Systems of bounded arithmetic from descriptive complexity

by

Antonina Kolokolova

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Systems of bounded arithmetic from descriptive complexity

Antonina Kolokolova
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2005

In this thesis we discuss a general method of constructing systems of bounded arithmetic from descriptive complexity logics of known complexity. We discuss the conditions under which the resulting systems capture the same complexity class in the bounded arithmetic setting as the corresponding logic in the descriptive complexity setting. Our method works for small complexity classes (P and below) which have simple proofs of closure under complementation. Additionally, we require proofs of membership and co-membership for instances of decision problems to be constructible within the same complexity class.

More formally, given a logic L capturing complexity class C, the corresponding second-order system V-L of arithmetic consists of a system for $\mathtt{AC}^0$ together with comprehension over L-formulae. If the class is provably in V-L closed under $\mathtt{AC}^0$ reductions and every formula or its (possibly semantic) negation can be witnessed in C, then the resulting system captures C.

Based on this general theorem, we discuss systems of arithmetic for classes P and NL. We also give a system of arithmetic for SL, although the definability theorem for SL is weaker.

Acknowledgements

I am honoured to have done my thesis work in a wonderful atmosphere of the Computer Science department of the University of Toronto. I am tempted to call these years the best years of my life, and it is the theory group, my friends, and the beautiful city itself that made it so.

First, I would like to thank my supervisor, Steve Cook. I am most grateful to him for being an example that was a pleasure to follow, as students do tend to copy their supervisors. Aside from being an amazing researcher, with unparalleled scientific intuition supported by his hard work, he is a perfect supervisor, kind and patient. It is a great pleasure to work with somebody one can admire both as a person and as a scientist.

I am very grateful as well to the rest of my committee and my external examiner Sam Buss for reading my thesis and putting it into a general perspective. I owe a lot to Toni Pitassi – without her I would not be in Toronto to begin with. I am very glad that Toni is in Toronto herself now – I feel that Toronto theory group is an appropriate place for somebody as good and as nice as she is. I am very grateful to Alasdair Urquhart for his vast knowledge that he shared with me and others, and for his ability to make a vague idea into a scientific statement. The theory group would not be what it is without Charlie Rackoff, the sharpness and precision of his thinking – and its applications to both science and life. And luckily for me Leonid Libkin came to the University of Toronto as well during my study: a lot of what I know of finite model theory I learned from him.

I also want to thank Alan Borodin and Faith Fich, as well as the postdocs and the students in the group and in the department – just for being there, being part of it. I am especially grateful to Valentine Kabanets for helping me with everything from my first day in Toronto, and being my closest friend all these years. Many of the results of my thesis came, at least in their final form, from my conversations with Valentine; and his comments on a draft of my thesis improved it a great deal. I am very thankful to my hardworking officemate Tsuyoshi Morioka, and my other "academic siblings" Alan Skelley, Mark Braverman and Phuong Nguyen, Michael Soltys, Francois Pitt and Josh Buresh-Oppenheim. It is always inspiring to be surrounded by people like that. I am grateful to my friends: Natasha Przulj, Wayne Hayes and Travis Gagie, who introduced me to Ontario wilderness; to David Mould and Luis Dissett for books and conversations, to Vlad Kolesnikov, Kleoni Ioannidou, Steve Myers, Dana Rosu and Cristiana Chitic, Matei David, Evgenia Ternovskaya and Dave Mitchell, Lucia Moura, UT Nguyen, Genevieve Arboit and Ani Popivanova; the list can continue...

It would be unfair for me to thank only the Toronto group, though. I learned a great deal from my communication with the rest of scientific community – it is a great inspiration to find out that what one is doing is actually interesting to others. The conferences provide us that window to the community; but more so, the summer schools. I was very lucky to attend the IAS Summer School in Computational Complexity in 2000: this three-week school was not only a chance to get an overall view of the state-of-art complexity theory, but spend time with researchers and students in the area. Another such school was Krajicek's Fall Logic School: it was an opportunity to communicate with European mathematical community, in particular logicians.

Lastly, I want to express my gratitude to the city of Toronto itself. Most cities resent foreigners, but Toronto seems to accept and make feel at home everybody who comes with good intentions, and Torontonians pay back by taking good care of their city. This is the place where one wishes to return...

Contents

Chapter 1

Introduction

Complexity theory, like its precursor computability theory, has its roots in mathematical logic. Throughout the history of these fields, logic has been used to formalize complexity theoretic notions and study them in machine-independent frameworks.

Currently, two major approaches to complexity from a logical perspective are descriptive complexity (finite model theory) and bounded arithmetic; the latter is closely related to proof complexity. There has been intensive research in each of these two areas and their relations to the traditional structural complexity. In particular, the relationship between bounded arithmetic and proof complexity is well-studied. However, little is known about the direct connection between descriptive complexity and bounded arithmetic. The only work that makes a similar connection is a PhD thesis of Albert Atserias [Ats02], in which he connects expressive power of Datalog with lower bounds on the Resolution proof system.

Just as putting bounds on computational resources gives rise to complexity theory, bounding all quantified variables by arithmetic terms brings us down to bounded arithmetic. In bounded arithmetic, the objects are weak fragments of arithmetic; complexity classes are represented by classes of *functions provably total* in these systems. In descriptive complexity, the objects are classes of formulae (logics) that can *express properties* of certain complexity. Informally, we say that a system of bounded arithmetic *captures* complexity class C if C is exactly the set of functions provably total in that system. A logic captures C if the set of predicates expressible in that logic is C.

Another representation of complexity classes is provided by (propositional) proof systems. There, the objects are proof systems of various power; the complexity is defined in terms of the size of proofs of tautologies in these systems. There is a direct relationship between many proof systems and the corresponding systems of bounded arithmetic. This is despite the fact that the proof systems fall under the non-uniform framework, and the systems of bounded arithmetic under the uniform framework. A proof system P corresponds to a system of bounded arithmetic S if 1) soundness of P is provable in S, and for any P' if soundness of P' is provable in S then P p-simulates P', and 2) proofs in S can be naturally translated into polysize proofs of a family of tautologies in P.

The goal of this work is to suggest a similar connection between the systems of bounded arithmetic and logics of descriptive complexity. Both talk about classes of formulae corresponding to complexity classes. Bounded arithmetic studies the complexity of proving properties of these classes of formulae, whereas descriptive complexity is concerned with their expressive power. The most important distinction between different systems of bounded arithmetic is the strength of their induction (or comprehension) axiom schemes. This leads to the following question: how does the expressive power of the class of formulae in the induction axioms of a system relate to the power of the resulting system?

More precisely, let C be a complexity class, and let Φ_C be a class of formulae that captures C in the descriptive complexity setting. Define the theory of bounded arithmetic V-Φ_C to be Robinson's Q together with comprehension over bounded Φ_C. The following is an informal statement of the main result of this thesis.

Claim. *Consider a class of formulae Φ_C capturing C in the descriptive complexity setting, and a system of arithmetic V-Φ that has comprehension (induction) over Φ_C. Suppose that Φ_C is closed under first-order operations provably in V-Φ. Also, suppose that for every formula $\phi \in \Phi_C$ there is a function F computable in C which takes as its input the values of the free variables of ϕ, and whenever ϕ is true on these values, F returns a "witness" for that. Then V-Φ captures C, that is, the class of provably total functions of V-Φ is the class of functions computable in C.*

This claim and its proof are the subject of chapter 3. The formal statement of the general definability theorem, theorem 3.3.13, appears on page 42. The properties under which the theorem holds are the Closure Property on page 38 and the Constructiveness Property on page 40.

Our setting is more appropriate for small complexity classes, such as classes between $\mathtt{AC}^0$ and $\mathtt{P}$. Most larger complexity classes are not known to be closed under complementation, and correspond to complexity classes that are, under complexity-theoretic assumptions, strictly weaker than the classes of predicates for which they have comprehension. Notably, the most well-known theory of arithmetic S^1_2, which has induction over NP predicates, captures P rather than NP, and does not capture NP unless P = NP.

Restricting our attention to small classes allows us to use definability by NP predicates for the definition of definability: we consider exactly the functions defined by NP predicates that are provably total in our systems. The provability of existence and uniqueness of a value for such functions depends on the power of the systems, and gives us the main measure of that power. In most other systems of arithmetic, functions are introduced either by their recursion-theoretic characterization (see [Coo75] for an original such result or [Zam96]), or by adding one axiom or rule for a property complete for the class (see a system of arithmetic for Logspace in [Zam97]). Since we know the exact expressive power of the formulae in comprehension, and are trying to define in V-Φ the same function class, we introduce function symbols by setting their bitgraphs to be formulae from Φ_C (bitgraphs for string functions, graphs for number functions). We still need to prove existence and uniqueness, but having comprehension over function definitions makes proofs much easier.

Traditionally, a majority of logicians work in the first-order setting. Some of the commonly used theories are restrictions of Peano Arithmetic with additional axioms: such is $I\Delta_0 + \Omega_1$, which has induction over bounded formulae and an axiom proving totality of $x^{|x|}$. Another class of theories, which are the most well-known theories of bounded arithmetic in the complexity theory community, come from the seminal PhD thesis of Samuel Buss [Bus86]. The most cited contribution of his work is the hierarchy of theories S^i_2, which correspond to polynomial-time hierarchy. The distinguishing feature of Buss's systems is the presence in the language of a "smash" function $x\#y = 2^{|x|\cdot|y|}$. This function gives just enough power to code polynomial-size computations. The induction axiom in S^i_2 is $PIND$ with a step from $\lfloor x/2 \rfloor$ to x. The system S^1_2 has the same power as V^1 described later in this thesis; the union of all S^i_2, called S_2, is equivalent in power to $I\Delta_0$ over a language with $\#$.

We find it more convenient to work in the second-order setting both in bounded arithmetic and finite model theory. Buss already introduced second-order theories in his thesis, but the presence of the smash function made them powerful enough to capture PSPACE and EXP. Later, Razborov [Raz93] and independently Takeuti [Tak93] proved the equivalence between second-order theories without smash function and first-order theories with smash; this equivalence is called RSUV isomorphism after the title of Takeuti's paper

("RSUV" stands for translation of classes of theories: R to U and S to V). In particular, this isomorphism takes S_2^i into V_1^i, replacing "large" numbers of S_2^i with strings, "small" numbers with numbers, $PIND$ with induction on the length of a string, and using operations on strings instead the $\#$ function. This framework of V-theories provides the basis for our systems of arithmetic. Note that S_2^0 is not equivalent to V^0, since the former allows number multiplication for arbitrarily large (bounded) numbers.

There are two motivations for choosing the second-order language for our work. The first motivation is that in the second-order framework we talk about properties of strings (or, equivalently, sets of numbers), which seems more natural from the point of view of conventional computational models, such as Turing machines. The second motivation, which allows us to convert freely from the finite model theory setting to the bounded arithmetic setting, is the fact that uninterpreted relational variables in vocabularies of finite model theory are strings or k-ary arrays. They translate directly into free second-order variables in the bounded arithmetic setting. In particular, first-order logic translates into bounded first-order formulae with free second-order variables. Note that for our translation we always assume the presence of order and arithmetic operations in the vocabulary, and assume that they are given their standard interpretation; this is because we are translating the formulae into the language of arithmetic and we do not want to increase accidentally the power of our logic by adding more symbols to the vocabulary.

A question similar to the relation between second-order systems of bounded arithmetic and second-order logics is the relation between proof complexity and propositional satisfiability. The main distinction is that the setting of bounded arithmetic is uniform, whereas the setting of proof complexity is non-uniform. So the complexity classes that we are considering are defined in terms of resource-bounded Turing machines and uniform circuit families.

The outline of this thesis is as follows. Chapters 2 and 3 give the general setting. In chapter 2 we talk about the descriptive complexity setting and the motivation for our work. The following chapter, chapter 3, is the core of the thesis. There we describe the bounded arithmetic setting. Also, this is where we formally state and prove the main definability theorem (theorem 3.3.13) and discuss properties sufficient for theorem 3.3.13 to hold (properties 1 on page 38 and 2 on page 40). The first part of that chapter consists of definitions and proofs of basic properties of our class of systems, the second half contains the definability theorem and its proof. The main part of the proof of the definability theorem is theorem 3.4.2 (generalized witnessing theorem). A "strong" version of the theorem applies to classes closed under complementation (provably in the corresponding V-Φ); a weak version concerns classes for which we do not know the closure (or, in our case, cannot formalize the existing proofs of the closure). In the strong case, the witnessing functions come from the class itself, in the weak case they come from the closure of the function class under first-order operations. We use a system of arithmetic for $\mathtt{AC}^0$ as a running example.

The chapters 4 and 5 give in-depth analysis of two special cases for which the strong version of the main definability theorem applies. Chapter 4 concerns constructing and proving properties for a theory of arithmetic capturing P based on second-order version of Horn formulae. The results of this chapter are published as [CK01], later extended to a journal version [CK03]. Chapter 5 provides a similar treatment for a theory of arithmetic capturing NL. This system is based on a second-order version of 2CNFs. Most of the results from this chapter appear in preliminary form in [CK04]. All these publications are joint work with Stephen Cook.

Chapter 6 presents an example of a complexity class for which the current proof of closure under complementation does not seem to be formalizable in the corresponding system of arithmetic. The class is SL, symmetric logspace, which is weaker than NL. The only currently known proof that SL is closed under complementation uses expander graphs. However, the concept of expander is not known to be formalizable within SL. Still, we can define a system of arithmetic for SL for which a weaker version of the definability theorem holds. This version uses only constructiveness property and talks about the $\mathtt{AC}^0$ closure of SL functions.

The last short chapter describes possible future directions and gives conclusions. One such direction is a project of building a system of arithmetic for the class LOGCFL. This class does not attract much attention, although it has numerous natural characterizations such as semi-unbounded fan-in circuits, and alternating logspace-bounded Turing machines with polynomial-size computation trees. Fairly recently there appeared a result linking LOGCFL with the notion of acyclicity ([GLS01]). It seems an interesting project to pursue that notion to define "acyclic satisfiability", then its second-order variant, and then, perhaps, build a system of arithmetic and a proof system for this class.

Chapter 2

Descriptive complexity background

Descriptive complexity is an area of finite model theory that deals with relationship between complexity classes and expressive power of logics. A good source on descriptive complexity is a book “Descriptive complexity” by Immerman [Imm99]; for background on finite model theory in general see Ebbinghaus and Flum’s “Finite model theory” [EF95] or Libkin’s “Elements of finite model theory” [Lib04].

Definition 2.0.1. A *vocabulary* is a finite set that consists of relational symbols $P, Q, R, \dots$ and constant symbols $c, d, \dots$. We use symbols $\tau, \sigma, \dots$ to denote vocabularies.

Every relational symbol has *arity* (natural number ≥ 1) associated with it. A vocabulary that does not contain constants is called a *relational* vocabulary. Sometimes we want to add function symbols to our vocabularies, but this is equivalent to adding a relation encoding the graph of a function.

Definition 2.0.2. A structure A for a vocabulary τ consists of a nonempty set A, called *universe* and *interpretations* R^A for every relation $R \in \tau$ and c^A for every constant $c \in \tau$. An interpretation of a k-ary relation R on a structure A is a set of k-tuples of elements of A, and an interpretation of a constant symbol is an element from A.

Here we are only concerned with finite structures. We will use the letter n to denote the size of (a universe of) a structure.

Example 2.0.1. A classical example is a vocabulary $\tau = \{E\}$, where E is a binary relation symbol. Structures $G = \langle V^G, E^G \rangle$ over this τ are simple directed *graphs* if $\forall v \in V^G, \neg E^G(v, v)$. If, in addition, $\forall u, v \in V^G E(u, v) \to E(v, u)$ then the graph is undirected.

Example 2.0.2. Another useful example of a vocabulary is $\tau = \{\min, \max, S\}$. *Finite successor structures* are structures over τ in which S is interpreted as a successor relation with min and max as minimal and maximal elements with respect to S. So the universe can be thought of as $\{0, \ldots, n-1\}$, with $\min = 0$, $\max = n-1$ and $S = \{(x, x+1) | x < n-1\}$. A similar, though not equivalent, vocabulary is $\tau = \{\min, \max, \leq\}$; it gives rise to the class of *ordered structures*, if all symbols get standard interpretation.

Example 2.0.3. The vocabulary that we use most often is the relational vocabulary $\tau = \{\min, \max, +, \times, \leq\}$. Structures over τ with standard interpretations for these symbols are *arithmetic structures.*

Terms are variables or constants; if we have function symbols, a function over variables/constants is a term as well. Formulae are constructed from terms by applying relations and logical rules $(\vee, \wedge, \neg, \exists, \forall)$ to the terms. In second-order logic, we allow quantification over sets of (tuples of) elements of the universe: essentially, relations can be viewed as free second-order variables. We will use the term "logic" to refer to a class of formulae with some (usually syntactic) restrictions. An example of a logic is first-order logic, which is the class of formulae with no quantification over second-order variables; another class of examples is existential second-order logic with syntactic restrictions on the quantifier-free part of the formulae.

When we are talking about a complexity class in the descriptive complexity setting, the intended meaning of a language is a class of structures (models of a given formula). To relate this to the traditional definition of a complexity class, where languages are sets of binary strings, consider ordered structures over a vocabulary τ, and assume that there is one unary uninterpreted relational symbol X in the vocabulary that encodes the string. Then the set of models of a formula corresponds to the set of interpretations of X such that the characteristic string of X is in the language; the length of that string corresponds to the size of the model.

Example 2.0.4. Consider the graph vocabulary $\tau = \{E\}$, where E is a binary relation symbol. The set K of models of the following second-order existential formula ϕ corresponds to the set of all 3-colourable undirected graphs:

$$\exists P \exists Q \exists R \forall x \forall y (P(x) \vee Q(x) \vee R(x))$$

$$\wedge(\neg E(x,x)) \wedge (\neg E(x,y) \vee E(y,x)) \wedge (\neg E(x,y) \vee \neg P(x) \vee \neg P(y))$$

$$\wedge(\neg E(x,y) \vee \neg Q(x) \vee \neg Q(y)) \wedge (\neg E(x,y) \vee \neg R(x) \vee \neg R(y))$$

That is, ϕ is true on exactly the structures in which E is interpreted as a 3-colourable graph. The condition that the endpoints are different colours guarantees $\neg E(x,x)$ for all x.

Thus, the traditional concept of a "language" as a set of binary strings corresponds to a "language" as a set of finite structures with one unary relation. A related question is how do we encode finite structures as binary strings. Here we need to make an assumption that the class of structures we are using is ordered, since for a binary string is an ordered sequence of bits. Now, we encode every k-ary relation by a binary string of length n^k where i^{th} bit is 1 iff i^{th} tuple in the ordering is in the relation. Call such encoding of a relation R $enc(R)$. Provided the machine knows the vocabulary, that is arities of the relational symbols, the only additional information needed is the size of the model. It can be inferred from the length of the encodings, but it is customary to start the encoding of a structure with a string of n 0's followed by 1 followed by sequence of encodings of the relations. For example, over a 3-element structure an undirected graph with edges $(0,1)$ and $(0,2)$ but not $(1,2)$ will be encoded by the string "0001011100100".

Now we are ready to define a notion of logic "capturing" a complexity class. For convenience, we restrict our definition of capture to ordered structures. We are mainly interested in captures over arithmetic structures for our connection with bounded arithmetic. The definition below follows [Lib04].

Definition 2.0.3. Let C be a complexity class, L a logic and K a class of finite structures. The L captures C on K if

1. For every L-sentence ϕ and every $\mathcal{A} \in K$, testing if $\mathcal{A} \models \phi$ can be done in C.

2. For every collection K' of structures closed under isomorphism, if this collection is decidable in C then there is a sentence $\phi_{K'}$ of L such that $\mathcal{A} \models \phi_{K'}$ iff $\mathcal{A} \in K'$, for every $\mathcal{A} \in K$.

For our purposes, we fix K to be the arithmetic structures. In particular, the universe of a structure is always considered to be $\{0, \ldots, n-1\}$.

2.1 Capturing complexity classes by logics

It seems that most descriptive complexity results are proven by a very similar strategy: encoding a computation of a Turing machine of a certain form by a corresponding class of formulae. The first such result was Trahtenbrot's theorem [Tra50]: if a vocabulary contains at least one binary symbol, then the class of sentences satisfiable by a finite structure is undecidable (and the class of sentences valid in finite structures is not r.e.). The proof of Trahtenbrot's theorem is based on encoding an arbitrary Turing machine M by a formula satisfiable on a finite structure iff M halts on blank input. Subsequent results concern resource-bounded Turing machines and their corresponding logics.

The smaller the complexity class, the more restricted the class of structures on which a logic captures the class. On the other side of the complexity spectrum, the first-order logic captures $\mathtt{AC}^0$ over arithmetic structures [BIS90]. The extensions of first-order logic by transitive closure, inflationary and partial fixed-point operators capture NL, P, and PSPACE, respectively [Imm82, Imm83].

The first result relating complexity classes and logics is due to Fagin [Fag74]. He noticed that the technique used in the proof of Cook's theorem can be adapted to show that second-order existential logic captures NP.

Theorem 2.1.1 (Fagin's theorem ([Fag74])). *The class of formulae $SO\exists$ captures* NP*.*

Proof sketch. For the model-checking direction, note that in NP we can guess the values of second-order variables and then evaluate the first-order part of the formula on these values. The evaluation can be done in polynomial time provided the number of quantified first-order variables is constant.

The capture direction resembles the proof of Cook's theorem. The first idea in the proof is that linear order can be defined by an existential second-order formula stating existence of a reflexive, transitive and antisymmetric relation

$$\exists P \forall x,y,z P(x,x) \wedge (P(x,y) \vee \neg P(x,y)) \wedge (P(x,y) \wedge x \neq y \rightarrow \neg P(y,x)) \qquad (2.1)$$
$$\wedge (P(x,y) \wedge P(y,z) \rightarrow P(x,z))$$

The main part of the proof is a construction of a second-order existential formula encoding a tableau of a run of an NP Turing machine M on an input string of length n. Suppose that n^k is the upper bound on the length of any computational path of M. Then such a path of M can be represented by an $(n^k+1) \times (n^k+1)$ tableau, in which every row is a configuration of M. That is, a cell (i,j) corresponds to the configuration on i^{th} step of the execution, and contains either a symbol encoding the state of M, if the head is in the j^th position, or a symbol in position j or $j+1$, depending whether it comes before or after the head position. For example, a configuration can be $010q_20\not{b}\not{b}\not{b}\not{b}\not{b}$, corresponding to M in state q_2 pointing to the last non-blank symbol on the tape. A tableau encodes an accepting computation of M if 1) the first line is an initial configuration $q_0w_1 \dots w_n\not{b} \dots \not{b}$, where $w_1 \dots w_n$ is an input string, 2) every subsequent configuration is obtained from the previous by a valid transition in δ, and 3) the state q_{accept} occurs in some configuration and thus in some cell of the tableau.

In the proof of Cook's theorem, each cell (i,j) is encoded by a set of propositional variables $x_{i,j,c}$, where $i,j \leq n^k$ and $c \in Q \cup \Gamma$. The intended meaning is that $x_{i,j,c}$ is true iff c is the symbol in the cell (i,j) of the tableau, either encoding a tape alphabet symbol or a state. Fagin's modification uses second-order $2k$-ary variables $C_1 \dots C_m$, where $m = |Q|+|\Gamma|$, to denote the cells: that is, $C_k(i,j)$ encodes a propositional variable $x_{i,j,k}$. On a structure of size n, a $2k$-ary variable encodes n^{2k} bits of information, each corresponding to a cell of the tableau. The formula itself is the same in both Cook's and Fagin's theorems, modulo respective variable substitution; it states that the tableau represented by the variables encodes an accepting computation of M. The last difference is that $C_1 \dots C_m$ are existentially quantified, so the question of propositional formula satisfiability is transformed into validity for a second-order existential formula, that is, whether there exist $C_1 \dots C_m$ satisfying the conditions. □

Corollary 2.1.2 ([Sto77]). *Second-order logic SO captures the polynomial-time hierarchy. Moreover, every level of PH is captured by the corresponding level of SO.*

2.2 Capturing feasible classes

For the purpose of this work we are more interested in feasible complexity classes, namely classes between $\mathtt{AC}^0$ and $\mathtt{P}$. The first successful characterizations of several such classes were obtained by Immerman in [Imm82, Imm83]. His approach is based on extending first-order logic by various fixed-point operators. In particular, he gives characterizations for $\mathtt{P}, \mathtt{NL}, \mathtt{L}, \mathtt{SL}$ and non-uniform $\mathtt{AC}^0$. The result that $\mathtt{P}$ is captured by FO+least fixed point operator was obtained independently by others: Sazonov [Saz80] has similar ideas, and Livchak [Liv82] and Vardi [Var82] gave the same characterizations of $\mathtt{P}$.

Before we show that extensions of first-order logic capture $\mathtt{NL}$, $\mathtt{P}$ and so on, we need to talk about the complexity of first-order logic itself. As mentioned before, the weaker the logic the smaller the class of structures on which it captures a complexity class. Barrington, Immerman and Straubing show in [BIS90] that DLOGTIME-uniform $\mathtt{AC}^0$ can be captured by first-order logic over arithmetic structures, or, equivalently, over vocabulary consisting of just one symbol $BIT(i,j)$ interpreted as "i^{th} bit of j is 1"; see [DDLW98] for defining order by BIT. Note that non-uniform $\mathtt{AC}^0$ corresponds to $FO(All)$, which is first-order logic augmented with all numeric predicates. However, our goals ask for a uniform framework, and so [BIS90] result is more useful for us.

Theorem 2.2.1 ([BIS90]). *Over arithmetic structures, FO captures DLOGTIME-uniform* $\mathtt{AC}^0$.

Now we can show how to extend first-order logic by fixed-point operators to capture more powerful complexity classes.

Definition 2.2.2. A *fixed point* of a formula is defined as follows. Let $\phi(X,\bar{x})$ be a (first-order) formula over vocabulary $\tau \cup X$ with $X \notin \tau$, and let $\mathcal{A}$ be a τ-structure. Define $\phi_{\mathcal{A}}(X)$ to be the set of all possible tuples $\bar{x}$ satisfying ϕ in $\mathcal{A}$ for a given value of X:

$$\phi_{\mathcal{A}} : X \mapsto \{\bar{a} \mid (\mathcal{A}, X) \models \phi(\bar{a})\}$$

In general, a set X is a fixed point of $\phi_{\mathcal{A}}$ if $\phi_{\mathcal{A}}(X) = X$, that is, the set of tuples satisfying $\phi_{\mathcal{A}}$ for that value of X is X itself.

Now take $X_0 = \emptyset$ and $X_{i+1} = \phi_{\mathcal{A}}(X_i)$. A set X_k is a *fixed point* of $\phi_{\mathcal{A}}$ if $X_k = X_{k+1}$. The fixed point X_k is often denoted by X^{ω}, since we do not care about the exact value of k.

A set X is called the *least fixed point* of ϕ (denoted $LFP_{X,\bar{x}}(\phi)[\bar{u}]$) if for every fixed point X_k of ϕ, $X \subseteq X_k$. Note that if X occurs only positively in ϕ then a fixed point always exists. In this case, if X_i is a fixed point, then $X_i \subseteq X_{i+1}$ for all i; then the fixed point is called *inflationary*. If the operator is not monotone in X, it might not have a fixed point; if it does, it is called a partial fixed-point.

