$\frac{N}{k^2} = \frac{x^2 - Dy^2}{k^2} = \left(\frac{x}{k}\right)^2 - D\left(\frac{y}{k}\right)^2.$$

□

It is now possible to generalise Lemma 6.5.8 to all solutions to a general Pell's equation.

Lemma 6.5.10. *Let S be the set of all solutions to the general Pell's equation $X^2 - DY^2 = N$, and $\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x + \beta y + \gamma} \# b^{\delta x + \epsilon y + \zeta} \mid (x,\ y) \in S\}$ is EDT0L;*

2. *A #-separated EDT0L system $\mathcal{H}$ for L is constructible in* $\mathsf{NSPACE}(fg)$*, where* f *is logarithmic in* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$*, and* g *is linear in* D*;*
3. *The system* $\mathcal{H}$ *is* h_1h_2*-bounded, where* h_1 *is linear in* $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$ *and* h_2 *is exponential in* D*.*

Proof First note that the following are equivalent:

1. $(x,\ y) \in S$;
2. $(x,\ -y) \in S$;
3. $(-x,\ y) \in S$;
4. $(-x,\ -y) \in S$.

Since we can use Lemma 6.3.3 to take finite unions of EDT0L languages, and preserve space complexity of EDT0L systems, it therefore suffices to show that

$$M = \{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta} \mid (x,\ y) \in \text{ is a non-negative integer solution to } X^2 - DY^2 = N\}$$

is accepted by an EDT0L system that satisfies (2) and (3). Using Lemma 6.5.9, all non-negative integer solutions to $X^2 - DY^2 = N$ are of the form $(xk,\ yk)$ where $(x,\ y)$ is a primitive solution to $X^2 - DY^2 = \frac{N}{k^2}$, for some k such that $k^2|N$. Moreover, if $(x,\ y)$ is a primitive solution to $X^2 - DY^2 = \frac{N}{k^2}$, then $(xk,\ yk)$ is a solution to $X^2 - DY^2 = N$. We will therefore show two claims:

1. The language $M_k = \{a^{\alpha kx+\beta ky+\gamma}\#b^{\delta kx+\epsilon ky+\zeta} \mid (x,\ y)$ is a primitive solution to $X^2 - DY^2 = \frac{N}{k^2}\}$ is EDT0L for all $k \in \mathbb{Z}_{>1}$ such that $k^2|N$;
2. The union of the languages M_k is EDT0L, and accepted by a #-separated EDT0L system $\mathcal{H}$ that satisfies (2) and (3);
3. The extended alphabet and endomorphisms in $\mathcal{H}$ satisfy the conditions in (3).

Since M equals this union, the result follows.

First note that if $k \in \mathbb{Z}_{\geq 2}$ is such that $k^2|N$, then $k < N$. Thus $\log(\alpha k) = \log(\alpha) + \log(k) \leq \log(\alpha) + \log(N)$. If we use β, δ or ϵ in place of α, this inequality will still hold. Thus the first claim follows from Lemma 6.5.8.

For the second claim, we can apply Lemma 6.3.3 repeatedly, once for each $k \in \mathbb{Z}_{\geq 2}$ such that $k^2 | N$. We need to do this for all such k. This could be done by cycling through all $k \in \{2, \ldots, N-1\}$, checking if $k^2 | N$, and then applying the lemma in those cases. We would need to store the 'current' k to do this, which would use at most $\log(N)$ bits.

□

Before attempting to tackle the general two-variable quadratic equations, we mention the result we can obtain so far for a general Pell's equation. The space complexity in this case is log-linear, which is better than the log-quartic space complexity we have for arbitrary two-variable quadratic equations.

Proposition 6.5.11. *The solution language to the general Pell's equation $X^2 - DY^2 = N$ is EDT0L, accepted by an EDT0L system that is constructible in non-deterministic log-linear space, with $\max(D, |N|)$ as the input size.*

Proof This follows by first taking $\alpha = \beta = \delta = \epsilon = 1$ and $\gamma = \zeta = 0$ in Lemma 6.5.10, and then applying a free monoid homomorphism that maps b to a, using Theorem 3.3.2. □

In order to understand the solutions to a generic two-variable quadratic equation, we must first know the solutions to the equation $X^2 + DY^2 = N$, where $N, D \in \mathbb{Z}$. Whilst we have considered the 'hardest' case of $D < 0$, $-D$ non-square and $N \neq 0$, it remains to consider the remaining cases. We start with the case when $D \geq 0$.

Lemma 6.5.12. *Let S be the set of all solutions to the equation $X^2 + DY^2 = N$, with $N \in \mathbb{Z}$, $D \in \mathbb{Z}_{\geq 0}$, and $\alpha, \beta, \gamma, \delta, \epsilon, \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x + \beta y + \gamma} \# b^{\delta x + \epsilon y + \zeta} \mid (x, y) \in S\}$ is EDT0L;*
2. *A #-separated EDT0L system $\mathcal{H}$ for L is constructible in $\mathsf{NSPACE}(f)$, where f is logarithmic in $\max(|\alpha|, |\beta|, |\gamma|, |\delta|, |\epsilon|, |\zeta|, |N|, D)$;*
3. *The system $\mathcal{H}$ is h-bounded, where h is linear in $\max(|\alpha|, |\beta|, |\gamma|, |\delta|, |\epsilon|, |\zeta|, |N|, D)$.*

Proof If $N < 0$, there is nothing to prove, as the equation has no solutions. So suppose $N \geq 0$. Then all solutions $(x,\ y)$ to this equation satisfy $|x| + |y| \leq N$. Let $(x,\ y)$ be such a solution. Then $\{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta}\}$ is accepted by the EDT0L system $(\{a,\ a^{-1},\ b,\ b^{-1},\ \#\},\ \{a,\ a^{-1},\ b,\ b^{-1},\ \#\} \cup \{\bot_1,\ \bot_2\},\ \bot_1 \# \bot_2,\ \varphi)$, where φ is defined by

$$c\varphi = \begin{cases} a^{\alpha x+\beta y+\gamma} & c = \bot_1 \\ b^{\delta x+\epsilon y+\zeta} & c = \bot_2 \\ c & \text{otherwise.} \end{cases}$$

We have that this EDT0L system satisfies the conditions stated in (2) and (3). Thus we can use Lemma 6.3.3 to obtain the result. □

We now consider the solutions to the equation $X^2 - DY^2 = N$ when D is square.

Lemma 6.5.13. *Let S be the set of all solutions to the equation $X^2 - DY^2 = N$, with $N \in \mathbb{Z}$, $D \in \mathbb{Z}_{\geq 0}$, such that D is a perfect square. Let $\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta} \mid (x,\ y) \in S\}$ is EDT0L;*
2. *A $\#$-separated EDT0L system $\mathcal{H}$ for L is constructible in $\mathsf{NSPACE}(f)$, where f is logarithmic in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$;*
3. *The system $\mathcal{H}$ is h-bounded, where h is linear in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$.*

Proof As D is square, we have that $D = E^2$ for some $E \in \mathbb{Z}_{\geq 0}$. Define a new variable $V = EY$. Substituting this into $X^2 - DY^2 = N$ gives $X^2 - V^2 = N$, which is again equivalent to $(X - V)(X + V) = N$. If $(x,\ v)$ is a solution, then $x + v$ and $x - v$ must both divide N, and thus $|x + v| \leq |N|$ and $|x - v| \leq |N|$. It follows that there are finitely many solutions $(x,\ v)$ to $X^2 - V^2 = N$, all of which satisfy $|x| \leq |N|$ and $|v| \leq N$. Thus there are finitely many solutions $(x,\ y)$ to $X^2 - DY^2 = N$, all of which satisfy $|x| \leq |N|$ and $|y| \leq |V| \leq |N|$.

We can use the same argument we used in Lemma 6.5.12, to show that each of the singleton languages $\{a^{\alpha x+\beta y+\gamma}\#b^{\delta x+\epsilon y+\zeta}\}$ are accepted by EDT0L systems satisfying the conditions in (2) and (3). We can again use Lemma 6.3.3 to union these to form L. □

We finally need to consider the case when $N = 0$.

Lemma 6.5.14. *Let S be the set of all solutions to the equation $X^2 - DY^2 = 0$, with $D \in \mathbb{Z}_{\geq 0}$, and $\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta \in \mathbb{Z}$. Then*

1. *The language $L = \{a^{\alpha x + \beta y + \gamma} \# b^{\delta x + \epsilon y + \zeta} \mid (x,\ y) \in S\}$ is EDT0L;*
2. *A #-separated EDT0L system $\mathcal{H}$ for L is constructible in $\mathsf{NSPACE}(f)$, where f is logarithmic in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ D)$;*
3. *The system $\mathcal{H}$ is h-bounded, where h is linear in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|,\ |N|,\ D)$.*

Proof First note that $(x,\ y)$ is a solution if and only if $x = \sqrt{D}y$. It follows that if D is non-square, then this admits no solutions, and there is nothing to prove. If D is square, then the result follows from Lemma 6.5.13. □

Combining the different cases of the equation $X^2 + DY^2 = N$ allows us to use Lagrange's method to show the following. The proof follows the arguments given in [89], Section 1.

Theorem 6.5.15. *Let*

$$\alpha X^2 + \beta XY + \gamma Y^2 + \delta X + \epsilon Y + \zeta = 0 \tag{6.2}$$

be a two-variable quadratic equation in the ring of integers, with a set S of solutions. Then

1. *The language $L = \{a^x \# b^y \mid (x,\ y) \in S\}$ is EDT0L;*
2. *Taking the input size to be $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|,\ |\zeta|)$, an EDT0L system for L is constructible in $\mathsf{NSPACE}(n^4 \log n)$.*

Proof Let $D = \beta^2 - 4\alpha\gamma$, $E = \beta\delta - 2\alpha\epsilon$ and $F = \delta^2 - 4\alpha\zeta$, and define new variables $U = DY + E$ and $V = 2\alpha X + \beta Y + \delta$. Then

1. $V^2 = 4\alpha^2 X^2 + \beta^2 Y^2 + \delta^2 + 4\alpha\beta XY + 4\alpha\delta X + 2\beta\delta Y$;
2. $DY^2 = \beta^2 Y^2 - 4\alpha\gamma Y^2$;
3. $2EY = 2\beta\delta Y - 4\alpha\epsilon Y$;

4. $F = \delta^2 - 4\alpha\zeta$.

Thus

$$V^2 - DY^2 - 2EY - F = 4\alpha^2 X^2 + 4\alpha\beta XY + 4\alpha\delta X + 4\alpha\gamma Y^2 + 4\alpha\epsilon Y + 4\alpha\zeta.$$

It follows that (6.2) can be rewritten as

$$V^2 = DY^2 + 2EY + F.$$

This is equivalent to

$$DV^2 = (DY + E)^2 + DF - E^2.$$

By substituting U for $DY + E$, and setting $N = E^2 - DF$, we can conclude that (6.2) can be written as

$$U^2 - DV^2 = N. \tag{6.3}$$

Note that

$$Y = \frac{U - E}{D}, \qquad X = \frac{V - \beta Y - \delta}{2\alpha} = \frac{VD - \beta U + \beta E - \delta D}{2\alpha D}.$$

Let T be the set of solutions to (6.3). By Lemma 6.5.10, Lemma 6.5.12, Lemma 6.5.13 or Lemma 6.5.14 (dependent on whether D is positive and non-square, positive and square, or non-positive, and whether or not $N = 0$) we have that

$$M = \{a^{DV - \beta U + \beta E - \delta D} \# b^{U-E} \mid (u,\ v) \in T\}$$

is accepted by a #-separated EDT0L system $\mathcal{H}$, which is constructible in $\mathsf{NSPACE}(fg)$, where f is logarithmic in $\max(|D|,\ |\beta|,\ |\beta E - \delta D|,\ |E|)$, and g is linear in $|D|$. Let C be the extended alphabet of $\mathcal{H}$, and let B be an alphabet the rational control of $\mathcal{H}$ is regular. Using Lemma 6.5.10, 6.5.10, Lemma 6.5.12, Lemma 6.5.13 or Lemma 6.5.14, we also have that $|C|$ and $\max\{|c\phi| \mid c \in C,\ \phi \in B\}$ are bounded by $h_1 h_2$, where h_1 is linear in $\max(|D|,\ |\beta|,\ |\beta E - \delta D|,\ |E|)$, and h_2 is exponential in $|D|$.

We have that $D = \beta^2 - 4\alpha\gamma$ and $N = E^2 - DF = (\beta\delta - 2\alpha\epsilon)^2 - (\beta^2 - 4\alpha\gamma)(\delta^2 - 4\gamma\zeta)$. It follows that f is logarithmic in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|)$, and g is quadratic in $\max(|\alpha|,\ |\beta|,\ |\gamma|)$.

In addition, h_1 is quartic in $\max(|\alpha|,\ |\beta|,\ |\gamma|,\ |\delta|,\ |\epsilon|)$, and h_2 is $\mathcal{O}(2^{n^2})$ in $\max(|\alpha|,\ |\beta|,\ |\gamma|)$.

Note that $DY = U - E$ and $2\alpha DX = DV - \beta U + \beta E - \delta D$. Thus we have that

$$M = \{a^{2\alpha Dx} \# b^{Dy} \mid (x,\ y) \in S\}.$$

By Lemma 6.3.5, it follows that L is EDT0L, and accepted by an EDT0L system that is constructible in $\mathsf{NSPACE}(n^4 \log n)$. $\square$

Using Theorem 3.3.2 to apply the free monoid homomorphism that maps b to a to a language described in Theorem 6.5.15 gives the following:

Corollary 6.5.16. *The solution language to a two-variable quadratic equation in integers is EDT0L, accepted by an EDT0L system that is constructible in* $\mathsf{NSPACE}(n^4 \log n)$*, with the input size taken to be the maximal absolute value of a coefficient.*

6.6 From Heisenberg equations to integer equations

This section aims to prove that the solution language to an equation in one variable in the Heisenberg group is EDT0L. We do this by showing that a single equation $\mathcal{E}$ in the Heisenberg group is 'equivalent' to a system $S_{\mathcal{E}}$ of quadratic equations in the ring of integers. The idea of the proof is to replace each variable in $\mathcal{E}$ with a word representing a potential solution, and then convert the resulting word into Mal'cev normal form. The equations in $S_{\mathcal{E}}$ occur by equating the exponent of the generators to 0.

We start with an example of an equation in the Heisenberg group.

Example 6.6.1. We will transform the equation $XYX = 1$ in the Heisenberg group into a system over the integers. Using the Mal'cev normal form we can write $X = a^{X_1}b^{X_2}c^{X_3}$ and $Y = a^{Y_1}b^{Y_2}c^{Y_3}$ for variables $X_1,\ X_2,\ X_3,\ Y_1,\ Y_2,\ Y_3$ over the integers. Replacing X and Y in $XYX = 1$ in these expressions gives

$$a^{X_1}b^{X_2}c^{X_3}a^{Y_1}b^{Y_2}c^{Y_3}a^{X_1}b^{X_2}c^{X_3} = 1. \tag{6.4}$$

After manipulating this into Mal'cev normal form, we obtain

$$a^{2X_1+Y_1}b^{2X_2+Y_2}c^{2X_3+Y_3+X_1Y_2+X_1X_2+Y_1X_2} = 1. \tag{6.5}$$

As this normal form word is trivial if and only if the exponents of a, b and c are all equal to 0, we obtain the following system over $\mathbb{Z}$:

$$\begin{aligned} &2X_1 + Y_1 = 0 \\ &2X_2 + Y_2 = 0 \\ &2X_3 + Y_3 + X_1Y_2 + X_1X_2 + Y_1X_2 = 0. \end{aligned} \tag{6.6}$$

Note that the variables corresponding to the exponent of c in X and Y, namely X_3 and Y_3, only appear in linear terms in the above system.

In this specific example it is not hard to enumerate the solutions in a somewhat reasonable manner. We can start by replacing occurrences of Y_1 and Y_2 in the third equation of (6.6) with $-2X_1$ and $-2X_2$, respectively, to give that (6.6) is equivalent to

$$\begin{aligned} &Y_1 = -2X_1 \\ &Y_2 = -2X_2 \\ &2X_3 + Y_3 - 2X_1X_2 + X_1X_2 - 2X_1X_2 = 0. \end{aligned}$$

This simplifies to

$$Y_1 = -2X_1$$

$$Y_2 = -2X_2$$

$$2X_3 + Y_3 = 3X_1X_2.$$

We can now enumerate all values of $(X_1,\ X_2,\ X_3)$ (across $\mathbb{Z}$), and each such choice will fix the values of Y_1, Y_2 and Y_3, for which there will always exist a solution. Using this method, we have that the solution set to (6.6) is equal to

$$\{(x_1,\ x_2,\ x_3,\ -2x_1,\ -2x_2,\ 3x_1x_2 - 2x_3) \mid x_1,\ x_2,\ x_3 \in \mathbb{Z}\}.$$

Translating this back into the language of the Heisenberg group gives that the solution set to $XYX = 1$ is

$$\{(a^{x_1}b^{x_2}c^{x_3},\ a^{-2x_1}b^{-2x_2}c^{3x_1x_2-2x_3}) \mid x_1,\ x_2,\ x_3 \in \mathbb{Z}\}.$$

The following definition allows us to transform an equation in a single variable in the Heisenberg group into a system of equations in the ring of integers. This is done by representing the variables as expressions in Mal'cev normal form, plugging these expressions back into the equation, and then converting the resulting word into Mal'cev normal form. After doing this, the exponents of the generators can the be equated to 0, which yields a system of equations in the ring of integers.

Definition 6.6.2. If $w = 1$ is an equation in a class 2 nilpotent group, consider the system of equations over the integers defined by taking the variable X, and viewing it in Mal'cev normal form by introducing new variables: $X = a^{X_1}b^{Y_1}c^{Z_1}$, where the X_1, X_2 and X_3 take values in $\mathbb{Z}$. The resulting system of equations over $\mathbb{Z}$ obtained by setting the expressions in the exponents equal to zero is called the $\mathbb{Z}$-*system* of $w = 1$.

Example 6.6.3. The $\mathbb{Z}$-system of the equation (6.4) from Example 6.6.1 is

$$2X_1 + Y_1 = 0$$

$$2X_2 + Y_2 = 0$$

$$2X_3 + Y_3 + X_1Y_2 + X_1X_2 + Y_1X_2 = 0.$$

We now explicitly calculate the $\mathbb{Z}$-system of an arbitrary equation in one variable in the Heisenberg group.