Example 2.2.1 *(Transitive closure).* The canonical example of a fixed point is a transitive closure over graph structures ($\tau = \{E\}$). Let

$$\phi(x,y,X) = E(x,y) \vee \exists z(X(x,z) \wedge E(z,y))$$

Since X occurs positively in ϕ, the least fixed point of ϕ exists. Let $TC_{x,y}E(x,y)[a,b]$ be the formula true if the pair (a,b) is in the transitive closure in a graph with edge relation

E. In this notation X is implicit. In general, let $\psi(x, y)$ be some formula; then transitive closure over the graph defined by ψ is $TC_{x,y}\psi(x, y)[a, b]$, where

$$\phi(x, y, X) = \psi(x, y) \vee \exists z(X(x, z) \wedge \psi(z, y))$$

A logic can be augmented with a fixed-point operator to make it more powerful.

Definition 2.2.3. Define FO+LFP to be the first-order logic together with a positive least fixed point operator $LFP_{X,\bar{x}}(\phi)[\bar{u}]$. This is interpreted as a formula of vocabulary τ which is true whenever $\bar{u}$ is in the fixed point of the first-order formula ϕ over X. Similarly, define FO+TC to be the first-order logic augmented with transitive closure as defined in the example 2.2.1.

Theorem 2.2.4 ([Imm82, Imm83]). *Over the class of ordered structures (that is, $\leq \in \tau$), FO+LFP captures* `P` *and FO+TC captures* `NL`*.*

Proof sketch. The proofs resemble the proof of Trahtenbrot's theorem: a corresponding logic is used to encode computations of respective Turing machines. In case of polynomial-time Turing machine, LFP operator states that accepting configuration is in the fixed-point of a formula defining transition function, where variables encode cells of computation tableau (a constant number of variables of arity $2k$ for each cell, where k is the degree of the polynomial). The `NL` case is slightly more interesting: there, the resulting formula states that there is a pair of the form (initial configuration, some accepting configuration) in the transitive closure of the transition function. In case of `NL`, the number of configurations is polynomial, so each configuration is represented by a constant number of variables. □

A different approach is presented in the seminal '91 paper by Grädel [Grä91, Grä92]. There, he gave characterizations for `L`, `SL`, `NL` and `P` by fragments of second-order logic over successor structures. Together with Immerman, Barrington and Straubing's result from [BIS90] that first-order logic captures DLOGTIME-uniform $\mathtt{AC}^0$ over arithmetic structures, these characterizations are the basis for the class of systems of arithmetic considered here.

Definition 2.2.5. We will use the term *restricted* $SO\exists$ to refer to formulae of the form

$$\exists P_1 \dots P_k \forall x_1 \dots x_l \psi(\bar{P}, \bar{x}, \bar{a}, \bar{Y}), \tag{2.2}$$

where k, l are constants, and ψ is a CNF in which use of the quantified second-order variables $\bar{P}$ is somehow restricted. The variables $\bar{a}$ and $\bar{Y}$ are free variables: the restrictions on $\bar{P}$ do not apply to them.

Note that there are no occurrences of existential first-order quantifiers in restricted $SO\exists$ formulae. This is because even when the class of ψ is restricted to 2CNF with at most one occurrence of a positive literal, with presence of an existential quantifier it is possible to capture all of $SO\exists$ [Grä91]. However, universal first-order and quantifier-free formulae are restricted $SO\exists$.

In particular, we are most interested in the following restrictions of ψ:

Definition 2.2.6. A formula $\psi(\bar{x}, \bar{P}, \bar{a}, \bar{Y})$ is *Horn* with respect to the second-order variables $P_1, ..., P_k$ if ψ is quantifier-free in conjunctive normal form and in every clause there is at most one positive literal of the form $P_i(\bar{x})$. It is *Krom* with respect to $\bar{P}$ if ψ is a CNF with at most two occurrences of a P-literal per clause. It is *SymKrom* if it is Krom with $\oplus$ instead of $\vee$ in every clause (so every clause is of the form $(\phi_i \rightarrow L_i \oplus L'_i)$, where the only P-literals are $L_i and L'_i$).

Following Grädel, we call define classes $SO\exists$ Horn and $SO\exists$ Krom to be restricted $SO\exists$, in which ψ is Horn with respect to $\bar{P}$ for $SO\exists$ Horn and Krom with respect to $\bar{P}$ for $SO\exists$ Krom, respectively. $SO\exists$ SymKrom is defined analogously.

Here, we are assuming that the vocabulary is relational, so the only P-literals are $P_i(\bar{x})$. This definition easily extends to allow functions in vocabularies; then the P-literals can be $P(t(\bar{x}))$ for a term $t(\bar{x})$.

Example 2.2.2 *(*PARITY(X)*).* This is a $SO\exists$-Horn formula over successor structures $(\min, \max, S \in \tau)$, models of which have interpretations of X as sets with an odd number of 1's. It encodes a dynamic-programming algorithm for computing parity of X: $P_{odd}(i)$

is true (and $P_{even}(i)$ is false) iff the prefix of X of length i contains an odd number of 1's.

$$\begin{aligned}
&\exists P_{even} \exists P_{odd} \forall i \\
&\qquad P_{even}(\min) \wedge \neg P_{odd}(\min) \wedge (P_{odd}(\max) \leftrightarrow \neg X(max)) \\
&\quad \wedge(\neg P_{even}(S(i)) \vee \neg P_{odd}(S(i))) \\
&\quad \wedge(P_{even}(i) \wedge X(i) \rightarrow P_{odd}(S(i))) \wedge (P_{odd}(i) \wedge X(i) \rightarrow P_{even}(S(i))) \\
&\quad \wedge(P_{even}(i) \wedge \neg X(i) \rightarrow P_{even}(S(i))) \wedge (P_{odd}(i) \wedge \neg X(i) \rightarrow P_{odd}(S(i)))
\end{aligned}$$

We state the following theorem only for $SO\exists$ Horn and $SO\exists$ Krom; the case of $SO\exists$ SymKrom is similar to the Krom case. This is an analog of Fagin's theorem for several feasible classes; just as Fagin's theorem corresponds to `NP`-completeness of 3SAT, Grädel's theorem relates to complexity of HornSat and 2SAT.

Theorem 2.2.7 (Grädel's theorem[Grä91]). *Over successor structures $SO\exists$ Horn captures* `P` *and $SO\exists$ Krom captures* `NL`*.*

Proof sketch. For one direction, we show that $SO\exists$ Horn and Krom formulae can be evaluated in `P` and `NL`, respectively. Fix the values of free variables. Let ϕ be as in the definition 2.2.5. First turn ϕ into a propositional formula by taking a conjunction of n^l copies of ψ, each with a different value for the first-order variables $x_1 \dots x_l$. Evaluate first-order terms and terms involving free variables in each copy of ψ. The second-order quantified literals of the form $P_i(t(\bar{x}, \bar{a}, BY))$ become propositional variables, which are independent unless P_i are the same and the terms evaluate to the same value (on possibly different values of $\bar{x}$). The resulting formula is Horn (respectively, 2CNF) if ψ is Horn (Krom) with respect to $\bar{P}$. Since satisfiability for propositional Horn formulae is in `P` and for propositional 2CNF is in `NL`, so is satisfiability for $SO\exists$ Horn and $SO\exists$ Krom.

Example 2.2.3. Let structure be $\{0, 1\}; f(0) = 1, f(1) = 1$. That is, the universe contains two elements and a function symbol f. This is how a formula (which happens to be both

Krom and Horn) is translated into its propositional equivalent over that structure:

$$\begin{aligned}
&\exists P \forall x, y(x = y \vee P(x) \vee \neg P(y)) \wedge P(f(x)) \\
\equiv&(0 = 0 \vee P(0) \vee \neg P(0)) \wedge P(1)) \wedge (0 = 1 \vee P(0) \vee \neg P(1)) \wedge P(1)) \wedge \\
&(1 = 0 \vee P(1) \vee \neg P(0)) \wedge P(1)) \wedge (1 = 1 \vee P(1) \vee \neg P(1)) \wedge P(1)) \\
\equiv&p_1 \wedge (p_1 \vee \neg p_0) \wedge (p_0 \vee \neg p_1)
\end{aligned}$$

Here, p_0 and p_1 are independent propositional variables. The satisfying assignment to this formula is $p_0 = T, p_1 = T$, so $P(0) = P(1) = T$. By evaluating the first-order terms, the clause $(0 = 0 \vee p_0 \vee \neg p_0)$ was removed because it evaluates to true by $0 = 0$; the clause $(0 = 1 \vee p_0 \vee \neg p_1)$ lost its $0 = 1$ part because $0 = 1$ is false.

The more detailed descriptions and formalizations of the satisfiability algorithms are given later, in the corresponding chapters.

For the proof of the other direction, Grädel uses the result from theorem 2.2.4, and shows that $SO\exists$ Horn and Krom correspond to FO+LFP and FO+TC, respectively. Note that without the successor relation the correspondences do not hold: the restrictions of SO are strictly weaker than the corresponding fixed point logics.

The easier part of the proof is defining transitive closure by a $SO\exists$ Krom formula. That implies that, in the presence of successor, $SO\exists$ Krom is (at least) as powerful as FO+TC, in particular it captures `NL`. Since `NL` is closed under complementation, it is sufficient to show that $SO\exists$ Krom can define negated transitive closure. Consider a formula $\neg TC_{\bar{x},\bar{y}}\phi(\bar{x}, \bar{y})[\bar{a}, \bar{b}]$. Since we need a CNF for the quantifier-free part of the $SO\exists$ Horn we are constructing, and since we are going to negate ϕ, take the disjunctive normal form of $\phi = \bigvee_j \phi_j$. Now the $SO\exists$ Horn formula encoding the negated transitive closure formula above is

$$\exists R \forall \bar{x}, \bar{y}, \bar{z} \neg R(\bar{a}, \bar{b}) \wedge (R(\bar{x}, \bar{x}) \wedge \bigwedge_j ((R(\bar{x}, \bar{y}) \wedge \phi_j(\bar{y}, \bar{z})) \rightarrow R(\bar{x}, \bar{z})))$$

This formula is true if there exists R that contains transitive closure over ϕ and does not contain the pair $(\bar{a}, \bar{b})$, which is equivalent to the statement that $(\bar{a}, \bar{b})$ is not in the transitive closure of ϕ.

Now we show that $SO\exists$ Horn can define FO+LFP (in presence of successor). Every formula in FO+LFP is has, in presence of successor, a normal form $LFP_{P,\bar{x}} \exists \bar{y} \phi(P, \bar{x}, \bar{y})[\bar{0}]$. Since FO+LFP is closed under complementation, for every language L its complement

$\bar{L}$ is definable in FO+LFP. Let $\bar{L}$ be defined by a formula in normal form, and let ϕ from the definition of $\bar{L}$ be equivalent to a disjunction $\bigvee_j \phi_j$, for the same reason as in the $SO\exists$ Krom case. Now $SO\exists$ Horn formula defining L is

$$\exists P \forall \bar{x} \forall \bar{y} \neg P(\bar{0}) \bigwedge_j (\phi_j(P, \bar{x}, \bar{y}) \to P(\bar{x}))$$

The first part of the formula states that P does not contain the tuple $\bar{0}$. The second part states that P contains every tuple that a fixed-point over ϕ has. Since ϕ came from the definition of $\bar{L}$, the resulting expression implies that there is a set P containing the fixed-point of P (and, therefore, containing its least fixed point), but not the tuple $\bar{0}$. So our formula is true iff the formula defining $\bar{L}$ is false. $\square$

Note that the descriptive complexity results above appeal to results about complexity of syntactically restricted subclasses of the propositional satisfiability. Fagin's theorem is a variation on Cook's proof that 3SAT is NP-complete, and Grädel's theorem refers to the fact that satisfiability of propositional Horn formulae is complete for P and 2SAT is complete for (co-)NL, as 2SymSat (2SAT with $\oplus$ instead of $\vee$) is complete for co-SL. The following major theorem by Schaefer [Sch78] summarizes the syntactic subclasses of $3SAT$ and their complexity. We use different names for some classes since some Schaefer's classes were combined or defined using conventional models such as circuits and Turing machines since 1978 when the paper was published. In Schaefer's original paper, the first class is stated as "decidable in L"; we strengthen it to be decidable in uniform $\mathtt{AC}^0$ (thanks to Valentine Kabanets for noticing this).

Let S be a set of k-ary relations, where relations can be thought of as types of clauses (e.g, S for $3CNF$ formulae consists of four ternary relations $(x \vee y \vee z)$, $(\neg x \vee y \vee z)$, $(\neg x \vee \neg y \vee z)$ and $(\neg x \vee \neg y \vee \neg z)$.) Then the following theorem holds:

Theorem 2.2.8 (Schaefer's dichotomy theorem ([Sch78])). *The complexity of the satisfiability problem for a conjunction of relations from S belongs to one of the following categories, depending on the types of relations (that is, clauses) occurring in S.*

1. *Decidable in $\mathtt{AC}^0$ (for example, in case every clause in S that has a negated variable is a unit clause.)*

2. *Logspace-complete for SL (e.g., when $S = \{(x \oplus y), (\neg x \oplus y)\}$, giving symmetric 2CNFs).*

3. *Logspace-complete for* NL *(e.g.,* $S = \{(x \vee y), (\neg x \vee y), (\neg x \vee \neg y)\}$*, so resulting formulae are 2CNF).*

4. *Logspace-equivalent to the consistency problem for systems of linear equations over the boolean domain (e.g., if* S *is a set of all* k*-clauses for some* $k \geq 3$ *of the form* $(x_1 \oplus x_2 \oplus \cdots \oplus x_k)$ *or* $(\neg x_1 \oplus x_2 \oplus \cdots \oplus x_k)$ *). This may correspond to complexity class* $DET(0,1)$ *of problems equivalent to testing if determinant over* $GF(2)$ *is 0;* DET *lies in* NC^2*, but its exact complexity is unknown.*

5. *Logspace-complete for* P *(e.g.,* S *is a set of* k*-ary Horn clauses or a set of dual Horn clauses,* $k \geq 3$*).*

6. *Logspace-complete for* NP *(e.g.,* S *is all four 3-clauses as above).*

Lemma 2.2.9. *The first of the classes in the theorem 2.2.8 is decidable in uniform* $\mathtt{AC}^0$*.*

Proof. Given some encoding of a CNF formula in which every clause either has only positive literals or exactly one negative literal, we construct an $\mathtt{AC}^0$ circuit to decide its satisfiability. The idea is to check, for every clause with positive literals, if all of its literals occur negated; if so, the formula evaluates to $\perp$, otherwise to $\top$.

The output node is an AND over all clauses. First, check if a clause is a unit clause with negated literal; if so, send 1 to the output node. Otherwise, the gate above the output node is a NOT which has as its input an AND gate. There, we take an AND over all variables in a corresponding clause. For every variable it is an OR over all clauses which checks whether that variable occurs negated. □

The logspace-completeness in Schaefer's theorem may be strengthened to be $\mathtt{AC}^0$-completeness.

Chapter 3

Bounded arithmetic and generalized witnessing

The goal of this work is to relate descriptive complexity to bounded arithmetic. Whereas in descriptive complexity we are concerned with the complexity of expressing properties, in bounded arithmetic it is proving totality of functions of varying complexity. The stronger the system, the more complex functions can be proven total. The systems of arithmetic we are using here follow the approach of [Coo02], the notes from the course taught by Cook at the University of Toronto in Spring 2002. The second-order systems in these notes are based, in turn, on Zambella's [Zam96] class of systems Σ^b_i-comp. Our second-order systems are based on the descriptive complexity characterizations described in chapter 2. For a general reference on bounded arithmetic, see the comprehensive book by Krajíček , [Kra95]. A classical book introducing bounded arithmetic is the PhD thesis of Sam Buss [Bus86]. For the treatment of first-order systems, as well as other fragments of arithmetic, see Hájek and Pudlák's book [HP93].

In descriptive complexity, a language in the traditional complexity theory setting is thought of as a set of interpretations of a unary predicate X (representing a binary string) in structures that are models of a given formula. A class of languages then naturally corresponds to a class of formulae: each language in the class corresponds to a formula which has, as its set of models, the structures with X interpreted as strings from the language. In the bounded arithmetic setting, the relationship with complexity classes is slightly different. Here, we consider representations of languages in the standard model of arithmetic $\mathbb{N}_2$, defined below. Instead of a set of structures with one predicate getting different interpretation we are talking about one fixed structure and different (second-order) elements of it satisfying the formula.

Even though the results of this chapter are stated in a more general form, the main focus of this work is systems of arithmetic based on classes of formulae that came from restricted SO$\exists$, as well as first-order logic.

3.1 Translation from descriptive complexity to bounded arithmetic

We would like to be able to talk about the same "class of formulae" both in descriptive complexity and bounded arithmetic context. However, there are several differences, the most obvious being the language (vocabulary). In the descriptive complexity setting, the vocabularies are allowed to vary, although some symbols have to be present and get standard interpretations in order to capture complexity classes. But the descriptive complexity results we are considering here hold for logics over arithmetic structures. That makes translation into the language of bounded arithmetic much easier.

3.1.1 The bounded arithmetic framework

The language of our systems of arithmetic is $\mathcal{L}_A^2 = \{0, 1, +, \cdot, |\ |; <, =, \in\}$, a natural second-order extension of the language of Peano Arithmetic $\mathcal{L}_A = \{0, 1, +, \cdot; <, =\}$. Let $\mathbb{N}_2$ be a standard structure with natural numbers and finite sets of natural numbers in the universe; our first-order objects (denoted by lower-case letters) are natural numbers; second-order objects (denoted by upper-case letters) are binary strings or, equivalently, (finite) sets of numbers. Treating a second-order variable X as a set, its "length" $|X|$ is defined to be the largest element $y \in X$ plus one, or 0 if X is an empty set.

Arithmetic terms are constructed using $+$ and $\times$ are from first-order variables, constants 0 and 1, and terms of the form $|X|$ where X is a second-order variable. The atomic formulae of $\mathcal{L}_A^2$ have one of the forms $s = t, s \leq t, t \in X$, where s and t are terms and X is a second-order variable. We usually write $X(t)$ instead of $t \in X$. Formulae are built from atomic formulae using the propositional connectives $\wedge, \vee, \neg$, the first-order quantifiers $\forall x, \exists x$ and the second-order quantifiers $\forall X, \exists X$.

We use the usual abbreviations $s \neq t$ for $\neg s = t$ and $s < t$ for $s \leq t \wedge s \neq t$. Bounded first-order quantifiers get their usual meaning: $\forall x \leq t\phi$ stands for $\forall x(x \leq t \rightarrow \phi)$ and $\exists x \leq t\phi$ stands for $\exists x(x \leq t \wedge \phi)$. Second-order quantifiers are strings of bounded length.

In complexity theory a member of a language is a binary string. We relate it to the second-order arithmetic view of second-order objects as finite subsets S of $\mathbb{N}$ as follows. Given a string $w = x_0 \ldots x_{n-1}$, define a corresponding set S by $S = \{i | x_i = 1\} \cup \{n\}$. For example, the string 01100 becomes the set $\{1, 2, 5\}$, and the empty string becomes the set $\{0\}$.

For the other direction, if S is the empty set then there is no string associated with it. Otherwise, $w(S)$ is defined to be a characteristic string of S of length $\max_i i \in S$ (the last element becomes length). With this correspondence, if $|S| = n$ where $|S|$ is as defined above, then $|w(S)| = n$: that is the length of the string is exactly the value of the largest element of S. For example, a set $\{2, 3, 7\}$ becomes a string 0011000.

The first-order objects in our theories play the role of indices of binary strings. Thus in determining the complexity of a set of natural numbers we code a natural number i using unary notation, that is, as a string of 1s of length i.

Let $w(S)$ be the finite binary string associated with $S \subset \mathbb{N}$ as above. We will say that a formula $A(X)$ *represents* the language L if $L = \{w(S) | \mathbb{N}_2 \models A(S)\}$. More generally, $A(\bar{x}, \bar{Y})$ represents a relation $R(\bar{x}, \bar{Y})$ which holds on $\bar{x}, \bar{Y}$ iff $\mathbb{N}_2 \models A(\bar{x}, \bar{Y})$.

Definition 3.1.1. If $\phi(\bar{z}, \bar{Y})$ is a formula of $\mathcal{L}_A^2$ whose free variables are among $z_1, ..., z_k$, $Y_1, ..., Y_\ell$ then ϕ *represents* a $k+\ell$-ary relation R^ϕ as follows. If $a_1, ..., a_k$ are natural numbers and $B_1, ..., B_\ell$ are finite sets of natural numbers, then $\langle a_1, ..., a_k, B_1, ..., B_\ell \rangle$ satisfies R^ϕ iff $\phi(a_1, ..., a_k, B_1, ..., B_\ell)$ is true in the standard model.

If $\mathbf{C}$ is a complexity class, then we make sense of the statement "R^ϕ is in $\mathbf{C}$" using the string encodings described above. In particular, a relation $R(x_1, ..., x_k, Y_1, ..., Y_m)$ is in $\mathtt{P}$ iff it is recognizable in time bounded by a polynomial in $(x_1, ..., x_k, |Y_1|, ..., |Y_m|)$. Generalizing the notion of representing, a class of formulae Φ represents a complexity class $\mathbf{C}$ iff every relation R from $\mathbf{C}$ is representable by a formula from Φ, and every formula from Φ can be evaluated within $\mathbf{C}$. This notion is parallel to the notion of "capture" from descriptive complexity (see definition 2.0.3); essentially, they have the same meaning of describing the expressive power of formulae. But the notion of "capture" we will be using for systems of bounded arithmetic will be quite different.

We now define the classes Σ_i^B and Π_i^B of bounded second-order formulae. (A formula is *bounded* if all its quantifiers are bounded.) Σ_0^B and Π_0^B both denote the class of bounded formulae with no second-order quantifiers. We define inductively Σ_{i+1}^B as the least class of formulae containing Π_i^B and closed under disjunction, conjunction, and bounded existential second-order quantification. The class Π^B_{i+1} is defined dually. Additionally, we define the classes $\Delta^B_i \equiv \Sigma^B_i \cap \Pi^B_i$.

The classes Σ^B_i and Π^B_i are the formulae in our (Zambella's) simplified language $\mathcal{L}^2_A$ which correspond to the classes $\Sigma^{1,b}_i$ and $\Pi^{1,b}_i$ in Buss's prototype second-order language [Bus86, Kra95], except that Buss's language contains the # function, and our language does not. Our Σ^B_i and Π^B_i are the second-order analogs of the first-order formula classes Σ^b_i and Π^b_i, where sharply bounded quantifiers correspond to our bounded first-order quantifiers.

Definition 3.1.2. Analogously to Definition 2.2.5, we use the term *restricted* Σ^B_1 to refer to the formulae of the form $\exists P_1 \dots P_k \forall x_1 < t_1 \dots x_l < t_l \psi(\bar{P}, \bar{x}, \bar{a}, \bar{Y})$, where ψ is quantifier-free with syntactic restrictions on the P_is. In particular, the variables P_i are only allowed to occur as $P(t)$ or $\neg P(t)$ for some terms t; occurrences of $|P_i|$ are not allowed, although the use of length function on free variables is not restricted.

Most of our restricted Σ^B_1 formulae are translations of restricted SO$\exists$ formulae in the bounded arithmetic context. All examples we study in detail, except for Σ^B_0, come from restricted Σ^B_1 classes of formulae.Note that first-order universally quantified and quantifier-free formulae are always equivalent to any subclass of restricted Σ^B_1, since restrictions only apply to quantified second-order variables.

Remark 3.1.3. In the classes Σ^B_i and Π^B_i defined above the second-order quantifiers are explicitly bounded: that is, all occurrences of $\exists X$ and $\forall X$ are of the form $\exists X < t$ and $\forall X < t$, where t is some term bounding the length of X. However, when we talk about restricted Σ^B_1 formulae, we do not explicitly put bounds on second-order quantifiers (although we do explicitly bound all first-order quantifiers). We can do that because the only literals involving P_is are of the form $P_i(t)$; in particular, $|P_i|$ is not allowed. Therefore, we only care about the values of P_i up to the largest value of an indexing term. So every quantified second-order variable is *implicitly bounded* by $\max_j t_j(\bar{b})$, where t_j are terms and $\bar{b}$ is the upper bound on the variables. Since restricted Σ^B_1 formulae behave as bounded formulae rather than unbounded, we will abuse notation and use Σ^B_1 instead of Σ^1_1 to refer to them.

3.1.2 The translation method

In the descriptive complexity setting the vocabulary is usually different from $\mathcal{L}^2_A$, and all variables are implicitly bounded by the size of the model. One consequence is that there is a difference between expressive power of formulae with monadic versus k-ary relations. In bounded arithmetic we have a pairing function, easily definable even in the weakest of our systems V^0, which can be used to encode a multi-dimensional array as one binary string.

We first illustrate the translation using $SO\exists$-Horn formulae as an example, and then give the full rules for the translation in the general case.

Recall Grädel's definition of $SO\exists$-Horn formulae in the descriptive complexity setting (definition 2.2.6). Consider $SO\exists$-Horn formulae over arithmetic structures. We obtain Σ_1^B-Horn by setting the language of ϕ to be $\mathcal{L}_A^2$ and bounding all first-order quantifiers by terms in free variables. Now a class of restricted SO∃ formulae ($SO\exists$-Horn) becomes the class of restricted Σ_1^B formulae (Σ_1^B-Horn).

Definition 3.1.4. A formula is Σ_1^B-Horn if it is of the form

$$\exists P_1 ... \exists P_k \forall x_1 < t_1(\bar{a}) ... \forall x_m < t_m(\bar{a}) \psi(\bar{x}, \bar{P}, n, \bar{a}, \bar{Y}), \tag{3.1}$$

where ψ is Horn with respect to $P_1, \dots, P_k$. If the vocabulary τ contains $+, \cdot, =$, they get standard interpretations; also, $<$ gets standard interpretation and $S(x, y)$ is interpreted as $x = y+1$. The uninterpreted variables $\bar{a}, \bar{Y}$ in τ occurring in ϕ become free variables of ψ, with an additional free variable n corresponding to the size of the structure. Usually we do not treat n differently from other free variables; however, there must be at least one free variable to use as a bound on the first-order quantified variables (if there are only second-order variables, use their length in the bounding terms).

Note that our definition of Σ_1^B-Horn formulae is somewhat more general than the original $SO\exists$-Horn: in particular, we allow first-order variables to be bounded by arbitrary terms, and we allow any arithmetic terms in our formulae, as long as they do not contain occurrences of P_i (e.g, no $|\ |$ on quantified second-order variables).

Example 3.1.1 *(*PARITY(X)*).* This is a Σ_1^B-Horn formula which is a translation of the $SO\exists$-Horn formula for $PARITY$ in the example 2.2.2. Since we could refer to values of variables past n, the clause $(P_{odd}(max) \leftrightarrow X(max))$ could be simplified to $P_{odd}(|X|)$, where $|X|$ is essentially max $+1$.

$$\begin{aligned}\textsc{Parity}(X) \equiv \exists P_{even} \exists P_{odd} \forall i < |X| \\ & P_{even}(0) \wedge \neg P_{odd}(0) \wedge P_{odd}(|X|) \wedge (\neg P_{even}(i+1) \vee \neg P_{odd}(i+1)) \\ & \wedge (P_{even}(i) \wedge X(i) \rightarrow P_{odd}(i+1)) \wedge (P_{odd}(i) \wedge X(i) \rightarrow P_{even}(i+1)) \\ & \wedge (P_{even}(i) \wedge \neg X(i) \rightarrow P_{even}(i+1)) \wedge (P_{odd}(i) \wedge \neg X(i) \rightarrow P_{odd}(i+1))\end{aligned}$$

Example 3.1.2 *(*$3\textsc{Color}(n, E)$*).* The following Σ_1^B formula is a translation of the 3-colourability predicate from the example 2.0.4. It asserts that the graph with edge relation E on nodes $\{0, 1, ..., n-1\}$ is three-colourable. We write $E(x, y)$ like a binary relation, although it can be coded as a unary relation using a pairing function. The three colors are P, Q, and R.

$$\begin{aligned}\exists P \exists Q \exists R \forall x < n \forall y < n (P(x) \vee Q(x) \vee R(x)) \wedge (\neg E(x,y) \vee \neg P(x) \vee \neg P(y)) \\ \wedge (\neg E(x,y) \vee \neg Q(x) \vee \neg Q(y)) \wedge (\neg E(x,y) \vee \neg R(x) \vee \neg R(y))\end{aligned}$$

This formula is Σ_1^B-Horn except for the first clause. Since graph 3-colourability is $\mathtt{NP}$-complete, it cannot be represented by a Σ_1^B-Horn formula unless $\mathtt{P} = \mathtt{NP}$. This example illustrates why we cannot allow bounded first-order existential quantifiers after the universal quantifiers in Σ_1^B-Horn formulae, since the first clause could be replaced by $\exists i < 3P(i, x)$ where now $P(0, x), P(1, x), P(2, x)$ represent the three colors.

Now we generalize these examples. Let Φ be a descriptive logic over a vocabulary τ. For every $\phi \in \Phi$, we can define a translation ϕ^* into $\mathcal{L}_A^2$ with the following properties:

1. Every interpreted symbol from τ that occurs in $\mathcal{L}_A^2$ gets the standard interpretation. Successor, min, etc are translated into appropriate arithmetic operations such as $+1$ for successor, 0 for min.

2. Translate max as n for a free variable n. For every quantified first-order variable, set $n+1$ (more generally, a polynomial of n) as a bound. Note that then $|X| = n+1$ for a unary second-order predicate.

3. Translate uninterpreted relational symbols of τ occurring in ϕ as free variables of ϕ^*. If a variable is k-ary, use pairing function (see definition 3.2.5 below) to encode the relational symbol as a unary second-order variable. Then any occurrence of $R(x_1, \ldots, x_k)$ becomes $R^*(\langle x_1, \ldots, x_k \rangle)$, where $\langle x_1, \ldots, x_k \rangle$ is a value obtained by applying the pairing function to $x_1, \ldots, x_k$.

3.1.3 Representation theorems

Let Φ^* denote a generalization of translated formulae in Φ which allows arbitrary arithmetic terms, in particular as bounds (see Σ_1^B-Horn example). Now, Φ and Φ^* satisfy the following property:

Property. *For a complexity class C, let Φ be a descriptive logic capturing C in the descriptive complexity setting. Then Φ^* corresponds to the same complexity class C in the bounded arithmetic setting. That is, $L \in C$ iff some formula of Φ^* represents L in the standard structure $\mathbb{N}_2$.*

For example, Σ_0^B (for which Φ is first-order logic) formulae represent DLOGTIME-uniform $\mathtt{AC}^0$, and for any $i > 0$, Σ_i^B formulae, translated from second-order formulae with i alternations of second-order quantifiers, represent i^{th} level of the polynomial-time hierarchy (in particular, Σ_1^B represent $\mathtt{NP}$ relations). This follows from theorem 2.1.1 and its corollary 2.1.2. See [Bus86, Kra95] for the full proof that formulae Σ_1^B represent precisely the $\mathtt{NP}$ relations, and more generally for $i > 1$ the Σ_i^B formulae represent the Σ_i^p relations in the polynomial hierarchy and Π_i^B represent the Π_i^p relations.

There are two directions of proof for the representation property. The first is showing that the value of formulae $\phi^* \in \Phi^*$ on its free variables can be checked in the respective complexity class C. This follows from the fact that the formulae of the corresponding descriptive logic Φ can be evaluated on a given structure by an algorithm from C. The terms of formulae in Φ^* all have value at most polynomial in the length of the free string variables (and value of free number variables), and so can be easily computed even in logarithmic time. The rest of the evaluation algorithm usually reduces to propositional satisfiability problem.

The other direction follows from the fact that Φ can express predicates complete for C. In particular, formulae in Φ^* can encode the run of a corresponding resource-bounded Turing machine on its input. As an example, we give the representation theorem for Σ_1^B-Horn formulae; the rest are similar.

Theorem 3.1.5. *A relation $R(z_1, ..., z_k, Y_1, ..., Y_m)$ is in $\mathtt{P}$ iff it is representable by a Σ_1^B-Horn formula ϕ. Further ϕ can be chosen with only one existentially quantified second-order variable, and only two universally quantified first-order variables.*

Proof of theorem. For the if direction, let $\phi(\bar{z}, \bar{Y})$ be a Σ_1^B-Horn formula which represents

$R(\bar{z}, \bar{Y})$. Then ϕ has the form

$$\exists P_1 ... \exists P_r \forall x_1 \leq t_1 ... \forall x_s \leq t_s \psi(\bar{x}, \bar{P}, \bar{z}, \bar{Y}) \tag{3.2}$$

where ψ is Horn with respect to $P_1, ..., P_r$. We outline a polynomial-time algorithm which, given numbers $a_1, ..., a_k$ (coded in unary) and finite sets $B_1, ..., B_m$ (coded by binary strings) determines whether $\phi(\bar{a}, \bar{B})$ is true in the standard model. First note since $\bar{a}$ and $\bar{B}$ are given, each first-order term u in $\psi(\bar{x}, \bar{P}, \bar{a}, \bar{B})$ becomes a polynomial $u(x_1, ..., x_k)$, and the coefficients can be computed in polynomial-time. Each P_i can occur only in the context $P_i(u(\bar{x}))$ for some such term u, and the terms $t_1, ..., t_s$ bounding the x_i's evaluate to constants.

The algorithm proceeds by computing for each possible $\bar{x}$-value $\bar{b} = (b_1, ..., b_s)$, $0 \leq b_i \leq t_i$, a simplified form $\psi[\bar{b}]$ of the instance $\psi(\bar{b}, \bar{P}, \bar{a}, \bar{B})$ of ψ. In this form all first-order terms and all atomic formulae not involving the P_i's are evaluated, and the result is a Horn formula $\psi[\bar{b}]$ with all of its atoms in the list $P_i(0), ..., P_i(T), i = 1, ..., r$, where T is the largest possible argument of any P_i in any instance. By taking the conjunction over all $\bar{b}$ of these instances, we obtain a propositional Horn formula PROP$[\psi, \bar{a}, \bar{B}]$. It is not hard to see that $\phi(\bar{a}, \bar{B})$ is true in the standard model iff PROP$[\psi, \bar{a}, \bar{B}]$ is satisfiable.

Finally, there is a standard polynomial-time algorithm to test satisfiability of a given propositional Horn formula ψ. Namely, initialize a truth assignment α to set all atoms to false. Now repeatedly, for each clause C in ψ not satisfied by the current α, either C has no positive occurrence of an atom P, in which case ψ is unsatisfiable, or C has a unique positive occurrence of some atom P, in which case flip the value of α on P from false to true.

The proof of the only-if direction resembles the proof of Cook's theorem that SAT is NP-complete, and of Fagin's theorem of finite model theory that second-order existential formulae capture NP. Let M be a deterministic Turing machine that recognizes a relation $R(x_1, ..., x_k, Y_1, ..., Y_m)$ within time n^ℓ, where $n = x_1 + ... + x_k + |Y_1| + ... + |Y_m|$ is the length of the input. The entire computation of M on this input can be represented by a two-dimensional array $P(i, j)$ with $t(n)$ rows and columns, for some polynomial t, where the i-th row specifies the tape configuration at time i. (P can be represented by a one-dimensional array using a pairing function.) Thus $R(\bar{x}, \bar{Y})$ is represented by the Σ_1^B-Horn formula

$$\exists P \exists \tilde{P} \forall i \leq t(n) \forall j \leq t(n) \psi(P, \tilde{P}, i, j, \bar{x}, \bar{Y}) \tag{3.3}$$

Here the variable $\tilde{P}$ is forced to be $\neg P$ in the same way that P_{even} and P_{odd} are forced to be complementary in the parity example above. The formula $\psi(P, \tilde{P}, i, j, \bar{x}, \bar{Y})$ is Horn with respect to P and $\tilde{P}$, and each clause specifies a local condition on the computation. These conditions are (1) the first row of P codes the initial tape configuration for the inputs $\bar{x}, \bar{Y}$, (2) for $i < t(n)$ the $i+1$-st row represents the i-th row after one step, and (3) the final state is accepting. To make (2) easier to specify, it is convenient to represent the state at time i at the beginning of row i by a string of fixed length, and after the code for the symbol stored at each tape position there is a bit specifying whether that square is currently scanned by the Turing machine head. In this way rows i and $i+1$ will be identical except for the state codes at the beginning and the bits coding the old and new tape squares scanned.

To see that each clause can be designed to meet the Horn condition of at most one positive occurrence among the atoms of the form $P(u), \tilde{P}(u)$, we include the clause $(\neg P(i,j) \vee \neg\tilde{P}(i,j))$. Then every bit in row 0 is specified using a clause with a positive literal of one of the forms $P(0,u)$ or $\tilde{P}(0,u)$, possibly together with other literals involving input variables. For example, if 15 bits are reserved at the beginning of each row to specify the state, and 3 bits code each tape square, then one of the clauses might be $(5 \leq j \wedge j \leq 5 + x_1 \rightarrow P(0, 3 \cdot j + 1))$. In general every bit in row $i+1$ is specified conditional on a fixed number of bits in row i. A clause is included for each possible state of these conditional bits, and the conditions are specified using $\neg P$ and $\neg\tilde{P}$ as appropriate. In this way at least one of $P(i,j), \tilde{P}(i,j)$ must be true for each (i,j) (and hence exactly one). Note however that if M were nondeterministic, then row $i+1$ would have more than one possible value, and some clauses would require more than one positive literal so the formula would not be Horn.

To meet the "further" condition stated in the theorem, the two arrays P and $\tilde{P}$ can be combined into one array $Q(i,j,k)$, where $k = 0$ for P and $k = 1$ for $\tilde{P}$. □

Note that above proof also shows that every `NP` relation can be represented by a Σ_1^B formula of the form (3.3), except that ψ is not Horn.

3.1.4 Evaluating restricted Σ_1^B formulae

Evaluation algorithms for restricted Σ_1^B formulae follow a general schema. Suppose we need to evaluate a formula of the form

$$\phi(\bar{z}, \bar{Y}) \equiv \exists \bar{P} \forall \bar{x} < \bar{t} \psi(\bar{P}, \bar{x}, \bar{z}, \bar{Y}),$$

where $\bar{t}$ are terms of $\bar{z}$ and $\bar{Y}$, for a given assignment $\bar{z} = \bar{a}$, $\bar{Y} = \bar{B}$. Here, ψ is a CNF with restrictions on occurrences of $\bar{P}$; for example, ψ can be Horn or Krom with respect to $\bar{P}$. Now, the evaluation algorithm follows these steps:

1. Take a conjunction of as many copies of ψ as there are tuples of bounded first-order variables. For example, if ϕ is of the form $\exists \bar{P} \forall x_1 < t_1 \dots \forall x_s < t_s \psi(\dots)$, then there are $t_1 \times \cdots \times t_s$ possible tuples of values for quantified first-order variables.

2. For each copy of $\psi(\bar{x}, \bar{P}, \bar{a}, \bar{B})$, where $\bar{x}$ are now fixed, evaluate all first-order terms and replace them with their values. For example, if there is a clause in ψ of the form $(x_1 = x_2 + x_3 \vee P(x_2 \times x_2 + 2))$, and $x_1 = 5$, $x_2 = 3$ and $x_3 = 2$, then replace it with $(\top \vee P(11))$.

3. Remove all clauses that became true because of the evaluation of first-order atoms, such as the clause $(\top \vee P(11))$.

4. Remove all first-order atoms that evaluated to false. If a clause in some copy of ψ became empty as the result of the evaluation of the first-order terms, then the whole formula is false.

5. If the formula did not become false at the previous step, it is now a CNF with variables of the form $P_i(c)$ for some values c. For example, a formula

$$\exists P_1 \exists P_2 \forall x_1 < 3(x_1 = 2 \vee P_1(x_1) \vee \neg P_2(x_1 + 1))(x_1 + 1 < 5)$$

   becomes $(P_1(0) \vee \neg P_2(1)) \wedge (P_1(1) \vee \neg P_2(2)$. The atoms $0 = 2$ and $1 = 2$ are removed from the clauses, and the clause with $2 = 2$, as well as the clauses $(0 < 5)$, $(1 < 5)$ and $(2 < 5)$, are removed from the formula.

6. Replace every atom of the form $P_i(t_j)$ by a different propositional variable. That is, if there is a subformula $(P_1(2) \vee \neg P_2(2)) \wedge (\neg P_2(3) \vee \neg P_1(2))$, then make it $(p_{1,2} \vee \neg p_{2,2}) \wedge (\neg p_{2,3} \vee p_{1,2})$. Note that even if originally the terms in $P_1(2)$ and $\neg P_1(2)$ were different, but they evaluated to the same value on possibly different tuples, then the two atoms become the same propositional variable.

7. Now the problem is reduced to propositional satisfiability, which is solved by running a respective propositional satisfiability algorithm, depending on the restriction, such as Horn or 2CNF satisfiability. Note that this is the only place in which the form of ψ becomes relevant.

Axioms for + and $\cdot$	B1	$x+1 \neq 0$	B2	$x+1=y+1 \to x=y$
	B3	$x+0=x$	B4	$x+(y+1)=(x+y)+1$
	B5	$x \cdot 0 = 0$	B6	$x \cdot (y+1) = (x \cdot y)+x$
Axioms for $\leq$	B7	$0 \leq x$	B8	$x \leq x+y$
	B9	$x \leq y \wedge y \leq z \to x \leq z$	B10	$(x \leq y \wedge y \leq x) \to x=y$
	B11	$x \leq y \vee y \leq x$	B12	$x \leq y \leftrightarrow x < y+1$
Predecessor	B13	$x \neq 0 \to \exists y(y+1=x)$		
Upper bound	L1	$X(y) \to y < \lvert X \rvert$	L2	$y+1=\lvert X \rvert \to X(y)$

Table 3.1: The 2-BASIC axioms

8. If the resulting formula is satisfiable, then construct witnesses for $\bar{P}$ from the values of corresponding propositional variables. If some variable $p_{i,j}$ does not occur in the formula, take any value for $P_i(j)$ (e.g, false).

Later we will show specifically how to encode formulae of different kinds by a free variable, and how to encode a satisfiability algorithm for each kind by a corresponding formula.

3.2 Systems of arithmetic V-Φ

We will say "a system of bounded arithmetic" to mean a theory over the language of arithmetic, as a set of consequences of explicitly listed axioms and axiom schemas. The theories we are considering are second-order with all quantifiers bounded, axiomatized by axioms similar to those of Peano Arithmetic with a set of axioms corresponding to a restricted version of induction scheme.

In the case of our family V-Φ of systems of arithmetic, each system in the family corresponds to a class Φ of formulae over $\mathcal{L}_A^2$, usually resulting from translation of some class of formulae in the descriptive complexity setting.

Definition 3.2.1. Let Φ be a class of formulae over $\mathcal{L}_A^2$ (that is, in the bounded arithmetic setting). The corresponding system V-Φ is axiomatized by the set of basic axioms 2-BASIC (see table 3.1) together with a comprehension scheme

$$\Phi\text{-COMP} : \exists X \leq b(\forall z < b(X(z) \leftrightarrow \phi(z, \bar{a}, \bar{Y}))),$$

where $\phi(\bar{a}, \bar{Y}) \in \Phi$, and $\bar{x}, \bar{Y}$ are free variables. For every instantiation of free variables there would be a different string X. To be consistent with the common notation, we will use V^0 instead of V-Σ_0^B and V^1 instead of V-Σ_1^B.

Since we are concerned with Σ_1^B-definability, Φ in our framework come from first-order and restricted SO$\exists$ logics. In particular, Φ is restricted Σ_1^B in the sense of Definition 3.1.2, and Φ contains all quantifier-free formulae and prenex formulae with just bounded universal first-order quantifiers. Usually we are interested in the systems powerful enough to handle $\texttt{AC}^0$ reasoning, so Φ has to either include or be able to simulate Σ_0^B. However, our set of function symbols does not include any string operations other than membership and length; that differentiates our setting from the first-order theories and allows us to capture classes weaker than $\texttt{TC}^0$: although string addition is definable by first-order formulae, string multiplication is not.

Example 3.2.1. The main focus of this work is systems V_1-Horn, V-Krom and V-SymKrom, with Φ being, respectively, Σ_1^B-Horn, Σ_1^B-Krom and Σ_1^B-SymKrom. However, all of the V_1^i hierarchy can be thought of as V-Φ with $\Phi \equiv \Sigma_i^B$. In particular, the system V^0, that is V-Φ with $\Phi \equiv \Sigma_0^B$, is the basis of all our systems; it corresponds to the complexity class $\texttt{AC}^0$. Also, the system V^1 which is equivalent in power to S_2^1 is V-Φ with comprehension over Σ_1^B formulae, which represent $\texttt{NP}$ predicates by theorem 3.1.5 (see the comment at the end of the proof).

3.2.1 Properties of systems V-Φ

An induction axiom is a standard feature of most standard systems of bounded arithmetic. We will show here that the axioms for the length function give us induction in V-Φ.

Lemma 3.2.2. *Suppose that Φ contains all first-order universal formulae. Then the least number principle is a theorem of V-Φ.*

$$0 < |X| \to \exists x < |X|(X(x) \land \forall y < x \neg X(y)) \tag{LNP}$$

Proof. By the comprehension schema there is a set Y such that $|Y| \leq |X|$ and for all $z < |X|$, $Y(z) \leftrightarrow \forall i < |X|(X(i) \to z < i)$. Thus the set Y consists of those elements smaller than every element in X. We claim that $|Y|$ satisfies the LNP for X; that is (i) $|Y| < |X|$, (ii) $X(|Y|)$ and (iii) $\forall y < |Y| \neg X(y)$. First suppose that Y is empty. Then

$|Y| = 0$ by B13 and L2. By assumption $0 < |X|$, so (i) holds in this case. Also $X(0)$, since otherwise $Y(0)$ by B7 and the definition of Y, so (ii) holds. Since $\neg y < 0$ by B7 and B10 we conclude (iii) holds vacuously.

Now suppose $Y(y)$ for some y. Then $y < |Y|$ by L1, so $|Y| \neq 0$ so by B13 $|Y| = z+1$ for some z and hence $Y(z)$ by L2. Then $\neg Y(z+1)$ by L1. Thus $X(z+1)$ by B11, B12 and the definition of Y, so (ii) holds. Also $\neg X(z)$, so (i) holds. Finally (iii) holds by the definition of Y and B10. □

Lemma 3.2.3. *Suppose that Φ contains all first-order universal formulae. Then induction on the length of a string is a theorem of V-Φ.*

$$(X(0) \wedge \forall y < z(X(y) \to X(y+1))) \to X(z) \qquad \text{(Induction)}$$

Proof. We show that negation of induction implies negation of LNP. By negation of induction, we have $X(0) \wedge \forall y < z(X(y) \to X(y+1))$, and $\neg X(z)$. By the comprehension schema there is a set Y such that $\forall y < z+1(Y(y) \leftrightarrow \neg X(y))$. Then $Y(z)$, so $0 < |Y|$. By LNP Y has a least element y_0. Then $y_0 \neq 0$ because $X(0)$, so $y_0 = x_0 + 1$ for some x_0, by B13. But then we must have $X(x_0)$ and $\neg X(x_0+1)$, which contradicts our assumption. □

It is easy to generalize Lemma 3.2.3 to allow induction with an arbitrary value k as a basis, not just $k = 0$.

If follows from the above lemma that each of the theories that we have presented proves an induction axiom for each formula in its comprehension scheme. Using one instance of Φ-comprehension to define $X \leftrightarrow \phi$, we get

Corollary 3.2.4. *V-Φ proves the induction axioms for formulae from Φ.*

$$(\phi(0) \wedge \forall y < z(\phi(y) \to \phi(y+1))) \to \phi(z) \qquad (\Phi\text{-induction})$$

where $\phi \in \Phi$.

The system V^0 with comprehension over Σ_0^B formulae (first-order with free second-order variables), which correspond exactly to DLOGTIME-uniform $\mathtt{AC}^0$ relations, is contained in all our other systems. It is already powerful enough to prove the simple properties of addition and multiplication, such as commutativity and associativity. One very useful property is the existence of a pairing function. It can be used to treat second-order objects as multi-dimensional arrays, instead of one-dimensional strings or sets.

Definition 3.2.5 (Pairing function). The pairing function $\langle x, y\rangle$ can be defined by

$$\langle x, y\rangle = (x+y)(x+y+1) + 2y \qquad \text{(Pairing function)}$$

This function is a one-one map from $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$, and it is represented by a term in our language. It is easily generalized to k-tuples by defining $\langle x_1, ..., x_k\rangle$ recursively: setting $\langle x\rangle = x$, and $\langle x_1, ..., x_{k+1}\rangle = \langle\langle x_1, ..., x_k\rangle, x_{k+1}\rangle$

Thus, any finite set P can be treated as a set of k-tuples of variables; $P(x_1, ..., x_k)$ is defined to be $P(\langle x_1, ..., x_k\rangle)$.

Notation 3.2.6. We use $P^{[b]}$ to denote the "b-th row" when P is being used as a 2-dimensional array. If $\phi(P)$ is a formula with no occurrence of $|P|$, then $\phi(P^{[b]})$ is obtained from $\phi(P)$ by replacing every atomic formula $P(t)$ by $P(b, t)$ (i.e. $P(\langle b, t\rangle)$: see (3.2.5)).

Note that in the descriptive complexity setting the logic with only unary relations is weaker than one with relations of arbitrary arity. But since in bounded arithmetic there is no strict bound on the size of the structure, this is not a restriction. Therefore, a class of formulae in the descriptive complexity setting with second-order variables of arbitrary (constant) arity is equivalent to a class of formulae in the bounded arithmetic setting with unary second-order variables, which can be interpreted as k-ary for a constant k by using the pairing function defined above.

Although 2-$BASIC$ does not include an explicit induction axiom, L2 asserts that a nonempty set has a largest element. This can be turned into a least number principle, from which induction follows.

Standard arguments show that induction on open formulae using axioms B1 to B13 is enough to prove simple algebraic properties of $+$ and $\cdot$ such as commutativity, associativity, distributive laws, and cancellation laws involving $+$, $\cdot$, and $\leq$. Hence all of our theories prove these properties, and in the sequel we take them for granted. These simple properties suffice to prove that the pairing function defined in (3.2.5) is one-one, so these theories all prove

$$\langle x_1, ..., x_k\rangle = \langle x'_1, ..., x'_k\rangle \rightarrow (x_1 = x'_1 \wedge ... \wedge x_k = x'_k) \qquad (3.4)$$

A useful application of the pairing function is a slightly more general version of the comprehension axiom. If Φ includes all open and universally first-order quantified formulae, we get the following statement.

Lemma 3.2.7 (k-ary Comprehension). *If $\phi(x_1, ..., x_k) \in \Phi$ with no free occurrence of Y, then V-Φ proves the k-ary comprehension formula*

$$\exists Y \leq \langle b_1, ..., b_k \rangle \forall x_1 < b_1 ... \forall x_k < b_k (Y(x_1, ..., x_k) \leftrightarrow \phi(x_1, ..., x_k)) \tag{3.5}$$

Proof outline. The proof is by induction on the number of variables k. For the base case, suppose that the formula is $\phi(x, y)$. Now by induction on y we can show that there exists $Y(x, y)$. For the base case we have comprehension over $\psi(i) \equiv (i = \langle x, 0 \rangle \wedge X(x))$, where $X(x) \leftrightarrow \phi(x, 0)$ exists by Φ-comprehension. For the induction step, take the formula $\psi(i) \equiv \forall x < n \forall z \leq y (i = \langle x, z \rangle \wedge (z < y \wedge Y_y(x, z) \vee z = y \wedge Z(z)))$, where $Y(x, y)$ exists by induction hypothesis and $Z(z)$ is obtained by comprehension over $\phi(x, y + 1)$ with y fixed and x as an index. Since by assumption we have comprehension over universal closure of quantifier-free formulae, we can do comprehension over ψ.

The case of $k > 2$ follows by induction on k. Take the characteristic string of ϕ on the first $k - 1$ variables, and apply comprehension over two variables to get the k-ary comprehension. □

The following lemma gives another application of the pairing function.

Lemma 3.2.8. *Every formula $\phi \in \Phi$ is provably equivalent in V-Φ to a formula in Φ with at most one second-order existential quantifier. Specifically, V-Φ proves*

$$\exists P_1 ... \exists P_m \phi(P_1, ..., P_m) \leftrightarrow \exists P \phi(P^{[1]}, ..., P^{[m]})$$

Proof. For the left-to-right direction, use k-ary comprehension (Lemma 3.2.7) to define P satisfying

$$P(i, x) \leftrightarrow (i = 1 \wedge P_1(x)) \vee ... \vee (i = m \wedge P_m(x))$$

For the other direction, for $i = 1, ..., m$ use comprehension over open formulae to define P_i such that $P_i(x) \leftrightarrow P(i, x)$. □

Definition 3.2.9. We use notation $\Sigma_0^B(\Phi)$ to refer to the closure of Φ under Σ_0^B operations: that is, under $\vee, \wedge, \neg$ and bounded first-order $\forall$ and $\exists$.

For example, if $\phi_1(i)$ and $\phi_2(i)$ are formulae from Φ, then the formula $\forall i < t(\phi_1(i) \wedge \neg\phi_2(i))$ is in the Σ_0^B closure of Φ. Usually, $\Sigma_0^B(\Phi)$ is not equal to Φ syntactically ($\Sigma_0^B(\Sigma_0^B)$ is an exception), but in some cases for every formula in $\Sigma_0^B(\Phi)$ there is an equivalent formula in Φ. We will extensively use this property later.

It is easy to show that V^0, and thus theories V-Φ extending V^0, prove the basic properties of $+$ and $\cdot$ such as commutativity and associativity. The following theorem summarizes some other useful properties that V-Φ proves. We will often refer to it, or use it implicitly, throughout the thesis.

Theorem 3.2.10. *If $V^0 \subseteq V$-Φ, then V-Φ proves induction and comprehension over $\Sigma_0^B(\Phi)$ formulae.*

The main part of the proof is to show that V-Φ that contains V^0 proves comprehension over $\Sigma_0^B(\Phi)$. Then, we extend the statement of corollary 3.2.4 to $\phi \in \Sigma_0^B(\Phi)$ rather than just $\phi \in \Phi$, alluding to the fact that comprehension over $\Sigma_0^B(\Phi)$ is available.