Lemma 6.6.4. *Let*

$$X^{\epsilon_1}a^{i_1}b^{j_1}c^{k_1}\cdots X^{\epsilon_n}a^{i_n}b^{j_n}c^{k_n} = 1 \tag{6.7}$$

be a single equation in one variable in the Heisenberg group, where $\epsilon_1, \ \ldots, \ \epsilon_n \in \{-1, \ 1\}$, *and* $i_1, \ \ldots i_n, \ j_1, \ \ldots, \ j_n, \ k_1, \ \ldots, \ k_n \in \mathbb{Z}$. *Define*

$$\delta_r = \begin{cases} 0 & \epsilon_r = 1 \\ 1 & \epsilon_r = -1. \end{cases}$$

Writing $X = a^{X_1}b^{X_2}c^{X_3}$ *with* X_1, X_2 *and* X_3 *over* $\mathbb{Z}$ *gives that the* $\mathbb{Z}$*-system of* (6.7) *is*

$$\sum_{r=1}^{n}(\epsilon_r X_1 + i_r) = 0$$

$$\sum_{r=1}^{n}(\epsilon_r X_2 + j_r) = 0$$

$$\sum_{r=1}^{n}(\epsilon_r X_3 + k_r + \delta_r X_1 X_2) + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r\epsilon_s X_1X_2 + \epsilon_r X_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r\epsilon_s X_2 + i_r j_s) = 0.$$

Proof We proceed as in Example 6.6.1. Replacing each occurrence of X in (6.7) with $a^{X_1}b^{X_2}c^{X_3}$ gives

$$(a^{X_1}b^{X_2}c^{X_3})^{\epsilon_1}a^{i_1}b^{j_1}c^{k_1}\cdots(a^{X_1}b^{X_2}c^{X_3})^{\epsilon_n}a^{i_n}b^{j_n}c^{k_n} = 1. \tag{6.8}$$

Since c is central, we can push all occurrences of c and c^{-1} to the right, and then freely reduce, thus showing that 6.7 is equivalent to

$$(a^{X_1}b^{X_2})^{\epsilon_1}a^{i_1}b^{j_1}\cdots(a^{X_1}b^{X_2})^{\epsilon_n}a^{i_n}b^{j_n}c^{\sum_{r=1}^{n}(\epsilon_r X_3+k_r)}=1. \tag{6.9}$$

Note that for all $x_1,\ x_2\in\mathbb{Z}$, $(a^{x_1}b^{x_2})^{-1}=b^{-x_2}a^{-x_1}=a^{-x_1}b^{-x_2}c^{x_1x_2}$. Using this, together with the fact that c is central, gives that (6.9) is equivalent to

$$a^{\epsilon_1X_1}b^{\epsilon_1X_2}a^{i_1}b^{j_1}\cdots a^{\epsilon_nX_1}b^{\epsilon_nX_2}a^{i_n}b^{j_n}c^{\sum_{r=1}^{n}(\epsilon_rX_3+k_r+\delta_rX_1X_2)}=1. \tag{6.10}$$

We now push all as in (6.10) to the left. The as at the beginning do not need to move. The as with exponent i_1 will need to move past $b^{\epsilon_1X_2}$, thus increasing the exponent of c by $i_1\epsilon_1X_2$. The as with exponent ϵ_2X_1 will need to move past b^{j_1} and $b^{\epsilon_1X_2}$, thus increasing the exponent of c by $j_1\epsilon_2X_1+\epsilon_1\epsilon_2X_1X_2$. This continues up to the as with exponent i_n, which will need to move past all bs, thus increasing the exponent of c by $i_n(\sum_{r=1}^{n}\epsilon_rX_2)+i_n\sum_{r=1}^{n-1}j_r$. Overall, we have that (6.10) is equivalent to

$$a^{\sum_{r=1}^{n}(\epsilon_rX_1+i_r)} \tag{6.11}$$

$$b^{\sum_{r=1}^{n}(\epsilon_rX_2+j_r)}$$

$$c^{\sum_{r=1}^{n}(\epsilon_rX_3+k_r+\delta_rX_1X_2)+\sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r\epsilon_sX_1X_2+\epsilon_rX_1j_s)+\sum_{r=1}^{n}\sum_{s=1}^{r}(i_r\epsilon_sX_2+i_rj_s)}=1.$$

Equating each of the exponents to 0 (as we are now in Mal'cev normal form) gives that the $\mathbb{Z}$-system of (6.7) is

$$\sum_{r=1}^{n}(\epsilon_r X_1 + i_r) = 0$$

$$\sum_{r=1}^{n}(\epsilon_r X_2 + j_r) = 0$$

$$\sum_{r=1}^{n}(\epsilon_r X_3 + k_r + \delta_r X_1 X_2) + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r\epsilon_s X_1 X_2 + \epsilon_r X_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r\epsilon_s X_2 + i_r j_s) = 0.$$

□

We have now collected the results we need to prove the main theorem of this section.

Theorem 6.6.5. *Let L be the solution language to a single equation with one variable in the Heisenberg group, with respect to the Mal'cev generating set and normal form. Then*

1. *The language L is EDT0L;*
2. *An EDT0L system for L is constructible in* $\mathsf{NSPACE}(n^8(\log n)^2)$.

Proof Let

$$X^{\epsilon_1}a^{i_1}b^{j_1}c^{k_1}\cdots X^{\epsilon_n}a^{i_n}b^{j_n}c^{k_n} = 1 \tag{6.12}$$

be an equation in the Heisenberg group in a single variable. By Lemma 6.6.4, we have that the $\mathbb{Z}$-system of (6.12) is

$$\sum_{r=1}^{n}(\epsilon_r X_1 + i_r) = 0 \tag{6.13}$$

$$\sum_{r=1}^{n}(\epsilon_r X_2 + j_r) = 0$$

$$\sum_{r=1}^{n}(\epsilon_r X_3 + k_r + \delta_r X_1 X_2) + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r\epsilon_s X_1 X_2 + \epsilon_r X_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r\epsilon_s X_2 + i_r j_s) = 0.$$

We consider two cases: when $\sum_{r=1}^{n}\epsilon_r = 0$ and when $\sum_{r=1}^{n}\epsilon_r \neq 0$.

Case 1: $\sum_{r=1}^{n}\epsilon_r = 0$.

Applying our case assumption to (6.13) gives that (6.13) is equivalent to

$$\sum_{r=1}^{n} i_r = 0 \tag{6.14}$$

$$\sum_{r=1}^{n} j_r = 0$$

$$\sum_{r=1}^{n}(k_r + \delta_r X_1 X_2) + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r \epsilon_s X_1 X_2 + \epsilon_r X_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r \epsilon_s X_2 + i_r j_s) = 0.$$

The first two of the above identities only involve constants. If one of these is not satisfied, then (6.12) has no solutions. In such a case, L is empty, and there is nothing to prove. So we suppose that these are satisfied. It follows that they are redundant, and the above system is equivalent to the third equation in it (with the addition that X_3 can be anything, regardless of X_1 and X_2). Note that this is a quadratic equation in integers, with variables X_1 and X_2. So by Theorem 6.5.15

$$K = \{a^{x_1} \# b^{x_2} \mid (x_1,\ x_2) \text{ is part of a solution (6.14) for } (X_1,\ X_2)\}$$

is EDT0L, and accepted by an EDT0L system that is constructible in $\mathsf{NSPACE}(n \mapsto n^4 \log n)$ in terms of the coefficients of the equation. These are

$$\sum_{r=1}^{n} k_r + \sum_{r=1}^{n}\sum_{s=1}^{r} i_r j_s,\ \sum_{r=1}^{n} \delta_r + \sum_{r=1}^{n}\sum_{s=1}^{r} \epsilon_r \epsilon_s,\ \sum_{r=1}^{n}\sum_{s=1}^{r} \epsilon_r j_s,\ \sum_{r=1}^{n}\sum_{s=1}^{r} i_r \epsilon_s.$$

Note that $|\epsilon_r| = 1$ and $|\delta_r| \leq 1$ for all r. In addition, as exponents of constants in (6.12), each sum $\sum_{r=1}^{n} i_r$, $\sum_{r=1}^{n} j_r$ and $\sum_{r=1}^{n} k_r$ is linear in our input. It follows that the above expression is quadratic in our input, and so an EDT0L system for K is constructible in $\mathsf{NSPACE}(n \mapsto n^8(\log n)^2)$. Applying the monoid homomorphism that maps $\#$ to ε, followed by concatenating the above language with the EDT0L language $\{c\}^*$, which is constructible in constant space, allows us to apply Theorem 3.3.2 to show

$$\{a^{x_1} b^{x_2} c^{x_3} \mid (x_1,\ x_2,\ x_3) \text{ is a solution (6.14)}\}$$

is EDT0L, accepted by an EDT0L system that is constructible in $\mathsf{NSPACE}(n \mapsto n^8(\log n)^2)$. Since this language is L, the result follows.

Case 2: $\sum_{r=1}^{n} \epsilon_r \neq 0$.

Let $\alpha = \sum_{r=1}^{n} \epsilon_r$, $\beta = \sum_{r=1}^{n} i_r$, $\gamma = \sum_{r=1}^{n} j_r$ and $\zeta = \sum_{r=1}^{n} k_r$. Then we can rewrite (6.13) as

$$\alpha X_1 + \beta = 0 \tag{6.15}$$

$$\alpha X_2 + \gamma = 0$$

$$\alpha X_3 + \zeta + \sum_{r=1}^{n} \delta_r X_1 X_2 + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r \epsilon_s X_1 X_2 + \epsilon_r X_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r \epsilon_s X_2 + i_r j_s) = 0.$$

If either of the first two equations have no solution, then neither does (6.12), and so L is empty, and there is nothing to prove. We will therefore suppose that both of these equations admit a solution. Since these are both single linear equations with one variable, they can both admit a single solution. Let x_1 be the solution for X_1, and x_2 be the solution for X_2. Plugging these into the third equation gives

$$\alpha X_3 + \zeta + \sum_{r=1}^{n} \delta_r x_1 x_2 + \sum_{r=1}^{n}\sum_{s=1}^{r}(\epsilon_r \epsilon_s x_1 x_2 + \epsilon_r x_1 j_s) + \sum_{r=1}^{n}\sum_{s=1}^{r}(i_r \epsilon_s x_2 + i_r j_s) = 0. \tag{6.16}$$

Note that this is a linear equation in integers with single variable X_3. Hence by [47], Corollary 3.13 and Proposition 3.16, the language

$$M = \{c^{x_3} \mid x_3 \text{ is a solution to (6.16)}\}$$

is EDT0L, and accepted by an EDT0L system that is constructible in non-deterministic quadratic space in terms of an input of length

$$|\alpha| + |\zeta| + \sum_{r=1}^{n} |\delta_r x_1 x_2| + \sum_{r=1}^{n}\sum_{s=1}^{r}(|\epsilon_r \epsilon_s x_1 x_2| + |\epsilon_r x_1 j_s|) + \sum_{r=1}^{n}\sum_{s=1}^{r}(|i_r \epsilon_s x_2| + |i_r j_s|).$$

As the sums of the lengths of constants in our original equation, $|\alpha|$, $|\beta|$, $|\gamma|$ and $|\zeta|$ are all linear in our input. As the number of constants in our equation, n is also linear in our input. We have that $|x_1| = \left|\frac{\beta}{\alpha}\right| \leq |\beta|$ and $|x_2| = \left|\frac{\gamma}{\alpha}\right| \leq |\alpha|$ are both linear in our input. Since $|\epsilon_r| = 1$ and $|\delta_r| \leq 1$ for all r, and the above expression is quartic in our input, it follows that M is constructible in $\mathsf{NSPACE}(n^4)$. Applying Theorem 3.3.2 to concatenate M with the singleton language $\{a^{x_1}b^{x_2}\}$, which is constructible in linear space, gives that

$$\{a^{x_1}b^{x_2}c^{x_3} \mid (x_1,\ x_2,\ x_3) \text{ is a solution to (6.14)}\}$$

is EDT0L, and accepted by an EDT0L system that is constructible in $\mathsf{NSPACE}(n^4)$. Since this language is L, the result follows. □

Chapter 7

Equations in class 2 nilpotent groups

This chapter is based on work in [64].

In this chapter we consider equations in class 2 nilpotent groups with virtually cyclic commutator subgroups, and use this to show that the satisfiability of single equations in virtually the Heisenberg group is decidable. This follows the method of Duchin, Liang and Shapiro [34], although we explicitly construct the quadratic equations in the ring of integers that equations in these class 2 nilpotent groups are 'equivalent'.

In Section 7.1 we generalise the Mal'cev generating set and normal form from Section 6.2.2 to all class 2 nilpotent groups with a virtually cyclic commutator subgroup. We then construct a $\mathbb{Z}$-system for an equation in such a group in Section 7.2, analogous to what was done in Section 6.6. This section follows the work of Duchin, Liang and Shapiro [34]. We then use this to show that the satisfiability of single equations in virtually the Heisenberg group is decidable in Section 7.3.

The explicit calculations used in Section 7.2 could be used in future work to attempt to generalise the work in Chapter 6 to show equations have EDT0L solutions in more class 2 nilpotent groups, as an exact definition of the sets will probably be needed to say anything about solution languages. In addition, the fact that the satisfiability of equations is decidable in virtually the Heisenberg group could be extended in future work to cover any group that is virtually a class 2 nilpotent groups with a virtually cyclic commutator subgroup - thus generalising the work of Duchin, Liang and Shapiro [34] to finite extensions.

7.1 Mal'cev generators

This section covers the normal form we will be using. This is a generalisation of the Mal'cev normal form introduced in Section 6.2.2. We include the proof of uniqueness and existence for completeness.

Lemma 7.1.1. *Let G be a class 2 nilpotent group. Then G has a generating set*

$$\{a_1, \ldots, a_n, b_1, \ldots, b_r, c_1, \ldots, c_s, d_1, \ldots, d_t\},$$

where $n, r, s, t \in \mathbb{N}\backslash\{0\}$, such that the d_is have finite order, the c_is and d_is are central, for each b_i, there exists $l_i \in \mathbb{N}\backslash\{0\}$, such that $b_i^{l_i} \in [G, G]$, and $[G, G] = \langle c_1, \cdots, c_s, d_1, \ldots, d_t\rangle$.

Proof Using the fundamental theorem for finitely generated abelian groups, the short exact sequence $\{1\} \to [G, G] \to G \to G/[G, G] \to \{1\}$ becomes

$$\{1\} \longrightarrow \mathbb{Z}^s \oplus (\mathbb{Z}_{k_1} \oplus \cdots \oplus \mathbb{Z}_{k_t}) \longrightarrow G \longrightarrow \mathbb{Z}^n \oplus (\mathbb{Z}_{l_1} \oplus \cdots \oplus \mathbb{Z}_{l_r}) \longrightarrow \{1\},$$

where $n, r, s, t \in \mathbb{N}\backslash\{0\}$. Let $a_1, \ldots, a_n$ be lifts in G of standard generators for $\mathbb{Z}$, $b_1, \ldots, b_r$ be lifts of generators of $\mathbb{Z}_{l_1}, \ldots, \mathbb{Z}_{l_r}$, respectively. Let $c_1, \ldots, c_s$ be a generating set for $\mathbb{Z}^s$, and $d_1, \ldots, d_t$ be generators of $\mathbb{Z}_{k_1}, \ldots, \mathbb{Z}_{k_t}$, respectively. Then using our short exact sequence, it follows that $\{a_1, \ldots, a_n, b_1, \ldots, b_r, c_1, \ldots, c_s, d_1, \ldots, d_t\}$ generates G. We have that $d_i^{k_i} = 1$, for all i. As $\{c_1, \ldots, c_s, d_1, \ldots, d_t\}$ generates $[G, G]$, the result follows. □

Definition 7.1.2. A generating set defined as in Lemma 7.1.1 is called a *Mal'cev generating set.*

Lemma 7.1.3. *Let G be a class 2 nilpotent group and*

$$\{a_1,\ \ldots,\ a_n,\ b_1,\ \ldots,\ b_r,\ c_1,\ \ldots,\ c_s,\ d_1,\ \ldots,\ d_t\},$$

be a Mal'cev generating set for G, where again, l_i is minimal (and exists) such that $b_i^{l_i} \in [G,\ G]$, and the order of d_i is k_i. Then every element of G can be expressed uniquely as an element of the set

$$\{a_1^{i_1} \cdots a_n^{i_n} b_1^{j_1} \cdots b_r^{j_r} c_1^{p_1} \cdots c_s^{p_s} d_1^{q_1} \cdots d_t^{q_t} \mid i_1,\ \ldots,\ i_n,\ p_1, \ldots,\ p_s \in \mathbb{Z}, \tag{7.1}$$

$$j_x \in \{0,\ \ldots, l_x - 1\},\ q_x \in \{0,\ \ldots,\ k_x - 1\} \text{ for each } x\}.$$

Proof Existence: Let $g \in G$, and w be a word over our generating set that represents g. As all c_is and d_is are central, we can push them to the back of w, and into the desired order. As $[a_i,\ a_j]$, $[b_i,\ b_j]$, and $[a_i,\ b_j]$ can be written as expressions using the c_is and d_is, for all i and j, we have that reordering the a_is and b_is to the desired form simply creates expressions using the c_is and d_is, which can then be pushed to the back of w, and into the stated order. Let $i \in \{1,\ \ldots,\ r\}$. By definition, $[G,\ G]b_i^{l_i} = [G,\ G]$, so we can reduce b_i modulo l_i by creating an expression over the c_is and d_is, which, again, can be pushed to the back and into the desired form. Since the d_is have finite order, we can reduce their exponents modulo these orders.

Uniqueness: It suffices to show that any expression of this form is non-trivial whenever at least one of the exponents is non-zero. So let $i_1,\ \ldots,\ i_{n+r+s+t} \in \mathbb{Z}$ be such that

$$a_1^{i_1} \cdots a_n^{i_n} b_1^{i_{n+1}} \cdots b_r^{i_{n+r}} c_1^{i_{n+r+1}} \cdots c_s^{i_{n+r+s}} d_1^{i_{n+r+s+1}} \cdots d_t^{i_{n+r+s+t}} = 1.$$

If $\phi \colon G \to G/[G,\ G]$ is the quotient map, then applying this to the above expression gives

$$(\phi(a_1))^{i_1} \cdots (\phi(a_n))^{i_n} (\phi(b_1))^{i_{n+1}} \cdots (\phi(b_r))^{i_{n+r}} = 1.$$

As the a_is and b_is are lifts to G of generators of the corresponding cyclic groups, we have that the above expression of $\phi(a_i)$s and $\phi(b_i)$s is in the (standard) normal form for $\mathbb{Z}^n \oplus (\mathbb{Z}_{l_1} \oplus \cdots \oplus \mathbb{Z}_{l_r})$, and so $i_1 = \cdots = i_{n+r} = 0$. It follows that

$$c_1^{i_{n+r+1}} \cdots c_s^{i_{n+r+s}} d_1^{i_{n+r+s+1}} \cdots d_t^{i_{n+r+s+t}} = 1.$$

But this expression is in the normal form for $\mathbb{Z}^s \oplus (\mathbb{Z}_{k_1} \oplus \cdots \oplus \mathbb{Z}_{k_t})$, and so $i_{n+r+1} = \cdots = i_{n+r+s+t} = 0$. □

Definition 7.1.4. The normal form defined in Lemma 7.1.3 is called the *Mal'cev normal form.*

Notation 7.1.5. We define a number of invariants for a group G with the Mal'cev generating set

$$\{a_1, \ldots, a_n, b_1, \ldots, b_r, c_1, \ldots, c_s, d_1, \ldots, d_t\},$$

where again, l_i is minimal (and exists) such that $b_i^{l_i} \in [G, G]$ and the order of d_i is k_i.

1. From Lemma 7.1.1, we have that $[a_i, a_j], [b_i, b_j], [a_i, b_j] \in c_1^* \cdots c_s^* d_1^* \cdots d_t^*$, for all i, j, with $i < j$ in the first two expressions. For all such i and j, $k \in \{1, \ldots, s\}$, and $l \in \{1, \ldots, t\}$, we can therefore define π_{ijk}, τ_{ijk}, υ_{ijk}, α_{ijl}, β_{ijl} and γ_{ijl} to be the unique integers satisfying the following normal form expressions in $[G, G]$:

$$[a_j, a_i] = c_1^{\pi_{ij1}} \cdots c_s^{\pi_{ijs}} d_1^{\alpha_{ij1}} \cdots d_t^{\alpha_{ijt}}, \quad (i < j)$$

$$[b_j, b_i] = c_1^{\tau_{ij1}} \cdots c_s^{\tau_{ijs}} d_1^{\beta_{ij1}} \cdots d_t^{\beta_{ijt}}, \quad (i < j)$$

$$[a_i, b_j] = c_1^{\upsilon_{ij1}} \cdots c_s^{\upsilon_{ijs}} d_1^{\gamma_{ij1}} \cdots d_t^{\gamma_{ijt}}.$$

2. Since $b_i^{l_i} \in [G, G]$, we can define ξ_{ik} and η_{il} for all $i \in \{1, \ldots, r\}$, $k \in \{1, \ldots, s\}$ and $l \in \{1, \ldots, t\}$, to be the unique integers such that

$$b_i^{l_i} = c_1^{\xi_{i1}} \cdots c_s^{\xi_{is}} d_1^{\eta_{i1}} \cdots d_t^{\eta_{it}}.$$

7.2 Transforming equations in nilpotent groups into equations over $\mathbb{Z}$

This section aims to prove that a single equation $\mathcal{E}$ in a class 2 nilpotent group is equivalent to a system $S_{\mathcal{E}}$ over the ring $\mathbb{Z}$ of quadratic equations and congruences that may also contain 'floor terms'. The idea of the proof is to replace each variable in $\mathcal{E}$ with a word representing a potential solution, and then convert the resulting word into Mal'cev normal form. The linear equations in $S_{\mathcal{E}}$ occur by equating the exponent of each generator a_i to 0, and the linear congruences, quadratic equations and quadratic congruences occur when the same is done for the b_is, c_is and d_is, respectively. Recall that Example 6.6.1 gave an example of an equation in the Heisenberg group, and a method of solving it using Mal'cev normal form.