Lemma 3.2.11. *If $V^0 \subseteq V$-Φ, then V-Φ has comprehension over $\Sigma_0^B(\Phi)$ formulae.*

Proof. We want to show that if $\phi^*(\bar{a}, \bar{Y}) \in \Sigma_0^B(\Phi)$, then

$$V\text{-}\Phi \vdash \exists Z \leq t \forall i < t(Z(i)\phi^*(i, \bar{a}, \bar{Y}).$$

The proof is by structural induction on ϕ.

The base case is when $\phi^* = \phi$ where $\phi \in \Phi$. Then there is comprehension over ϕ by definition of V-Φ.

Now suppose that there is comprehension over $\phi_1, \phi_2 \in \Sigma_0^B(\Phi)$. Then there are strings $X_1 \leq b, X_2 \leq b$ that are characteristic strings of ϕ_1 and ϕ_2, respectively. Now, Σ_0^B comprehension applies to formulae ϕ' of the form $X_1(i) \vee X_2(i)$, $X_1(i) \wedge X_2(i)$, $\neg X_1(i)$. In case of $\exists x < t\phi(x, i)$ we use k-ary comprehension 3.2.7 to obtain strings $X(x, i)$. Now, we apply Σ_0^B comprehension to $\exists x < tX(x, i)$ and $\forall x < tX(x, i)$. This completes the proof. □

The statement of lemma 3.2.11 is not very strong. It only shows that on a meta-level a proof can be carried out in V-Φ using multiple application of the comprehension axiom. In particular, a statement that every formula in $\Sigma_0^B(\Phi)$ is equivalent to a formula from Φ, which we sometimes need, is a much stronger statement which requires additional closure conditions on Φ.

In theories below V^1 it is not possible to prove replacement axioms of the form $\forall y < t\exists \bar{P}\phi(y, \bar{P}) \leftrightarrow \exists \bar{P} \forall y < t\phi(y, \bar{P}^{[y]})$ if ϕ is an arbitrary Σ_1^B formula. However, if $\phi \in \Phi$ and original Φ is a restricted SO$\exists$ (see definition 2.2.5), then it is possible to prove a version of the replacement axiom.

Lemma 3.2.12 (Replacement). *Let Φ be a class of restricted Σ_1^B formulae. Then for every formula $\exists \bar{P}\phi(y, \bar{P}) \in \Phi$, where ϕ can have additional free variables, V-Φ proves*

$$\forall y < t \exists \bar{P}\phi(y, \bar{P}) \leftrightarrow \exists \bar{P} \forall y < t \phi(y, \bar{P}^{[y]}) \qquad \text{(Replacement)}$$

where $\bar{P}^{[y]}$ is $P_1^{[y]}, ..., P_k^{[y]}$.

Proof. Appealing to the pairing function and the fact that every term t can be bounded by the value b of the term on the maximal values of its variables, it is sufficient to show

$$V\text{-}\Phi \vdash \forall y < b \exists P \phi(y, P) \leftrightarrow \exists P \forall y < b \phi(y, P^{[y]})$$

To prove the right-to-left implication, assume that P satisfies the existential quantifier on the right and suppose $y < b$. Use the V-Φ comprehension axiom to define P' such that $\forall i < b(P'(i) \leftrightarrow P(y, i))$ Then P' satisfies the existential quantifier on the left.

The left-to-right direction is proven by the induction on the number of second-order existential quantifiers in ϕ. Define $\psi(z) \equiv \exists P \forall y < z \phi(y, P^{[y]})$. If ϕ does not have any second-order existential quantifiers, then $\psi(z) \in \Phi$, since Φ is restricted Σ_1^B. Otherwise, use induction hypothesis to say that ψ is equivalent to a formula from Φ, by moving existential second-order quantifiers of ϕ past $\forall y < z$. Now we may use the IND scheme (Corollary 3.2.4) to conclude $\psi(b)$. It suffices to prove that the LHS $\forall y < b \exists P \phi(y, P)$ implies the basis and induction steps. The basis is trivial, since when $b = 0$ $\psi(0)$ is vacuously true.

For the induction step, by the induction hypothesis $\psi(z)$ we may assume $z < b$ and P satisfies $\forall y < z \phi(y, P^{[y]})$. Setting $y = z$ in the LHS we have Q such that $\phi(z, Q)$. Now we use k-ary comprehension 3.2.7 on two variables (y, i) to define $P'(y, i)$ by

$$P'(y, i) \leftrightarrow \begin{cases} P(y, i) & \text{if } y < z \\ Q(i) & \text{if } y = z \end{cases}.$$

Then we conclude in V-Φ the formula $\forall y < z + 1 \phi(y, P'^{[y]})$, and hence $\psi(z + 1)$. $\square$

3.3 Definability

Our complexity classes, both in the descriptive complexity setting and in bounded arithmetic, are relational. However, in bounded arithmetic we would like to be able to talk about functions. We can use relations as graphs to define number functions and as bit graphs to define string functions. The following definition is very general, but sometimes does not produce a robust function class: for example, there is nothing in this definition that would force the functions to be closed under composition. In order to make the function classes defined this way meaningful, we will need additional restrictions.

Definition 3.3.1. Let C be a complexity class. We define the corresponding class FC of functions of C as follows:

A number function $f : \mathbb{N}^k \times (\{0,1\}^*)^l \to \mathbb{N}$ is in FC iff there is a relation R in C and a polynomial p such that

$$f(\bar{x}, \bar{Y}) = \min z < p(\bar{x}, |\bar{Y}|) R(z, \bar{x}, \bar{Y})$$

A string function $F : \mathbb{N}^k \times (\{0,1\}^*)^l \to \{0,1\}^*$ is in FC iff there is a relation R in C and a polynomial p such that

$$F(\bar{x}, \bar{Y})(i) \leftrightarrow i < p(\bar{x}, |\bar{Y}|) \wedge R(i, \bar{x}, \bar{Y}) \text{ for all } i \in \mathbb{N}$$

For the string function, we are only concerned with the bits with indices smaller than $p(\bar{x}, \bar{Y})$. Therefore, a string corresponding to the value of a function will be of length less than $p(\bar{x}, \bar{Y})$. In particular, by the length axioms, all bits with indices larger than $p(\bar{x}, \bar{Y})$ are 0.

A power of a system of arithmetic can be defined either as a class of relations the system proves Δ_1^B-definable , or a class of Σ_1^B-definable functions the system proves total.

Definition 3.3.2. A relation $R(\bar{x}, \bar{Y})$ is Δ_1^B-definable in V-Φ iff there exists formulae $\phi, \tilde{\phi} \in \Sigma_1^B$ such that $R(\bar{x}, \bar{Y})$ is represented by $\phi(\bar{x}, \bar{Y})$ and

$$V\text{-}\Phi \vdash \phi(\bar{x}, \bar{Y}) \leftrightarrow \neg\tilde{\phi}(\bar{x}, \bar{Y})$$

A number (resp. string) function f (resp. F) is Σ_1^B-definable in V-Φ if it has a defining axiom

$$z = f(\bar{x}, \bar{Y}) \leftrightarrow \phi(z, \bar{x}, \bar{Y}) \qquad (\text{resp. } Z = F(\bar{x}, \bar{Y}) \leftrightarrow \phi(Z, \bar{x}, \bar{Y}))$$

with $\phi \in \Sigma_1^B$ such that

$$V\text{-}\Phi \vdash \forall \bar{x} \forall \bar{Y} \exists! z \phi(z, \bar{x}, \bar{Y}) \qquad (\text{resp. } V\text{-}\Phi \vdash \forall \bar{x} \forall \bar{Y} \exists! Z \phi(Z, \bar{x}, \bar{Y})).$$

Using definition 3.3.2, we can state the definition of “capture” in the bounded arithmetic setting. This gives us a way of measuring the power of a system of arithmetic.

Definition 3.3.3 (Capture). A system of arithmetic V *captures* a complexity class C if the class of Σ_1^B-definable functions of V is exactly FC.

Note that this is quite different from the descriptive complexity notion of "capture". The reason we are using the same word is that in both cases we are relating a logic (system of arithmetic) and a complexity class; "capture" here is a generic name for such a connection.

Lemma 3.3.4. *There is an exact correspondence between Δ_1^B-definable relations and a Σ_1^B-definable boolean functions of a system (characteristic functions).*

Proof. We describe this correspondence following [Coo04]. Let $R(\bar{x}, \bar{Y})$ be representable by a Σ_1^B formula $\phi(\bar{x}, \bar{Y}) \equiv \exists \bar{Z} \psi(\bar{x}, \bar{Y}, \bar{Z})$, and let $\tilde{\phi}(\bar{x}, \bar{Y}) \equiv \exists \bar{Z} \tilde{\psi}(\bar{x}, \bar{Y}, \bar{Z})$, where $\psi, \tilde{\psi}$ are Σ_0^B, such that

$$V\text{-}\Phi \vdash \phi(\bar{x}, \bar{Y}) \leftrightarrow \neg\tilde{\phi}(\bar{x}, \bar{Y}). \tag{3.6}$$

The defining axiom for the characteristic function of R is $z = f(\bar{x}, \bar{Y}) \leftrightarrow \exists \bar{Z}(z = 1 \wedge \psi(\bar{x}, \bar{Y}, \bar{Z})) \vee (z = 0 \wedge \tilde{\psi}(\bar{x}, \bar{Y}, \bar{Z}))$. The existence of z follows from the right-to-left direction of the equation 3.6, and the uniqueness from the left-to-right direction.

For the other direction, suppose that $f(\bar{x}, \bar{Y})$ is definable by an axiom $z = f(\bar{x}, \bar{Y}) \leftrightarrow \phi^*(z, \bar{x}, \bar{Y})$ and $V\text{-}\Phi \vdash \exists! z \phi^*(z, \bar{x}, \bar{Y})$. Then $R(\bar{x}, \bar{Y})$ is defined by the formula $\phi(\bar{x}, \bar{Y}) \equiv \exists z(z \neq 0 \wedge \phi^*(z, \bar{x}, \bar{Y}))$ and $\tilde{\phi}(\bar{x}, \bar{Y}) \equiv \phi^*(0, \bar{x}, \bar{Y})$. □

Now we need a method to introduce function symbols into the system. A traditional way is to use recursion-theoretic characterizations of classes, such as Cobham's characterization of FP as a class of functions closed under $\mathtt{AC}^0$ operations and limited recursion on notation. Here, however, we use the fact that our systems are built on classes of formulae which express complexity classes in the descriptive setting. That is, the relation in Definition 3.3.1 are representable by formulae from Φ.

Definition 3.3.5. Let Φ represent a complexity class C. We define a corresponding function class FC by setting

$$z = f(\bar{x}, \bar{Y}) \leftrightarrow \phi(z, \bar{x}, \bar{Y}) \qquad F(\bar{x}, \bar{Y})(i) \leftrightarrow i < t \wedge \phi(i, \bar{x}, \bar{Y})$$

where f and F are number and string functions, respectively, and $\phi \in \Phi$. That is, we define functions by formulae from Φ by stating that the graphs of number functions and bit graphs of string functions are representable by formulae from Φ.

What we would like to have is a property that a class of functions defined in this manner is robust: i.e., closed under composition, substitution of a term for a variable, and Σ_0^B operations. There are some natural complexity classes for which this is not the case. For example, $\mathtt{NP}$ predicates translate into Σ_1^B formulae defining a class of functions FNP which is not closed under complement unless $\mathtt{NP} = \mathtt{coNP}$. It seems that closure under complementation is a necessary condition for a complexity class to have a robust function class. If Φ is not syntactically closed under complementation and boolean operations, we need to prove the closures within V-Φ.

The first requirement that we will have in most of our systems will be not only that they include V^0, but also that they are closed under Σ_0^B operations. The respective complexity class, in this case, has to be closed under uniform $\mathtt{AC}^0$ reductions, or, equivalently, first-order reductions. Recall that Σ_0^B is the first-order logic translated into the language of bounded arithmetic.

We start with a weaker notion of Σ_0^B definitions. Let $\mathcal{L} \supseteq \mathcal{L}_A^2$ be a collection of two-sorted functions and relations. Assume that functions in $\mathcal{L}$ are closed under composition and substitution of a term for a variable.

Definition 3.3.6. A string function $F(\bar{x}, \bar{Y})$ is in $\Sigma_0^B(\mathcal{L})$ if there is a formula $\phi(i, \bar{x}, \bar{Y})$ which is Σ_0^B with occurrences of symbols from $\mathcal{L}$ and a term $t(\bar{x}, \bar{Y})$ over $\mathcal{L}_A^2$ such that

$$F(\bar{x}, \bar{Y})(i) \leftrightarrow i < t(\bar{x}, \bar{Y}) \wedge \phi(i, \bar{x}, \bar{Y}).$$

The definition for the number function is similar:

$$f(\bar{x}, \bar{Y}) = i \leftrightarrow i < t(\bar{x}, \bar{Y}) \wedge \phi(i, \bar{x}, \bar{Y}).$$

That is, ϕ is a bit-graph of a string function or a graph of a number function.

Now, allowing repetitions of this operation, we obtain the notion of an $\mathtt{AC}^0$ reduction.

Definition 3.3.7. A string function $F(\bar{x}, \bar{Y})$ is $\mathtt{AC}^0$ reducible to $\mathcal{L}$ iff there is a sequence $F_1 \dots F_n$ of string functions such that $F_n = F$ and F_i is in $\Sigma_0^B(\mathcal{L} \cup \{F_1 \dots F_{i-1}\})$ for $i = 1, \dots, n$. A number function $f(\bar{x}, \bar{Y})$ is $\mathtt{AC}^0$-reducible to $\mathcal{L}$ if there is a string function $F(\bar{x}, \bar{Y})$ that is $\mathtt{AC}^0$-reducible to $\mathcal{L}$ and $f(\bar{x}, \bar{Y}) = |F(\bar{x}, \bar{Y})|$.

Definition 3.3.8. We say that $\mathcal{L}$ is closed under $\mathtt{AC}^0$ reductions iff for any string or number function $F(\bar{x}, \bar{Y})$, it is $\mathtt{AC}^0$-reducible to $\mathcal{L}$ iff $F(\bar{x}, \bar{Y}) \in \mathcal{L}$. We denote an $\mathtt{AC}^0$ closure of $\mathcal{L}$ by $\mathtt{AC}^0(\mathcal{L})$.

The definition 3.3.6 is, in general, weaker than the notion of an $\mathtt{AC}^0$ reduction. One example of it is the class $\mathtt{TC}^0$, in which everything can be $\mathtt{AC}^0$-reduced to MAJORITY function, but it is not known if it is possible to capture the whole class without nesting calls to MAJORITY. However, the notions of *closure* under $\mathtt{AC}^0$ reductions and Σ_0^B closure coincide, because if $\Sigma_0^B(\mathcal{L}) = \mathcal{L}$, then nesting the definitions does not take us outside of $\mathcal{L}$. See, for example, [NC04] for a discussion of this.

We need to be able to talk about the $\mathtt{AC}^0$ closure of a class of formulae, whereas the definition above is stated for functions. A class Φ of formulae is closed under $\mathtt{AC}^0$ reduction if any Σ_0^B combination of formulae from Φ is equivalent to a formula from Φ (see definition 3.2.9). We can restate this definition by introducing a string function symbol with its bitgraph a formula from Φ for every formula from Φ, obtaining the language $\mathcal{L} = \mathcal{L}_A^2 \cup \{F_\phi\}$, applying the definition of $\mathtt{AC}^0$ reducibility, and then stating that if a resulting function symbol is in $\mathcal{L}$, it must have as its bitgraph a formula from Φ.

We are now ready to state the "robustness" properties that we assume for our systems. Note that even though the properties are stated for the classes of formulae, they have to apply to the original classes as well. For example, if Σ_1^B-Horn is closed under $\mathtt{AC}^0$ reductions, then $\mathtt{P}$ definitely is.

Property 1 (Closure). *Let Φ represent a complexity class C and let FC be as in the definition 3.3.1. Then the* closure property *holds if Φ is closed under* $\mathtt{AC}^0$ *reductions, that is, $\Sigma_0^B(\Phi) \equiv \Phi$. In particular FC is closed under composition and substitution of a term for a variable. In addition, Φ is* strongly *closed if for every $\phi^* \in \Sigma_0^B(\Phi)$ there exists $\phi \in \Phi$ such that $V\text{-}\Phi \vdash \phi^* \leftrightarrow \phi$.*

In particular, then the corresponding C is closed under complementation and Φ extends Σ_0^B (that is, defines all of first-order). For some Φ, notably restricted Σ_1^B, it is not syntactically true that $\Sigma_0^B \subseteq \Phi$, but it can be proved that for any Σ_0^B formula there is an equivalent formula of Φ.

It seems that Property 1, although strong, is not sufficient for our purposes. It is not enough to be able to evaluate the formulae and their combinations within the corresponding complexity class. In addition to that, we need to be able to construct, within the same class, the "certificates" or "proofs" that the formulae are true or false. This leads to an additional property, that we will call "Constructiveness".

Definition 3.3.9. Existential quantifiers in a formula $\exists\bar{Z}\phi(\bar{Z},\bar{x},\bar{Y})$ are said to be *witnessed* in V-Φ by functions $\bar{F}(\bar{x},\bar{Y})$ with defining axioms $AX(\bar{F})$ if $V\text{-}\Phi, AX(\bar{F}) \vdash \phi(\bar{F}(\bar{x},\bar{Y}),\bar{x},\bar{Y})$.

Example 3.3.1 *(Witnessing parity).* Recall the Σ_1^B-Horn formula for the PARITY problem from the Example 3.1.1. Note that without the clause $(P_{odd}(|X|))$ the formula is always true; moreover, it can be proven in V_1-Horn (by induction on the length of X). Call this formula PARITY-COUNT(X).

To compute $P_{odd}(k)$ and $P_{even}(k)$ all we need to know is the parity of the prefix of X of length k. Let formula PARITY-PREFIX-ODD(k,X) be PARITY with $|X|$ replaced with k in $\forall i < |X|$ and in $P_{odd}(|X|)$, and P_{even} and P_{odd} renamed to P_e and P_o. Let PARITY-PREFIX-EVEN(k,X) be PARITY-PREFIX-ODD(k,X) with clause $(P_o(k))$ replaced with $(P_e(k))$. It is easy to see that $P_{odd}(i)$ and $P_{even}(i)$ in PARITY-COUNT(X) hold iff so do PARITY-PREFIX-ODD(i,X) and PARITY-PREFIX-EVEN(i,X), respectively. Also, note that PARITY-PREFIX-ODD(i,X) and PARITY-PREFIX-EVEN(i,X) are Σ_1^B-Horn formulae.

Introduce the witnessing functions F_{odd} and F_{even} for P_{odd} and P_{even}, respectively, in PARITY-COUNT by the following defining axioms:

$$\begin{aligned} AX(F_{odd}): &\quad F_{odd}(X)(i) \leftrightarrow i \leq |X| \wedge \text{PARITY-PREFIX-ODD}(i,X) \\ AX(F_{even}): &\quad F_{even}(X)(i) \leftrightarrow i \leq |X| \wedge \text{PARITY-PREFIX-EVEN}(i,X) \end{aligned}$$

Now,

$$\begin{aligned} &V_1\text{-Horn}, AX(F_{odd}), AX(F_{even}) \vdash \\ &\forall i < |X| F_{even}(X)(0) \wedge \neg F_{odd}(X)(0) \wedge (\neg F_{even}(X)(i+1) \vee \neg F_{odd}(X)(i+1)) \\ &\wedge (F_{even}(X)(i) \wedge X(i) \rightarrow F_{odd}(X)(i+1)) \wedge (F_{odd}(X)(i) \wedge X(i) \rightarrow F_{even}(X)(i+1)) \\ &\wedge (F_{even}(X)(i) \wedge \neg X(i) \rightarrow F_{even}(X)(i+1)) \wedge (F_{odd}(X)(i) \wedge \neg X(i) \rightarrow F_{odd}(X)(i+1)) \end{aligned}$$

Since F_{odd} and F_{even} are bit-defined by Σ_1^B-Horn formulae, they are polynomial-time functions. So not only V_1-Horn can prove that P_{odd} and P_{even} exist, but also that the values of P_{odd} and P_{even} can be computed by polynomial-time functions.

The idea behind the constructiveness property is that whenever a formula from Φ is true (resp. false) there is a (short) proof (resp. counterexample) for that. Additionally, these proof and counterexamples can be constructed from a first-order (that is, Σ_0^B) combination of functions in the corresponding class FC.

Property 2 (Constructiveness). *Let Φ be a class of restricted Σ_1^B formulae, and let Φ represent C. This Φ has the* constructiveness property *if the following two conditions hold. Firstly, every $\phi \in \Phi$ defines a relation R that is Δ_1^B-definable in V-Φ, with ϕ being its Σ_1^B definition. That is, for every $\phi \in \Phi$ there exists $\tilde{\phi} \in \Sigma_1^B$ such that $V\text{-}\Phi \vdash \phi(\bar{a}, \bar{Y}) \leftrightarrow \neg\tilde{\phi}(\bar{a}, \bar{Y})$. Secondly, there are witnessing functions $\bar{F}$ with bit graphs in $\Sigma_0^B(\Phi)$ such that $\bar{F}(\bar{a}, \bar{Y})$ witness the existential quantifiers of the prenex form of $\phi \vee \tilde{\phi}$.*

Remark 3.3.10. If, additionally, Φ is strongly closed, that is, has the property 1, then the conclusion of the constructiveness property can be stated simpler as follows.

For every $\phi \equiv \exists \bar{P}\psi(\bar{P}, \bar{a}, \bar{Y}) \in \Phi$ such that $V\text{-}\Phi \vdash \phi$ there are functions $\bar{F}$ witnessing $\bar{P}$ such that bitgraphs of $\bar{F}$ are in Φ. It is enough to consider ϕ-theorems of V-Φ because if Φ is closed, then $\tilde{\phi} \in \Phi$ and so is $\phi \vee \tilde{\phi}$. Also, the assumption that bitgraphs of $\bar{F}$ are in $\Sigma_0^B(\Phi)$ becomes bitgraphs $\in \Phi$.

Another way of stating the constructiveness property is to say that the witnesses (satisfying assignments and proofs of tautologies, in case of satisfiability problems) can be computed in $\mathtt{AC}^0$ with oracle access to C (or, equivalently, to the validity problem of Φ, since it is complete for C). For strongly closed classes, such as NL and P, constructiveness states that the witnesses can be computed within the class. For weakly closed classes, such as SL, the constructiveness property states that witnesses can be computed within the class, but it is not provable in the corresponding system.

In the systems that we are considering Φ is restricted Σ_1^B. Then by Lemma 3.2.8 every $\phi \in \Phi$ is equivalent to a formula of the form $\exists P\psi(P, \bar{a}, \bar{Y})$, where ψ does not have second-order quantifiers. Suppose that Φ represents a class of Δ_1^B-definable relations in V-Φ, so for every $\phi \in \Phi$ there exists its opposite $\tilde{\phi} \equiv \exists Q\tilde{\psi}(Q, \bar{a}, \bar{Y})$, where $\tilde{\phi} \in \Sigma_1^B$, such that $V\text{-}\Phi \vdash \phi(\bar{a}, \bar{Y}) \leftrightarrow \neg\tilde{\phi}(\bar{a}, \bar{Y})$. Then there exists a witnessing function $F(\bar{a}, \bar{Y})$ such that $V\text{-}\Phi, AX(F) \vdash \psi(F(\bar{a}, \bar{Y}), \bar{a}, \bar{Y}) \leftrightarrow \neg\tilde{\psi}(F(\bar{a}, \bar{Y}), \bar{a}, \bar{Y})$ where $AX(F)$ is $F(\bar{a}, \bar{Y})(i) \leftrightarrow \phi_F(i, \bar{a}, \bar{Y})$ for $\phi_F \in \Sigma_0^B(\Phi)$.

We would like to work with formulae that contain function symbols. For that, we define a system $V\text{-}\Phi(FC)$ by augmenting V-Φ with a function symbol for every function definable by a formula from Φ using definition 3.3.5. That is, string functions are introduced by their bit graphs and number functions by their graphs. Since V-Φ has comprehension for Φ, $V\text{-}\Phi(FC)$ is conservative over V-Φ. Moreover, the following lemma holds:

Lemma 3.3.11. *If $V^0 \subseteq V\text{-}\Phi$, then $V\text{-}\Phi(\Sigma_0^B(FC))$ is conservative over V-Φ.*

Proof. In order to prove that adding function symbols to a system results in a conservative extension we need to show the following. Let $F(\bar{a}, \bar{X})$ be a function with defining axiom $AX(F) : F(\bar{a}, \bar{Y}) = Z \leftrightarrow \phi^*(Z, \bar{a}, \bar{X})$ (we make no assumptions about the structure of ϕ^* at this point). Then adding F to a theory V results in a conservative extension if

$$V + AX(F) \vdash \forall \bar{a} \forall \bar{X} \exists ! Z \phi^*(Z, \bar{a}, \bar{X}).$$

Then we can argue that every model of V can be extended to a model of $V + AX(F)$.

Our goal is to show that V-$\Phi(\Sigma_0^B(FC))$ is a conservative extension of V-Φ. First, note that functions from FC are bit-definable by formulae from Φ, therefore V-Φ proves the existence and uniqueness of Z satisfying $\phi^*(Z, \bar{a}, \bar{X}) \leftrightarrow |Z| \leq t \wedge \forall i < t(Z(i) \leftrightarrow \phi(i, \bar{a}, \bar{X}))$. The existence of such Z follows from Φ comprehension axiom.

Now suppose that the bit-defining axiom for F is $\phi \in \Sigma_0^B(FC)$. Then the proof is by structural induction on ϕ using Σ_0^B-COMP. We omit the technical details. □

It is useful to note that lemma 3.3.11, as stated, does not show that V-$\Phi(\mathtt{AC^0}(FC))$ is conservative over V-Φ. For that we need to consider the case of function composition, which requires additional properties on Φ.

The following lemma is essential in the proof of both directions of the definability theorem. This is a version of the Replacement Lemma (lemma 3.2.12).

Lemma 3.3.12. *Let Φ be restricted Σ_1^B, $V^0 \subseteq V$-Φ, and let Φ capture FC. Suppose that Φ is constructive. That is, for every $\phi \equiv \exists \bar{X} \psi(i, \bar{X}, \bar{a}, \bar{Y}) \in \Phi$ there exists $\tilde{\phi} \equiv \exists \bar{X} \tilde{\psi}(i, \bar{X}, \bar{a}, \bar{Y}) \in \Sigma_1^B$ such that V-$\Phi \vdash \phi \leftrightarrow \neg\tilde{\phi}$ and the quantified variables $\bar{X}$ in $\exists \bar{X}(\psi(i, \bar{X}, \bar{a}, \bar{Y}) \vee \tilde{\psi}(i, \bar{X}, \bar{a}, \bar{Y}))$ can be witnessed by functions $\bar{F}(i, \bar{a}, \bar{Y})(j)$ with bit-defining axioms $\phi_F(j, i, \bar{a}, \bar{Y}) \in \Sigma_0^B(\Phi)$. Then V-Φ proves the following equivalence, for any free variables t and Z:*

$$\begin{aligned} V\text{-}\Phi \vdash & \forall i < t \exists \bar{X}_1 \exists \bar{X}_2 (Z(i) \wedge \psi(i, \bar{X}_1, \bar{a}, \bar{Y}) \vee (\neg Z(i) \wedge \tilde{\psi}(i, \bar{X}_2, \bar{a}, \bar{Y}))) \\ \leftrightarrow & \exists \bar{W} \exists \bar{U} \forall i < t(Z(i) \wedge \psi(i, \bar{W}^{[i]}, \bar{a}, \bar{Y}) \vee (\neg Z(i) \wedge \tilde{\psi}(i, \bar{U}^{[i]}, \bar{a}, \bar{Y}))) \end{aligned} \tag{3.7}$$

Proof. The right-to-left direction of the proof is trivial. The bulk of the proof is the left-to-right direction.

By pairing, we can assume that ϕ and $\tilde{\phi}$ have only one existential second-order quantifier; the proof easily generalizes to multiple quantifiers, but the presentation with one quantifier is cleaner. Thus, we are proving the statement for ϕ and $\tilde{\phi}$ with a single quantified second-order variable X and its witnessing function F, and for a single array variable W.

Recall that there are formulae $\phi_F(j,i,\bar{a},\bar{Y}) \in \Sigma_0^B(\Phi)$ and $\phi_G(j,i,\bar{a},\bar{Y}) \in \Sigma_0^B(\Phi)$ which are the bit-defining axioms for functions F and G witnessing X_1 and X_2 in $\phi \vee \tilde{\phi}$, respectively. Consider the first formula in the equation 3.7 with occurrences of X_1 replaced by $F(i,\bar{a},\bar{Y})$ and occurrences of X_2 by $G(i,\bar{a},\bar{Y})$. Then this formula becomes $\Sigma_0^B(\Phi)$.

Let $F_W(\bar{a},\bar{Y})(i,j) \leftrightarrow \phi_F(j,i,\bar{a},\bar{Y})$ and $F_U(\bar{a},\bar{Y})(i,j) \leftrightarrow \phi_G(j,i,\bar{a},\bar{Y})$. Then by k-ary comprehension over $\Sigma_0^B(\Phi)$ there exist characteristic strings W of F_W, and U of F_U, given bounds on i and j. Consider the second statement with $W^{[i]}$ replaced by $F_W^{[i]}(\bar{a},\bar{Y})$. and $U^{[i]}$ replaced by $F_U^{[i]}(\bar{a},\bar{Y})$. By construction, for every i,j $F(i,\bar{a},\bar{Y})(j) \leftrightarrow F_W(\bar{a},\bar{Y}(i,j)$ and $G(i,\bar{a},\bar{Y})(j) \leftrightarrow F_U(\bar{a},\bar{Y}(i,j)$

Therefore, the second statement holds when W,U are replaced by F_W, F_U. That implies that this statement also holds for W and U that are characteristic strings of the respective functions.

We appeal to lemma 3.3.11 to conclude that the statement of lemma 3.3.12 is already provable in V-Φ. □

Now we can state the general definability theorem that can be used to characterize the power of closed systems based on constructive descriptive logics.

Theorem 3.3.13 (Definability theorem). *Suppose that Φ is restricted Σ_1^B or Σ_0^B, constructive, and represents a complexity class C. Then all functions from FC are Σ_1^B-definable in V-Φ and all Σ_1^B-definable functions of V-Φ are in $\mathtt{AC}^0(FC)$.*

Suppose, additionally, that Φ is strongly closed. Then the class of Σ_1^B-definable functions of V-Φ coincides with FC provably in V-Φ.

We will refer to the first statement as "weak definability" and the second statement as "strong definability".

First we prove the easy direction of this theorem.

Lemma 3.3.14. *Suppose that Φ is constructive. Then every function from FC is Σ_1^B-definable in V-Φ.*

Proof. First consider the number functions. Since V-Φ extends V^0, the least number principle is a theorem of V-Φ. Therefore, for any $\phi(z,\bar{a},\bar{Y}) \in \Phi$ there exists a unique minimal element z satisfying ϕ, or by definition of min, $z = t$.

The case of string functions is somewhat different. Every string function from FC is defined by a formula from Φ, for which there is comprehension axiom; that guarantees the existence of a string Z which is a characteristic string of the bitgraph of a function. Let

$$F(\bar{a},\bar{Y})(i) \leftrightarrow i < t(\bar{a},\bar{Y}) \wedge \phi(i,\bar{a},\bar{Y}),$$

where $\phi \in \Phi$. By assumption, there exists a Σ_1^B formula $\tilde{\phi}$ such that $V\text{-}\Phi \vdash \phi \leftrightarrow \neg\tilde{\phi}$. Take a formula

$$\phi^*(Z,\bar{a},\bar{Y}) \leftrightarrow \forall i < |Z|(Z(i) \wedge \phi(i,\bar{a},\bar{Y}) \vee \neg Z(i) \wedge \tilde{\phi}(i,\bar{a},\bar{Y})).$$

Placing quantifiers of ϕ and $\tilde{\phi}$ right after $\forall i < t$, we obtain a formula of the form used in lemma 3.3.12. By that lemma, this formula is Σ_1^B, and thus the defining axiom ϕ^* of F gives a Σ_1^B definition.

In $V\text{-}\Phi$ we can prove that $\phi^*(Z,\bar{a},\bar{Y})$ is true iff $\forall i < |Z|, Z(i) \leftrightarrow \phi(i,\bar{a},\bar{Y})$. Therefore, the existence of Z satisfying $\phi^*(Z,\bar{a},\bar{Y})$ follows by comprehension over the formula $\phi(i,\bar{a},\bar{Y})$. Its uniqueness follows from the fact that it is a characteristic string of ϕ. $\square$

The other direction of Theorem 3.3.13 requires using a powerful technique called *witnessing*, which is the subject of the rest of this chapter.

3.4 Witnessing

The origins of the witnessing method lie in the Skolem functions. Skolem was not concerned with the complexity of witnessing functions, he just needed a method to replace existential quantifiers by function symbols. Later, it was shown that the complexity of witnessing functions for Σ_1 formulae, that is, a version of Σ_1^B formulae without bounds on quantifiers, is primitive recursive [Par68, Min73].

3.4.1 Buss's witnessing theorem

The major work establishing the relation between complexity theory and bounded arithmetic was the 1986 PhD thesis of Sam Buss [Bus86], where a number of first-order and second-order theories that characterize the polytime hierarchy (starting with the first level), `PSPACE` and `EXPTIME` were developed. These systems are based on first-order framework in which numbers can be "large" or "small" (logarithmic); large numbers correspond to strings in our notation. Buss has multiplication for both kinds of numbers, so his systems cannot capture $\mathtt{AC}^0$. However, his goal was to capture the polynomial-time hierarchy and above, for which his systems were appropriate.

A major feature of these systems is that the language includes a # function: $x\#y = 2^{|x|\cdot|y|}$. This allows for coding of sequences and referencing bits of numbers without making the codes exponentially large. All systems consist of a set BASIC of axioms (Q, augmented with defining axioms for #) and various induction schemes. In Buss's systems there are two types of bounded variables: polynomially bounded (treated as "strings"), and logarithmically ("sharply") bounded. When determining levels of hierarchy (Σ_i^b) only alternations of polynomially bounded quantifiers are counted. The induction axiom schemes are

$$\text{IND: } \phi(0) \wedge \forall x(\phi(x) \to \phi(x+1)) \to \forall x \phi(x)$$
$$\text{PIND: } \phi(0) \wedge \forall x(\phi(\lfloor x/2 \rfloor) \to \phi(x)) \to \forall x \phi(x).$$

The four main theories (or, more precisely, hierarchies of theories) in this work are two first-order and two second-order systems based on IND and $PIND$. The hierarchies T_2^i and V_2^i have IND, restricted to Σ_i^b and its second-order version, respectively (where in the second-order version of Σ_i^b we count the number of alterations of second-order quantifiers, omitting first-order ones). Similarly, S_2^i and U_2^i are first- and second-order systems based on $PIND$. The subscript "2" in these theories denotes usage of # function. It can be shown that $T_2^0 = S_2^0$, and for $i \geq 1$ $S_2^i \subseteq T_2^i \subseteq S_2^{i+1}$.

The main connection with complexity theory is proved about the first-order theory S_2^1, consisting of a set of 32 axioms and a Σ_1^B-PIND induction scheme (length induction on $\mathtt{NP}$ predicates). Similar witnessing theorems show that U_2^1 captures $\mathtt{PSPACE}$ and V_2^1 captures $\mathtt{EXPTIME}$.

Theorem 3.4.1 (Buss's witnessing theorem). *Let $i \geq 1$ and let A be a Σ_i^b formula. s.t. $S_2^i \vdash \forall \bar{x} \exists y A(\bar{x}, y)$. Then there is a term t, a Σ_i^b formula B and a function $f \in P^{\Sigma_{i-1}^p}$ such that*

1. $S_2^i \vdash (\forall \bar{x})(\exists y \leq t) B(\bar{x}, y)$
2. $S_2^i \vdash (\forall \bar{x})(\forall y)(B(\bar{x}, y) \to A(\bar{x}, y))$
3. $S_2^i \vdash (\forall \bar{x})(\forall y)(\forall z)(B(\bar{x}, y) \wedge B(\bar{x}, z) \to y = z)$
4. *For all $\bar{n}$, $\mathbb{N} \models B(\bar{n}, f(\bar{n}))$*

For $i = 1$ this implies that predicates are polynomial iff they are Δ_1^B-definable in S_2^1. Conditions 1,3 and 4 state that f is Σ_i^b-definable in S_2^i by the formula B.

In general, this theorem gives a correspondence between polynomial hierarchy and hierarchy of S_2^i. An important open problem is whether the union for all i of S_2^i, called S_2, is finitely axiomatizable. A positive answer to this question would imply collapse of the polynomial hierarchy. Buss shows that S_2^1 is Σ_1^b conservative over PV; however, general conservativity of S_2^1 over QPV would imply collapse of polynomial hierarchy to $\Sigma_2^p = \Pi_2^p$ [Bus95, KPT91, Zam96]. More generally, if $T_2^i = S_2^{i+1}$, then PH collapses to $\Sigma_{i+2}^p = \Pi_{i+2}^p$, and provably collapses to $\Sigma_{i+3}^p = \Pi_{i+3}^p$. In [KPT91] Krajiček, Pudlák and Takeuti use an interesting technique that became known as "KPT witnessing theorem", witnessing $\exists\forall\exists\Pi_i^p$ formulae with a list of functions from $P^{\Sigma_{i+1}^p}$.

3.4.2 Witnessing for V-Φ

Buss was interested in more powerful theories than ones considered in this work. In his theories, the classes of predicates in induction and comprehension axioms come from complexity classes not known to be closed under $\mathtt{AC}^0$ reductions. That is, his classes of formulae do not satisfy property 1 unless NP=co-NP. Therefore, the complexity of witnessing functions there is strictly smaller than what can be defined by the formulae in comprehension scheme. In particular, in his theory S_2^1 which has the same power as V^1, the comprehension (induction) is over NP predicates. However, the class of Δ_1^B-definable predicates of S_2^1 is P.

The classes we are dealing with are much nicer, so the definability theorem for them is simpler. The main part of the proof of the definability theorem 3.3.13 is the following statement.

Theorem 3.4.2 (Generalized witnessing theorem). *Let Φ be a class of restricted Σ_1^B formulae representing C. Suppose that Φ is constructive. Then Σ_1^B-theorems of V-Φ can be witnessed by functions from $\mathtt{AC}^0(FC)$ provably in V-Φ. That is, if V-$\Phi \vdash \exists Z\phi(\bar{x}, \bar{Y}, Z)$, where $\phi \in \Sigma_1^B$, then there is a string function $F(\bar{x}, \bar{Y})$ in $\mathtt{AC}^0(FC)$ such that*

$$V\text{-}\Phi, AX(F) \vdash \phi(\bar{x}, \bar{Y}, F(\bar{x}, \bar{Y})),$$

where $AX(F)$ is a $\Sigma_0^B(\Phi)$-bit-defining axiom for F. If Φ is strongly closed and constructive, then V-Φ proves that the defining axiom for F is equivalent to a formula from Φ.

The witnessing theorem is stronger than the constructiveness property. Constructiveness is concerned with witnessing an existential quantifier in a $\phi \in \Phi$ (or finding a counterexample to ϕ). On the other hand, the witnessing theorem describes the power of a system in terms of the strength of Σ_1^B-theorems that the system in question can prove. In all our systems the witnessing theorem describes functions witnessing Σ_1^B-theorems, however in weaker systems the class of witnessing functions, and thus the number of Σ_1^B-theorems, is smaller.

The theorem 3.4.2 is a generalization of the witnessing theorem for V^0 as presented in [Coo02] (hence the name "Generalized witnessing"). The three main examples for which theorem 3.4.2 applies are V^0 itself, V_1-Horn and V-Krom. The simplest is V^0, because for it the properties are trivial; we will use it as a running example for this section. Proving the properties for V_1-Horn and V-Krom are the content of chapters 4 and 5. We also apply this theorem to a system for $\mathtt{SL}$ based on symmetric Krom formulae (see chapter 6); in that case, we use the weaker statement.

Example 3.4.1. [Coo02, Zam96] Functions bit-definable by Σ_0^B formulae in V_1^0 are $\mathtt{AC}^0$ functions, and Σ_0^B formulae correspond to the first-order logic which captures $\mathtt{AC}^0$ in the descriptive sense (see theorem 2.2.1). There is a witnessing theorem stating that the class of witnessing functions for Σ_1^B theorems in V_1^0 is $\mathtt{AC}^0$ as well.

Example 3.4.2. [CK01] The class of Σ_1^B-Horn formulae comes from $SO\exists$-Horn formulae capturing $\mathtt{P}$ in descriptive setting. V_1-Horn defines polynomial-time functions by Σ_1^B-Horn formulae, and is equivalent in power to Zambella's $\mathtt{P}$-def. In this case, the Theorem 3.4.2 holds with $\Phi = \Sigma_1^B$-Horn and $FC = FP$. So by the definability theorem Σ_1^B-definable functions of V_1-Horn are precisely polynomial-time functions. The bulk of work is proving closure of Σ_1^B-Horn formulae under complementation; the techniques required for that give constructiveness.

Note that if the conditions do not hold, then the class of witnessing functions can be smaller than definable by Σ_1^B formulae.

Example 3.4.3. By Fagin's theorem, Σ_1^B formulae capture $\mathtt{NP}$, which is not believed to be closed under complementation. Consider a system V^1, in which $\Phi = \Sigma_1^B$. We could try to define"$\mathtt{NP}$ functions" by setting their bitgraphs to be Σ_1^B formulae, as in the definition 3.3.5. Therefore, such functions are not (known to be) Σ_1^B-definable. Let the function $f_{3col}(n, E)$ return 1 if subgraph of E on the first n vertices is 3-colourable, and 0 if it is not 3-colourable. There is a Σ_1^B formula which is true on 3-colourable graphs (see example

2.0.4), but we do not know any Σ_1^B formula which would be true on exactly those graphs that are not 3-colourable. If $\mathtt{NP} = co\mathtt{NP}$, then for 3-colouring we will have Σ_1^B formulae $\phi = \exists Z\psi(Z, n, E)$ and $\tilde{\phi} = \exists\tilde{Z}\tilde{\psi}(\tilde{Z}, n, E)$, where ϕ is true on 3-colourable graphs and $\tilde{\phi}$ true on not 3-colourable graphs. Then we can give the defining axiom for f_{3col} as follows:

$$f_{3col}(n, E) = y \leftrightarrow \exists Z(y = 1 \wedge \psi(Z, n, E) \vee y = 0 \wedge \tilde{\psi}(Z, n, E)) \qquad \text{(Ax3COL)}$$

Note that the formula in the defining axiom is Σ_1^B. If, in addition, $V^1 \vdash \phi(n, E) \leftrightarrow \neg\tilde{\phi}(n, E)$ (that is, if V^1 "proves" that NP=coNP), then $V^1 \vdash \exists Z(\psi(Z, n, E) \vee \tilde{\psi}(Z, n, E))$. Then, by Buss's witnessing theorem, there exists a polynomial-time function $f(n, E)$ witnessing Z. By testing which of the two formulae is satisfied by the witnessing function on a given input (n, E), we can decide in polynomial time if the input is 3-colourable or not.

Remark 3.4.3. Note that if $\mathtt{NP} = coNP$, then the formula $\exists Z(\psi(Z, n, E) \vee \tilde{\psi}(Z, n, E))$ is a true Σ_1^B formula. However, it may not be theorem of V^1, in which case functions defined by $\Phi = \Sigma_1^B$ formulae are not Σ_1^B-definable in V^1. This is the case when Lemma 3.3.14 does not apply. If it were definable, then as we just saw, we would get $\mathtt{P} = \mathtt{NP}$.

The two major approaches to proving witnessing theorems are model-theoretic and proof-theoretic. The first works especially well in case when the system of arithmetic is a universal theory: then a simple Herbrand-style proof suffices. An example of a witnessing theorem of that kind is witnessing for Cook's equational theory PV ([Coo75]; see [CU93] for a more extensive treatment of PV-like theories). In his thesis, Buss used the second, proof theoretic approach, which we follow as well; however Zambella's proof of Buss's witnessing theorem in the second-order setting used model-theoretic argument [Zam96].

The approach that we describe here is a slight modification of the proof of witnessing theorem for V^0 from [Coo02]. All that was done was to notice that the properties above are sufficient for the proof to work, and prove that several major complexity classes can be characterized by classes of formulae satisfying these properties.

3.4.3 Quantified Gentzen proof system LK^2

The proof system used in [Coo02] for the proof of the V^0 witnessing theorem is a version of Gentzen's propositional calculus PK, extended with quantifier introduction rules for both first-order and second-order quantifiers. Here we follow [Coo02] in description of LK^2.

A line of a proof in LK^2 is a sequent of the form

$$A_1, \ldots, A_k \longrightarrow B_1, \ldots, B_l,$$

where $A_1, \ldots A_k$ and $B_1 \ldots B_l$ are sequences of formulae. The sequence to the left of the $\longrightarrow$ symbol is called "antecedent"; the sequence to the right is called the succedent. The sequent can be semantically interpreted as

$$(A_1 \wedge \cdots \wedge A_k) \rightarrow (B_1 \vee \cdots \vee B_l),$$

or, equivalently,

$$\neg A_1 \vee \cdots \vee \neg A_k \vee B_1 \vee \cdots \vee B_l.$$

A truth assignment satisfies the sequent iff it satisfies the corresponding formula.

Definition 3.4.4. For a theory V-Φ, we define a proof system LK^2-V-Φ, which consists of a set of rules of inference given in the Table 3.4.3, together with logical axioms

$$A \longrightarrow A, \qquad \longrightarrow \top \qquad \bot \longrightarrow$$

and nonlogical axioms; in our case, equality axioms and axioms of V-Φ.

A proof in LK^2-V-Φ can be represented as a directed acyclic graph in which the nodes are sequents, every edge corresponds to a rule of inference, and leaves are axioms. A proof is called *anchored* if the only formulae A allowed to be cut out by the cut rule are instances of nonlogical axioms.

Please see, for example, [Coo02] or Buss's chapters in the Handbook of Proof Theory for the proof that LK^2 is sound and complete (including the anchored version).

3.4.4 Σ_1^B-axiomatizable version of V-Φ

We would like to analyze an anchored LK^2 proof of a Σ_1^B sequent in V-Φ. It would be very useful to be able to say that no formula in the proof is more complex than the endsequent. In a cut-free proof, it follows from the subformula property: every formula appearing in the proof eventually becomes a part of the endsequent. However, when cuts are allowed this is no longer true: a complex formula can be cut out by the cut rule. Since the proofs we are considering are anchored, only nonlogical axioms can be cut by the cut rule. So we want to make sure that nonlogical axioms are simple; in particular, that they are Σ_1^B formulae.

Rule name	**Left**	**Right**
Weakening	$\dfrac{\Gamma \longrightarrow \Delta}{A, \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A}$
Exchange	$\dfrac{\Gamma_1, A, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, B, A, \Gamma_2 \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta_1, A, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, B, A, \Delta_2}$
Contraction	$\dfrac{\Gamma, A, A \longrightarrow \Delta}{\Gamma, A \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A, A}{\Gamma \longrightarrow \Delta, A}$
$\neg$ introduction	$\dfrac{\Gamma \longrightarrow \Delta, A}{\neg A, \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, \neg A}{A, \Gamma \longrightarrow \Delta}$
$\wedge$ introduction	$\dfrac{A, B, \Gamma \longrightarrow \Delta}{(A \wedge B), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A \quad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, (A \wedge B)}$
$\vee$ introduction	$\dfrac{A, \Gamma \longrightarrow \Delta \quad B, \Gamma \longrightarrow \Delta}{(A \vee B), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A, B}{\Gamma \longrightarrow \Delta, (A \vee B)}$
Cut	$\dfrac{\Gamma \longrightarrow \Delta, A \quad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$	
$\forall$ introduction	$\dfrac{A(t), \Gamma \longrightarrow \Delta}{\forall x A(x), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A(b)}{\Gamma \longrightarrow \Delta, \forall x A(x)}$
$\exists$ introduction	$\dfrac{A(b), \Gamma \longrightarrow \Delta}{\exists x A(x), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A(t)}{\Gamma \longrightarrow \Delta, \exists x A(x)}$
String $\forall$ introduction	$\dfrac{A(\alpha), \Gamma \longrightarrow \Delta}{\forall X A(X), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A(\beta)}{\Gamma \longrightarrow \Delta, \forall X A(X)}$
String $\exists$ introduction	$\dfrac{A(\beta), \Gamma \longrightarrow \Delta}{\exists X A(X), \Gamma \longrightarrow \Delta}$	$\dfrac{\Gamma \longrightarrow \Delta, A(\alpha)}{\Gamma \longrightarrow \Delta, \exists X A(X)}$

Restriction: The free variables b (β) must not occur in the conclusion of (string) $\forall$-**right** and $\exists$-**left**.

Table 3.2: LK^2 rules of inference

In the case of V^0, this statement holds. However, if the formulae in the comprehension rule have string quantifiers, the comprehension axiom becomes Σ_2^B. In order to avoid this, we will define a new equivalent theory $\tilde{V}$-Φ in which all axioms are Σ_1^B formulae, and work with $\tilde{V}$-Φ instead of V-Φ.

Let V-Φ satisfy the constructiveness property. Then for every formula $\phi \in \Phi$ there is a formula $\tilde{\phi} \in \Sigma_1^B$ such that $V\text{-}\Phi \vdash \phi \leftrightarrow \neg\tilde{\phi}$. For strongly closed classes, $\tilde{\phi}$ is in Φ and it is obtained by formalizing the construction in the proof of closure of the corresponding complexity class under complementation. We will refer to $\tilde{\phi}$ as a *tilde-counterpart* of ϕ.

Definition 3.4.5. Suppose that Φ is constructive. Let $\phi \in \Phi$, let $\tilde{\phi}$ be the Σ_1^B formula equivalent, provably in V-Φ, to $\neg\phi$, and let t be any term. A theory $\tilde{V}$-Φ corresponding to V-Φ is axiomatized by the axioms 2-BASIC together with rules $\longrightarrow \phi, \tilde{\phi}$ and $\phi, \tilde{\phi} \longrightarrow$ for every $\phi \in \Phi$, Σ_0^B comprehension axiom, and a conditional Φ-comprehension axiom of the form

$$\exists Z \leq t \forall i < t(\phi(i) \wedge Z(i)) \vee (\tilde{\phi}(i) \wedge \neg Z(i)) \qquad (\Phi\text{-}\widetilde{comp})$$

Lemma 3.4.6. *If $V\text{-}\Phi \vdash \Sigma_0^B(\Phi) \equiv \Phi$, and $V^0 \subseteq V\text{-}\Phi$, then V-Φ and $\tilde{V}$-Φ have the same Σ_1^B theorems.*

Proof. First we show that $V\text{-}\Phi \subseteq \tilde{V}\text{-}\Phi$. For that, we only need to show that $\tilde{V}$-Φ proves the comprehension axiom of V-Φ for every ϕ.

Let $\tilde{\phi}$ be the tilde-counterpart of ϕ. Then $\tilde{V}$-Φ proves the Φ-comp for ϕ and $\tilde{\phi}$. But since $\phi \leftrightarrow \neg\tilde{\phi}$ by the rules, $\tilde{\phi}$ can be replaced by $\neg\phi$ in Φ-$\widetilde{comp}$, giving the bottom sequent of the comprehension rule of V-Φ.

The proof that $\tilde{V}\text{-}\Phi \subseteq V\text{-}\Phi$ relies on constructiveness. It guarantees, for every $\phi \in \Phi$, the existence of $\tilde{\phi} \in \Sigma_1^B$ equivalent to the negation of ϕ. That is, V-Φ proves $\phi \leftrightarrow \neg\tilde{\phi}$, which gives $\longrightarrow \phi, \tilde{\phi}$ and $\phi, \tilde{\phi} \longrightarrow$. From there, comprehension on ϕ existence of its characteristic string Z. Since $\phi \leftrightarrow \neg\tilde{\phi}$, $V\text{-}\Phi \vdash \forall i < tZ(i) \leftrightarrow \neg\tilde{Z}(i)$. As a consequence, V-Φ proves that Z satisfies the comprehension axiom of $\tilde{V}$-Φ for ϕ. □

The comprehension axiom of $\tilde{V}$-Φ is not a strict Σ_1^B formula. However, it is provably equivalent to a Σ_1^B formula, so for every ϕ the comprehension scheme can be thought of as strict Σ_1^B.

Lemma 3.4.7. *The conditional comprehension axiom Φ-$\widetilde{comp}$ of $\tilde{V}$-Φ is equivalent in $\tilde{V}$-Φ to a Σ_1^B formula.*

Proof. Consider the subformula of Φ-$\widetilde{comp}$ with Z as a free variable. It consists of a formula from Φ preceded by a universal first-order quantifier. Let $\phi \equiv \exists \bar{P}' \forall \bar{x} \leq b(\bar{a}, \bar{X}) \psi(i, \bar{x}, \bar{P}', \bar{a}, \bar{X})$ and $\tilde{\phi} \equiv \exists \bar{Q}' \forall \bar{x}' \leq b'(\bar{a}, \bar{X}) \tilde{\psi}(i, \bar{x}', \bar{Q}', \bar{a}, \bar{X})$; assume without loss of generality that $b = b'$. Putting the subformula under $\exists Z \forall i < t$ in prenex form, and encoding, using pairing function, vectors of second-order variables as single variables, get

$$\forall i < t[\exists P' \exists Q' \forall \bar{x}, \bar{x}' \leq b(\bar{a}, \bar{X})(Z(i) \to \psi(i, \bar{x}, P')) \\ \wedge(\neg Z(i) \to \tilde{\psi}(i, \bar{x}', Q'))].$$

Applying lemma 3.3.12, obtain

$$\exists P \exists Q \forall i < t \forall \bar{x}, \bar{x}' \leq b(\bar{a}, \bar{X})(Z(i) \to \psi(i, \bar{x}, P^{[i]})) \\ \wedge(\neg Z(i) \to \tilde{\psi}(i, \bar{x}', Q^{[i]})).$$

Since all free variables, in particular Z, are implicitly universally quantified in this formula, existence of Z satisfying the first formula implies existence of Z satisfying the second (and, in fact, Z can be the same). $\square$

3.4.5 Proof of the generalized witnessing theorem

Take a Σ_1^B theorem of $\tilde{V}$-Φ. Since every axiom of $\tilde{V}$-Φ is (equivalent to) a Σ_1^B formula, every sequent in the proof contains only Σ_1^B formulae: this follows from the subformula property and the fact that the only cuts are on the axioms of $\tilde{V}$-Φ, and the comprehension axiom is equivalent to a Σ_1^B formula by the replacement lemma (lemma 3.2.12). We will combine the comprehension rule and the replacement and always treat the comprehension axiom as a Σ_1^B formula.