Example 7.2.1. Let G be the class 2 nilpotent group with the presentation

$$\langle a_1,\ a_2,\ b,\ c,\ d \mid c = [a_1,\ a_2],\ d = [a_1,\ b] = [a_2,\ b],\ b^2 = c,\ d^2 = 1,$$
$$[a_1,\ c] = [a_1,\ d] = [a_2,\ c] = [a_2,\ d] = [b,\ c] = [b,\ d] = 1\rangle.$$

Consider the equation

$$Xba_1cXa_2c^{-3}a_1X = 1 \tag{7.2}$$

We first convert the constants into Mal'cev normal form, and push the commutators to the right to obtain

$$Xa_1bXa_1a_2Xc^{-3}d^{-1} = 1.$$

Using the fact that $d^2 = 1$ gives that this is equivalent to

$$Xa_1bXa_1a_2Xc^{-3}d = 1. \tag{7.3}$$

As in Example 6.6.1, we set $X = a_1^{X_1}a_2^{X_2}b^{X_3}c^{X_4}d^{X_5}$ using our Mal'cev normal form, for new variables $X_1, \dots, X_5$ over $\mathbb{Z}$. Plugging this into (7.3) gives

$$a_1^{X_1}a_2^{X_2}b^{X_3}c^{X_4}d^{X_5}a_1ba_1^{X_1}a_2^{X_2}b^{X_3}c^{X_4}d^{X_5}a_1a_2a_1^{X_1}a_2^{X_2}b^{X_3}c^{X_4}d^{X_5}c^{-3}d = 1. \tag{7.4}$$

We can the transform this into Mal'cev normal form, to (first) obtain

$$a_1^{3X_1+2}a_2^{3X_2+1}b^{3X_3+1}c^{3X_4+X_1(1+X_2+X_2)+X_1X_2-3} \tag{7.5}$$

$$d^{3X_5+X_2(X_3+1+X_3)+X_1(X_3+1+X_3)+(X_3+1+X_3)+(X_3+1+X_3)+X_2(1+X_3)+X_1(1+X_3)+X_3+2+1} = 1.$$

Simplifying this gives

$$a_1^{3X_1+2}a_2^{3X_2+1}b^{3X_3+1}c^{3X_1X_2+X_1+3X_4-3}d^{3X_1X_3+3X_2X_3+4X_1+4X_2+5X_3+3X_5+5} = 1. \tag{7.6}$$

Using the relations $b^2 = c$ and $d^2 = 1$, we can conclude

$$a_1^{3X_1+2}a_2^{3X_2+1}b^{(X_3+1) \bmod 2}c^{3X_1X_2+X_1+3X_4-3+\lfloor \frac{X_3+1}{2} \rfloor+X_3}d^{(X_1X_3+X_2X_3+X_3+X_5+1) \mod 2} = 1. \tag{7.7}$$

This results in the following system of equations over (the ring) $\mathbb{Z}$

$$3X_1 + 2 = 0 \tag{7.8}$$

$$3X_2 + 1 = 0$$

$$X_3 + 1 \equiv 0 \mod 2$$

$$3X_1X_2 + X_1 + X_3 + \left\lfloor \frac{X_3+1}{2} \right\rfloor + 3X_4 - 3 = 0$$

$$X_1X_3 + X_2X_3 + X_3 + X_5 + 1 \equiv 0 \mod 2.$$

As $3X_1 + 2 = 0$ admits no integer solutions, we can conclude that (7.2) does not admit a solution.

Definition 7.2.2. Let w be a word over an alphabet of the form Σ, where every letter in Σ has an assigned inverse letter. The *exponent sum* of w with respect to a letter $a \in \Sigma$ is defined by

$$\mathrm{expsum}_a(w) = \#_a(w) - \#_{a^{-1}}(w).$$

Notation 7.2.3. Let G be a class 2 nilpotent group, $X_1, \ldots, X_N$ be variables, where $N \in \mathbb{Z}_>$, and

$$X_{p_1}^{\epsilon_1} \cdots X_{p_K}^{\epsilon_K} = 1 \tag{7.9}$$

be an equation over G with no constants, where $p_1, \ldots, p_K \in \{1, \ldots, N\}$ and $\epsilon_1, \ldots, \epsilon_K \in \{-1, 1\}$. We will use the notation introduced in Lemma 7.1.1 for the generators, and $(\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N)$ will be a potential solution to (7.9), with each $\boldsymbol{\nu}_z$ expressed as a word in Mal'cev normal form. For each Mal'cev generator a, define

$$\boldsymbol{\nu}_{z,a} = \mathrm{expsum}_a(\boldsymbol{\nu}_z).$$

We denote these in bold in order to make clear these represent variables (or at least potential solutions to variables) as opposed to the constants that appear from the choice of G.

The following lemma converts an equation of the form of (7.9) into a system of equations and congruences over $\mathbb{Z}$. This is done by expressing the variables as expressions in Mal'cev normal form, plugging these expressions back into the equation, and then converting the resulting word into Mal'cev normal form. After doing this, the exponents of the generators can be equated to 0 or set congruent to 0, which yields the system stated.

Lemma 7.2.4. *The words $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N$ form a solution to (7.9) in G if and only if* *the following equations and congruences hold:*

$$\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,a_m} = 0, \text{ for all } m \in \{1, \ldots, n\}, \tag{7.10}$$

$$\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,b_m} \equiv 0 \mod l_m, \text{ for all } m \in \{1, \ldots, r\}, \tag{7.11}$$

$$\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{z,c_m} + \sum_{u<z=1}^{K} \sum_{i<j=1}^{n} \pi_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{u<z=1}^{K} \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \upsilon_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} \tag{7.12}$$

$$+ \sum_{u<z=1}^{K} \sum_{i<j=1}^{r} \tau_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_i,b_j} + \sum_{z=1}^{K} \sum_{i=1}^{r} \xi_{im} \left\lfloor \frac{\epsilon_z \boldsymbol{\nu}_{p_z,b_i}}{l_i} \right\rfloor = 0, \text{ for all } m \in \{1, \ldots, s\},$$

$$\sum_{z=1}^{K} \boldsymbol{\nu}_{z,d_m} + \sum_{u<z=1}^{K} \sum_{i<j=1}^{n} \alpha_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{u<z=1}^{K} \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \gamma_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} \tag{7.13}$$

$$+ \sum_{u<z=1}^{K} \sum_{i<j=1}^{r} \beta_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_u,b_j} + \sum_{z=1}^{K} \sum_{i=1}^{r} \eta_{im} \left\lfloor \frac{\epsilon_z \boldsymbol{\nu}_{p_z,b_i}}{l_i} \right\rfloor \equiv 0 \mod k_m,$$

for all $m \in \{1, \ldots, t\}$,

where the π_{ijk}*s,* τ_{ijk}*s,* υ_{ijk}*s,* α_{ijk}*s,* β_{ijk}*s, and* γ_{ijk}*s are defined as in Notation 7.1.5 and represent constants. The* ϵ_z*s are defined as in Notation 7.2.3, and represent constants. The potential solutions for the variables are the* $\boldsymbol{\nu}_{z,a}$*s, for a generator* a *and are defined in Notation 7.2.3.*

Proof Consider (7.9) with the potential solution $(\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N)$ plugged in. We obtain

$$\boldsymbol{\nu}_{p_1}^{\epsilon_1} \cdots \boldsymbol{\nu}_{p_K}^{\epsilon_K} = 1.$$

As $\boldsymbol{\nu}_z = a_1^{\boldsymbol{\nu}_{z,a_1}} \cdots a_n^{\boldsymbol{\nu}_{z,a_n}} b_1^{\boldsymbol{\nu}_{z,b_1}} \cdots b_r^{\boldsymbol{\nu}_{z,b_r}} c_1^{\boldsymbol{\nu}_{z,c_1}} \cdots c_s^{\boldsymbol{\nu}_{z,c_s}} d_1^{\boldsymbol{\nu}_{z,d_1}} \cdots d_t^{\boldsymbol{\nu}_{z,d_t}}$, replacing these in the above equation gives

$$(a_1^{\boldsymbol{\nu}_{p_1,a_1}} \cdots a_n^{\boldsymbol{\nu}_{p_1,a_n}} b_1^{\boldsymbol{\nu}_{p_1,b_1}} \cdots b_r^{\boldsymbol{\nu}_{p_1,b_r}} c_1^{\boldsymbol{\nu}_{p_1,c_1}} \cdots c_s^{\boldsymbol{\nu}_{p_1,c_s}} d_1^{\boldsymbol{\nu}_{p_1,d_1}} \cdots d_t^{\boldsymbol{\nu}_{p_1,d_t}})^{\epsilon_1} \cdots \tag{7.14}$$
$$(a_1^{\boldsymbol{\nu}_{p_K,a_1}} \cdots a_n^{\boldsymbol{\nu}_{p_K,a_n}} b_1^{\boldsymbol{\nu}_{p_K,b_1}} \cdots b_r^{\boldsymbol{\nu}_{p_K,b_r}} c_1^{\boldsymbol{\nu}_{p_K,c_1}} \cdots c_s^{\boldsymbol{\nu}_{p_K,c_s}} d_1^{\boldsymbol{\nu}_{p_K,d_1}} \cdots d_t^{\boldsymbol{\nu}_{p_K,d_t}})^{\epsilon_K} = 1.$$

We now convert the left hand side of (7.14) into Mal'cev normal form. This is done from (6.4) to (6.5) in Example 6.6.1, and from (7.4) to (7.7) in Example 7.2.1. We start by pushing all commutators to the right, which gives that (7.14) is equivalent to

$$(a_1^{\boldsymbol{\nu}_{p_1,a_1}} \cdots a_n^{\boldsymbol{\nu}_{p_1,a_n}} b_1^{\boldsymbol{\nu}_{p_1,b_1}} \cdots b_r^{\boldsymbol{\nu}_{p_1,b_r}})^{\epsilon_1} \cdots (a_1^{\boldsymbol{\nu}_{p_K,a_1}} \cdots a_n^{\boldsymbol{\nu}_{p_K,a_n}} b_1^{\boldsymbol{\nu}_{p_K,b_1}} \cdots b_r^{\boldsymbol{\nu}_{p_K,b_r}})^{\epsilon_K}$$

$$\prod_{m=1}^{s} c_m^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,c_m}} \prod_{m=1}^{t} d_m^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,d_m}} = 1.$$

Using Notation 7.1.5, if $i < j$, then $a_j a_i = a_i a_j [a_j,\ a_i] = a_i a_j c_1^{\pi_{ij1}} \cdots c_s^{\pi_{ijs}} d_1^{\alpha_{ij1}} \cdots d_t^{\alpha_{ijt}}$ and $b_j b_i = b_i b_j c_1^{\tau_{ij1}} \cdots c_s^{\tau_{ijs}} d_1^{\beta_{ij1}} \cdots d_t^{\beta_{ijt}}$. Similarly, for any i and j,

$$b_j a_i = a_i b_j c_1^{\upsilon_{ij1}} \cdots c_s^{\upsilon_{ijs}} d_1^{\gamma_{ij1}} \cdots d_t^{\gamma_{ijt}}.$$

We will use this to reorder all of the subwords $(a_1^{\boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{\boldsymbol{\nu}_{p_z,a_n}} b_1^{\boldsymbol{\nu}_{p_z,b_1}} \cdots b_r^{\boldsymbol{\nu}_{p_z,b_r}})^{\epsilon_z}$ into a word within $(a_1^{\pm})^* \cdots (a_n^{\pm})^* (b_1^{\pm})^* \cdots (b_r^{\pm})^* (c_1^{\pm})^* \cdots (c_s^{\pm})^* (d_1^{\pm})^* \cdots (d_t^{\pm})^*$, subject to 'creating' some additional commutators, which are then pushed to the right. Note that if $\epsilon_z = 1$, then the word is already in the desired form, so consider when $\epsilon_z = -1$. Let w be such a subword. Then

$$w = b_r^{-\boldsymbol{\nu}_{p_z,b_r}} \cdots b_1^{-\boldsymbol{\nu}_{p_z,b_1}} a_n^{-\boldsymbol{\nu}_{p_z,a_n}} \cdots a_1^{-\boldsymbol{\nu}_{p_z,a_1}}.$$

We will start at the right, and push terms to the left. We have that the a_1s will have to be pushed past everything (except each other), the a_2s will need to be pushed past everything except the a_1s, and so on up to the b_{r-1}s, which will only need to be pushed past the b_rs, and the b_rs which will not need to be pushed past anything, as they will now be in the correct place. Thus

$$w = a_1^{-\boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{-\boldsymbol{\nu}_{p_z,a_n}} b_1^{-\boldsymbol{\nu}_{p_z,b_1}} \cdots b_r^{-\boldsymbol{\nu}_{p_z,b_r}}$$

$$\prod_{m=1}^{s} c_m^{\sum_{i<j=1}^{n} \pi_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \upsilon_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \tau_{ijm}\boldsymbol{\nu}_{p_z,b_i}\boldsymbol{\nu}_{p_z,b_j}}$$

$$\prod_{m=1}^{t} d_m^{\sum_{i<j=1}^{n} \alpha_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \gamma_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \beta_{ijm}\boldsymbol{\nu}_{p_z,b_i}\boldsymbol{\nu}_{p_z,b_j}}.$$

Now consider the general case for $\epsilon_z \in \{-1,\ 1\}$. Let $\delta_i = 1$ if $i = -1$ and $\delta_i = 0$ otherwise. We have

$$w = a_1^{\epsilon_z\boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{\epsilon_z\boldsymbol{\nu}_{p_z,a_n}} b_1^{\epsilon_z\boldsymbol{\nu}_{p_z,b_1}} \cdots b_r^{\epsilon_z\boldsymbol{\nu}_{p_z,b_r}}$$

$$\prod_{m=1}^{s} c_m^{\delta_{\epsilon_z}\left(\sum_{i<j=1}^{n} \pi_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \upsilon_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \tau_{ijm}\boldsymbol{\nu}_{p_z,b_i}\boldsymbol{\nu}_{p_z,b_j}\right)}$$

$$\prod_{m=1}^{t} d_m^{\delta_{\epsilon_z}\left(\sum_{i<j=1}^{n} \alpha_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \gamma_{ijm}\boldsymbol{\nu}_{p_z,a_i}\boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \beta_{ijm}\boldsymbol{\nu}_{p_z,b_i}\boldsymbol{\nu}_{p_z,b_j}\right)}.$$

We now use the rules $a_j a_i = a_i a_j [a_j,\ a_i] = a_i a_j c_1^{\pi_{ij1}} \cdots c_s^{\pi_{ijs}} d_1^{\alpha_{ij1}} \cdots d_t^{\alpha_{ijt}}$, for $i < j$ and $b_j a_i = a_i b_j c_1^{\upsilon_{ij1}} \cdots c_s^{\upsilon_{ijs}} d_1^{\gamma_{ij1}} \cdots d_t^{\gamma_{ijt}}$, for any i and j to push all of the a_is to the left, ordered from a_1 to a_n, and push all commutators 'created' by this action to the right, to obtain that (7.14) is equivalent to

$$1 = a_1^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{\sum_{z=1}^{K} \epsilon \boldsymbol{\nu}_{p_z,a_n}} b_1^{\epsilon_1 \boldsymbol{\nu}_{p_1,b_1}} \cdots b_r^{\epsilon_1 \boldsymbol{\nu}_{p_1,b_r}} \cdots b_1^{\epsilon_z \boldsymbol{\nu}_{p_z,b_1}} \cdots b_r^{\epsilon_z \boldsymbol{\nu}_{p_z,b_r}}$$

$$\prod_{m=1}^{s} c_m^{\sum_{z=1}^{K} \left(\epsilon_z \boldsymbol{\nu}_{p_z,c_m} + \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \pi_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \upsilon_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \tau_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right) \right)}$$

$$+ \sum_{u<z=1}^{K} \sum_{i<j=1}^{n} \pi_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{u<z=1}^{K} \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \upsilon_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j}$$

$$\prod_{m=1}^{t} d_m^{\sum_{z=1}^{K} \left(\epsilon_z \boldsymbol{\nu}_{p_z,d_m} + \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \alpha_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \gamma_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \beta_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right) \right)}$$

$$+ \sum_{u<z=1}^{K} \sum_{i<j=1}^{n} \alpha_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{u<z=1}^{K} \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \gamma_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{z,a_i} \boldsymbol{\nu}_{u,b_j} .$$

We now reorder the b_is. Again, we use Notation 7.1.5 to say $b_j b_i = b_i b_j c_1^{\tau_{ij1}} \cdots c_s^{\tau_{ijs}} d_1^{\beta_{ij1}} \cdots d_t^{\beta_{ijt}}$ for $i < j$. Reordering the b_is, and pushing all commutators to the right, gives that (7.14) is equivalent to

$$1 = a_1^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{\sum_{z=1}^{K} \epsilon \boldsymbol{\nu}_{p_z,a_n}} b_1^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,b_1}} \cdots b_r^{\sum_{z=1}^{K} \epsilon \boldsymbol{\nu}_{p_z,b_r}}$$

$$\prod_{m=1}^{s} c_m^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,c_m} + \sum_{z=1}^{K} \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \pi_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \upsilon_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \tau_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right) + \sum_{u<z=1}^{K} \left(\sum_{i<j=1}^{n} \pi_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \upsilon_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} + \sum_{i<j=1}^{r} \tau_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_u,b_j} \right)}$$

$$\prod_{m=1}^{t} d_m^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,d_m} + \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \alpha_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \gamma_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \beta_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right) + \sum_{u<z=1}^{K} \left(\sum_{i<j=1}^{n} \alpha_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{\substack{i=1\\j=1}}^{\substack{n\\r}} \gamma_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} + \sum_{i<j=1}^{r} \beta_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_u,b_j} \right)}$$

Next, we reduce each b_i modulo l_i, as in (7.6) to (7.7) in Example 7.2.1. We have from Notation 7.1.5 that $b_i^{l_i} = c_1^{\xi_{i1}} \cdots c_s^{\xi_{is}} d_1^{\eta_{i1}} \cdots d_t^{\eta_{it}}$. Doing this, then pushing all commutators to the right implies that (7.14) is equivalent to

$$1 = a_1^{\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,a_1}} \cdots a_n^{\sum_{z=1}^{K} \epsilon \boldsymbol{\nu}_{p_z,a_n}} b_1^{\left(\sum_{z=1}^{K} \epsilon_z \boldsymbol{\nu}_{p_z,b_1}\right) \bmod l_1} \cdots b_r^{\left(\sum_{z=1}^{K} \epsilon \boldsymbol{\nu}_{p_z,b_r}\right) \bmod l_r}$$

$$\prod_{m=1}^{s} c_m^{\sum_{z=1}^{K} \left(\epsilon_z \boldsymbol{\nu}_{p_z,c_m} + \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \pi_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \upsilon_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \tau_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right)\right)}$$

$$+ \sum_{u<z=1}^{K} \left(\sum_{i<j=1}^{n} \pi_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \upsilon_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} + \sum_{i<j=1}^{r} \tau_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_u,b_j} \right)$$

$$+ \sum_{z=1}^{K} \sum_{i=1}^{r} \xi_{im} \left\lfloor \frac{\epsilon_z \boldsymbol{\nu}_{p_z,b_i}}{l_i} \right\rfloor$$

$$\prod_{m=1}^{t} d_m^{\sum_{z=1}^{K} \left(\epsilon_z \boldsymbol{\nu}_{p_z,d_m} + \delta_{\epsilon_z} \left(\sum_{i<j=1}^{n} \alpha_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \gamma_{ijm} \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_z,b_j} + \sum_{i<j=1}^{r} \beta_{ijm} \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_z,b_j} \right)\right)}$$

$$+ \sum_{u<z=1}^{K} \left(\sum_{i<j=1}^{n} \alpha_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,a_j} + \sum_{\substack{i=1 \\ j=1}}^{\substack{n \\ r}} \gamma_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,a_i} \boldsymbol{\nu}_{p_u,b_j} + \sum_{i<j=1}^{r} \beta_{ijm} \epsilon_z \epsilon_u \boldsymbol{\nu}_{p_z,b_i} \boldsymbol{\nu}_{p_u,b_j} \right)$$

$$+ \sum_{z=1}^{K} \sum_{i=1}^{r} \eta_{im} \left\lfloor \frac{\epsilon_z \boldsymbol{\nu}_{p_z,b_i}}{l_i} \right\rfloor.$$

The right hand side of the identity above is in normal form, excepting the fact that the exponents of the d_is are not necessarily reduced with respect to their modularities. It follows that this identity is satisfied if and only if the exponents of the a_is, b_is, and c_is all equal zero, and the exponent of each d_i is congruent to 0 modulo k_i. Thus $(\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N)$ is a solution if and only if the exponents satisfy these conditions, and the result follows. □

Definition 7.2.5. If $w = 1$ is an equation in a class 2 nilpotent group, consider the system of equations over the integers defined by taking each variable V, and viewing it in Mal'cev normal form by introducing new variables: $V = a_1^{W_1} \cdots a_n^{W_n} b_1^{X_1} \cdots b_r^{X_r} c_1^{Y_1} \cdots c_s^{Y_s} d_1^{Z_1} \cdots d_n^{Z_t}$, where the W_is, X_is, Y_is and Z_is take values in $\mathbb{Z}$. The resulting system of equations and congruences over $\mathbb{Z}$ obtained by setting the expressions in the exponents equal to zero is called the *$\mathbb{Z}$-system* of $w = 1$.