The idea of this proof came from the proof of witnessing theorem for V^0 in [Coo02], and it follows the same structure. The proof proceeds by cases corresponding to the rules of inference of LK^2-$\tilde{V}$-Φ. That is, for every rule in the table 3.4.3, and for the rules of $\tilde{V}$-Φ as stated in definition 3.4.5, we will show how to construct the witnesses for the bottom sequent from witnesses of the top sequent(s). We omit the proof that (anchored) LK^2 is sound and complete; see [Coo02] for that.

Recall that we are trying to show that if $\tilde{V}$-$\Phi \vdash \exists Z B(\bar{x}, \bar{Y}, Z)$, then there is a function F with Φ-bit-defining axiom $AX(F) = \psi_F(\bar{x}, \bar{Y}, i)$. There is a term t such that $\tilde{V}$-$\Phi \vdash \exists Z\ |Z| < t \wedge B(\bar{x}, \bar{Y}, Z)$, Let π be an anchored LK^2-$\tilde{V}$-Φ proof of

$$\longrightarrow \exists Z \le t\ B(\bar{a}, \bar{\alpha}, Z)$$

Then cut formulae in π are restricted to formulae in the axioms of $\tilde{V}$-Φ. Since the endsequent of ϕ is Σ_1^B, and all axioms are Σ_1^B, every formula occurring in the proof is of the form

$$\exists X \le r\ \phi(X), \qquad \phi \in \Sigma_0^B. \tag{3.8}$$

Thus every sequent S in π has the form

$$\exists X_1 \le t_1 \phi_1(X_1), ..., \exists X_m \le t_m \phi_m(X_m), \Gamma \longrightarrow \Delta, \exists Y_1 \le s_1 \psi_1(Y_1), ..., \exists Y_n \le s_n \psi_n(Y_n) \tag{3.9}$$

for $m, n \ge 0$, where all formulae in Γ and Δ are Σ_0^B. We will prove by induction on the depth of S in π that there are functions $F_1, ..., F_n \in FC$, that are bit-definable by formulae from Φ, and a proof in LK^2-$\tilde{V}$-Φ of the following S':

$$|\beta_1| \le t_1 \wedge \phi_1(\beta_1), ..., |\beta_m| \le t_m \wedge \phi_m(\beta_m), \Gamma \longrightarrow \Delta, |F_1| \le s_1 \wedge \psi_1(F_1), ..., |F_n| \le s_n \wedge \psi_n(F_n) \tag{3.10}$$

where F_i stands for $F_i(\bar{a}, \bar{\alpha}, \bar{\beta})$, and $\bar{a}, \bar{\alpha}$ is a list of exactly those variables with free occurrences in S. (This list may be different for different sequents.) Here $\beta_1, ..., \beta_m$ are distinct new free variables corresponding to the bound variables $X_1, ..., X_m$, although the latter variables may not be distinct.

Specifically, if we write (3.10) in the form

$$S' = \quad A_1, ..., A_k \longrightarrow B_1, ..., B_\ell$$

then we assert

$$\tilde{V}\text{-}\Phi + AX(FC) \vdash\ \forall \vec{a} \forall \bar{\alpha} \forall \bar{\beta} [(A_1 \wedge ... \wedge A_k) \supset (B_1 \vee ... \vee B_\ell)] \tag{3.11}$$

In general it is useful to introduce the constant string function λ whose intended interpretation is the empty string. Thus λ has the bit-defining axiom

$$\lambda(i) \leftrightarrow i < 0 \tag{3.12}$$

Our inductive proof has several cases, depending on whether S is a $\tilde{V}$-Φ axiom, or which rule is used to generate S.

1. **Axioms of V^0:** S is an axiom of V^0.
   If the axiom only involves Σ_0^B formulae, then no witnessing functions are needed. Otherwise S comes from a Σ_0^B-comp axiom:
   $$S = \longrightarrow \exists X \leq b \forall z < b(X(z) \leftrightarrow \psi_F(z, b, \bar{a}, \bar{\alpha}))$$
   Then a function witnessing X has bit-defining axiom
   $$F(b, \bar{a}, \bar{\alpha})(i) \leftrightarrow i < b \wedge \psi_F(i, b, \bar{a}, \bar{\alpha})$$
   Since $\tilde{V}$-$\Phi \vdash \Sigma_0^B(\Phi) \equiv \Phi$, even if Σ_0^B is not syntactically a subset of Φ for every Σ_0^B formula ψ_F there is a provably equivalent formula $\psi'_F \in \Phi$ which can be used as a witness instead of ψ_F. Thus, the defining axiom for F is (equivalent to) a formula from Φ.

2. **Complementary pairs:** S is of the form $i < t, \phi, \tilde{\phi} \longrightarrow$ or $i < t \longrightarrow \phi, \tilde{\phi}$.
   In the first rule, there is nothing on the right side, and so no witnessing is required. The second rule is harder. If $\Phi = \Sigma_0^B$, then this case is not a problem, since there are no quantifiers to witness. However, if ϕ contains second-order quantifiers, as in the case of Φ being restricted Σ_1^B, then these quantifiers need to be witnessed. Here is the place where we need the strong constructiveness property (property 2.) This part is the hardest to prove for specific systems, and requires formalizing the respective satisfiability algorithms.

   Let ϕ and $\tilde{\phi}$ be restricted Σ_1^B. Then by existence of a pairing function we can assume that each of them has only one second-order quantifier. That is, ϕ and $\tilde{\phi}$ are of the form
   $$\phi(\bar{a}, \bar{\alpha}, i) \equiv \exists P \forall \bar{x} \leq t'(\bar{a}, \bar{\alpha}, i)\psi(i, \bar{x}, P, \bar{a}, \bar{\alpha})$$
   $$\tilde{\phi}(\bar{a}, \bar{\alpha}, i) \equiv \exists Q \forall \bar{x}' \leq t''(\bar{a}, \bar{\alpha}, i)\tilde{\psi}(i, \bar{x}', Q, \bar{a}, \bar{\alpha})$$
   By constructiveness property, there are terms t_P and t_Q and formulae $\phi_P(\bar{a}, \bar{\alpha}, i, j)$ and $\phi_Q(\bar{a}, \bar{\alpha}, i, j)$ such that if ϕ holds on its variables, then P is defined by ϕ_P, and if $\tilde{\phi}$ is holds, then ϕ_Q defines Q. Take F_P and F_Q defined by
   $$F_P(i, \bar{a}, \bar{\alpha})(j) \equiv j < t_P \wedge \phi(\bar{a}, \bar{\alpha}, i) \wedge \phi_P(j, i, \bar{a}, \bar{\alpha})$$
   and
   $$F_Q(i, \bar{a}, \bar{\alpha})(j) \equiv j < t_Q \wedge \tilde{\phi}(\bar{a}, \bar{\alpha}, i) \wedge \phi_Q(j, i, \bar{a}, \bar{\alpha})$$
   as witnessing functions.

3. **Conditional comprehension axiom:** Then S is of the form

$$\longrightarrow \exists Z \forall i \leq t(\phi(i) \wedge Z(i) \vee \tilde{\phi}(i) \wedge \neg Z(i))$$

The quantified second-order variables in this axiom come from two sources: the variables of ϕ and $\tilde{\phi}$ and the new variable Z. The witnessing functions for the variables of ϕ and $\tilde{\phi}$ are constructed just as in the case of complementary pairs rule. The witnessing function for Z becomes $F_Z(\bar{a}, \bar{\alpha})(i) \equiv \phi(i, \bar{a}, \bar{\alpha})$.

4. **String $\exists$-right:** Then S is the bottom of the inference

$$\frac{\Lambda \longrightarrow \Pi, |\gamma| \leq t \wedge A(\gamma)}{\Lambda \longrightarrow \Pi, \exists X \leq t A(X)}$$

Here, Λ and Π are the rest of the formulae in the sequents, Σ_0^B with the appropriate witnessing functions. If the variable γ occurs free in S, then we may witness the new quantifier $\exists X$ by the function F with bit-defining axiom

$$F(\bar{a}, \gamma, \bar{\alpha}, \bar{\beta})(i) \leftrightarrow i < t \wedge \gamma(i)$$

If γ has no occurrence in S, then we may take $\gamma = \lambda$ and define

$$F(\bar{a}, \bar{\alpha}, \bar{\beta})(i) \leftrightarrow i < 0$$

Also in this case we modify each witnessing function F_i for formulae in Π by substituting λ for the argument γ, where λ is the constant string function defined in (3.12).

5. **String $\exists$- left:** Then S is the bottom of the inference

$$\frac{\gamma \leq t \wedge A(\gamma), \Lambda \longrightarrow \Pi}{\exists X \leq t A(X), \Lambda \longrightarrow \Pi}$$

Note that γ cannot occur in S, by the restriction for this rule, but S' has a new variable β' available corresponding to $\exists X$ (see (3.10)). No new witnessing function is required. Each witnessing function $F_j(\bar{a}, \gamma, \bar{\alpha}, \bar{\beta})$ for the top sequent is replaced by the witnessing function

$$F'_j(\bar{a}, \bar{\alpha}, \beta', \bar{\beta}) = F_j(\bar{a}, \beta', \bar{\alpha}, \bar{\beta})$$

for S'.

6. **Number $\exists$-right and number $\forall$- left:**
   No new witnessing functions are required, but these rules may eliminate some free variables, and these variables will not be available in S' as arguments of the witnessing functions. Simply replace each number variable eliminated by 0 and each string variable eliminated by λ.

7. **Number $\exists$- left and number $\forall$- right:** We consider $\exists$-**left**, since $\forall$-**right** is similar. Then S is the bottom sequent in the inference

$$\frac{b \leq t \wedge B(b), \Lambda \longrightarrow \Pi}{\exists x \leq tB(x), \Lambda \longrightarrow \Pi}$$

   No new witnessing function is needed, but the free variable b is eliminated as an argument to the existing witnessing functions, and it must be given a value. We give it a value which satisfies the new existential quantifier, if one exists. Thus define the $\mathtt{FAC}^0$ number function

$$g(\bar{a}, \bar{\alpha}) = \min b \leq t \ B(b)$$

   For each witnessing function $F_j(b, \bar{a}, \bar{\alpha}, \bar{\beta})$ for the top sequent define the corresponding witnessing function for the bottom sequent by

$$F'_j(\bar{a}, \bar{\alpha}, \bar{\beta}) = F_j(g(\bar{a}, \bar{\alpha}), \bar{a}, \bar{\alpha}, \bar{\beta})$$

8. **Cut:**
   Then S is the bottom of the inference

$$\frac{\Lambda \longrightarrow \Pi, A \quad A, \Lambda \longrightarrow \Pi}{\Lambda \longrightarrow \Pi}$$

   We use the notation

$$S_1 = \Lambda \longrightarrow \Pi, A \qquad\qquad S_2 = A, \Lambda \longrightarrow \Pi$$

   No new witnessing function is required. As in the case of string $\exists$ right, any free variables eliminated can be replaced by either 0 or λ.

   Let $F_1, ..., F_n$ be the witnessing functions for Π in S'_1 and let $F'_1, ..., F'_n$ be the witnessing functions for these same formulae in S'_2. Assume first that A is Σ_0^B.

Then we define witnessing functions $F_1'', ..., F_n''$ for these formulae in the conclusion S' by the defining axioms

$$F_j''(...)(i) \leftrightarrow (\neg A \wedge F_j(...)(i) \ \vee \ (A \wedge F_j'(...)(i)))$$

Now assume that A is not Σ_0^B, so A has the form

$$A \equiv \exists X \leq t B(X) \tag{3.13}$$

where $B(X)$ is Σ_0^B. Let G be the witnessing function for $\exists X$ in S_1' and let β be the variable in S_2' corresponding to X. Then the witnessing function $F_j'(\beta)$ for a formula in S_2' from Π will have an argument β missing from the corresponding witnessing function $F_j()$ in S_1'. The corresponding witnessing function F_j'' in S' has defining axiom

$$F_j''()(i) \leftrightarrow [\neg(|G()| \leq t \wedge B(G())) \wedge F_j()(i)] \vee [|G()| \leq t \wedge B(G()) \wedge F_j'(G())(i)]$$

Since Φ is closed, this is definable by a formula from Φ.

9. $\vee$- **left and** $\wedge$-**right**

These are both handled in the same manner. Consider $\wedge$-**right**.

$$\frac{\Lambda \longrightarrow \Pi, A \quad \Lambda \longrightarrow \Pi, B}{\Lambda \longrightarrow \Pi, (A \wedge B)}$$

Let S_1 and S_2 be the left and right sequents on top, and let S be the sequent on bottom. Suppose the j-th Σ_1^B-formula in Π is

$$\exists Y_j \leq t_j \psi_j(Y_j)$$

and suppose $F_j()$ and $F_j'()$ witness Y_j in S_1' and S_2', respectively. Then we define the witness $F_j''()$ for Y_j in S' to be $F_j()$ or $F_j'()$, depending on whether $F_j()$ works as a witness. That is,

$$F_j''()(i) \leftrightarrow [|F_j()| \leq t_j \wedge \psi_j(F_j()) \wedge F_j()(i)] \vee [\neg(|F_j()| \leq t_j \wedge \psi_j(F_j())) \wedge F_j'()(i)]$$

10. $\wedge$- **left and** $\vee$-**right:**

Nothing to do. Keep the same witnesses.

11. **All other rules:**

Weakening is easy. There is nothing to do for exchange and $\neg$ introduction. The contraction rules can be derived from cut and exchanges.

This completes the proof of theorem 3.4.2. □.

Corollary 3.4.8. *Existential quantifier in every Σ_1^B theorem of V-Φ can be witnessed by a function defined by a formula from Φ.*

Proof. This follows immediately from the equivalence of $\tilde{V}$-Φ and V-Φ (lemma 3.4.6) and the witnessing theorem for $\tilde{V}$-Φ. □

Since we have the pairing function, we could assume that there is always at most one quantified string variable. However, it is not necessary: if there are several quantified second-order variables, each of them can be witnessed by a function from FC.

Corollary 3.4.9. *If V-$\Phi \vdash \exists Z_1 ... \exists Z_k\ B(\bar{x}, \bar{Y}, \bar{Z})$, where B is a Σ_0^B formula, then there are string functions $F_1, ..., F_k \in FC$ such that*

$$V\text{-}\Phi, AX(F_1, ..., F_k) \vdash B(\bar{x}, \bar{Y}, \bar{F}(\bar{x}, \bar{Y}))$$

Proof. From the hypothesis and Lemma we conclude

$$V\text{-}\Phi \vdash \exists Z\ B(\bar{x}, \bar{Y}, Z^{[1]}, ..., Z^{[k]})$$

By the Witnessing Theorem for V-Φ we conclude there is a string function $F \in FC$ such that

$$V\text{-}\Phi, AX(F) \vdash B(\bar{x}, \bar{Y}, F^{[1]}(\bar{x}, \bar{Y}), ..., F^{[k]}(\bar{x}, \bar{Y}))$$

The Corollary follows by defining $F_i(\bar{x}, \bar{Y}) = F^{[i]}(\bar{x}, \bar{Y})$, for $i = 1, ..., k$. □

3.5 Example: witnessing for V^0

The following example illustrates the basic outline of the proof. First, we show the properties, then apply the definability theorem (theorem 3.3.13) to conclude that the class of Σ_1^B-definable functions of V^0 is DLOGTIME-uniform $\mathtt{AC}^0$.

3.5.1 Capturing $\mathtt{AC}^0$ descriptively

The first part is to show that the descriptive analog of Σ_0^B formulae captures $\mathtt{AC}^0$ in descriptive setting. Recall that translation from descriptive setting to bounded arithmetic turns uninterpreted variables of a vocabulary into free second-order variables. So the counterpart of Σ_0^B in the descriptive complexity setting is just a first-order logic. By the result of Immerman [Imm83, Imm87], later made uniform by Barrington, Immerman and Straubing [BIS90], first-order logic captures DLOGTIME-uniform $\mathtt{AC}^0$ over arithmetic structures. Therefore, Σ_0^B represents DLOGTIME-uniform $\mathtt{AC}^0$ in the standard model.

3.5.2 Strong closure and constructiveness

Since Σ_0^B formulae do not have string quantifiers, the constructiveness is trivial. The closure of Σ_0^B formulae under Σ_0^B operations is trivial as well. The only part that needs to be proven is the closure of $\mathtt{AC}^0$ functions under composition and substitution of a term a function for a variable.

Lemma 3.5.1. *The class of* $\mathtt{AC}^0$ *functions is closed under* $\mathtt{AC}^0$ *reductions.*

Proof. See the proof in [Coo04]. □

3.5.3 Applying the generalized witnessing theorem

Since $\tilde{V}$-Φ has a Σ_0^B-comp axiom scheme, when $\Phi = V^0$ the complementation pairs rules and thus the conditional comprehension axioms do not occur in any proof, making constructiveness property irrelevant: it is vacuously true. Since Σ_0^B satisfies the closure property, the theorem 3.3.13 applies. Therefore, the class of Σ_1^B-definable functions of V^0 is exactly $\mathtt{AC}^0$.

3.6 V-Φ is finitely axiomatizable

Since the validity of formulae from Φ is complete for C (by definition of descriptive capture), the comprehension scheme can be replaced by a finite number of axioms. The comprehension axioms consist of axioms needed to finitely axiomatize V^0, together with an axiom specific for Φ. In general, this axiom is comprehension over a formula $\phi \in \Phi$ which takes as one free variable an encoding of an arbitrary formula $\psi \in \Phi$, and its free variables as tuples. In the examples in the next chapters, the formulae in comprehension are encodings of the satisfiability algorithm for Φ.

The remaining part is to prove the finite axiomatizability of V^0.

Theorem 3.6.1. V^0 *is finitely axiomatizable.*

Proof. We must show that all Σ_0^B-COMP axioms follow from finitely many theorems of V^0.

Let $2-BASIC^+$ (or simply B^+) denote the $2-BASIC$ axioms along with finitely many theorems of V^0 asserting basic properties of $+$ and $\cdot$ such as commutativity, associativity, distributive laws, and cancellation laws involving $+$, $\cdot$, and $\leq$. These can be proved from the $2-BASIC$ axioms by induction on Σ_0^B formulae, as discussed in section 3.2.1.

It suffices to show that k-ary comprehension (3.5) for all Σ_0^B formulae follow from B^+ and finitely many such comprehension instances. We use the notation $\Phi[\bar{a},\bar{Q}](\bar{x})$ to indicate that the Σ_0^B formula Φ can contain the free variables $\bar{a},\bar{Q}$ in addition to $\bar{x}=x_1,...,x_k$. Then $COMP_\Phi(\bar{a},\bar{Q},\bar{b})$ denotes the comprehension formula

$$\exists Y \leq \langle b_1,...,b_k\rangle \forall x_1 < b_1 ... \forall x_k < b_k (Y(\bar{x}) \leftrightarrow \Phi(\bar{x})) \tag{3.14}$$

We will show that $COMP_\Phi$ for the following 12 formulae Φ will suffice.

$$\begin{array}{lll}
\Phi_1(x_1,x_2) & \equiv & \exists y \leq x_1(x_1 = \langle x_2, y\rangle) \\
\Phi_2(x_1,x_2) & \equiv & \exists z \leq x_1(x_1 = \langle z, x_2\rangle) \\
\Phi_3[Q_1,Q_2](x_1,x_2) & \equiv & \exists y \leq x_1(Q_1(x_1,y) \wedge Q_2(y,x_2)) \\
\Phi_4[a](x,y) & \equiv & y = a \\
\Phi_5[Q_1,Q_2](x,y) & \equiv & \exists z_1 \leq y \exists z_2 \leq y(Q_1(x,z_1) \wedge Q_2(x,z_2) \wedge y = z_1 + z_2) \\
\Phi_6[Q_1,Q_2](x,y) & \equiv & \exists z_1 \leq y \exists z_2 \leq y(Q_1(x,z_1) \wedge Q_2(x,z_2) \wedge y = z_1 \cdot z_2) \\
\Phi_7[Q_1,Q_2,c](x) & \equiv & \exists y \leq c(Q_1(x,y) \wedge Q_2(x,y)) \\
\Phi_8[Q_1,Q_2,c](x) & \equiv & \exists y_1 \leq c \exists y_2 \leq c(Q_1(x,y_1) \wedge Q_2(x,y_2) \wedge y_1 \leq y_2) \\
\Phi_9[X,Q,c](x) & \equiv & \exists y \leq c(Q(x,y) \wedge X(y)) \\
\Phi_{10}[Q](x) & \equiv & \neg Q(x) \\
\Phi_{11}[Q_1,Q_2](x) & \equiv & Q_1(x) \wedge Q_2(x) \\
\Phi_{12}[Q,c](x) & \equiv & \forall y \leq c Q(x,y)
\end{array}$$

In the following lemmas, we abbreviate $COMP_{\Phi_i}(...)$ by C_i.

Lemma 3.6.2. *For each $k \geq 2$ and $1 \leq i \leq k$ let*

$$\Psi_{ik}(y,z) \equiv$$
$$\exists x_1 \leq y ... \exists x_{i-1} \leq y \exists x_{i+1} \leq y ... \exists x_k \leq y (y = \langle x_1,...,x_{i-1},z,x_{i+1},...,x_k\rangle)$$

Then

$$B^+, C_1, C_2, C_3 \vdash COMP_{\Psi_{ik}}$$

Proof. We proceed by induction on k. For $k = 2$ we have $\Psi_{1,2} \equiv \Phi_1$ and $\Psi_{2,2} \equiv \Phi_2$. For $k > 2$, recall $\langle x_1, ..., x_k \rangle = \langle \langle x_1, ..., x_{k-1} \rangle, x_k \rangle$. Thus $\Psi_{kk} \equiv \Phi_2$. For $1 \leq i < k$ use $COMP_{\Phi_3}$ with Q_1 defined by $COMP_{\Phi_1}$ and Q_2 defined by $COMP_{\Psi_{i,k-1}}$. $\square$

Lemma 3.6.3. *Let $t(\bar{x})$ be a term which in addition to variables $\bar{x}$ may involve other variables $\bar{a}, \bar{Q}$. Let $\Psi_t[\bar{a}, \bar{Q}](\bar{x}, y) \equiv y = t(\bar{x})$. Then*

$$B^+, C_1, ..., C_6 \vdash COMP_{\Psi_t}(\bar{a}, \bar{Q}, \bar{b}, d)$$

Proof. By using algebraic theorems in B^+ we may suppose that $t(\bar{x})$ is a sum of monomials in $x_1, ..., x_k$, where the coefficients are terms involving $\bar{a}, \bar{Q}$. The case $t \equiv u$, where u does not involve any x_i is obtained from $COMP_{\Phi_4}$ with $a \leftarrow u$. The cases $t \equiv x_i$ are obtained from Lemma 3.6.2. We then build monomials using $COMP_{\Phi_6}$ repeatedly, and build the general case by repeated use of $COMP_{\Phi_5}$. $\square$

Lemma 3.6.4. *Let $t_1(\bar{x}), t_2(\bar{x})$ be terms with variables among $\bar{x}, \bar{a}, \bar{Q}$. Suppose*

$$\begin{aligned}
\Psi_1[\bar{a}, \bar{Q}](\bar{x}) &\equiv t_1(\bar{x}) = t_2(\bar{x}) \\
\Psi_2[\bar{a}, \bar{Q}](\bar{x}) &\equiv t_1(\bar{x}) \leq t_2(\bar{x}) \\
\Psi_3[\bar{a}, \bar{Q}, X](\bar{x}) &\equiv X(t_1(\bar{x}))
\end{aligned}$$

Then $B^+, C_1, ..., C_9 \vdash COMP_{\Psi_i}$, for $i = 1, 2, 3$.

Proof. $COMP_{\Psi_1}(\bar{a}, \bar{Q}, \bar{b})$ follows from $COMP_{\Phi_7}(P_1, P_2, c, b)$ with for $i = 1, 2$, P_i defined from $COMP_{\Psi_{t_i}}$ in Lemma 3.6.3 with $d \leftarrow t_1(\bar{b}) + t_2(\bar{b}) + 1$, so

$$\forall \bar{x} < \bar{b} \forall y < t_1(\bar{b}) + t_2(\bar{b}) + 1 (P_i(\bar{x}, y) \leftrightarrow y = t_i(\bar{x}))$$

In $COMP_{\Phi_7}$ we take $c \leftarrow t_1(\bar{b})$ and $b \leftarrow \langle b_1, ..., b_k \rangle$. We proceed similarly for $COMP_{\Psi_2}$, using $COMP_{\Phi_8}$.

For $COMP_{\Psi_3}(\bar{a}, \bar{Q}, X, \bar{b})$ we use $COMP_{\Phi_9}(X, P, c, b)$ with $c \leftarrow t_1(\bar{b})$ and $b \leftarrow \langle b_1, ..., b_k \rangle$ and P defined from Lemma 3.6.3 similarly to P_1 above. $\square$

Now we can complete the proof of the theorem. Lemma 3.6.4 takes care of the case when Φ is an atomic formula. Then by repeated applications of $COMP_{\Phi_{10}}$ and $COMP_{\Phi_{11}}$ we handle the case in which Φ is quantifier-free.

Now suppose $\Phi(\bar{x}) \equiv \forall y \leq t(\bar{x}) \phi(\bar{x}, y)$. We assume as an induction hypothesis that we can define Q satisfying

$$\forall \bar{x} < \bar{b} \forall y < t(\bar{b}) + 1 [Q(\bar{x}, y) \leftrightarrow (y \leq t(\bar{x}) \rightarrow \phi(\bar{x}, y)]$$

Then $COMP_{\Phi}(\bar{b})$ follows from $COMP_{\Phi_{12}}(Q, c, b)$ with $c \leftarrow t(\bar{b})$ and $b \leftarrow \langle b_1, ..., b_k \rangle$. $\square$

3.7 History

In 1971, Parikh [Par71] proposed the first system of bounded arithmetic, called $I\Delta_0$. There the basic axioms are similar to Robinson's Q, and the induction scheme is restricted to bounded (Δ_0) formulae. He showed that all functions that are Δ_0-definable in $I\Delta_0$ are polynomially bounded; i.e., if ϕ is a Δ_0 formula and $I\Delta_0 \vdash \forall \bar{x} \exists y \phi(\bar{x}, y)$, then the value of y is bounded by a polynomial in $\bar{x}$. However, $I\Delta_0$ does not have some important properties: though it is possible to code sequences in $I\Delta_0$, proving substitution and coding of polynomial length proofs cannot be done. To allow for that, Paris and Wilkie [PW81a, PW81b] later extended $I\Delta_0$ by adding the axiom Ω_1, stating the totality of the function $x^{|x|}$.

The first theory that was explicitly designed in order for all proofs to be feasibly constructible (i.e., constructible in polynomial time) was the equational theory PV, proposed by Cook in 1975 [Coo75]. There, Cobham's recursion-theoretic characterization of polytime was used to construct polytime functions. One motivation for PV was its close relation with the Extended Frege proof system: theorems of PV correspond to families of tautologies with polynomial length proofs. Another candidate for the system with feasibly constructive proofs was the intuitionistic version of PV, IPV, presented in [CU93]. This system, as well as its classical version CPV, includes an induction on NP predicates; the class of Σ_1^B-definable functions of CPV is the same as that of V_1^1 and S_2^1.

Later, Cook [Coo98] used a quantified version of PV called QPV to study the relationship between $\mathtt{NC}^1$ and P from the point of view of corresponding theories, where $QALV$ represents $\mathtt{NC}^1$ in the same sense as QPV represents P. Since PV includes the induction on notation, the system QPV, axiomatized by the universal closures of the theorems of PV, has enough power for polytime reasoning; however, it is possibly weaker than CPV in that it might not prove the induction on NP predicates from CPV.

3.7.1 Second-order theories of arithmetic

In his thesis, Buss gave two examples of second-order theories V_2^1 and U_2^1, capturing EXP and PSPACE, respectively. These systems are S_2^1 and T_2^1 augmented with induction on second-order objects.

Razborov and, at the same time, Takeuti [Raz93, Tak93] show the equivalence between first-order and second-order hierarchies, named RSUV isomorphism, which can be used to show that V_1^1 is equivalent in power to S_2^1 [Raz93]. The important consequence of this result is the appearance of V_1^i hierarchy, equivalent to S_2^i but without the # function and with fewer axioms. Here, the first level of the hierarchy, V_1^0, captures $\mathtt{AC}^0$; V_1^1, like S_2^1, captures P and so on.

This led to development of second-order (two-sorted) systems, where strings are second-order objects rather than large first-order objects of Buss's thesis. A nice presentation of such second-order systems appears in [Zam96]. There, Zambella introduced a hierarchy of second-order theories P_i-def, and studied its relation with the hierarchy of theories of Σ_i^p-comp. The theories Σ_i^p-comp consist of Robinson's Q together with few other axioms, augmented with a comprehension scheme for Σ_i^p formulae. There Σ_i^p are formulae with i alternations of second-order quantifiers, where all quantifiers are polynomially bounded (for second-order objects the bound is on length). The first level of this hierarchy, the theory of Σ_0^p-comp, captures $\mathtt{AC}^0$; the rest are equivalent to the corresponding S_2^i by RSUV isomorphism. The other hierarchy consists of Σ_0^p-comp together with defining axioms for the functions from the corresponding levels of polynomial hierarchy. In his paper Zambella proves several interesting results, such as above mentioned result that if P_i-def $\vdash \Sigma_{i+1}^p$-comp, then polytime hierarchy provably collapses to Σ_{i+3}^p.

In another paper [Zam97] Zambella defines a very elegant theory capturing $\mathtt{L}$, Σ_0^p-rec. There he augments Σ_0^p-comp with an axiom scheme stating that there is a set coding an arbitrarily long path through any given Σ_0^p-definable directed graph without terminating nodes. He then proves that every model of linear arithmetic has an end extension to a model of Σ_0^p-rec. In this paper the collapse result of the previous work is extended to the following: Σ_0^p-rec proves Σ_1^p-comp iff it proves $\Sigma_1^p \subset \Pi_1^p/poly$.

3.7.2 Clote-Takeuti systems

In [CT86, CT92, CT95] Clote and Takeuti describe several other (first-order) systems of bounded arithmetic. Their goal was to design theories of bounded arithmetic for some interesting complexity classes. The main tool for that was bounded recursion on notation with different (e.g., logarithmic) bounds, similar to the Cobham characterization. This was the basis for different induction principles used in the theories.

In [CT86], which came soon after Buss's thesis and uses the techniques from there, in particular witnessing, they characterize multi-exponential complexity classes. In [CT92] there are second-order systems corresponding to $\mathtt{NC}$, $\mathtt{ALOGTIME}$, $\mathtt{L}$ and $\mathtt{NL}$. The first two are based on previous work by Clote on recursion-theoretic characterizations of these classes. The definitions of classes for $\mathtt{L}$ and $\mathtt{NL}$ are not as clean.

In [CT95], a series of theories capturing small complexity classes is defined. The weakest of them is TAC^0, capturing $\mathtt{AC}^0$. This is the smallest class of functions closed under some set of elementary operations. They similarly define theories for $\mathtt{AC}^0(2)$ and $\mathtt{AC}^0(6)$. The other theories defined in this paper are TTC^0, TLS and TPT, for $\mathtt{TC}^0$, $\mathtt{L}$ and $\mathtt{P}$, respectively. The definition of a theory for $\mathtt{TC}^0$, TTC^0, is interesting in that it uses a result from [BIS90] that the descriptive complexity of logic based on majority is $\mathtt{TC}^0$ to define the system.

Chapter 4

V_1-Horn: a system of arithmetic for P.

The class P is one of the most well-studied class in bounded arithmetic. The first theory that was explicitly designed in order for all proofs to be feasibly constructible (i.e., constructible in polynomial time) was the equational theory PV, proposed by Cook in 1975 [Coo75]. There, Cobham's characterization of polynomial-time was used to construct polynomial-time functions. One motivation for PV was its close relation with Extended Frege proof systems for the propositional calculus: theorems of PV give rise to families of tautologies with polynomial-length proofs. Later, Buss showed that Σ_1^B-definable functions of S_2^1, discussed at the end of chapter 3, are precisely the polynomial-time functions. Other characterizations of P include Zambella's P-def (V^0 augmented with functions constructed by Cobham's characterization) from [Zam96], TPV from Clote and Takeuti's [CT95] and several other. Recently Nguyen [NKC04] suggested a yet another system of arithmetic for P using alternating reachability.

Here we describe a system of arithmetic for P based on Grädel's descriptive complexity characterization of P by second-order Horn formulae (see definition 2.2.6). This material was published as [CK01], and later the full version as [CK03].

Recall the definition of Σ_1^B-Horn formulae (definition 3.1.4): a formula $\phi(\bar{a}, \bar{Y})$ is Σ_1^B-Horn if it is of the form

$$\exists P_1 ... \exists P_k \forall x_1 < t_1(\bar{a}, \bar{Y}) ... \forall x_m < t_m(\bar{a}, \bar{Y}) \psi(\bar{x}, \bar{P}, \bar{a}, \bar{Y}),$$

where ψ is a CNF with no more than one positive occurrence of a literal of the form $P_i(t)$ per clause. Using this definition, we define the system of arithmetic V_1-Horn as ϕ

with $\Phi = \Sigma_1^B$-Horn. That is, the formulae in the comprehension scheme are restricted to Σ_1^B-Horn formulae. Our goal is to apply the definability theorem 3.3.13 to V_1-Horn to show that the class of Σ_1^B-definable functions of V_1-Horn is FP, the class of all polytime functions. In addition to that, we show that it is equivalent to Zambella's system P-def, and thus to Cook's QPV, and is therefore a minimal theory for P, as opposed to V^1

The outline of the proof is similar to the proof of definability theorem for V^0 in section 3.5. However, the properties that hold trivially for V^0, such as closure under Σ_0^B reductions and constructiveness, require much more work in V_1-Horn.

There are two restrictions in the Σ_1^B-Horn formulae. The first is common to all restricted Σ_1^B formulae: there are no existential first-order quantifiers. The second, specific to Σ_1^B-Horn, is that no more than one positive literal is allowed per clause. The first restriction only matters in the proof that Σ_1^B-Horn can simulate Σ_0^B; later, we use second-order variables and the (semantic) closure of Σ_1^B-Horn under complementation to simulate first-order existential quantifiers.

To handle the second restriction we introduce the technique of tilde-counterparts. It is used slightly differently in different contexts, but the main idea behind it is to use pairs of variables to represent one variable, where the second variable of each pair is a tilde-counterpart of the first: it is semantically equivalent to the negation of the first variable. If the original variable is called P, we use the notation $\tilde{P}$ to refer to its semantic opposite. Usually our formulae contain a clause $(\neg P \vee \neg \tilde{P})$ for every such pair of variables; the conditions that replace $(P \vee \tilde{P})$ depend on the context.

Notation 4.0.1. If P is a second-order variable, then its "tilde-counterpart" $\tilde{P}$ denotes a second-order variable whose intended interpretation is $\neg P$.

We start by showing that Σ_0^B formulae can be simulated by Σ_1^B-Horn formulae provably in V_1-Horn. That implies $V^0 \subset V_1$-Horn. Then, we show the strong constructiveness condition by formalizing the Horn satisfiability proof in V_1-Horn. From that, we get the closure under complementation and thus under $\mathtt{AC}^0$ reductions. At this point we can apply the Definability Theorem (theorem 3.3.13) to conclude that V_1-Horn captures FP. We explicitly show the equivalence between V_1-Horn and more conventional systems capturing P, which are based on Cobham recursion-theoretic characterization of P. Lastly, we show that V_1-Horn is finitely axiomatizable by replacing Σ_1^B-Horn comprehension with comprehension over the satisfiability predicate, and adding the finite set of axioms of V^0.

4.1 V_1-Horn extends V^0

We start by showing that V_1-Horn can prove Σ_0^B comprehension. For that, we show how to simulate Σ_0^B formulae by Σ_1^B-Horn formulae, provably in V_1-Horn. We start by simple observation:

Lemma 4.1.1. *If ϕ_1 and ϕ_2 are Σ_1^B-Horn formulas, then $\phi_1 \wedge \phi_2$ is logically equivalent to a Σ_1^B-Horn formula.*

Proof. Take a suitable prenex form of $\phi_1 \wedge \phi_2$. □

4.1.1 Simulating first-order bounded existential quantification

A major inconvenience of restricted Σ_1^B formulae, which makes non-trivial the proof that V-Φs based on restricted Σ_1^B contain V^0, is lack of first-order existential quantifiers in restricted Σ_1^B formulae. In general we cannot allow such quantifiers without increasing the apparent expressive power of the formulas, as pointed out in the 3-colourability example (example 2.0.4). However, it is possible to introduce bounded existential quantifiers in some contexts.

We now introduce formulae $\textsc{Search}_k$, which have only universal first-order quantifiers and are Horn with respect to all of their second-order variables. They will allow a Σ_1^B-Horn formula to represent $\exists z < bX(\bar{y}, z)$. $\textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$ asserts that $S(\bar{y}, i)$ holds iff $X(\bar{y}, z)$ holds for some $z < i$, where $\bar{b}$ stands for $b_1, ..., b_k$, and $\bar{y}$ stands for $y_1, ..., y_k$. We use $\bar{y} < \bar{b}$ for $y_1 < b_1 \wedge ... \wedge y_k < b_k$.

Definition 4.1.2. For each $k \geq 1$ $\textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$ is the Π_1^b- Horn formula

$$\begin{aligned}
\forall \bar{y} < \bar{b} \forall i < b(\neg S(\bar{y}, 0) \wedge \tilde{S}(\bar{y}, 0))& \\
\wedge(\neg S(\bar{y}, i+1) \vee \neg \tilde{S}(\bar{y}, i+1))& \\
\wedge(S(\bar{y}, i) \rightarrow S(\bar{y}, i+1))& \\
\wedge(X(\bar{y}, i) \rightarrow S(\bar{y}, i+1))& \\
\wedge(\tilde{S}(\bar{y}, i) \wedge \tilde{X}(\bar{y}, i) \rightarrow \tilde{S}(\bar{y}, i+1))&
\end{aligned}$$

Lemma 4.1.3. *V_1-Horn proves the following:*

$$\text{(i) } \forall z < b(X(\bar{y}, z) \leftrightarrow \neg\tilde{X}(\bar{y}, z)) \wedge \bar{y} < \bar{b} \rightarrow \exists S \exists \tilde{S} \text{ SEARCH}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$$

$$\text{(ii) } \forall z < b(X(\bar{y}, z) \leftrightarrow \neg\tilde{X}(\bar{y}, z)) \wedge \text{SEARCH}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X}) \wedge \bar{y} < \bar{b}$$
$$\rightarrow (S(\bar{y}, b) \leftrightarrow \exists z < bX(\bar{y}, z)) \wedge (\tilde{S}(\bar{y}, b) \leftrightarrow \forall z < b\tilde{X}(\bar{y}, z))$$

Proof. First we prove part (i). Arguing in V_1-Horn, there are two cases. If $\forall z < b\tilde{X}(\bar{y}, b)$ then use $k+1$-ary comprehension (Lemma 3.2.7) to define $S(\bar{y}, z)$ false and $\tilde{S}(\bar{y}, z)$ true, for all $z < b$. The clauses in the definition of SEARCH_k are clearly satisfied in this case. Otherwise, by the LNP there is a least number $z_0 < b$ such that $X(\bar{y}, x_0)$. Use $k+1$-ary comprehension to define $S(\bar{y}, z)$ false for $z \leq z_0$ and true for $z_0 < z < b$, and define $\tilde{S}(\bar{y}, z) \leftrightarrow \neg S(\bar{b}, z)$. Again $\text{SEARCH}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$ holds.

To prove (ii) we use the same two cases as for (i). If $\forall z < b\tilde{X}(\bar{y}, b)$ we use the definition of SEARCH_k to show by induction on z that $S(\bar{y}, z)$ is false and $\tilde{S}(\bar{y}, z)$ is true for $z \leq b$, so (ii) holds in this case. For the second case we know from above what S and $\tilde{S}$ must be, and we again prove our claim by induction on z. Again (ii) follows. □

4.1.2 The Σ_0^B formulas are equivalent to Σ_1^B-Horn

Consider a Σ_0^B formula $Q_1 y_1 < b_1 ... Q_k y_k < b_k \phi(\bar{y})$, where each Q_i is either $\forall$ or $\exists$. The proof of the following lemma shows how to conjoin copies of $\text{SEARCH}(...)$ to define arrays $S_0, ..., S_k$ such that

$$S_i(y_1, ..., y_{k-i}) \leftrightarrow Q_{k-i+1} y_{k-i+1} < b_{k-i+1} \phi(\bar{y}).$$

These are used to form an equivalent Σ_1^B-Horn formula.

Theorem 4.1.4. *Let $\psi(\bar{y})$ be a Σ_0^B formula which may have other free variables besides $\bar{y}$ but does not involve any of the variables $S, \tilde{S}, \bar{W}$. Then there is a formula $\psi^*(\bar{b}, S, \tilde{S}, \bar{W})$ not involving $\bar{y}$ but which may have other variables of ψ not indicated and which is Π_1^b-Horn with respect to $S, \tilde{S}, \bar{W}$ such that V_1-Horn proves the following:*

$$(i) \;\; \exists S \exists \tilde{S} \exists \bar{W} \psi^*(\bar{b}, S, \tilde{S}, \bar{W})$$
$$(ii) \;\; \psi^*(\bar{b}, S, \tilde{S}, \bar{W}) \rightarrow \forall \bar{y} < \bar{b}[(S(\bar{y}) \leftrightarrow \psi(\bar{y})) \wedge (\tilde{S}(\bar{y}) \leftrightarrow \neg\psi(\bar{y}))]$$

Proof. We may assume that ψ is in prenex form, and proceed by induction on the number of quantifiers. For the base case ψ is quantifier-free, and we take $\psi^*(\bar{b}, S, \tilde{S})$ to be equivalent to

$$\forall \bar{y} < \bar{b}[(S(\bar{y}) \leftrightarrow \psi(\bar{y})) \wedge (\tilde{S}(\bar{y}) \leftrightarrow \neg\psi(\bar{y}))]$$

The formula in brackets can be written in conjunctive normal form, in which case $\psi^*(\bar{b}, S, \tilde{S})$ is Π_1^b-Horn with respect to S and $\tilde{S}$ and obviously satisfies (ii). Also (i) is easily proved by defining S and $\tilde{S}$ using Σ_1^B-Horn comprehension.

For the induction step, assume that $\psi(\bar{y})$ is $\exists z < t\phi(\bar{y}, z)$, where t is a term not involving z. By the induction hypothesis applied to ϕ there is a formula $\phi^*(\bar{b}, b, S_1, \tilde{S}_1, \bar{W})$ not involving $\bar{y}, z$ which is Π_1^b-Horn with respect to $S_1, \tilde{S}_1, \bar{W}$ which satisfies (i) and (ii) (with ϕ, ϕ^*, S_1 for ψ, ψ^*, S). In fact the induction hypothesis (ii) states

$$\phi^*(\bar{b}, b, S_1, \tilde{S}_1, \bar{W}) \to \forall \bar{y} < \bar{b} \forall z < b(S_1(\bar{y}, z) \leftrightarrow \phi(\bar{y}, z)) \wedge (\tilde{S}_1(\bar{y}, z) \leftrightarrow \neg\phi(\bar{y}, z))$$

We define $\psi^*(\bar{b}, S, \tilde{S}, S_1, \tilde{S}_1, \bar{W})$ to be the prenex form of

$$\phi^*(\bar{b}, t, S_1, \tilde{S}_1, \bar{W}) \wedge \textsc{Search}_k(\bar{b}, t, S, \tilde{S}, S_1, \tilde{S}_1) \tag{4.1}$$

Note that this is Π_1^b-Horn with respect to the displayed second-order variables. By the induction hypothesis (i) there exists $S_1, \tilde{S}_1, \bar{W}$ satisfying ϕ^*. By the induction hypothesis (ii) we have $S_1 \leftrightarrow \neg\tilde{S}_1$. Hence by (i) of Lemma 4.1.3 we know $S, \tilde{S}$ exist satisfying (i) in the present lemma for ψ^* as defined above.

To prove (ii), assume $\bar{y} < \bar{b}$ and $\psi^*(\bar{b}, S, \tilde{S}, S_1, \tilde{S}_1, \bar{W})$. By the induction hypothesis (ii) for ϕ^* and (ii) of Lemma 4.1.3 we have $S(\bar{y}, t) \leftrightarrow \exists z < t\phi(\bar{y}, z)$ and $\tilde{S}(\bar{y}, t) \leftrightarrow \forall z < b\neg\phi(\bar{y}, z)$, as required.

For the induction step in case $\psi(\bar{y})$ is $\forall z < t\phi(\bar{y}, z)$ we simply modify the arguments of $\textsc{Search}_k$ in (4.1) by interchanging S with $\tilde{S}$ and S_1 with $\tilde{S}_1$. □

Corollary 4.1.5. *Every Σ_0^B formula is provably equivalent in V_1-Horn to a Σ_1^B-Horn formula.*

Proof. Let ψ be a Σ_0^B formula not involving y and let $\psi^*(b, S, \tilde{S}, \bar{W})$ result from applying the above Lemma to $\psi(y)$. Then $\psi(y) \leftrightarrow \psi(0)$ so V_1-Horn proves

$$\psi(y) \leftrightarrow \exists S \exists \tilde{S} \exists \bar{W}(\psi^*(1, S, \tilde{S}, \bar{W}) \wedge S(0))$$

The right hand side is easily equivalent to a Σ_1^B-Horn formula. □

Thus V_1-Horn proves the induction and comprehension schemes for Σ_0^B formulas, and hence it is an extension of V^0.

4.1.3 Collapse of V-Σ_i^B-Horn hierarchy to V_1-Horn

Grädel [Grä91] showed that it is possible to represent a $SO\exists$-Horn formula preceded by alternating SO quantifiers by a $SO\exists$-Horn formula, which implies the collapse of the SO-Horn hierarchy to $SO\exists$-Horn. Similarly, we define a hierarchy of theories with comprehension over Σ_i^B-Horn. Let V-Horn $\equiv \bigcup V$-Σ_i^B-Horn. In this section, we show that V-Horn is fully conservative over V_1-Horn. The proof proceeds by formalizing Grädel's argument in V_1-Horn.

Theorem 4.1.6. *Every Σ_1^B-Horn formula preceded by a sequence of (possibly alternating) second-order quantifiers is provably equivalent in V_1-Horn to a Σ_1^B-Horn formula. Moreover, a prefix of alternating second-order quantifiers can be replaced with a single existential second-order quantifier.*

This follows from the Replacement Lemma 3.2.12 and the following Lemma.

Lemma 4.1.7. *If $\phi(P,\bar{Q})$ is Π_1^b-Horn with respect to $P,\bar{Q}$ then V_1-Horn proves*

$$\forall P\exists\bar{Q}\phi(P,\bar{Q}) \leftrightarrow \forall y \le u\exists\bar{Q}\phi'(y,\bar{Q})$$

where if $P(t_1),...,P(t_k)$ is a list of all occurrences of P in ϕ, then u is the term $t_1+...+t_k+1$, and $\phi'(y,\bar{Q})$ is obtained from $\phi(P,\bar{Q})$ by replacing each $P(t_i)$ by $t_i \neq y$.

Proof. First note that V_1-Horn proves $t_i < u$, for $i = 1,...,k$. To prove the left-to-right direction, for each y simply use comprehension to define P by the condition

$$\forall i \le u(P(i) \leftrightarrow i \neq y)$$

The proof of the converse is more complicated. Given P we use Σ_0^B comprehension to define the sets $\bar{Q}$ in terms of P and the $\bar{Q}$ from the RHS. There are two cases. The easy case is that $\forall z < uP(z)$ holds. Then take $y = u$, and the $\bar{Q}$ which satisfy the RHS will also satisfy the LHS, since $t_i \neq y$ for each i.

Now suppose $\exists z < u\neg P(z)$. By the Replacement Lemma applied to the RHS there are $\bar{Q}'$ satisfying $\forall y \le u\phi'(y,\bar{Q}'^{[y]})$. For each $Q_j \in \bar{Q}$ use Σ_0^B comprehension to define Q_j by the condition

$$\forall z < u_j(Q_j(z) \leftrightarrow \forall y < u(P(y) \vee Q_j'^{[y]}(z)))$$

where u_j is an upper bound on all terms v such that $Q_j(v)$ occurs in ϕ.

It remains to show in V_1-Horn that this definition of $\bar{Q}$ satisfies $\phi(P, \bar{Q})$. We argue the contrapositive: If $\neg\phi(P, \bar{Q})$ then $\neg\phi'(y, \bar{Q}'^{[y]})$ for some y. Recall that ϕ begins with a string of bounded universal quantifiers $\forall \bar{x} \leq \bar{w}$, followed by a quantifier-free formula ψ which is Horn with respect to $P, \bar{Q}$. Fix values for the variables $\bar{x}$ which cause some clause $C(\bar{x}, P, \bar{Q})$ in ψ to be false. We will show that the corresponding clause $C'(\bar{x}, y, \bar{Q}'^{[y]})$ in ϕ' is false for a suitable choice of y. If the head of C is $P(t_i)$, then take $y = t_i$. If the head of C is $Q_j(v)$, then choose $y \leq u$ satisfying $(\neg P(y) \wedge \neg Q_j'^{[y]}(v))$. Such a y must exist because $\neg Q_j(v)$. Otherwise choose any $y \leq u$. In each case it is easy to see that $C'(\bar{x}, y, \bar{Q}'^{[y]})$ is false. $\square$

4.2 Encoding the Horn SAT algorithm by a Σ_1^B-Horn formula

In order to prove the constructiveness property for the Σ_1^B-Horn formulae, we will show how to encode the Horn satisfiability algorithm by a Σ_1^B-Horn formula. This formula can be used to define a function giving a satisfying assignment for the formula. Since the algorithm is deterministic, there will always be a particular satisfying assignment (or a proof that there is none) outputted by the algorithm; thus, the value of the resulting function is uniquely defined.

4.2.1 The Σ_1^B-Horn evaluation algorithm

The polynomial-time algorithm for evaluation of the Σ_1^B-Horn formulae follows the outline in section 3.1.4. That is, we start with a Σ_1^B-Horn formula and convert it to a polynomial-length propositional Horn formula. Now, the propositional Horn satisfiability algorithm proceeds as follows:

1. Initially, set all variables to false ($\perp$).

2. Find all unsatisfied clauses. If an unsatisfied clause has a positive literal, set the corresponding variable to $\top$.

3. Repeat the previous step until either there is an unsatisfied clause with no positive literals (then the formula is unsatisfiable) or there are no unsatisfied clauses (then the formula is satisfiable, and the current assignment is a satisfying assignment).

Note that the maximal number of steps this algorithm makes is equal to the number of propositional variables, and thus is bounded from above by the upper bound on all terms $t()$ occurring as $P(t)$.

4.2.2 The Σ_1^B-Horn constructiveness theorem

Here we show that a run of the Horn satisfiability algorithm described in the proof of Theorem 3.1.5 can be represented by a Σ_1^B-Horn formula RUN. A simple corollary is that the negation of a Σ_1^B-Horn formula is provably equivalent to a Σ_1^B-Horn formula. In other words, V_1-Horn proves that P is closed under complementation.

Theorem 4.2.1. *Let ϕ be a Σ_1^B-Horn formula which does not involve R or $\tilde{R}$. Then there is a formula $\text{RUN}_\phi(R, \tilde{R})$ whose free variables include those of ϕ in which the only atomic subformulas involving R and $\tilde{R}$ are $R(0)$ and $\tilde{R}(0)$ and such that $\exists R \exists \tilde{R} \text{RUN}_\phi(R, \tilde{R})$ is a Σ_1^B-Horn formula and V_1-Horn proves the following:*

$$(i)\ \exists R \exists \tilde{R}\ \text{RUN}_\phi(R, \tilde{R})$$
$$(ii)\ \text{RUN}_\phi(R, \tilde{R}) \to [(R(0) \leftrightarrow \phi) \wedge (\tilde{R}(0) \leftrightarrow \neg\phi)]$$

Corollary 4.2.2. *If ϕ is Σ_1^B-Horn, then $\neg\phi$ is provably equivalent in V_1-Horn to a Σ_1^B-Horn formula NEG_ϕ.*

Proof. We may take NEG_ϕ to be $\text{RUN}_\phi(\bot, \top)$; that is $\text{RUN}_\phi(R, \tilde{R})$ with each occurrence of the formula $R(0)$ replaced by $\bot$ (FALSE) and each occurrence of the formula $\tilde{R}(0)$ replaced by $\top$ (TRUE). □

Corollary 4.2.3. *Σ_1^B-Horn has the strong closure property (property 1 from page 38). That is, the class of formulas provably equivalent in V_1-Horn to a Σ_1^B-Horn formula is closed under $\neg$, $\wedge$, $\vee$, and bounded first-order quantification.*

Proof. The preceding corollary handles the case of $\neg$, Lemma 4.1.1 handles the case of $\wedge$, and the Replacement Lemma 3.2.12 handles the case of $\forall y < t$. The other cases follow by DeMorgan's laws. □

Theorem 4.2.1 can be generalized to the case in which arrays $R(\bar{y})$ and $\tilde{R}(\bar{y})$ code values of $\phi(\bar{y})$ and $\neg\phi(\bar{y})$.