The following example demonstrates the method used in the proof of Lemma 7.2.7; that is that any equation in G is equivalent to a system of equations that consists of one equation with no constants, and finitely many equations of the form $X = g$, for some variable X, and a constant $g \in G$.

Example 7.2.6. Let G be a group, and consider the equation

$$X^2 g Y^{-1} X h Z = 1, \tag{7.15}$$

where $g,\ h \in G$. Then (7.15) is equivalent to the system

$$X^2 W_1 Y^{-1} X W_2 Z = 1$$

$$W_1 = g$$

$$W_2 = h.$$

To do this, notice that we can obtain (7.15) from the above system by substituting W_1 and W_2 for g and h, respectively.

Lemma 7.2.7. *The $\mathbb{Z}$-system of any equation in a class 2 nilpotent group is equivalent to a system of the form of Lemma 7.2.4, together with finitely many (additional) linear equations and linear congruences.*

Proof We have that any equation in a class 2 nilpotent group G is equivalent to a system of comprising one equation $X_{p_1}^{\epsilon_1} \cdots X_{p_K}^{\epsilon_K} = 1$ with no constants, and equations of the form $X_z = g$, for some variable X_z and $g \in G$. Let $g \in G$, and X_z be a variable. We have that g is represented by a word

$$g = a_1^{g_{a_1}} \cdots a_n^{g_{a_n}} b_1^{g_{b_1}} \cdots b_r^{g_{b_r}} c_1^{g_{c_1}} \cdots c_s^{g_{c_s}} d_1^{g_{d_1}} \cdots d_t^{g_{d_t}}$$

in Mal'cev normal form, where $g_\mu \in \mathbb{Z}$ for any generator μ. Then the $\mathbb{Z}$-system of $X_z = g$ is

$$X_{z,a_i} = g_{a_i} \text{ for each } i \in \{1, \ldots, n\}$$

$$X_{z,b_i} \equiv g_{b_i} \mod l_i \text{ for each } i \in \{1, \ldots, r\}$$

$$X_{z,c_i} = g_{c_i} \text{ for each } i \in \{1, \ldots, s\}$$

$$X_{z,d_i} \equiv g_{d_i} \mod k_i \text{ for each } i \in \{1, \ldots, n\},$$

These equations and congruences are linear, as required. □

We now restate Lemma 7.2.7 up to grouping constants, and renaming constants and variables. We also add new constants κ_{ji} and χ_{ji}, which any system of the form of Lemma 7.2.7 will have as 0, however, after 'plugging' linear equations back into these equations, these constants will not necessarily be 0. We also note that if there is a variable X in the equation $w = 1$, such that $\text{expsum}_X(w) \neq 0$, then the coefficient in of ν_{X,c_m} in the exponent of c_m will be non-zero for all m. This follows as this coefficient equals $\text{expsum}_X(w)$.

Proposition 7.2.8. *The $\mathbb{Z}$-system of a single equation $w = 1$ in a class 2 nilpotent group is equivalent to a finite system of linear equations and congruences in $\mathbb{Z}$, together with the following equations and congruences for finitely many j and k:*

$$\sum_{i=1}^{n} -\alpha_{ji} Y_{ji} + f_j(X_1, \ldots, X_m) + \sum_{i=1}^{m} \epsilon_{ji} \left\lfloor \frac{\beta_{ji} X_i + \kappa_{ji}}{\gamma_{ji}} \right\rfloor = 0, \tag{7.16}$$

$$g_k(X_1, \ldots, X_m) + \sum_{i=1}^{m} \zeta_{ki} \left\lfloor \frac{\mu_{ki} X_i + \chi_{ki}}{\lambda_{ki}} \right\rfloor \equiv 0 \mod \delta_k, \tag{7.17}$$

where the values with Greek alphabet names are all constants, easily computed from *the class 2 nilpotent group and the single equation, $X_1, \ldots, X_m$ are variables which appear in the linear equations and congruences, $Y_{j1}, \ldots, Y_{jn}$ are variables which do not appear in any of the linear equations and congruences, for each j and the f_js and g_ks are quadratic functions.*

Moreover, if there is a variable X such that $\operatorname{expsum}_X(w) \neq 0$, then there exists i such that $\alpha_{ji} \neq 0$ for all j.

7.3 Equations in virtually Heisenberg groups

Within this section, we look at how equations behave when passing to a finite index overgroup. From [34], we know that the single equation problem is decidable in any class 2 nilpotent group with a virtually cyclic commutator subgroup. We show that the satisfiability of single equations in groups that are virtually the Heisenberg group is decidable, which is due to the straightforward nature of its automorphisms. Further work could be to extend this to virtually any class 2 nilpotent group with a virtually cyclic commutator subgroup, or to attempt to show the solutions in virtually the Heisenberg group are EDT0L. Throughout this section, we will use the following presentation for the Heisenberg group $H(\mathbb{Z})$:

$$H(\mathbb{Z}) = \langle a_1,\ a_2,\ c \mid [a_1,\ a_2] = c \rangle.$$

Lemma 7.3.1 ([46], Proposition 4.4.3)**.** *Let θ be any automorphism of the Heisenberg group. Then there exist linear functions $f,\ g \colon \mathbb{Z}^2 \to \mathbb{Z}$, a quadratic function $h \colon \mathbb{Z}^2 \to \mathbb{Z}$ and $\alpha \in \mathbb{Z}$ such that*

$$\theta(a_1^i a_2^j c^k) = a_1^{f(i,\ j)} a_2^{g(i,\ j)} c^{\alpha k + h(i,\ j)}. \tag{7.18}$$

Moreover, f, g, h and k can be computed from the action of ϕ on the generators a_1, a_2 and c.

Remark 7.3.2. In the statement of [46], Proposition 4.4.3, the fact that f, g, h and k can be computed from the action of θ on the generators is not mentioned within the statement of the proposition. However, the proof explicitly calculates them from a matrix representative of the image of θ within the outer automorphism group, together with an inner automorphism, both of which can be computed from the action of θ on the generators, which gives the required statement.

Definition 7.3.3. Let G be a finitely generated group. The *single (twisted) equation problem* in G is the decidability question as to whether there is a terminating algorithm that accepts a (twisted) equation $w = 1$ as input, returns YES if $w = 1$ admits a solution and NO otherwise, where elements of G within w are represented by words over a finite generating set, and automorphisms are represented by their action on the finite generating set.

The following lemma is widely known, although often not stated explicitly. Variations of it have been used to show systems of equations in virtually free groups, or virtually abelian groups are decidable, or to describe the structure of solution sets (see for example [26] or [31]). We include a proof for completeness.

Lemma 7.3.4. *Let G be a group with a finite index normal subgroup H, such that H has decidable single twisted equation problem. Then G has decidable single equation problem.*

Proof Let $w = 1$ be an equation in G. By Lemma 4.3.7, there is a finite set $\mathcal{S}$ of equations in H such that $w = 1$ admits a solution if and only if some equation in $\mathcal{E}$ does. Since we can compute whether or not a given equation in H admits a solution, the result follows. □

Now that we have Lemma 7.3.4, the following is (almost) all that is required to prove that single equations in virtually Heisenberg groups are decidable.

Lemma 7.3.5. *The single twisted equation problem in the Heisenberg group is decidable.*

Proof Let the functions f, g and h, and the integer k be defined as in Lemma 7.3.1. Let $w = 1$ be a twisted equation in the Heisenberg group. We first convert this into a $\mathbb{Z}$-system as in Section 7.2. Doing so will yield the system from Lemma 7.2.4, except $\boldsymbol{\nu}_{z,a_1}$, $\boldsymbol{\nu}_{z,\ a_2}$ and $\boldsymbol{\nu}_{z,\ c}$ will be replaced with $f(\boldsymbol{\nu}_{z,a_1},\ \boldsymbol{\nu}_{z,a_2})$, $g(\boldsymbol{\nu}_{z,a_1},\ \boldsymbol{\nu}_{z,a_2})$ and $h(\boldsymbol{\nu}_{z,a_1},\ \boldsymbol{\nu}_{z,a_2}) + k\boldsymbol{\nu}_{z,c}$, respectively. Let $\alpha_f,\ \alpha_g,\ \beta_f,\ \beta_g,\ \gamma_f,\ \gamma_g \in \mathbb{Z}$ be such that

$$f(i,\ j) = \alpha_f i + \beta_f j + \gamma_f \text{ and } g(i,\ j) = \alpha_g i + \beta_g j + \gamma_g.$$

Applying these to an equation with no constants, as in Lemma 7.2.4, gives that the associated $\mathbb{Z}$-system of $w = 1$ with the potential solutions $\boldsymbol{\nu}_1,\ \ldots,\ \boldsymbol{\nu}_N$ plugged in, is

$$\sum_{z=1}^{K}(\alpha_f\epsilon_z\boldsymbol{\nu}_{p_z,a_1} + \beta_f\epsilon_z\boldsymbol{\nu}_{p_z,a_2} + \gamma_f) = 0, \tag{7.19}$$

$$\sum_{z=1}^{K}(\alpha_g\epsilon_z\boldsymbol{\nu}_{p_z,a_1} + \beta_g\epsilon_z\boldsymbol{\nu}_{p_z,a_2} + \gamma_g) = 0,$$

$$\sum_{z=1}^{K} h(\epsilon_z\boldsymbol{\nu}_{p_z,a_1},\ \epsilon_z\boldsymbol{\nu}_{p_z,a_2}) + k\epsilon_z\boldsymbol{\nu}_{p_z,c} + \sum_{u<z=1}^{N} f(\epsilon_z\boldsymbol{\nu}_{p_z,a_1},\ \epsilon_z\boldsymbol{\nu}_{p_z,a_2})g(\epsilon_u\boldsymbol{\nu}_{p_u,a_1},\ \epsilon_u\boldsymbol{\nu}_{p_u,a_2}) = 0,$$

with the addition of finitely many linear equations and congruences, by Lemma 7.2.7. Here, as we only have one commutator generator, $c = c_1$, the constant π_{12} is defined to be π_{121}, as defined in Lemma 7.2.4 and Notation 7.1.5. By rearranging and renaming constants and variables, it follows that (7.19) is equivalent to

$$\sum_{i=1}^{2N}\delta_i X_i + \lambda = 0,$$

$$\sum_{i=1}^{2N}\eta_i X_i + \mu = 0,$$

$$\phi(X_1,\ \ldots,\ X_{2N}) = \sum_{i=1}^{N}\xi_i Y_i,$$

As the first two equations are linear, these can be plugged back into the third equation to obtain another equation, which will still be quadratic, that will admit a solution if and only if the above system does. We can use the algorithm of Siegel [91] to determine if this single quadratic equation admits a solution. $\square$

Theorem 7.3.6. *The single equation problem in a virtually Heisenberg group is decidable.*

Proof This follows by Lemmas 7.3.4 and 7.3.5, together with the fact that every virtually Heisenberg group has a finite index normal Heisenberg group subgroup, a proof of which can be found in [46], Lemma 4.5.2. □

Appendix A

Context-free and indexed languages

A.1 Context-free languages

The next class up from regular in Chomsky's hierarchy of languages is the class of context-free languages. Unlike regular languages, automata accepting context-free languages have a certain amount of memory, beyond the states. We do not need to use context-free languages in this thesis, however we give the basic definitions, as they are a standard class of languages, and give a good background for where the classes that we do use fit into the general picture.

We start with the definition.

Definition A.1.1. A *context-free grammar* is a tuple $\mathcal{G} = (V,\ \Sigma,\ P,\ \mathbf{S})$, where

1. V is an alphabet, called the *non-terminal alphabet*;
2. Σ is an alphabet, called the *terminal alphabet*, disjoint from V;
3. $P \subseteq V \times (V \cup \Sigma)^*$ is a finite set called the set of *productions*. An element $(\mathbf{A},\ \omega)$ is usually written $\mathbf{A} \to \omega$;
4. $\mathbf{S} \in V$ is called the *start symbol*.

The production $\mathbf{A} \to \omega$ acts on words in $\nu \in (V \cup \Sigma)^*$ by replacing a single occurrence of $\mathbf{A}$ within ν with ω.

A word $w \in \Sigma^*$ is *generated* by $\mathcal{G}$ if we can apply a sequence of productions in $\mathcal{G}$ to $\mathbf{S}$ to obtain w. The *language generated* by $\mathcal{G}$ is the language of all words generated by $\mathcal{G}$.

A language is called *context-free* if it is generated by a context-free grammar.

We now give some examples of context-free languages. None of the languages mentioned are regular.

Example A.1.2. If w is a word, we will use $\overleftarrow{w}$ to denote the word obtained by reading w backwards. We will show that the language $L = \{w\overleftarrow{w} \mid w \in \{a,\ b\}^*\}$ is context-free.

We define a context-free grammar for L. Our set of non-terminals will be $V = \{\mathbf{S}\}$, our terminal alphabet will be $\{a,\ b\}$, our start symbol will be $\mathbf{S}$, and our productions will be

$$\{\mathbf{S} \to a\mathbf{S}a,\ \mathbf{S} \to b\mathbf{S}b,\ \mathbf{S} \to \varepsilon\}.$$

Since every word that can be made starting with $\mathbf{S}$ using the above productions, if we apply the third production we will end up with a word in $\{a,\ b\}^*$, and thus we will not be able to apply any more productions. Before we do this, the first two productions will allow us to generate a word to the left of the $\mathbf{S}$ always adding to the end, and a word to the right of the $\mathbf{S}$ obtained by always adding the same letter to the beginning. Thus any word generated will lie in L. Moreover, since we can create any word to the left of the $\mathbf{S}$, because we can add a or b whenever we want, we can generate all words in L.

Example A.1.3. Consider the language W over all words w over $\{a,\ a^{-1}\}$ such that w has the same number of occurrences of a as a^{-1}. Equivalently, W is the language of all words over $\{a,\ a^{-1}\}$ that represent the identity of the group $\mathbb{Z}$, with the presentation $\langle a \mid \rangle$. This language is called the *word problem* of $\mathbb{Z}$ with respect to $\{a\}$. Whilst we will only show that W is context-free, Muller, Schupp and Dunwoody ([74], [37]) showed that a group has a context-free word problem if and only if it is virtually free.

To show that W is context-free, consider the context-free grammar with non-terminals $V = \{\mathbf{S}\}$, terminal alphabet $\{a,\ a^{-1}\}$, start symbol $\mathbf{S}$, and productions

$$\{\mathbf{S} \to \mathbf{S}a\mathbf{S}a^{-1}\mathbf{S},\ \mathbf{S} \to \mathbf{S}a^{-1}\mathbf{S}a\mathbf{S},\ \mathbf{S} \to \varepsilon\}.$$

If we start using the first two productions, we will always end up with a word with $\mathbf{S}$ in (at least) every other position. Thus, if occurrences of $\mathbf{S}$ are ignored, we can use these productions to 'insert' aa^{-1} or $a^{-1}a$ anywhere. Afterwards, we can use $\mathbf{S} \to \varepsilon$ repeatedly to remove all non-terminals. We have now shown that the language generated by this context-free grammar contains W.

If we generate a word using the grammar, it must have the same number of as as a^{-1}s, since the productions always add the same number of as as a^{-1}s. Thus W is accepted by this grammar, and is therefore context-free.

Like regular languages, context-free languages form a full abstract family of languages, that is they are closed under the operations in the following lemma.

Lemma A.1.4 ([58], Section 6.2)**.** *Let L and M be context-free languages over alphabets Σ_L and Σ_M. Let $\phi\colon \Sigma_L^* \to \Sigma_M^*$ be a free monoid homomorphism. Then the following languages are context-free:*

1. *$L \cup M$ (union);*
2. *$L \cap K$, for any regular language K (intersection with regular languages);*
3. *LM (concatenation);*
4. *L^* (Kleene star closure);*
5. *$L\phi$ (homomorphism);*
6. *$L\phi^{-1}$ (inverse homomorphism).*

Using the fact that all regular languages are defined by rational expressions (Lemma 2.4.6) with Lemma A.1.4, we can now show the following.

Lemma A.1.5. *Regular languages are context-free.*

An alternative method of defining context-free languages is using pushdown automata. These are a generalisation of finite-state automata, and thus imply the regular languages are context-free. We omit the definition here, however we refer the reader to [56] or [58] for more information about context-free languages, including pushdown automata.

A.2 Indexed languages

Indexed languages were introduced in the 1960s by Aho ([2], [3]) as a generalisation of context-free languages. Whilst they are probably not as well-studied as context-free and regular languages, they still occupy an important place as a full abstract family of languages that is less restrictive than context-free, but not nearly as broad as the large class of context-sensitive languages. We refer the reader to Aho's papers, and also Gilman's exposition of formal languages [51], Section 6 for more information. We follow the notation of [1].

Definition A.2.1. An *indexed grammar* is a tuple $\mathcal{G} = (V,\ \Sigma,\ \chi,\ P,\ \mathbf{S})$, where

1. V is an alphabet, called the *non-terminal alphabet*;
2. Σ is an alphabet, called the *terminal alphabet*, disjoint from V;
3. χ is an alphabet, called the *flag alphabet* or *indices*, disjoint from $V \cup \Sigma$. Elements of $V \times \chi^*$ are called *indexed non-terminals*, and the element $(\mathbf{A},\ f) \in V \times \chi$ is denoted $\mathbf{A}_f$. We consider $\mathbf{A}_\varepsilon$ and $\mathbf{A}$ as equal.
4. $P \subseteq (V \times \chi) \times ((V \times \chi) \cup \Sigma)^*$ is a finite set called the set of *productions*. An element $(\mathbf{A}_f,\ \omega)$ is usually written $\mathbf{A}_f \to \omega$. All productions must be of one of the following forms, where $\mathbf{A},\ \mathbf{B} \in V$, $\omega \in (V \cup \Sigma)^*$ and $f \in \chi^*$:
   (a) $\mathbf{A} \to \omega$. This production acts on words of indexed non-terminals by replacing a single occurrence of $\mathbf{A}_g$ for any $g \in \chi^*$, with ω, where every non-terminal in ω is indexed with g;
   (b) $\mathbf{A} \to \mathbf{B}_f$, called a *push* production. This production acts on words of indexed non-terminals by replacing a single occurrence of $\mathbf{A}_g$ for any $g \in \chi^*$, with $\mathbf{B}_{gf}$;
   (c) $\mathbf{A}_f \to \omega$, called a *pop* production. This production acts on words of indexed non-terminals by replacing a single occurrence of $\mathbf{A}_{gf}$ for any $g \in \chi^*$, with ω, where every non-terminal in ω is indexed with g.

5. $\mathbf{S} \in V$ is called the *start symbol*.

A word $w \in \Sigma^*$ is *generated* by $\mathcal{G}$ if we can apply a sequence of productions in $\mathcal{G}$ to $\mathbf{S}$ to obtain w. The *language generated* by $\mathcal{G}$ is the language of all words generated by $\mathcal{G}$.

A language is called *indexed* if it is generated by an indexed grammar.

We give a standard example of an indexed language.

Example A.2.2. The language $L = \{w^2 \mid w \in \Sigma^*\}$ is indexed over the alphabet Σ, but not context-free. We construct an indexed grammar for L. Let $V = \{\mathbf{S},\ \mathbf{T},\ \mathbf{U}\}$. Let $\hat{\Sigma} = \{\hat{a} \mid a \in \Sigma\}$ be a disjoint copy of Σ. Let $\chi = \hat{\Sigma} \cup \{\$\}$. Consider the indexed grammar $\mathcal{G} = (V,\ \Sigma,\ \chi,\ P,\ \mathbf{S})$, where the set P of productions equals

$$\{\mathbf{S} \to \mathbf{T}_{\$}\} \cup \{\mathbf{T} \to \mathbf{T}_{\hat{a}} \mid a \in \Sigma\} \cup \{\mathbf{T} \to \mathbf{U}\mathbf{U},\ \mathbf{U}_{\hat{a}} \to \mathbf{U}a,\ \mathbf{U}_{\$} \to \varepsilon\}.$$

Any word generated by $\mathcal{G}$ starts with $\mathbf{S}$, and we are then forced to go through the production $\mathbf{S} \to \mathbf{T}_{\$}$. After this, a (possibly empty) sequence of productions of the form $\mathbf{T} \to \mathbf{T}_{\hat{a}}$ will be performed to obtain $\mathbf{T}_{\$\hat{w}}$ for some $w \in \Sigma^*$. Following this, we must apply $\mathbf{T} \to \mathbf{U}\mathbf{U}$, as otherwise we are in the case we just considered. Thus we have $\mathbf{U}_{\$\hat{w}}\mathbf{U}_{\$\hat{w}}$. We are now forced pop off each index from $\mathbf{U}$ to obtain $\mathbf{U}_{\$}u\mathbf{U}_{\$}u$, where u is the reverse of w. We can now only apply $\mathbf{U}_{\$} \to \varepsilon$ twice, and we end up with u^2, and so $\mathcal{G}$ only generates words in L. Since we can choose the order that flags are pushed to $\mathbf{T}$, we can obtain $\mathbf{T}_{\$\hat{w}}$ for all $w \in \Sigma^*$, and thus all words in L are generated by $\mathcal{G}$.

Since indexed grammars are a generalisation of context-free grammars, we have the following:

Lemma A.2.3. *Context-free languages are indexed.*

Similar to regular and context-free languages, indexed languages also form a full abstract family of languages.

Lemma A.2.4 ([51], Theorem 6.3 and Theorem 6.10)**.** *Let L and M be indexed languages over alphabets Σ_L and Σ_M. Let $\phi\colon \Sigma_L^* \to \Sigma_M^*$ be a free monoid homomorphism. Then the following languages are indexed:*

1. *$L \cup M$ (union);*
2. *$L \cap K$, for any regular language K (intersection with regular languages);*
3. *LM (concatenation);*
4. *L^* (Kleene star closure);*
5. *$L\phi$ (homomorphism);*
6. *$L\phi^{-1}$ (inverse homomorphism).*

Indexed languages also have a definition using an automaton, called a ' nested stack automaton', for which we refer the reader to [51], Section 6.

Appendix B

L-Systems

We start with the definition of ET0L languages in Section B.1. Section B.2 covers a definition of ET0L languages using automata instead of grammars. The next few sections cover alternative definitions of EDT0L languages, however the equivalence of these definitions is non-trivial. Section B.8 proves that ET0L languages are indexed, and presents some corollaries that can be drawn from this.

Whilst we will focus on EDT0L languages, we will mention a number of other L-systems. The acronym T0L stands for *Table 0-interaction Lindenmayer*, and defines an L-system that is equivalent to an ET0L system with no extended alphabet. We will use L-systems that are T0L systems with the following additional letters:

1. E (extended) - there is an extended alphabet;
2. D (deterministic) - tables are free monoid endomorphisms;
3. P (propagating) - tables or endomorphisms are non-erasing; that is, they map no letter to the empty word;
4. H (homomorphism) - this language is the image of the remainder of the acronym under a free monoid homomorphism;
5. N (non-erasing homomorphism) - this language is the image of the remainder of the acronym under a non-erasing free monoid homomorphism;
6. W (weak coding) - this language is the image of the remainder of the acronym under a weak coding;

7. C (coding) - this language is the image of the remainder of the acronym under a coding.

B.1 ET0L languages

ET0L languages are a generalisation of EDT0L languages, and can be thought of as a 'non-deterministic' variant on EDT0L languages.

In order to define an ET0L language, we first need to define a table.

Definition B.1.1. Let C be an alphabet. A *table* t over C is a set of tuples $(c,\ K_c)$, for each $c \in C$, and where K_c is a finite set of words over C.

Tables act as rewrite rules on words in C^* as follows. If $\omega = c_1 \cdots c_n$ is a word over C, where each $c_i \in C$, then the set of images of ω under a table t, denoted ωt, is defined by

$$\omega t = \{\nu_1 \cdots \nu_n \mid (c_i,\ K_{c_i}) \in t \text{ and } \nu_i \in K_{c_i} \text{ for all } i\}.$$

Tables $t_1 = \{(c,\ K_c) \mid c \in C\}$ and $t_2 = \{(c,\ M_c) \mid c \in C\}$ over C can be composed to create a new table

$$t_1 t_2 = \left\{ \left(c,\ \bigcup_{d \in K_c} M_d \right) \;\middle|\; c \in C \right\}.$$

Remark B.1.2. Tables are often presented as larger sets of tuples of letters an words, rather than tuples of letters and sets. For the work we do later on, the latter definition is easier to generalise, so we have used this.

We give some examples of tables, and how to compose them.

Example B.1.3. Consider the alphabet $C = \{\perp,\ a,\ b\}$, and two tables over C:

$$t_1 = \{(\perp,\ \{aba\}),\ (a,\ \{a,\ a^2\}),\ (b,\ \{b\})\}, \qquad t_2 = \{(\perp,\ \{\perp\}),\ (a,\ \{a,\ b\}),\ (b,\ \{b^2\})\}.$$

These can be written as

$$t_1\colon \begin{array}{c|c} \perp & aba \\ a & a,\ a^2 \\ b & b \end{array} \qquad t_2\colon \begin{array}{c|c} \perp & \perp \\ a & a,\ b \\ b & b^2 \end{array}.$$

The composition is

$$t_1t_2\colon \begin{array}{c|c} \perp & ab^2a,\ ab^3,\ b^3a,\ b^4 \\ a & a,\ b,\ a^2,\ ab,\ ba,\ b^2 \\ b & b^2 \end{array}.$$

We can now define ET0L languages which are a generalisation of EDT0L languages. We base our definitions on our definition of EDT0L languages; however, there are a number of equivalent definitions used elsewhere, as discussed in Section 3.3.

Definition B.1.4. An *ET0L system* is a tuple $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$, where

1. Σ is an alphabet, called the *(terminal) alphabet*;
2. C is a finite superset of Σ, called the *extended alphabet* of $\mathcal{H}$;
3. $\omega \in C^*$ is called the *start word*;
4. $\mathcal{R}$ is a regular (as a language) set of tables over C, called the *rational control* of $\mathcal{H}$.

The language *accepted* by $\mathcal{H}$ is $L(\mathcal{H}) = \Sigma \cap \bigcup_{\phi\in\mathcal{R}} \omega\phi$.

A language that is accepted by some ET0L system is called an *ET0L language*.

We extend the notion of an endomorphism of a free monoid fixing a letter to tables.

Definition B.1.5. Let C be an alphabet, and let ϕ be a table. We say ϕ *fixes* a letter $a \in C$ if $a\phi = \{a\}$.

Remark B.1.6. EDT0L systems can be viewed as ET0L systems, where for every table ϕ in the rational control, and for every letter c in the extended alphabet, $|c\phi| = 1$. When talking about systems that could be ET0L systems or EDT0L systems, we will often treat them as ET0L systems, with the understanding that in the EDT0L case, we are adding the above restriction on the lengths of images.

B.2 CSPD Automata

Introduced by van Leeuwen [92], CSPD automata give an alternative method of describing ET0L languages, and are sometimes easier to work with than ET0L systems. We give a brief definition and explanation of how they work.

These machines are a generalisation of pushdown automata; they still have a finite state control and a pushdown stack, but they also possess a second stack called the check-stack. When attempting to read a word using a CSPD automaton, there are two stages. The first stage comprises choosing a check-stack from the regular language of allowed check-stacks. After it is chosen, it cannot be edited; the machine will move up and down it without changing it.

During the second stage, the word is read letter by letter. When reading a letter a, the machine looks at the state it is in, the letter at the top of the pushdown and the letter within the check-stack at the same height as the pushdown, in order to decide what to do next. It then removes the pushdown letter it looked at (possibly ε), places a new symbol on top of the pushdown (again, possibly ε), and moves up or down the check-stack stack, so that the length of the pushdown and the position on the check stack always remain the same.

Definition B.2.1. A *check-stack pushdown automaton (CSPD automaton)* is a tuple

$$\mathcal{A} = (Q,\ \Sigma,\ \Gamma,\ \Delta,\ \bot,\ \mathcal{R},\ \theta,\ q_0,\ F),$$

where

1. Q is a finite set, called the set of *states*;
2. Σ is an alphabet, called the *terminal alphabet*;
3. Γ is an alphabet, called the *pushdown alphabet*;
4. Δ is an alphabet, called the *check-stack alphabet*;
5. $\bot \notin \Delta \cup \Gamma$ is the *bottom of stack symbol*;
6. $\mathcal{R} \subseteq (\{\bot\} \cup \Delta)^*$ is a regular language, called the set of *allowed check-stacks.* All words in $\mathcal{R}$ must be of the form $\bot\, \omega$ for some $\omega \in \Delta^*$;

7. θ is a finite subset of

$$(Q \times (\Sigma \cup \{\varepsilon\}) \times ((\Delta \times \Gamma) \cup \{(\varepsilon,\ \varepsilon),\ (\perp,\ \perp)\})) \times (Q \times (\Gamma \cup \{\perp\})^*),$$

is called the *transition relation*. All elements of θ must be of one of three forms described below. Elements of the transition relation are called *transitions*, and the transition $((p,\ a,\ (x,\ \alpha)),\ (q,\ \omega))$ is usually denoted $(p,\ a,\ (x,\ \alpha)) \to (q,\ \omega)$;
8. $q_0 \in Q$ is called the *start state*;
9. $F \subseteq Q$ is called the set of *accept states*.

The three forms of transitions within the transition relation θ are:

1. $(p,\ a,\ (\perp,\ \perp)) \to (q,\ \omega\ \perp)$. In this case the machine will be in state p, with $\perp$ at the top of both stacks, and will see and consume $a \in \Sigma \cup \{\varepsilon\}$ as input. It will then move to the state q, push ω onto the pushdown, and also move up the check-stack by $|\omega|$, so that the length of the pushdown and the position on the check-stack remain the same.
2. $(p,\ a,\ (x,\ \alpha)) \to (q,\ \omega)$. This transition can be used when the machine is in the state p, and sees x on the check stack, and α at the top of the pushdown, whilst reading and consuming the input $a \in \Sigma \cup \{\varepsilon\}$. The machine then pops α from the pushdown, adds ω to the pushdown, moves $|\omega| - 1$ positions up the check-stack, and transfers to the state q.
3. $(p,\ a,\ (\varepsilon,\ \varepsilon)) \to (q,\ \omega)$. This transition can be used when the CSPD automaton is in the state p, with any possible symbols on both stacks, whilst seeing and consuming $a \in \Sigma \cup \{\varepsilon\}$ as input. The machine then pushes ω onto the pushdown, moves $|\omega|$ positions up the check-stack, and transitions into the state q.

A word $u \in \Sigma^*$ is *accepted* by the CSPD automaton $\mathcal{A}$, if there is an allowed check-stack $\omega \in \mathcal{R}$, together with a finite sequence of transitions in θ, starting at the state q_0 with $\perp$ on the pushdown, and at the bottom (also looking at $\perp$) of the check-stack, and terminating in a state within F, whilst reading u as input.

The *language accepted* by $\mathcal{A}$ is the set of all words accepted by $\mathcal{A}$.

Whilst first proved by van Leeuwen, Bishop and Elder have recently provided a proof of the result, using modern notation [10].

Theorem B.2.2 ([92])**.** *A language is ET0L if and only if it is accepted by a CSPD automaton.*

B.3 EPDT0L languages

EPDT0L languages provide another equivalent definition of EDT0L languages, when the empty word is removed. They allow us to place an assumption on EDT0L systems that endomorphisms are non-erasing; that is, they do not map any letters to the empty word. They have also been used to prove a pumping lemma of a sort for EDT0L languages ([86], Section IV.3).

Theorem B.3.1. *A language L is EDT0L if and only if $L \backslash \{\varepsilon\}$ is EPDT0L.*

Proof Using Theorem 3.3.2, (2), we can assume $\varepsilon \notin L$. Let $\mathcal{H} = (\Sigma,\ C,\ \omega,\ \mathcal{R})$ be an EDT0L system for L. We will define an EPDT0L system $\mathcal{G}$ based on $\mathcal{H}$.

By Theorem 3.3.1, we can assume that $\omega = \perp$ for some $\perp \in C$. The extended alphabet of $\mathcal{G}$ will be $D = \{[c,\ Z] \mid c \in C,\ Z \subseteq C\} \sqcup \{F\} \sqcup \Sigma$, where F is a symbol not already used, which we will use as a fail symbol. The terminal alphabet will be Σ, and the start word will be $[\perp,\ \varnothing]$.

Let $B \subseteq \operatorname{End}(C^*)$ be an alphabet of $\mathcal{R}$. We now define the rational control of $\mathcal{G}$. For each $\phi \in B$, define $\Phi_\phi \subseteq \operatorname{End}(D^*)$ as follows. We first define, for any $Z \subseteq C$,

$$\operatorname{suc}_\phi(Z) = \{U \subseteq C \mid Z\phi = U\}.$$

Let Φ_ϕ be the set of all $\psi \in \operatorname{End}(D^*)$, such that $F\psi = F$, and $a\psi = F$, for all $a \in \Sigma$, and $[c,\ Z]\psi$, where $c \in C$ and $Z \subseteq \Sigma$, is defined by:

1. If $c\phi = d \in C$, then $[c,\ Z]\psi = [d,\ Z']$ for any $Z' \in \operatorname{suc}_\phi(Z)$;

2. If $c\phi = d_1 \cdots d_k$, where $d_i \in C$ for all i, then

$$[c,\ Z]\psi = [d_{i_1},\ \{d_1,\ \ldots,\ d_{i_1-1}\} \cup Z_{i_1}][d_{i_2},\ Z_{i_2}] \cdots [d_{i_{p-1}},\ Z_{i_{p-1}}][d_{i_p},\ Z_{i_p} \cup Z'],$$

for any $Z' \in \text{suc}_\phi(Z)$, and any $1 \leq i_1 < i_2 < \cdots < i_p \leq k$, where

$$Z_{i_j} = \{d_{i_j+1},\ \ldots,\ d_{i_{j+1}-1}\},$$

for all $j \in \{1,\ \ldots,\ p\}$;

3. If $c\phi = \varepsilon$, then $[c,\ Z] = F$.

Note that Φ_ϕ is finite for all $\phi \in B$. Let $\bar{\mathcal{R}}$ be the regular language of endomorphisms of D^* obtained from C by replacing each ϕ with the finite set Φ_ϕ. Let $\theta \in \text{End}(D^*)$ be defined by $[a,\ \varnothing]\theta = a$, for all $a \in \Sigma$. We take the rational control of $\mathcal{G}$ to be $\bar{\mathcal{R}}\theta$.

First note that $\mathcal{G}$ is indeed an EPDT0L system, as neither θ nor an endomorphism in any Φ_ϕ maps anything to ε. The symbol F is used as a 'fail symbol'. That is to say, if $\phi \in \text{End}(D^*)$ is such that $[\bot,\ \varnothing]\phi = \sigma F \tau$, then all for all $\psi \in \text{End}(D^*)$, the word $[\bot,\ \varnothing]\phi\psi$ will contain the letter F, and thus will not be accepted by $\mathcal{G}$.

The idea of the construction is to delete any 'branches' of a derivation in $\mathcal{H}$ that do not contribute to the word being generated. To do this, we record the minimal alphabet of the branches we 'plan' on deleting within the letters of our extended alphabet, that is, the Z within the letter $[c,\ Z]$, and only when all the letters in Z have been deleted by $\mathcal{H}$, that is $Z = \varnothing$, can we replace our letters $[a,\ Z]$ for $a \in \Sigma$ with terminal letters a. If we every 'try' to delete a letter that was not marked as a letter for deletion, we instead map that letter to F, as we can only accept words that are accepted by $\mathcal{H}$ with deletion by marking the letters that are deleted for deletion first. Note that this method will create many new derivations that accept nothing. □

B.4 The Copying Lemma

Whilst the pre-image of an EDT0L language under a monoid homomorphism is not always an EDT0L language, sometimes this is the case. The Copying Lemma, proved by Ehrenfeucht, Rozenberg and Skyum ([40], [42]), allows this to be done in certain cases. Ciobanu and Elder recently noted that this result also preserves space complexity [19].

Theorem B.4.1 ([40], Theorem 1; [42], Theorem 3.3; [19], Lemma 5.2)**.** *Let Σ_1 and Σ_2 be disjoint alphabets, and $K_1 \subseteq \Sigma_1^*$ and $K_2 \subseteq \Sigma_2^*$ be languages. Let $f \colon K_1 \to K_2$ be a bijection. If*

$$K = \{w(wf) \mid w \in K_1\}$$

is ET0L, then K_1, K_2 and K are all EDT0L.

Moreover, if an ET0L system for K is constructible in $\mathsf{NSPACE}(f)$, where $f \colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$, then there are EDT0L systems for K_1, K_2 and K, all of which are definable in $\mathsf{NSPACE}(f)$.

B.5 DT0L languages

Before we embark on the sequence of proofs used to show that the class of HDT0L languages is equal to the class of EDT0L languages, it is convenient to first define DT0L languages. These are similar to EDT0L languages, except there is no extended alphabet, and the rational control is always expressed in the form B^*, for some finite set of endomorphisms B.

Definition B.5.1. A *DT0L system* is a tuple $(\Sigma,\ w,\ B)$, where

1. Σ is an alphabet;
2. $w \in \Sigma^*$ is called the start word;
3. B is a finite set of endomorphisms of Σ^*.

A *PDT0L system* is a DT0L system $(\Sigma,\ w,\ B)$ such that $a\phi \neq \varepsilon$ for all $a \in \Sigma$, $\phi \in B$.

A language accepted by some (P)DT0L system is called a *(P)DT0L language*.

Remark B.5.2. DT0L systems can be thought of as EDT0L systems, using the definition of EDT0L systems where the rational control is of the form B^* (Theorem 3.3.1), where the alphabet and extended alphabet coincide.

Notation B.5.3. Let Σ be an alphabet, and $w \in \Sigma^*$. We use $\text{alph}(w)$ to denote the set of letters that occur within w.

Lemma B.5.4. *Singleton languages are PDT0L.*

Proof Let $L = \{w\}$ be a singleton language. Then L is accepted by the PDT0L system $(\text{alph}(w),\ w,\ \{\text{id}\})$. □

B.6 HDT0L, NDT0L, WDT0L and CDT0L languages

In a series of papers ([76], [75], [38]) Ehrenfeucht, Nielsen, Rozenberg, Salomaa and Skyum considered whether using an extended alphabet for L-systems had the same effect as applying a homomorphism. They showed that this was the case for ET0L systems; every ET0L language can indeed be obtained as the homomorphic image of a T0L language. For the fact that EDT0L systems are equivalent to the set of languages that can be obtained as homomorphic images of DT0L language, they refer the reader to the analogous proof for ET0L and T0L. We include here a proof of this fact for EDT0L systems using modern notation. We also use most of this proof to consider the addition of the restriction that endomorphisms in the rational control are non-erasing, which we view in the subsequent section.

We start with the definitions of the types of homomorphisms we will be considering, and the corresponding L-systems we obtain using the images of DT0L languages under these homomorphisms.

Definition B.6.1. Let Σ and Δ be alphabets. A homomorphism $\phi \colon \Sigma^* \to \Delta^*$ is called:

1. A *coding* if $a\phi \in \Delta$ for all $a \in \Sigma$;
2. A *weak coding* if $a\phi \in \Delta \cup \{\varepsilon\}$ for all $a \in \Sigma$;
3. A *non-erasing* homomorphism if $a\phi \neq \varepsilon$ for all $a \in \Sigma$.

A language L is called:

1. *C(P)DT0L* if $L = M\theta$ for some (P)DT0L language M, and coding θ;
2. *W(P)DT0L* if $L = M\theta$ for some (P)DT0L language M, and weak coding θ;
3. *N(P)DT0L* if $L = M\theta$ for some (P)DT0L language M, and non-erasing homomorphism θ;
4. *H(P)DT0L* if $L = M\theta$ for some (P)DT0L language M, and homomorphism θ.

We now consider the concept of ultimately periodic sets of non-negative integers. These are sets of integers that are periodic, apart from a finite amount of transient behaviour.

Definition B.6.2. A subset $X \subseteq \mathbb{Z}_{\geq 0}$ is called *ultimately periodic* if there exists $p,\ t,\ i_1,\ \ldots,\ i_k \in \mathbb{Z}_{\geq 0}$ such that $i_j < p$ for all j, and a finite subset $F \subseteq \{0,\ \ldots,\ t-1\}$, such that

$$X = \{t + i_j + pn \mid j \in \{1,\ \ldots,\ k\},\ n \in \mathbb{Z}_{\geq 0}\} \cup F.$$

The smallest integers t and p, such that the above equality holds, are called the *threshold* and *period* of X, and denoted $\mathrm{Thres}(X)$ and $\mathrm{Per}(X)$, respectively.

We now define the spectrum of a finite-state automaton.

Definition B.6.3. Let $\mathcal{A}$ be a finite state automaton, and let q be a state in $\mathcal{A}$. The *spectrum* of q in $\mathcal{A}$, denoted $\mathrm{Spec}(\mathcal{A},\ q)$, is defined by

$$\mathrm{Spec}(\mathcal{A},\ q) = \{|x| \mid x \text{ labels a path in } \mathcal{A} \text{ from } q \text{ to an accept state}\}.$$

The spectrum of a finite-state can be used to show that regular languages are in some ways 'periodic', up to a finite amount of mess.

Lemma B.6.4. *Let $\mathcal{A}$ be a finite state automaton, and let q be a state in $\mathcal{A}$. Then* $\mathrm{Spec}(\mathcal{A},\ q)$ *is ultimately periodic.*

We now separate states of finite-state automata into those with infinite and finite spectra. This is because finite spectra will not have a period.

Definition B.6.5. Let $\mathcal{A}$ be a finite state automaton. A state q of $\mathcal{A}$ is called *weak* if $\mathrm{Spec}(\mathcal{A},\ q)$ is finite, and *strong* otherwise.

To avoid talking about period and threshold too frequently, and to deal with the difference between strong and weak, we combine these notions into the uniform period of a spectrum.

Definition B.6.6. Let $\mathcal{A}$ be a finite state automaton. The *uniform period* of $\mathcal{A}$, denoted $m_{\mathcal{A}}$, is the smallest $j \in \mathbb{Z}_{>0}$, such that

1. For each state q in $\mathcal{A}$, $j > \mathrm{Thres}(\mathrm{Spec}(\mathcal{A},\ q))$;
2. For each strong state q, j is a multiple of $\mathrm{Per}(\mathrm{Spec}(\mathcal{A},\ q))$.

We now define the spectrum of an EPDT0L system, based on our definition of a spectrum of a finite-state automaton.

Definition B.6.7. Let $\mathcal{G} = (\Sigma,\ C,\ \perp,\ B^*)$ be an EPDT0L system, and let $D \subseteq C$ be non-empty. The *spectrum* of D in $\mathcal{G}$, denoted $\mathrm{Spec}(\mathcal{G},\ D)$, is defined by

$$\mathrm{Spec}(\mathcal{G},\ D) = \{|\phi|_B \mid \phi \in B^*,\ \text{where there exists } w \in \Sigma^*,\ \text{with } \mathrm{alph}(w) = D,\ w\phi \in \Sigma^*\}.$$

We now define a finite-state automaton based on an EDT0L system. As we will see, this is constructed so that the spectrum of the finite-state automaton and that of the EDT0L system coincide.

Definition B.6.8. Let $\mathcal{G} = (\Sigma,\ C,\ \perp,\ B^*)$ be an EPDT0L system, and let $D \subseteq C$ be non-empty. The *D-spectral representation* of $\mathcal{G}$, denoted $\mathcal{A}_{\mathcal{G},D}$, is the finite state automaton where

1. $Q = \mathcal{P}(C) \backslash \varnothing$ is the set of states;

2. B is the alphabet;
3. $q_0 = D$ is the start state;
4. $F = \{q \in Q \mid q \subseteq \Sigma\}$ is the set of accept states;
5. For all $q \in Q$ and $\phi \in B$, there is a transition from q to $q\phi$, labelled by ϕ.

The *spectral representation* of $\mathcal{G}$, denoted $\mathcal{A}_{\mathcal{G}}$ is the $\{\perp\}$-spectral representation of $\mathcal{G}$.

Lemma B.6.9. *Let $\mathcal{G} = (\Sigma,\ C,\ \perp,\ B^*)$ be an EPDT0L system, and let $D,\ E \subseteq C$ be non-empty. Then*

$$\mathrm{Spec}(\mathcal{A}_{\mathcal{G},E},\ D) = \mathrm{Spec}(G,\ D).$$

Proof Note that $\mathrm{Spec}(\mathcal{G},\ D)$ is the set of $|\gamma|$ for all $\gamma \in B^*$ such that γ labels a path in $\mathcal{A}_{\mathcal{G},E}$ from D to a subset of Σ. Note that the choice of E only affects the start state of $\mathcal{A}_{\mathcal{G},E}$, which does not affect $\mathrm{Spec}(\mathcal{A}_{\mathcal{G},E},\ D)$. Thus

$$\begin{aligned}\mathrm{Spec}(\mathcal{A}_{\mathcal{G},E},\ D) &= \{|\phi|_B \mid \phi \in B^*,\ D\gamma \subseteq \Sigma\} \\ &= \{|\phi|_B \mid \phi \in B^*,\ \text{where there exists } w \in \Sigma^*,\ \text{with } \mathrm{alph}(w) = D,\ w\phi \in \Sigma^*\} \\ &= \mathrm{Spec}(\mathcal{G},\ D)\end{aligned}$$

□

We now combine the lemmas we have just shown to obtain the fact that spectra of EPDT0L systems are ultimately periodic.

Lemma B.6.10. *Let $\mathcal{G} = (\Sigma,\ C,\ \perp,\ B^*)$ be an EPDT0L system, and let $D \subseteq C$ be non-empty. Then* $\mathrm{Spec}(\mathcal{G},\ D)$ *is ultimately periodic.*

Proof This follows from Lemma B.6.9, together with the fact that the spectrum of a finite state automaton is ultimately periodic with respect to any state (Lemma B.6.4). □

We use the previous lemma to extend the notion of a uniform period to an EPDT0L system.

Definition B.6.11. Let $\mathcal{G}$ be an EPDT0L system. The *uniform period* of $\mathcal{G}$, denoted $m_{\mathcal{G}}$, is defined by $m_{\mathcal{G}} = m_{\mathcal{A}_{\mathcal{G}}}$.

We now define the indexed spectral representation of an EPDT0L system. This is another finite state automaton defined from an EPDT0L system, except we have added indices to the states and the productions in the alphabet, to allow us to track where we are with respect to the uniform period.

Definition B.6.12. Let $\mathcal{G} = (\Sigma,\ C,\ \perp,\ B^*)$ be an EPDT0L system, with spectral representation $\mathcal{A}_{\mathcal{G}} = (Q,\ B,\ \delta,\ q_0,\ F)$. Fix an ordering on Q: $Q = \{u_0,\ \ldots,\ u_p\}$, where $p = |Q| - 1$. For each $\phi \in B$, and $i,\ j \in \{0,\ \ldots,\ p\}$, define $\phi^{i,j} \in \mathrm{End}((C \times \{0,\ \ldots,\ p\}))$ (noting that the letter $(a,\ i) \in C \times \{0,\ \ldots,\ p\}$ will be denoted a^i), by

$$a^i\phi^{i,j} = \begin{cases} b_1^j \cdots b_n^j & a \in u_i \\ a^j & a \notin u_i. \end{cases}$$

where $a\phi = b_1 \cdots b_n$, with $b_l \in \Sigma$ for all l. The *indexed spectral representation* of $\mathcal{G}$, denoted $\mathcal{I}_{\mathcal{G}}$ is the finite state automaton $\mathcal{I}_{\mathcal{G}} = (\bar{Q},\ \bar{B},\ \bar{\delta},\ \bar{q}_0,\ \bar{F})$, where

1. $\bar{Q} = \{u_i \times \{i\} \mid i \in \{0,\ \ldots,\ p\}\}$;
2. $\bar{B} = \{\phi^{i,j} \mid \phi \in B,\ i,\ j \in \{0,\ \ldots,\ p\} \cup \{\psi^{i,j} \mid i,\ j \in \{0,\ \ldots,\ p\}\}$;
3. $\bar{q}_0 = q_0 \times \{0\}$ (note $u_0 = q_0$);
4. $\bar{F} = \{u_i \times \{i\} \mid u_i \in F\}$;
5. For all pairs of states $u_i \times \{i\},\ u_j \times \{j\} \in \bar{Q}$ and each $\phi \in B$ such that $u_i\phi = u_j$, there are two transitions from $u_i \times \{i\}$ to $u_j \times \{j\}$, labelled with $\phi^{i,j}$.

We will often refer to the indexed spectral representation of an EPDT0L system $\mathcal{G}$, where we assume some order on the states of $\mathcal{A}_{\mathcal{G}}$ has been fixed.

The indexed spectral representation of an EPDT0L system has the same spectrum as the spectral representation, thus giving that it is uniformly periodic of the same uniform period.

Lemma B.6.13. *Let $\mathcal{G}$ be an EPDT0L system, and let Q be the set of states of the* *spectral representation* $\mathcal{A}_{\mathcal{G}}$*. Then for each* $u_i \in Q$*,*

$$\mathrm{Spec}(\mathcal{A}_{\mathcal{G}},\ u_i) = \mathrm{Spec}(\mathcal{I}_{\mathcal{G}},\ u_i).$$

In particular, $m_{\mathcal{A}_{\mathcal{G}}} = m_{\mathcal{I}_{\mathcal{G}}}$*.*

Notation B.6.14. Let $\mathcal{G} = (\Sigma,\ C,\ \bot,\ B^*)$ be an EPDT0L system accepting a language L, and $\mathcal{A}_{\mathcal{G}} = (Q,\ B,\ \delta,\ q_0,\ F)$ be the spectral representation. Fix an order on $Q = \{u_0,\ \ldots,\ u_p\}$, and let $\mathcal{I}_{\mathcal{G}} = (\bar{Q},\ \bar{B},\ \bar{\delta},\ \bar{q}_0,\ \bar{F})$ be the indexed spectral representation.

Note that $\bigcup_{\bar{q}\in\bar{Q}} \bar{q} \subseteq \Sigma \times \{0,\ \ldots,\ p\}$. Define $A(\mathcal{G},\ k)$, for $k \in \{0,\ \ldots,\ m_{\mathcal{G}}\}$, to be the language of all $a_1^i \cdots a_t^i \in (\bigcup_{\bar{q}\in\bar{Q}})^*$ such that

1. $a_1 \cdots a_t =_\bot \phi$, for some $\phi \in B^*$ such that $|\phi|_B = m_{\mathcal{G}}$;
2. ϕ traces a path in $\mathcal{A}_{\mathcal{G}}$ from q_0 to u_i;
3. $m_{\mathcal{G}} + k \in \mathrm{Spec}(\mathcal{A}_{\mathcal{G}},\ u_i)$.

For every $r,\ s \in \{0,\ \ldots,\ p\}$, and $k \in \{0,\ \ldots,\ m_{\mathcal{G}}\}$, let $B^{r,s,k}$ be the set of all words $\phi^{v_0,v_1} \cdots \phi^{v_{m_{\mathcal{G}}-1},v_{m_{\mathcal{G}}}}$ over $\bar{B}$ of length $m_{\mathcal{G}}$, such that $r = v_0$ and $s = v_{m_{\mathcal{G}}}$. Let

$$B^k = \bigcup_{\substack{r,s\in\{0,\ldots,p\} \\ m_{\mathcal{G}}+k\in\mathrm{Spec}(\mathcal{A}_{\mathcal{G}},u_r)\cap\mathrm{Spec}(\mathcal{A}_{\mathcal{G}},u_s)}} B^{r,s,k}.$$

Let $\hat{C} = \{c^i \mid c \in C,\ i \in \{0,\ \ldots,\ p\}\}$. Define a homomorphism $\theta\colon \hat{C}^* \to C^*$ by $c^i\theta = c$.

Fix $k \in \{0,\ \ldots,\ m_{\mathcal{G}}\}$. For each $w \in A(\mathcal{G},\ k)$, let $\mathcal{H}_{\mathcal{G},k,w}$ be the PDT0L system $(\hat{C},\ w,\ B^k)$. Now let

$$M_{\mathcal{G},k,w} = \{x \in \Sigma^+ \mid x = y\theta\phi \text{ for some } y \in L(\mathcal{H}_{\mathcal{G},k,w}),\ \phi \in B^* \text{ with } |\phi|_B = m_{\mathcal{G}} + k\}.$$

We have now set up the notation and basic lemmas we need, and we can start the proof that EDT0L languages are CDT0L. We start by showing that languages accepted by EPDT0L systems are the (finite) union of a finite language with the languages $M_{\mathcal{G},k,w}$. We will then go on to prove that these languages themselves are also finite unions of CDT0L languages. At that point, it will be sufficient to show that finite unions of CDT0L languages are CDT0L.

Lemma B.6.15. *Let L be a non-empty EPDT0L language, accepted by an EPDT0L system $\mathcal{G}$. Then*

$$L = F_{\mathcal{G}} \cup \bigcup_{\substack{k \in \{0,\ \dots,\ m_{\mathcal{G}}\} \\ w \in A(\mathcal{G},\ k)}} M_{\mathcal{G},k,w}. \tag{B.1}$$

Proof Let $F_{\mathcal{G}} = \{w \in \Sigma^+ \mid w =\perp \phi \text{ for some } \phi \in B^* \text{ with } |\phi|_B < 2m_{\mathcal{G}}\}$. Note that $F_{\mathcal{G}}$ is finite. By construction,

$$F_{\mathcal{G}} \cup \bigcup_{\substack{k \in \{0,\ \dots,\ m_{\mathcal{G}}\} \\ w \in A(\mathcal{G},\ k)}} M_{\mathcal{G},k,w} \subseteq L.$$

We now show the other direction of containment. Let $u \in L$. If $u \in F_{\mathcal{G}}$, then there is nothing to prove. Otherwise, $u =\perp \phi$ for some $\phi \in B^*$, $|\phi| \geq m_{\mathcal{G}}$. Let $\phi = \psi_1 \cdots \psi_r$, with each $\psi_i \in B$. Write $r = l_r m_{\mathcal{G}} + k_r$, where $l_r \in \mathbb{Z}_{>0}$ and $k_r \in \{0,\ \dots,\ m_{\mathcal{G}} - 1\}$. Let $x_i = x\psi_i$ for every i.

Let $\mathcal{A}_{\mathcal{G}} = (Q,\ B,\ \delta,\ q_0,\ F)$ be the spectral representation of $\mathcal{G}$, and fix an order on $Q = \{u_0,\ \dots,\ u_p\}$. Let $i \in \{1,\ \dots,\ l_r\}$. Note that $x_{im_{\mathcal{G}}}\varphi = x$, for some $\varphi \in B^*$ with $|\varphi|_B = (l_r - i)m_{\mathcal{G}} + k_r$. So $(l_r - i)m_{\mathcal{G}} + k_r \in \operatorname{Spec}(\mathcal{A}_{\mathcal{G}},\ \operatorname{alph}(x_{m_{\mathcal{G}}}))$. Recall that $\operatorname{Spec}(\mathcal{A}_{\mathcal{G}},\ \operatorname{alph}(x_{m_{\mathcal{G}}}))$ is an ultimately periodic set (Lemma B.6.4), with $m_{\mathcal{G}}$ greater than the threshold, and a multiple of the period. Using this, together with the facts that $m_{\mathcal{G}} + k_r \geq m_{\mathcal{G}}$ and $(l_r - i)m_{\mathcal{G}} + k_r \in \operatorname{Spec}(\mathcal{A}_{\mathcal{G}},\ \operatorname{alph}(x_{m_{\mathcal{G}}}))$ gives that $m_{\mathcal{G}} + k_r \in \operatorname{Spec}(\mathcal{A}_{\mathcal{G}},\ \operatorname{alph}(x_{im_{\mathcal{G}}}))$.

By construction, there is a path in $\mathcal{A}_{\mathcal{G}}$ from q_0 to a vertex $u_{j_i} \in Q$, labelled with $\psi_1 \cdots \psi_i$, for all i. We have that

$$u_{j_i} = \operatorname{alph}(\perp)\psi_1 \cdots \psi_i = \operatorname{alph}(\perp \psi_1 \cdots \psi_i) = \operatorname{alph}(x_i).$$

Write $x_{m_{\mathcal{G}}} = c_1 \cdots c_t$ for letters $c_i \in C$. Since $\operatorname{alph}(x_{m_{\mathcal{G}}}) = u_{j_1}$, we now have that $c_1^{j_{m_{\mathcal{G}}}} \cdots c_t^{j_{m_{\mathcal{G}}}} \in A(\mathcal{G},\ k_r)$.

Let $z = c_1^{j_{m_\mathcal{G}}} \cdots c_t^{j_{m_\mathcal{G}}}$, and write $x_{(l_r-1)m_\mathcal{G}} = d_1 \cdots d_s$. Note that $c_1 \cdots c_t \phi = x_{(l_r-1)m_\mathcal{G}}$, for some $\phi \in B^*$ with $|\phi|_B = (l_r - 2)m_\mathcal{G}$. We can therefore write $\phi = \varphi_1 \cdots \varphi_{l_r-2}$, where each $\varphi_i \in B^{m_\mathcal{G}}$. Moreover, since $u_{j_i} = \mathrm{alph}(x_i)$ for all i, we have that $z\varphi_1^{j_1,j_2} \cdots \varphi_{l_r-2}^{j_{l_r-2},j_{l_r-1}} = d_1^{j_{l_r-1}} \cdots d_s^{j_{l_r-1}}$.

We now show that $d_1^{j_{l_r-1}} \cdots d_s^{j_{l_r-1}} \in L(\mathcal{H}_{\mathcal{G},k_r,w}))$. It suffices to show that each $\varphi_i^{j_i,j_{i+1}} \in B^{k_r}$. We have already shown that $m_\mathcal{G} + k_r \in \mathrm{Spec}(\mathcal{A}_\mathcal{G},\ \mathrm{alph}(x_{im_\mathcal{G}}))$ for all i, and by definition, $\varphi_i^{j_i,j_{i+1}} \in B^{j_i,j_{i+1},k_r}$.

Using this fact, together with the fact that there exists $\phi \in B^*$ with $|\phi|_B = m_\mathcal{G} + k_r$, such that

$$(d_1^{j_{l_r-1}} \cdots d_s^{j_{l_r-1}})\theta\phi = x_{(l_r-1)m_\mathcal{G}}\phi = x,$$

and so $x \in M_{\mathcal{G},k_r,w}$, as required. □

We now show that the languages $M_{\mathcal{G},k,w}$ are indeed finite unions of CDT0L languages.

Lemma B.6.16. *Let L be a non-empty EPDT0L language, accepted by an EPDT0L system $\mathcal{G}$. Let $k \in \{0,\ \ldots,\ m_\mathcal{G}\}$, and $w \in A(\mathcal{G},\ k)$. Then $M_{\mathcal{G},k,w}$ is a finite union of CDT0L languages.*

Proof For each non-empty $D \subseteq C$, and $i \in \mathbb{Z}_{\geq 0}$, let $\Phi_{D,i} = \{\phi \in B^* \mid |\phi|_B = m_\mathcal{G} + i,\ D\phi \subseteq \Sigma\}$. Let

$$Z = \{c^{i,\tau,a} \mid c^i \in B^k,\ \tau \in \Phi_D,\ a \in \Sigma\} \cup \{\mathbf{c}^{i,\tau,a} \mid c^i \in B^k,\ \tau \in \Phi_{D,k},\ a \in \Sigma\},$$

where the bold versions are distinct copies of each $c^{i,\tau,a}$. Let $\phi \in B^k$. Then $\phi \in B^{r,s,k}$ for some $r,\ s \in \{0,\ \ldots,\ p\}$. For each $\tau \in \Phi_{D,k}$, define $\phi^\tau \in \mathrm{End}(Z^*)$ by

$$\mathbf{c}^{r,\rho,a}\phi^\tau = d_1^{s,\tau,b_{11}} d_2^{s,\tau,b_{12}} \cdots \mathbf{d}_1^{s,\tau,b_{1n_1}} d_2^{s,\tau,b_{21}} d_2^{s,\tau,b_{22}} \cdots \mathbf{d}_2^{s,\tau,b_{2n_2}} \cdots d_v^{s,\tau,b_{v1}} d_v^{s,\tau,b_{v2}} \cdots \mathbf{d}_v^{s,\tau,b_{vn_v}}$$

$$c^{r,\rho a}\phi^\tau = \varepsilon,$$

where $c\phi = d_1 \cdots d_v$, $d_1\tau = b_{11} \cdots b_{1n_1},\ \ldots,\ d_v\tau = b_{v1} \cdots b_{vn_v}$.

Write $w = e_1^i \cdots e_g^i$. Let

$$W(w) = \{e_1^{i,\tau,b_{11}} e_2^{i,\tau,b_{12}} \cdots \mathbf{e}_1^{s,\tau,b_{1n_1}} \cdots e_g^{s,\tau,b_{g1}} e_g^{s,\tau,b_{g2}} \cdots \mathbf{e}_g^{s,\tau,b_{gn_g}} \mid \tau \in \Phi_{u_i,m_{\mathcal{G}}+k},\ b_{j1} \cdots b_{jn_j} = e_j\tau\}.$$

Define

$$\mathscr{B} = \{\phi^\tau \mid\ r,\ s \in \{0,\ \ldots,\ p\},\ \phi \in B^{r,s,k},\ \tau \in \Phi_{u_s,k}\}.$$

For each $y \in W(w)$, let $\mathcal{E}_{k,w,y}$ be the DT0L system $(Z,\ y,\ \mathscr{B})$. Let $\eta\colon Z^* \to \Sigma^*$ be the coding defined by $(c^{i,\tau,a})\eta = (\mathbf{c}^{i,\tau,a})\eta = a$. Note that $W(w)$ is finite. It therefore suffices to show that

$$M_{\mathcal{G},k,w} = \bigcup_{y \in W(w)} L(\mathcal{E}_{k,w,y})\eta.$$

Let $x \in L(\mathcal{E}_{k,w,y})$ for some $y \in W(w)$. Write $x = a_1 \cdots a_n$, where every $a_i \in \Sigma$. Then there exists $x = z\eta$, for some $z = (c_1^{i_1,\tau,a_{11}} \cdots \mathbf{c}_1^{i_1,\tau,a_{1n_1}} \cdots c_v^{i_v,\tau,a_{v1}} \cdots \mathbf{c}_v^{i_v,\tau,a_{vn_v}}) \in Z^+$. If $z = y$, then $z \in L(\mathcal{H}_{\mathcal{G},k,w})$, by the construction of $W(w)$. Otherwise, $c_i\tau = a_i$, for all i, and $|\tau| = m_{\mathcal{G}} + k$. In order to show $x \in M_{\mathcal{G},k,w}$, it therefore suffices to show that $c_1^{i_1} \cdots c_n^{i_n} \in L(\mathcal{H}_{\mathcal{G},k,w})$. But $c_1^{i_1} \cdots c_n^{i_n} = w\phi$, for some $\phi \in B^k$, as required.

Now suppose $x \in M_{\mathcal{G},k,w}$. Then $x = z\theta\phi$ for some $z \in L(\mathcal{H}_{\mathcal{G},k,w})$, $\phi \in B^*$ with $|\phi|_B = m_{\mathcal{G}} + k$, where $\theta\colon \hat{C}^* \to C^*$ is the projection map. Thus there exists $\psi \in B^k$ such that $w\psi = z$. If $|\psi|_{B^k} = 0$, then $z = w$, and we can simply choose $y = e_1^{i,\tau,b_{11}} e_2^{i,\tau,b_{12}} \cdots \mathbf{e}_1^{s,\tau,b_{1n_1}} \cdots e_g^{s,\tau,b_{g1}} e_g^{s,\tau,b_{g2}} \cdots \mathbf{e}_g^{s,\tau,b_{gn_g}} \in W(w)$ such that $b_{11} \cdots b_{gn_g} = x$, $x \in L(\mathcal{E}_{k,w,y})\eta$.

Otherwise, pick any $y \in W(w)$, and write $\psi = \psi_1 \cdots \psi_t$ with each $\psi_i \in B^{r_i,s_i,k}$, noting that $t \geq 1$. For each $i \in \{1,\ \ldots,\ t-1\}$, choose any $\tau_i \in \Phi_{u_{s_i},k}$. Let $\tau_t = \phi$. Then

$$y\psi_1^{\tau_1} \cdots \psi_t^{\tau_t} = c_1^{s,\tau_t,a_{11}} \cdots \mathbf{c}_1^{s,\tau_t,a_{1n_1}} \cdots c_v^{s,\tau_t,a_{v1}} \cdots \mathbf{c}_v^{s,\tau_t,a_{vn_v}},$$

where $c_1 \cdots c_n = w\psi = z\theta$, $a_{11} \cdots a_{vn_v} = z\theta\tau_t = z\theta\phi = x$. Thus $y\psi_1^{\tau_1} \cdots \psi_t^{\tau_t}\eta = x$, and $x \in L(\mathcal{E}_{k,w,y})\eta$, as required. □

Combining our previous lemmas gives the following.

Lemma B.6.17. *A non-empty EPDT0L language is a finite union of CDT0L languages.*

Proof Using the fact that singletons are CDT0L (Lemma B.5.4), the result follows from Lemma B.6.15 and Lemma B.6.16. □

In order to show that EDT0L languages are CDT0L, it remains to show the following.

Lemma B.6.18. *Finite unions of CDT0L languages are CDT0L.*

Proof Let L and M be CDT0L languages over an alphabet Σ. Suppose $L \cup M \neq \varnothing$. There exist DT0L languages K_L and K_M, accepted by DT0L systems $(C_L,\ \omega_L,\ B_L)$ and $(C_M,\ \omega_M,\ B_M)$, respectively, together with codings θ_L and θ_M, such that $L = K_L\theta_L$ and $M = K_M\theta_M$. By replacing C_M with a copy of itself that is disjoint from C_L, and updating ϕ_M accordingly, we can assume that C_L and C_M are disjoint.

Let $\perp \notin C_L \cup C_M$. Let $C = \{\perp\} \cup C_L \cup C_M$. For each $\psi \in B_L$, define $\bar{\psi} \in \mathrm{End}(C^*)$ by

$$c\bar{\psi} = \begin{cases} c\psi & c \in C_L \\ c & c \notin C_L. \end{cases}$$

Define $\bar{\psi}$ for each $\psi \in C_M$ analogously. Define $\phi_L,\ \phi_M \in \mathrm{End}(C^*)$ by

$$c\phi_L = \begin{cases} \omega_L & c = \perp \\ c & c \neq \perp, \end{cases} \qquad c\phi_M = \begin{cases} \omega_M & c = \perp \\ c & c \neq \perp\ . \end{cases}$$

Let $B = \{\bar{\psi} \mid \psi \in B_L \cup B_M\} \cup \{\phi\}$. Note that $L \neq \varnothing$, and so there exists $u \in L$. Define the coding $\theta \colon C^* \to \Sigma^*$ by

$$c\theta = \begin{cases} c\theta_L & c \in C_L \\ c\theta_M & c \in C_M \\ u & c = \perp\ . \end{cases}$$

Consider the DT0L system $\mathcal{H} = (C,\ \perp,\ B)$, and let K be the language it accepts. By construction, $K\theta = L \cup M$. □

The fact that codings are stronger than weak codings, non-erasing homomorphisms and homomorphisms allows us to show that that the classes listed are all equal.

Theorem B.6.19 ([76], Theorems 6.2-6.5). *Let L be a non-empty language. Then the following are equivalent:*

1. *L is EDT0L;*
2. *L is CDT0L;*
3. *L is WDT0L;*
4. *L is NDT0L;*
5. *L is HDT0L.*

Proof (1) ⇒ (2): Using the fact that the class of CDT0L languages is closed under finite unions (Lemma B.6.18), together with the fact that $\{\varepsilon\}$ is CDT0L (Lemma B.5.4), it suffices to show that $L\backslash\{\varepsilon\}$ is CDT0L. By Theorem B.3.1, $L\backslash\{\varepsilon\}$ is EPDT0L. So by Lemma B.6.17, $L\backslash\{\varepsilon\}$ is a finite union of CDT0L languages. The result now follows by Lemma B.6.18.

(2) ⇒ (3): This follows as codings are weak codings.

(3) ⇒ (1): Since DT0L languages are EDT0L, this follows by the fact that the class of EDT0L languages is closed under applying a homomorphism (Theorem 3.3.2 (5)).

(2) ⇒ (4): This follows as codings are non-erasing homomorphisms.

(4) ⇒ (5): This follows as non-erasing homomorphisms are homomorphisms.

(5) ⇒ (1). This also follows from Theorem 3.3.2 (5). □

B.7 WPDT0L and HPDT0L languages

We now consider what can be done if we want to assume our HDT0L systems have only non-erasing endomorphisms. We have seen that with EDT0L systems, this does not affect the expressive power. With NDT0L, and thus HDT0L, it does work. The author does not believe it would be difficult to modify this proof to show that it also works for WDT0L systems, however CDT0L systems appear harder. It may require a different approach in order to determine whether or not the addition of the assumption that endomorphisms are non-erasing affects the class of languages that CDT0L systems accept.

Whilst these classes of languages have not been used when studying equations so far, they have the potential to be used to show some solutions to equations in groups cannot be expressed as EDT0L languages, as their more rigid structure is easier to work with to show languages are not EDT0L, than the structure of EDT0L systems.

We begin by reproving Lemma B.6.16, except for WPDT0L languages, rather than CDT0L languages.

Lemma B.7.1. *Let L be a non-empty EPDT0L language, accepted by an EPDT0L system $\mathcal{G}$. Let $k \in \{0, \ldots, m_{\mathcal{G}}\}$, and $w \in A(\mathcal{G}, k)$. Then $M_{\mathcal{G},k,w}$ is a finite union of WPDT0L languages.*

Proof For each non-empty $D \subseteq C$, and $i \in \mathbb{Z}_{\geq 0}$, let $\Phi_{D,i} = \{\phi \in B^* \mid |\phi|_B = m_{\mathcal{G}} + i,\ D\phi \subseteq \Sigma\}$. Let

$$Z = \{c^{i,\tau,a} \mid c^i \in B^k,\ \tau \in \Phi_D,\ a \in \Sigma\} \cup \{\mathbf{c}^{i,\tau,a} \mid c^i \in B^k,\ \tau \in \Phi_{D,k},\ a \in \Sigma\} \cup \{\kappa\},$$

where the bold versions are distinct copies of each $c^{i,\tau,a}$. Let $\phi \in B^k$. Then $\phi \in B^{r,s,k}$ for some $r,\ s \in \{0, \ldots, p\}$. For each $\tau \in \Phi_{D,k}$, define $\phi^\tau \in \mathrm{End}(Z^*)$ by

$$\mathbf{c}^{r,\rho,a}\phi^\tau = d_1^{s,\tau,b_{11}} d_2^{s,\tau,b_{12}} \cdots \mathbf{d}_1^{s,\tau,b_{1n_1}} d_2^{s,\tau,b_{21}} d_2^{s,\tau,b_{22}} \cdots \mathbf{d}_2^{s,\tau,b_{2n_2}} \cdots d_v^{s,\tau,b_{v1}} d_v^{s,\tau,b_{v2}} \cdots \mathbf{d}_v^{s,\tau,b_{vn_v}}$$

$$c^{r,\rho a}\phi^\tau = \kappa\phi^\tau = \kappa,$$

where $c\phi = d_1 \cdots d_v$, $d_1\tau = b_{11} \cdots b_{1n_1},\ \ldots,\ d_v\tau = b_{v1} \cdots b_{vn_v}$.

Write $w = e_1^i \cdots e_g^i$. Let

$$W(w) = \{e_1^{i,\tau,b_{11}} e_2^{i,\tau,b_{12}} \cdots \mathbf{e}_1^{s,\tau,b_{1n_1}} \cdots e_g^{s,\tau,b_{g1}} e_g^{s,\tau,b_{g2}} \cdots \mathbf{e}_g^{s,\tau,b_{gn_g}} \mid \tau \in \Phi_{u_i,m_{\mathcal{G}}+k},\ b_{j1} \cdots b_{jn_j} = e_j\tau\}.$$

Define

$$\mathscr{B} = \{\phi^\tau \mid\ r,\ s \in \{0, \ldots, p\},\ \phi \in B^{r,s,k},\ \tau \in \Phi_{u_s,k}\}.$$

For each $y \in W(w)$, let $\mathcal{E}_{k,w,y}$ be the DT0L system $(Z,\ y,\ \mathscr{B})$. Let $\eta \colon Z^* \to \Sigma^*$

be the weak coding defined by $(c^{i,\tau,a})\eta = (\mathbf{c}^{i,\tau,a})\eta = a$, $\kappa\eta = \varepsilon$. Note that $W(w)$ is finite. It therefore suffices to show that

$$M_{\mathcal{G},k,w} = \bigcup_{y \in W(w)} L(\mathcal{E}_{k,w,y})\eta.$$

Let $x \in L(\mathcal{E}_{k,w,y})$ for some $y \in W(w)$. Write $x = a_1 \cdots a_n$, where every $a_i \in \Sigma$. Then there exists $x = z\eta$, for some

$$z = (\kappa^{\delta_0} c_1^{i_1,\tau,a_{11}} \kappa^{\delta_1 1} \cdots \mathbf{c}_1^{i_1,\tau,a_{1n_1}} \kappa^{\delta_{1n_1}} \cdots c_v^{i_v,\tau,a_{v1}} \kappa^{\delta_{v1}} \cdots \mathbf{c}_v^{i_v,\tau,a_{vn_v}} \kappa^{\delta_{vn_v}}) \in Z^+.$$

If $z = y$, then $z \in L(\mathcal{H}_{\mathcal{G},k,w})$, by the construction of $W(w)$. Otherwise, $c_i\tau = a_i$, for all i, and $|\tau| = m_{\mathcal{G}} + k$. In order to show $x \in M_{\mathcal{G},k,w}$, it therefore suffices to show that $c_1^{i_1} \cdots c_n^{i_n} \in L(\mathcal{H}_{\mathcal{G},k,w})$. But $c_1^{i_1} \cdots c_n^{i_n} = w\phi$, for some $\phi \in B^k$, as required.

Now suppose $x \in M_{\mathcal{G},k,w}$. Then $x = z\theta\phi$ for some $z \in L(\mathcal{H}_{\mathcal{G},k,w})$, $\phi \in B^*$ with $|\phi|_B = m_{\mathcal{G}} + k$, where $\theta \colon \hat{C}^* \to C^*$ is the projection map. Thus there exists $\psi \in B^k$ such that $w\psi = z$. If $|\psi|_{B^k} = 0$, then $z = w$, and we can simply choose $y = e_1^{i,\tau,b_{11}} e_2^{i,\tau,b_{12}} \cdots \mathbf{e}_1^{s,\tau,b_{1n_1}} \cdots e_g^{s,\tau,b_{g1}} e_g^{s,\tau,b_{g2}} \cdots \mathbf{e}_g^{s,\tau,b_{gn_g}} \in W(w)$ such that $b_{11} \cdots b_{gn_g} = x$, $x \in L(\mathcal{E}_{k,w,y})\eta$.

Otherwise, pick any $y \in W(w)$, and write $\psi = \psi_1 \cdots \psi_t$ with each $\psi_i \in B^{r_i,s_i,k}$, noting that $t \geq 1$. For each $i \in \{1, \ldots, t-1\}$, choose any $\tau_i \in \Phi_{u_{s_i},k}$. Let $\tau_t = \phi$. Then

$$y\psi_1^{\tau_1} \cdots \psi_t^{\tau_t} = \kappa_0^{\delta} c_1^{s,\tau_t,a_{11}} \kappa_{11}^{\delta} \cdots \mathbf{c}_1^{s,\tau_t,a_{1n_1}} \kappa_{1n_1}^{\delta} \cdots c_v^{s,\tau_t,a_{v1}} \kappa_{v1}^{\delta} \cdots \mathbf{c}_v^{s,\tau_t,a_{vn_v}} \kappa_{vn_v}^{\delta},$$

where $c_1 \cdots c_n = w\psi = z\theta$, $a_{11} \cdots a_{vn_v} = z\theta\tau_t = z\theta\phi = x$, $\delta_0, \delta_{11}, \ldots, \delta_{vn_v} \in \mathbb{Z}_{\geq 0}$. Thus $y\psi_1^{\tau_1} \cdots \psi_t^{\tau_t}\eta = x$, and $x \in L(\mathcal{E}_{k,w,y})\eta$, as required. $\square$

Using the results from Section B.6, we can now show that EPDT0L languages are finite unions of WPDT0L languages.

Lemma B.7.2. *A non-empty EPDT0L language is a finite union of WPDT0L languages.*

Proof Using the fact that singletons are WPDT0L (Lemma B.5.4), the result follows from Lemma B.6.15 and Lemma B.7.1. □

It now remains to show that finite unions of WPDT0L languages are WPDT0L.

Lemma B.7.3. *Finite unions of WPDT0L languages are WPDT0L.*

Proof Let L and M be WPDT0L languages over an alphabet Σ. Suppose $L \cup M \neq \varnothing$. There exist PDT0L languages K_L and K_M, accepted by PDT0L systems $(C_L,\ \omega_L,\ B_L)$ and $(C_M,\ \omega_M,\ B_M)$, respectively, together with weak codings θ_L and θ_M, such that $L = K_L\theta_L$ and $M = K_M\theta_M$. By replacing C_M with a copy of itself that is disjoint from C_L, and updating ϕ_M accordingly, we can assume that C_L and C_M are disjoint.

Let $\perp \notin C_L \cup C_M$ and $C = \{\perp\} \cup C_L \cup C_M$. For all $\psi \in B_L$, define $\bar{\psi} \in \mathrm{End}(C^*)$ by

$$c\bar{\psi} = \begin{cases} c\psi & c \in C_L \\ c & c \notin C_L. \end{cases}$$

Define $\bar{\psi}$ for each $\psi \in C_M$ analogously. Define $\phi_L,\ \phi_M \in \mathrm{End}(C^*)$ by

$$c\phi_L = \begin{cases} \omega_L & c = \perp \\ c & c \neq \perp, \end{cases} \qquad c\phi_M = \begin{cases} \omega_M & c = \perp \\ c & c \neq \perp\ . \end{cases}$$

Let $B = \{\bar{\psi} \mid \psi \in B_L \cup B_M\} \cup \{\phi\}$. Note that $L \neq \varnothing$, and so there exists $u \in L$. Define the coding $\theta \colon C^* \to \Sigma^*$ by

$$c\theta = \begin{cases} c\theta_L & c \in C_L \\ c\theta_M & c \in C_M \\ u & c = \perp\ . \end{cases}$$

Consider the DPT0L system $\mathcal{H} = (C,\ \perp,\ B)$, and let K be the language it accepts. By construction, $K\theta = L \cup M$. □

As with Section B.6, we use the fact that non-erasing homomorphisms are homomorphisms to show that the classes of EDT0L, WPDT0L and HPDT0L are all equal.

Theorem B.7.4. *Let L be a non-empty language. Then the following are equivalent:*

1. *L is EDT0L;*
2. *L is WPDT0L;*
3. *L is HPDT0L.*

Proof (1) ⇒ (2): Using the fact that the class of CDT0L languages is closed under finite unions (Lemma B.7.3), together with the fact that $\{\varepsilon\}$ is WPDT0L (Lemma B.5.4), it suffices to show that $L\backslash\{\varepsilon\}$ is CDT0L. By Theorem B.3.1, $L\backslash\{\varepsilon\}$ is EPDT0L. So by Lemma B.7.2, $L\backslash\{\varepsilon\}$ is a finite union of WPDT0L languages. The result now follows by Lemma B.7.3.

(2) ⇒ (3): This follows as weak codings are homomorphisms.

(3) ⇒ (1): Since PDT0L languages are EDT0L, this follows by the fact that the class of EDT0L languages is closed under applying a homomorphism (Theorem 3.3.2 (5)). □

B.8 ET0L languages and indexed languages

The fact that ET0L languages are indexed was noted in works of Christensen and Salomaa in 1974 ([16], [88]), and seemingly first proved by Ehrenfeucht, Rozenberg and Skyum that year [42]. Also in the same year, Salomaa conjectured that this containment is proper [88], a fact that was proved by Ehrenfeucht, Rozenberg and Skyum ([42], [39]). We give our own argument below.

Theorem B.8.1. *ET0L languages are indexed.*

Proof Let L be an ET0L language accepted by an ET0L system $(\Sigma,\ C,\ \perp,\ B^*)$; we use Theorem 3.3.1 to assume that this ET0L system has a single-letter start word, and rational control of the form B^*. We will use this to define an indexed grammar for L.

We will define an indexed grammar to accept L. Our set of non-terminals will be $V = \{\mathbf{S},\ \mathbf{T}\} \sqcup \bar{C}$, where $\bar{C} = \{\bar{c} \mid c \in C\}$ is a disjoint copy of C. We extend the action of the tables in B on C to $\bar{C}$. Our start symbol will be $\mathbf{S}$. Our alphabet of indices will be $\chi = B \cup \{\$\}$, where \$ will be used as an end of stack marker. Our set of productions $\mathcal{P}$ will be:

$$\mathbf{S} \to \mathbf{T}_\$ \qquad \bar{c}_\phi \to \bar{w} \text{ for all } \bar{w} \in \overline{c\phi},\ \bar{c} \in \bar{C},\ \phi \in B$$

$$\mathbf{T} \to \mathbf{T}_\phi \text{ for all } \phi \in B \qquad \bar{c}_\$ \to c \text{ for all } c \in \Sigma$$

$$\mathbf{T} \to \bar{\perp}$$

Let $\mathcal{G} = (V,\ \Sigma,\ \mathcal{P},\ \mathbf{S})$ be an indexed grammar. We will show that $\mathcal{G}$ accepts L. For any word $w \in L$, we have that $w \in \perp \phi_1 \cdots \phi_n$ for some $\phi_1,\ \ldots,\ \phi_n \in B$. Thus we can use the sequence of productions

$$\mathbf{S} \to \mathbf{T}_\$ \to \mathbf{T}_{\$\phi_1} \to \cdots \to \mathbf{T}_{\$\phi_1 \cdots \phi_n} \to \bar{\perp}_{\$\phi_1 \cdots \phi_n}.$$

After this, we can pop the ϕ_n off the word to obtain a word in $\bar{\perp}\phi_n$, where every letter will have the index $\$\phi_1 \cdots \phi_{n-1}$. Proceeding in this manner to pop all of the $\phi_1 \cdots \phi_{n-1}$ from all of the letters will yield any choice of a word in $\bar{\perp}\phi_1 \cdots \phi_n$, where every letter has the flag \$. Since $\bar{w} \in \bar{\perp}\phi_1 \cdots \phi_n$, we can choose a sequence of productions to yield $\bar{w}$, where every letter is indexed using \$. We then apply the production $\bar{c}_\$ \to c$ to each letter in $\bar{w}$ to yield w.

Conversely, any word w accepted by $\mathcal{G}$ will have to satisfy $\bar{w} \in \bar{\perp}\phi_1 \cdots \phi_n$ for some $\phi_1,\ \ldots \phi_n \in B$. Thus $w \in L$. □

The fact that there exist indexed was originally proved by Rozenberg, Ehrenfeucht and Skyum (([42], [39])). We give an example here.

Theorem B.8.2 ([42], Theorem 3)**.** *Let K be a language over an alphabet Σ that is context-free but not EDT0L. Let $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$ be a distinct copy of Σ. Then*

$$M_K = \{w\bar{w} \mid w \in K\}$$

is indexed but not ET0L.

It was shown in [20] that the word problem of any free group of rank at least 2 is not EDT0L. As this language is context-free, we now have an example of a language that is indexed but not ET0L.

Bibliography

[1] J. Adams, E. Freden, and M. Mishna. From indexed grammars to generating functions. *RAIRO Theor. Inform. Appl.*, 47(4):325–350, 2013.

[2] A. V. Aho. Indexed grammars—an extension of context-free grammars. *J. Assoc. Comput. Mach.*, 15:647–671, 1968.

[3] A. V. Aho. Nested stack automata. *J. Assoc. Comput. Mach.*, 16:383–406, 1969.

[4] T. Andreescu and D. Andrica. *Quadratic Diophantine equations*, volume 40 of *Developments in Mathematics*. Springer, New York, 2015. With a foreword by Preda Mihăilescu.

[5] A. V. Anisimov. Group languages. *Cybernetics and Systems Analysis*, 7:594–601, 1971.

[6] P. Asveld. *A Characterization of ET0L and EDT0L Languages*. Number 129 in Memorandum / Department of Applied Mathematics. University of Twente, Department of Applied Mathematics, 1976.

[7] P. R. J. Asveld. Controlled iteration grammars and full hyper-AFL's. *Information and Control*, 34(3):248–269, 1977.

[8] M. Benson. Growth series of finite extensions of $\mathbf{Z}^n$ are rational. *Invent. Math.*, 73(2):251–269, 1983.

[9] A. Bishop. Geodesic growth in virtually abelian groups. *J. Algebra*, 573:760–786, 2021.

[10] A. Bishop and M. Elder. Bounded automata groups are co-ET0L. *arXiv e-prints*, 2018. arXiv:1811.10157.

[11] A. Bishop and M. 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.

[12] C. Bleak, F. Matucci, and M. Neunhöffer. Embeddings into Thompson's group V and $co\mathcal{CF}$ groups. *J. Lond. Math. Soc. (2)*, 94(2):583–597, 2016.

[13] O. Bogopolski, A. Martino, and E. Ventura. Orbit decidability and the conjugacy problem for some extensions of groups. *Trans. Amer. Math. Soc.*, 362(4):2003–2036, 2010.

[14] T. Brough, L. Ciobanu, M. Elder, and G. Zetzsche. Permutations of context-free, ET0L and indexed languages. *Discrete Math. Theor. Comput. Sci.*, 17(3):167–178, 2016.

[15] N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In *Computer programming and formal systems*, pages 118–161. North-Holland, Amsterdam, 1963.

[16] P. A. Christensen. Hyper-AFL's and ET0L systems. In *L systems (Third Open House, Comput. Sci. Dept., Aarhus Univ., Aarhus, 1974)*, pages 254–257, 327–338. Lecture Notes in Comput. Sci., Vol. 15. 1974.

[17] L. Ciobanu, V. Diekert, and M. Elder. Solution sets for equations over free groups are EDT0L languages. *Internat. J. Algebra Comput.*, 26(5):843–886, 2016.

[18] L. Ciobanu and M. Elder. Solutions sets to systems of equations in hyperbolic groups are EDT0L in PSPACE. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 110, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.

[19] L. Ciobanu and M. Elder. The complexity of solution sets to equations in hyperbolic groups. *Israel J. Math.*, 245(2):869–920, 2021.

[20] L. Ciobanu, M. Elder, and M. Ferov. Applications of L systems to group theory. *Internat. J. Algebra Comput.*, 28(2):309–329, 2018.

[21] L. Ciobanu, S. Hermiller, D. Holt, and S. Rees. Conjugacy languages in groups. *Israel J. Math.*, 211(1):311–347, 2016.

[22] L. Ciobanu, D. Holt, and S. Rees. Equations in groups that are virtually direct products. *J. Algebra*, 545:88–99, 2020.

[23] A. E. Clement, S. Majewicz, and M. Zyman. *The theory of nilpotent groups*. Birkhäuser/Springer, Cham, 2017.

[24] R. Cluckers, J. Gordon, and I. Halupczok. Integrability of oscillatory functions on local fields: transfer principles. *Duke Math. J.*, 163(8):1549–1600, 2014.

[25] M. Cordes, J. Russell, D. Spriano, and A. Zalloum. Regularity of Morse geodesics and growth of stable subgroups, 2020. arXiv:2008.06379.

[26] F. Dahmani. Existential questions in (relatively) hyperbolic groups. *Israel J. Math.*, 173:91–124, 2009.

[27] F. Dahmani and V. Guirardel. Foliations for solving equations in groups: free, virtually free, and hyperbolic groups. *J. Topol.*, 3(2):343–404, 2010.

[28] T. C. Davis and A. Y. Olshanskii. Relative subgroup growth and subgroup distortion. *Groups Geom. Dyn.*, 9(1):237–273, 2015.

[29] J. Denef. The rationality of the Poincaré series associated to the p-adic points on a variety. *Invent. Math.*, 77(1):1–23, 1984.

[30] V. Diekert. More than 1700 years of word equations. In *Algebraic informatics*, volume 9270 of *Lecture Notes in Comput. Sci.*, pages 22–28. Springer, Cham, 2015.

[31] V. Diekert and M. Elder. Solutions to twisted word equations and equations in virtually free groups. *Internat. J. Algebra Comput.*, 30(4):731–819, 2020.

[32] V. Diekert, A. Jeż, and M. Kufleitner. Solutions of word equations over partially commutative structures. In *43rd International Colloquium on Automata,*

Languages, and Programming, volume 55 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 127, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016.

[33] V. Diekert and A. Muscholl. Solvability of equations in free partially commutative groups is decidable. In *Automata, languages and programming*, volume 2076 of *Lecture Notes in Comput. Sci.*, pages 543–554. Springer, Berlin, 2001.

[34] M. Duchin, H. Liang, and M. Shapiro. Equations in nilpotent groups. *Proc. Amer. Math. Soc.*, 143(11):4723–4731, 2015.

[35] M. Duchin and M. Shapiro. The Heisenberg group is pan-rational. *Adv. Math.*, 346:219–263, 2019.

[36] A. Duncan, A. Evetts, D. Holt, and S. Rees. Using EDT0L systems to solve equations in the solvable Baumslag-Solitar groups. *arXiv e-prints*, arXiv:2204.13758, 2022.

[37] M. J. Dunwoody. The accessibility of finitely presented groups. *Invent. Math.*, 81(3):449–457, 1985.

[38] A. Ehrenfeucht and G. Rozenberg. Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems. *Acta Informat.*, 3:265–283, 1973/74.

[39] A. Ehrenfeucht and G. Rozenberg. On decomposing some etol languages into deterministic etol languages. 1974. University of Colorado Technical Report.

[40] A. Ehrenfeucht and G. Rozenberg. On inverse homomorphic images of deterministic ETOL languages. In *Automata, languages, development*, pages 179–189. 1976.

[41] 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.

[42] A. Ehrenfeucht, G. Rozenberg, and S. Skyum. A relationship between ETOL and EDTOL languages. *Theoret. Comput. Sci.*, 1(4):325–330, 1975/76.

[43] S. Eilenberg and M. P. Schützenberger. Rational sets in commutative monoids. *J. Algebra*, 13:173–191, 1969.

[44] J. L. Eršov. Elementary group theories. *Dokl. Akad. Nauk SSSR*, 203:1240–1243, 1972.

[45] A. Evetts. Rational growth in virtually abelian groups. *Illinois J. Math.*, 63(4):513–549, 2019.

[46] A. Evetts. *Aspects of growth in finitely generated groups.* PhD thesis, Heriot-Watt University, 2020.

[47] A. Evetts and A. Levine. Equations in virtually abelian groups: Languages and growth. *Internat. J. Algebra Comput.*, 32(3):411–442, 2022.

[48] D. Farley. Local similarity groups with context-free co-word problem. In *Topological methods in group theory*, volume 451 of *London Math. Soc. Lecture Note Ser.*, pages 67–91. Cambridge Univ. Press, Cambridge, 2018.

[49] P. Flajolet and R. Sedgewick. *Analytic combinatorics.* Cambridge University Press, Cambridge, 2009.

[50] A. Garreta, A. Miasnikov, and D. Ovchinnikov. Random nilpotent groups, polycyclic presentations, and Diophantine problems. *Groups Complex. Cryptol.*, 9(2):99–115, 2017.

[51] R. H. Gilman. Formal languages and their application to combinatorial group theory. In *Groups, languages, algorithms*, volume 378 of *Contemp. Math.*, pages 1–36. Amer. Math. Soc., Providence, RI, 2005.

[52] Z. Grunschlag. *Algorithms in geometric group theory.* PhD thesis, 1999.

[53] T. Herbst and R. M. Thomas. Group presentations, formal languages and characterizations of one-counter groups. *Theoret. Comput. Sci.*, 112(2):187–213, 1993.

[54] D. F. Holt and S. Rees. Regularity of quasigeodesics in a hyperbolic group. *Internat. J. Algebra Comput.*, 13(5):585–596, 2003.

[55] D. F. Holt, S. Rees, and C. E. Röver. Groups with context-free conjugacy problems. *Internat. J. Algebra Comput.*, 21(1-2):193–216, 2011.

[56] D. F. Holt, S. Rees, and C. E. Röver. *Groups, languages and automata*, volume 88 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2017.

[57] D. F. Holt, S. Rees, C. E. Röver, and R. M. Thomas. Groups with context-free co-word problem. *J. London Math. Soc. (2)*, 71(3):643–657, 2005.

[58] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Co., Reading, Mass., 1979.

[59] L. K. Hua. *Introduction to number theory*. Springer-Verlag, Berlin-New York, 1982. Translated from the Chinese by Peter Shiu.

[60] O. Kharlampovich, L. López, and A. Myasnikov. The Diophantine problem in some metabelian groups. *Math. Comp.*, 89(325):2507–2519, 2020.

[61] D. Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, pages 254–266. 1977.

[62] J. Lehnert and P. Schweitzer. The co-word problem for the Higman-Thompson group is context-free. *Bull. Lond. Math. Soc.*, 39(2):235–241, 2007.

[63] H. W. Lenstra, Jr. Solving the Pell equation. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 1–23. Cambridge Univ. Press, Cambridge, 2008.

[64] A. Levine. Solving equations in class 2 nilpotent groups. *arXiv e-prints*, arXiv:2009.10651, 2020.

[65] A. Levine. EDT0L solutions to equations in group extensions. *arXiv e-prints*, arXiv:2108.09390, 2021.

[66] A. Levine. Formal languages, quadratic Diophantine equations and the Heisenberg group. *arXiv e-prints*, arXiv:2203.04849, 2022.

[67] G. S. Makanin. Systems of equations in free groups. *Sibirsk. Mat. Ž.*, 13:587–595, 1972.

[68] G. S. Makanin. The problem of the solvability of equations in a free semigroup. *Mat. Sb. (N.S.)*, 103(145)(2):147–236, 319, 1977.

[69] G. S. Makanin. Equations in a free group. *Izv. Akad. Nauk SSSR Ser. Mat.*, 46(6):1199–1273, 1344, 1982.

[70] A. Mann. *How groups grow*, volume 395 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, Cambridge, 2012.

[71] P. Massazza. Holonomic functions and their relation to linearly constrained languages. *RAIRO Inform. Théor. Appl.*, 27(2):149–161, 1993.

[72] J. V. Matijasevič. The Diophantineness of enumerable sets. *Dokl. Akad. Nauk SSSR*, 191:279–282, 1970.

[73] D. Meuser. On the rationality of certain generating functions. *Math. Ann.*, 256(3):303–310, 1981.

[74] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983.

[75] M. Nielsen, G. Rozenberg, A. Salomaa, and S. Skyum. Nonterminals, homomorphisms and codings in different variations of OL-systems. II. Nondeterministic systems. *Acta Informat.*, 3:357–364, 1973/74.

[76] M. Nielsen, G. Rozenberg, A. Salomaa, and S. Skyum. Nonterminals, homomorphisms and codings in different variations of OL-systems. I. Deterministic systems. *Acta Informat.*, 4:87–106, 1974/75.

[77] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley Publishing Company, Reading, MA, 1994.

[78] M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes-rendus du I Congrès des Mathématiciens des Pays Slaves*, pages 92–101, 1929.

[79] A. A. Razborov. *On systems of equations in free groups*. PhD thesis, 1971. In Russian.

[80] A. A. Razborov. On systems of equations in free groups. In *Combinatorial and geometric group theory (Edinburgh, 1993)*, volume 204 of *London Math. Soc. Lecture Note Ser.*, pages 269–283. Cambridge Univ. Press, Cambridge, 1995.

[81] N. N. Repin. Solvability of equations with one indeterminate in nilpotent groups. *Izv. Akad. Nauk SSSR Ser. Mat.*, 48(6):1295–1313, 1984.

[82] E. Rips and Z. Sela. Canonical representatives and equations in hyperbolic groups. *Invent. Math.*, 120(3):489–512, 1995.

[83] V. A. Roman' kov. Unsolvability of the problem of endomorphic reducibility in free nilpotent groups and in free rings. *Algebra i Logika*, 16(4):457–471, 494, 1977.

[84] V. A. Roman'kov. Diophantine questions in the class of finitely generated nilpotent groups. *J. Group Theory*, 19(3):497–514, 2016.

[85] G. Rozenberg. Extension of tabled OL-systems and languages. *Internat. J. Comput. Information Sci.*, 2:311–336, 1973.

[86] G. Rozenberg and A. Salomaa. *The mathematical theory of L systems*, volume 90 of *Pure and Applied Mathematics*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1980.

[87] G. Rozenberg and A. Salomaa, editors. *Handbook of formal languages. Vol. 1*. Springer-Verlag, Berlin, 1997. Word, language, grammar.

[88] A. Salomaa. Parallelism in rewriting systems. In *Automata, languages and programming (Second Colloq., Univ. Saarbrücken, Saarbrücken, 174)*, pages 523–533. Lecture Notes in Comput. Sci., Vol. 14. 1974.

[89] R. E. Sawilla, A. K. Silvester, and H. C. Williams. A new look at an old equation. In *Algorithmic number theory*, volume 5011 of *Lecture Notes in Comput. Sci.*, pages 37–59. Springer, Berlin, 2008.

[90] M. Shapiro. A note on context-sensitive languages and word problems. *Internat. J. Algebra Comput.*, 4(4):493–497, 1994.

[91] C. L. Siegel. Zur Theorie der quadratischen Formen. *Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl. II*, pages 21–46, 1972.

[92] J. van Leeuwen. Variations of a new machine model. In *17th Annual Symposium on Foundations of Computer Science (Houston, Tex., 1976)*, pages 228–235. 1976.