Corollary 4.2.4. *Let $\phi(\bar{y})$ be a Σ_1^B-Horn formula which does not involve R or $\tilde{R}$. Then there is a formula $\text{RUN}_{\phi(\bar{y})}(\bar{b}, R, \tilde{R})$ which does not have $\bar{y}$ free but whose free variables include any other free variables of ϕ such that*

$$\exists R \exists \tilde{R} \text{RUN}_{\phi(\bar{y})}(R, \tilde{R})$$

is a Σ_1^B-Horn formula and V_1-Horn proves the following:

(*i*) $\exists R \exists \tilde{R}\ \text{RUN}_{\phi(\bar{y})}(\bar{b}, R, \tilde{R})$

(*ii*) $\text{RUN}_{\phi(\bar{y})}(\bar{b}, R, \tilde{R}) \rightarrow \forall \bar{y} < \bar{b}[(R(\bar{y}) \leftrightarrow \phi(\bar{y})) \wedge (\tilde{R}(\bar{y}) \leftrightarrow \neg\phi(\bar{y}))]$

Proof. We take $\text{RUN}_{\phi(\bar{y})}$ such that V_1-Horn proves

$$\text{RUN}_{\phi(\bar{y})}(\bar{b}, R, \tilde{R}) \leftrightarrow$$
$$\forall \bar{y} < \bar{b} \exists R' \exists \tilde{R}'[\text{RUN}_{\phi}(R', \tilde{R}') \wedge (R(\bar{y}) \leftrightarrow R'(0)) \wedge (\tilde{R}(\bar{y}) \leftrightarrow \tilde{R}'(0))]$$

We may take $\text{RUN}_{\phi(\bar{y})}$ to be Σ_1^B-Horn by placing the subformula enclosed in [...] above by a suitable prenex form and applying Corollary 4.2.3. To prove (i) we use Σ_1^B-Horn comprehension to define $R(\bar{y})$ satisfying $R(\bar{y}) \leftrightarrow \phi(\bar{y})$ and use Σ_1^B-Horn comprehension together with corollary 4.2.2 to define $R(\bar{y}) \leftrightarrow \neg\phi(\bar{y})$ and then apply (i) and (ii) of Theorem 4.2.1 to R' and $\tilde{R}'$. To prove (ii) we use (ii) in Theorem 4.2.1. □

We now turn to the proof of Theorem 4.2.1. This proof is long, however it shows constructiveness and strong closure of Σ_1^B-Horn, as well as giving us techniques to prove equivalence of V_1-Horn to other theories capturing P.

By the lemma 3.2.8, it suffices to prove Theorem 4.2.1 for Σ_1^B-Horn formulae with a single existential quantifier, that is formulae of the form

$$\phi \equiv \exists P \forall x_1 \leq t_1 ... \forall x_k \leq t_k \psi(\bar{x}, P) \tag{4.2}$$

where ψ is Horn with respect to P.

The algorithm we wish to represent has two main steps. First create a propositional Horn formula PROP^{ϕ} (which depends on the values for the free variables in ϕ), as described in section 3.1.4, and second apply the Horn Sat algorithm described above to determine whether PROP^{ϕ} is satisfiable.

There are three pieces of information we need to encode about each clause: its positive literal, its negative literals and whether it is already valid due to first-order atoms. Thus, we need three arrays, called, respectively, C, D and V to encode these three facts about each clause; so PROP^ϕ is encoded by C, D, V, and we will present a Σ_1^B-Horn formula $\text{PROP}_\phi(C, \tilde{C}, D, \tilde{D}, V, \tilde{V})$ which defines these arrays and their negations. Besides the indicated free variables, PROP_ϕ also has as free variables the free variables of ϕ. In the proof of Lemma 4.2.5, we will give Σ_0^B formulae ψ_C, ψ_D and ψ_V defining the respective arrays and their negations.

For the second step we present a Σ_1^B-Horn formula $\text{HORNSAT}(a, b, C, \tilde{C}, D, \tilde{D}, V, \tilde{V}, R, \tilde{R})$ (with all free variables indicated) which is independent of ϕ and which sets the result variable $R(0)$ true iff PROP^ϕ is satisfiable.

The arrays C, D, V together with the scalars a, b completely specify the formula PROP^ϕ as follows. The atoms of PROP^ϕ are $P(0), ..., P(a-1)$, and the clauses are $cl_0, ..., cl_{b-1}$. We allow both the empty clause and the special clause TRUE. The arrays C, D, V are defined as follows: For $0 \le x < b$, $0 \le v < a$

- $C(x, v)$ asserts that clause cl_x contains the negative literal $\neg P(v)$.
- $D(x, v)$ asserts that clause cl_x contains the positive literal $P(v)$.
- $V(x)$ asserts that clause cl_x is the clause TRUE.

Since PROP^ϕ is a Horn formula, for each x, $D(x, v)$ can be true for at most one v.

The array bounds a, b are represented by terms $\hat{a}, \hat{b}$ in the free variables of ϕ and are determined as follows. For each term s in $\psi(\bar{x}, P)$ in (4.2) let $\hat{s}$ be the result of replacing each variable $x_1, ..., x_k$ by its respective upper bound $t_1, ..., t_k$. Then the upper bound $\hat{a}$ on the arguments of $P()$ is

$$\hat{a} \equiv \hat{s}_1 + ... + \hat{s}_\ell$$

where $s_1, ..., s_\ell$ is a list of all terms such that $P(s_i)$ or $\neg P(s_i)$ occurs in ϕ.

The upper bound $\hat{b}$ on the number of clauses in PROP^ϕ is

$$\hat{b} \equiv \langle t_1, ..., t_s, m \rangle$$

where $t_1, ..., t_s$ are as in (4.2), m in the number of clauses in $\psi(\bar{x}, P)$, and $\langle ... \rangle$ is the tupling function (3.2.5).

Using the abbreviation

$$\bar{Q} \equiv C, \tilde{C}, D, \tilde{D}, V, \tilde{V}$$

we can now choose $\text{RUN}_\phi(R, \tilde{R})$ to be a Σ_1^B-Horn formula such that

$$\text{RUN}_\phi(R, \tilde{R}) \leftrightarrow \exists \bar{Q}[\text{PROP}_\phi(\bar{Q}) \wedge \text{HORNSAT}(\hat{a}, \hat{b}, \bar{Q}, R, \tilde{R})] \tag{4.3}$$

In fact we take $\text{RUN}_\phi(R, \tilde{R})$ to be a suitable prenex form of the right hand side.

4.2.3 Definition of $\mathbf{Prop}_\phi(C, \tilde{C}, D, \tilde{D}, V, \tilde{V})$

Below we define three Σ_0^B formulas $\psi_C(x, v), \psi_D(x, v), \psi_V(x)$ which characterize the three arrays C, D, V.

Lemma 4.2.5. $\text{PROP}_\phi(\bar{Q})$ *can be defined in such a way that* $\exists\bar{Q}\text{PROP}_\phi(\bar{Q})$ *is* Σ_1^B*-Horn and* V_1*-Horn proves*

(i) $\exists\bar{Q}\text{PROP}_\phi(\bar{Q})$

(ii) $\text{PROP}_\phi(\bar{Q}) \;\rightarrow\; \forall v < \hat{a}\forall x < \hat{b}$

$$[(C(x,v) \leftrightarrow \psi_C(x,v)) \wedge (D(x,v) \leftrightarrow \psi_D(x,v)) \wedge (V(x) \leftrightarrow \psi_V(x))$$
$$\wedge\, (\tilde{C}(x,v) \leftrightarrow \neg\psi_C(x,v)) \wedge (\tilde{D}(x,v) \leftrightarrow \neg\psi_D(x,v)) \wedge (\tilde{V}(x) \leftrightarrow \neg\psi_V(x))]$$

Proof. We apply Theorem 4.1.4 once each for ψ_C, ψ_D, ψ_V with S in the theorem taken to be C, D, V, respectively, to obtain three Σ_1^B-Horn formulas $\psi_C^*, \psi_D^*, \psi_V^*$, and then let $\text{PROP}_\phi(\bar{Q})$ be a prenex form of their conjunction. □

To define ψ_C, ψ_D, ψ_V let the Horn formula $\psi(\bar{x}, P)$ in (4.2) be the conjunction of the clauses $CL_0, ..., CL_{m-1}$. For $j = 0, ..., m-1$ let $\psi_j(\bar{x})$ be the quantifier-free formula which results by deleting all literals involving P from CL_j. Then we define

$$\psi_V(x) \equiv \forall x_1 \le t_1, ..., \forall x_k \le t_k$$
$$[(x = \langle x_1, ..., x_k, 0\rangle \rightarrow \psi_0(\bar{x})) \wedge ... \wedge (x = \langle x_1, ..., x_k, m-1\rangle \rightarrow \psi_{m-1}(\bar{x}))]$$

Now let $\mathcal{S}$ be the set of indices j such that the clause CL_j has a positive literal of the form $P(u)$, and let for $j \in \mathcal{S}$ let that literal be $P(u_j(\bar{x}))$. Then we define

$$\psi_D(x, v) \equiv \neg\psi_V(x) \wedge \exists x_1 \le t_1, ..., \exists x_k \le t_k \bigvee_{j\in\mathcal{S}} [x = \langle x_1, ..., x_k, j\rangle \wedge v = u_j(\bar{x})]$$

For $j = 0, ..., m-1$ let $\neg P(u_j^0), ..., \neg P(u_j^{n_j - 1)})$ be the literals involving $\neg P$ in CL_j. Then

$$\psi_C(x, v) \equiv \neg\psi_V(x) \wedge \exists x_1 \le t_1, ..., \exists x_k \le t_k \bigvee_{j=0}^{m-1} \bigvee_{i=0}^{n_j-1} [x = \langle x_1, ..., x_k, j\rangle \wedge v = u_j^i(\bar{x})]$$

4.2.4 Definition of $\mathbf{HornSat}(a, b, C, \tilde{C}, D, \tilde{D}, V, \tilde{V}, R, \tilde{R})$

Although the Horn satisfiability algorithm is easy to describe informally, it is not straightforward to formalize in V_1-Horn. The propositional Horn satisfiability problem is complete for P, [GHR95], and hence cannot be represented by a Σ_0^B formula. We need a more general form of Theorem 4.1.4 which allows us to use a Σ_1^B-Horn formula to define an array representing a given Σ_0^B formula, now in the presence of complementary variables $U, \tilde{U}$ which we want to existentially quantify.

Lemma 4.2.6. *Let $\psi(\bar{y}, U)$ be a Σ_0^B formula which may have free variables not indicated, but does not involve any of the variables $S, \tilde{S}, \bar{W}, \tilde{U}$ and has no occurrence of $|U|$. Then there is a formula $\psi^*(\bar{b}, S, \tilde{S}, \bar{W}, U, \tilde{U})$ not involving $\bar{y}$ but which may have other variables of ψ not indicated and which is Π_1^b-Horn with respect to $S, \tilde{S}, \bar{W}, U, \tilde{U}$ such that V_1-Horn proves the following:*

$$(i)\quad \exists S \exists \tilde{S} \exists \bar{W} \psi^*(\bar{b}, S, \tilde{S}, \bar{W}, U, \tilde{U})$$

$$\begin{aligned} (ii)\quad & \psi^*(\bar{b}, S, \tilde{S}, \bar{W}, U, \tilde{U}) \wedge \forall z < s(U(z) \leftrightarrow \neg \tilde{U}(z)) \\ & \rightarrow\ \forall \bar{y} < \bar{b}[(S(\bar{y}) \leftrightarrow \psi(\bar{y}, U)) \wedge (\tilde{S}(\bar{y}) \leftrightarrow \neg \psi(\bar{y}, U))] \end{aligned}$$

where the term s is a provable upper bound on all terms r such that $U(r)$ occurs in ψ. A similar statement applies more generally to formulas $\psi(\bar{y}, U_1, ..., U_\ell)$ where the arrays U_i may have various dimensions.

Proof. We proceed by induction on the number of quantifiers in ψ, as in the proof of Theorem 4.1.4. The induction step is the same as before, but the base case now becomes more interesting. In this case ψ is quantifier-free, and we observe that the formula $(S(\bar{y}) \leftrightarrow \psi(\bar{y}, U))$ can be put into a conjunctive normal form which is Horn with respect to $S, U, \tilde{U}$ by taking the original CNF and replacing each positive literal of the form $U(r)$ by $\neg \tilde{U}(r)$. A similar remark applies to the formula $(\tilde{S}(\bar{y}) \leftrightarrow \neg \psi(\bar{y}, U))$. $\square$

The algorithm represented by $\textsc{HornSat}(a, b, C, \bar{Q}, R, \tilde{R})$ attempts to find a satisfying assignment to the Horn formula $\textsc{Prop}^\phi$ described by the parameters a, b, C, D, V. This is done by filling in an array $T(t, v)$, where $T(t, v)$ is the truth value assigned to the atom $P(v)$ after step t, $0 \leq t, v < a$. Initially $T(0, v)$ is false, and at step $t + 1$ $T(t + 1, v)$ sets each $P(v)$ to true whenever $P(v)$ occurs positively in some clause not satisfied after step t. Once $P(v)$ is set to true, it is never changed to false.

The following Σ_0^B formulas describe the array T and its negation $\tilde{T}$. First, $\textsc{Init}$ initializes T.

$$\textsc{Init} \equiv \forall v < a(\tilde{T}(0, v) \wedge \neg T(0, v))$$

In general we need to define a Σ_0^B formula $\textsc{Step}(v, T^{[t]})$ which expresses the value of $T(t + 1, v)$ in terms of the values $T^{[t]}$ of T at time t. We define $\textsc{Step}$ using the one-dimensional array T_1 for $T^{[t]}$. First we need to define $\textsc{ClauseSat}(x, T_1)$ which asserts that assignment T_1 satisfies clause cl_x in $\textsc{Prop}^{\phi}$.

$$\textsc{ClauseSat}(x, T_1) \;\equiv\; V(x) \vee \exists v < a[(C(x,v) \wedge \neg T_1(v)) \vee (D(x,v) \wedge T_1(v))]$$

Now $\textsc{Step}(v, T_1)$ holds iff either $P(v)$ is true under T_1 or there is a clause not satisfied by T_1 which has a positive literal $P(v)$.

$$\textsc{Step}(v, T_1) \;\equiv\; T_1(v) \vee \exists x < b(\neg\textsc{ClauseSat}(x, T_1) \wedge D(x,v)) \tag{4.4}$$

Now we apply Lemma 4.2.6 taking ψ to be $\textsc{Step}$ and $\bar{U}$ to be C, D, V, T_1 to obtain the formula $\textsc{Step}^*(a, S, \tilde{S}, \bar{W}, \bar{Q}, T_1, \tilde{T}_1)$ which is Π_1^b-Horn with respect to all of its displayed second-order variables and for which V_1-Horn proves the following versions of (i) and (ii) in the lemma.

$$\begin{aligned}
&(i)' \;\; \exists S \exists \tilde{S} \exists \bar{W}\, \textsc{Step}^*(a, S, \tilde{S}, \bar{W}, \bar{Q}, T_1, \tilde{T}_1)\\
&(ii)' \;\; \textsc{Step}^*(a, S, \tilde{S}, \bar{W}, \bar{Q}, T_1, \bar{T}_1) \wedge \textsc{Neg} \wedge \forall v < a(T_1(v) \leftrightarrow \neg\tilde{T}_1(v))\\
&\qquad \rightarrow \;\forall v < a[(S(v) \leftrightarrow \textsc{Step}(v, T_1)) \wedge (\tilde{S}(v) \leftrightarrow \neg\textsc{Step}(v, T_1))]
\end{aligned}$$

where we define $\textsc{Neg}$ by

$$\begin{aligned}
\textsc{Neg}(a, b, \bar{Q}) \;\equiv\; & \forall v < a \forall x < b[(C(x,v) \leftrightarrow \neg\tilde{C}(x,v))\\
& \wedge (D(x,v) \leftrightarrow \neg\tilde{D}(x,v)) \wedge (V(x) \leftrightarrow \neg\tilde{V}(x))].
\end{aligned} \tag{4.5}$$

Next we use the following formula to define the array T, where we have substituted $T^{[t+1]}$ for S and $T^{[t]}$ for T_1 in $\textsc{Step}^*$.

$$\begin{aligned}
\textsc{TDef}(a, b, \bar{Q}, T, \tilde{T}) \;\equiv\; & \textsc{Init}(T, \tilde{T}) \wedge \forall t < a \exists \bar{W}\\
& \textsc{Step}^*(a, T^{[t+1]}, \tilde{T}^{[t+1]}, \bar{W}, T^{[t]}, \tilde{T}^{[t]}).
\end{aligned} \tag{4.6}$$

Lemma 4.2.7. *V_1-Horn proves*

$$\begin{aligned}
&(i) \;\; \exists T \exists \tilde{T} \;\textsc{TDef}(a, b, \bar{Q}, T, \tilde{T})\\
&(ii) \;\; \textsc{TDef}(a, b, \bar{Q}, T, \tilde{T}) \wedge \textsc{Neg} \rightarrow \;\forall t < a \forall v < a\\
&\qquad [(T(t+1, v) \leftrightarrow \textsc{Step}(v, T^{[t]})) \wedge (\tilde{T}(t+1, v) \leftrightarrow \neg\textsc{Step}(v, T^{[t]}))]
\end{aligned}$$

Proof. To prove (i), let $\textsc{TDef}'$ be obtained from $\textsc{TDef}$ by replacing the bounded quantifier $\forall t < a$ in the above definition of $\textsc{TDef}$ by $\forall t < y$. Define

$$\phi(y) \equiv \exists T \exists \tilde{T} \;\textsc{TDef}'(y, a, b, \bar{Q}, T, \tilde{T})$$

By the Replacement Lemma $\phi(y)$ is equivalent to a Σ_1^B-Horn formula, so we may use the induction scheme for $\phi(y)$. This will establish (i), which is simply $\phi(a)$.

For the base case $y = 0$ we need only satisfy INIT, so we use the comprehension scheme to define T to be identically false and $\tilde{T}$ to be identically true.

Now assume the induction hypothesis and suppose that $T, \tilde{T}$ satisfy the existential quantifiers in $\phi(y)$. Let $S, \tilde{S}$ satisfy the existential quantifiers in $(i)'$ when $T_1, \tilde{T}_1$ are replaced by $T^{[y]}, \tilde{T}^{[y]}$. Use comprehension to define the arrays $T', \tilde{T}'$ by

$$T'(t,v) \leftrightarrow \begin{cases} T(t,v) & \text{if } t \le y \\ S(v) & \text{if } t > y \end{cases}$$

and

$$\tilde{T}'(t,v) \leftrightarrow \begin{cases} \tilde{T}(t,v) & \text{if } t \le y \\ \tilde{S}(v) & \text{if } t > y. \end{cases}$$

It follows from $\phi(y)$ and $(i)'$ that $T', \tilde{T}'$ satisfy the existential quantifiers in $\phi(y+1)$.

To prove (ii) we first claim that V_1-Horn proves

$$\text{TDEF} \wedge \text{NEG} \rightarrow \forall t \le a \forall v < a (T(t,v) \leftrightarrow \neg \tilde{T}(t,v)) \tag{4.7}$$

V_1-Horn proves the RHS by induction on t, assuming TDEF $\wedge$ NEG. For the base case $t = 0$ this follows from $\text{INIT}(T, \tilde{T})$. The induction step $t \rightarrow t+1$ follows from $(ii)'$ above with $T^{[t+1]}, \tilde{T}^{[t+1]}$ substituted for $S, \tilde{S}$ and $T^{[t]}, \tilde{T}^{[t]}$ substituted for $T_1, \tilde{T}_1$.

Now (ii) follows from (4.7) and $(ii)'$ with this same substitution. □

Now we define $\text{SAT}(T_1)$ to assert that the truth assignment T_1 satisfies PROP^ϕ.

$$\text{SAT}(T_1) \;\equiv\; \forall x < b\, \text{CLAUSESAT}(x, T_1)$$

The next lemma asserts that if the formula PROP is satisfied at step t, then it remains satisfied for each subsequent step.

Lemma 4.2.8. *V_1-Horn proves*

$$\text{TDEF} \wedge \text{NEG} \;\rightarrow\; [t \le y \le a \wedge \text{SAT}(T^{[t]}) \rightarrow \text{SAT}(T^{[y]})]$$

Proof. This follows by applying induction on y to the RHS using Lemma 4.2.7(ii). □

Let $\text{SAT}^*(b, S, \tilde{S}, \bar{W}, \bar{Q}, T_1, \tilde{T}_1)$ be the result of applying Lemma 4.2.6 to $\text{SAT}(y, T_1)$, where we have introduced the new variable y as a placeholder. Now we define HORNSAT to assert that there are arrays $T, \tilde{T}$ which satisfy TDEF and such that $R(0)$ is true iff the truth assignment T at step a satisfies PROP$^\phi$. Thus

$$\begin{aligned} &\text{HORNSAT}(a, b, \bar{Q}, R, \tilde{R}) \equiv \\ &\quad \exists T \exists \tilde{T} [\text{TDEF}(a, b, \bar{Q}, T, \tilde{T}) \wedge \exists \bar{W} \text{SAT}^*(1, R, \tilde{R}, \bar{W}, \bar{Q}, T^{[a]}, \tilde{T}^{[a]})] \end{aligned} \tag{4.8}$$

It is clear from Lemma 4.2.6 that we may assume that the only atomic subformulas involving R or $\tilde{R}$ in HORNSAT are $R(0)$ and $\tilde{R}(0)$ (by replacing $R(y)$ by $R(0)$ and $\tilde{R}(y)$ by $\tilde{R}(0)$), as required by the statement of Theorem 4.2.1.

Lemma 4.2.9. *V_1-Horn proves* $\exists R \exists \tilde{R}$ HORNSAT$(a, b, \bar{Q}, R, \tilde{R})$.

Proof. This is immediate from Lemma 4.2.7 and Lemma 4.2.6 (i) applied to SAT. □

4.2.5 Proof of Theorem 4.2.1

Part (i) asserts that V_1-Horn proves $\exists R \exists \tilde{R}$ RUN$_\phi(R, \tilde{R})$, where RUN$_\phi$ is defined in (4.3). This follows immediately from Lemma 4.2.5 (i) and Lemma 4.2.9.

The proof of (ii) requires formalizing the correctness proof of the Horn Sat algorithm. Correctness asserts that assuming $\bar{Q}$ is a proper code for a Horn formula PROP, then HORNSAT implies $R(0)$ iff PROP is satisfiable. To clarify the formal statement of correctness we write SAT(T_1) as SAT$(a, b, \bar{Q}, T_1)$ with all of its free variables indicated.

Lemma 4.2.10 (Correctness of HornSat). *V_1-Horn proves*

$$\begin{aligned} &\text{HORNSAT}(a, b, \bar{Q}, R, \tilde{R}) \wedge \text{NEG} \\ &\to (R(0) \leftrightarrow \exists T_1 \text{SAT}(a, b, \bar{Q}, T_1)) \wedge (\tilde{R}(0) \leftrightarrow \neg \exists T_1 \text{SAT}(a, b, \bar{Q}, T_1)) \end{aligned}$$

Proof. Reasoning in V_1-Horn, assume the hypotheses HORNSAT and NEG, and let $T, \tilde{T}, \bar{W}$ satisfy the existential quantifiers in the definition (4.8) of HORNSAT. By Lemma 4.2.6 (ii) applied to SAT$(y, a, b, \bar{Q}, T_1)$ (where we have added the new variable y as a placeholder) with R for S and $T^{[a]}$ for T_1 we have

$$\begin{aligned} (ii)'' \quad &\text{SAT}^*(1, a, b, R, \tilde{R}, \bar{W}, \bar{Q}, T^{[a]}, \tilde{T}^{[a]}) \wedge \text{NEG} \wedge \forall z < a (T^{[a]}(z) \leftrightarrow \neg \tilde{T}^{[a]}(z)) \\ &\to (R(0) \leftrightarrow \text{SAT}(T^{[a]})) \wedge (\tilde{R}(0) \leftrightarrow \neg \text{SAT}(T^{[a]})) \end{aligned}$$

By (4.7), (4.8) and the hypotheses to the Correctness Lemma we conclude the hypotheses to $(ii)''$ and hence we conclude

$$(R(0) \leftrightarrow \text{SAT}(T^{[a]})) \wedge (\tilde{R}(0) \leftrightarrow \neg \text{SAT}(T^{[a]})) \tag{4.9}$$

From this we conclude $R(0) \to \exists T_1 \textsc{Sat}(T_1)$ thus establishing one direction each in the two equivalences on the RHS of the Correctness Lemma (since $(ii)'' \to (R(0) \leftrightarrow \neg\tilde{R}(0))$.

Showing the other direction amounts to showing that under our hypotheses, $\exists T_1 \textsc{Sat}(T_1) \to \textsc{Sat}(T^{[a]})$. In other words, we must show that if PROP is satisfiable, then it is satisfied by the final truth assignment given by the the Horn Sat algorithm. Formally it suffices to show that V_1-Horn proves

$$\text{TDEF} \wedge \text{NEG} \wedge \textsc{Sat}(T_1) \;\to\; \textsc{Sat}(T^{[a]}) \tag{4.10}$$

First we show that $T^{[a]}$ is contained in every truth assignment satisfying PROP.

Lemma 4.2.11. *V_1-Horn proves*

$$\text{TDEF} \wedge \text{NEG} \wedge \textsc{Sat}(T_1) \;\to\; \forall t < a \forall v < a(T(t,v) \to T_1(v))$$

Proof. The RHS is proved by induction on t. The base case $t = 0$ is vacuous because the condition $\textsc{Init}(T, \tilde{T})$ in the definition (4.6) of TDEF implies $T^{[0]}$ is identically false.

For the induction step we apply Lemma 4.2.7 (ii) and the definition (4.4) of $\textsc{Step}(v, T^{[t]})$. Thus the only way that $T(t+1, v)$ can hold but not $T(t, v)$ is if some clause cl_x is not satisfied by $T^{[t]}$ and contains a positive literal $P(v)$. (Recall that $cl_0, ..., cl_{b-1}$ are the clauses in PROP^{ϕ}, as explained in the paragraphs following equation (4.2).) But by the induction hypothesis and our assumption that T_1 satisfies cl_x we have $\neg\textsc{ClauseSat}(x, t^{[t]}) \to T_1(v)$. □

If $\textsc{Sat}(T_1)$ holds, but $\neg\textsc{Sat}(T^{[a]})$, then there is a clause cl_x such that $\textsc{ClauseSat}(x, T_1)$ but $\neg\textsc{ClauseSat}(x, T^{[a]})$. Hence by the above lemma cl_x contains a positive literal $P(v)$ such that $\neg T(a, v)$. Thus V_1-Horn proves

$$\text{TDEF} \wedge \text{NEG} \wedge \textsc{Sat}(T_1) \wedge \neg\textsc{Sat}(T^{[a]}) \;\to\; \exists v < a \neg T(a, v) \tag{4.11}$$

There are only a atoms $P(0), ..., P(a-1)$ to be set, and as long as at least one clause is not satisfied every step sets at least one atom. It follows that after a steps $T^{[a]}$ must be identically true, contradicting (4.11).

To formalize the last part of the argument we introduce in the next subsection a counting formula $\textsc{NumOnes}(a, y, X)$, which asserts that the number of true values among $X(0), ..., X(a-1)$ is at least y. Using results in that subsection we now claim that V_1-Horn proves

$$\text{TDEF} \wedge \text{NEG} \wedge \neg\textsc{Sat}(T^{[a]}) \wedge \textsc{Sat}(T_1) \;\to\; \textsc{NumOnes}(a, t, T^{[t]}) \tag{4.12}$$

This follows by applying induction on t to the RHS, using Lemma 4.2.13 (i) for the basis $t = 0$. For the induction step $t \to t+1$ we use Lemma 4.2.14 with $T^{[t]}$ for X, $T^{[t+1]}$ for Y, and t for y, and Lemma 4.2.7 (ii). The existence of v such that $\neg T(t, v) \wedge T(t+1, v)$ follows from our assumptions $\neg \textsc{Sat}(T^{[a]})$ (and hence $\neg \textsc{Sat}(T^{[t]})$ by Lemma 4.2.8) and $\textsc{Sat}(T_1)$ using Lemmas 4.2.7 (ii) and 4.2.11.

Finally (4.10) follows from (4.12) (with $t = a$) together with Lemma 4.2.13 (ii) and (4.11). This completes the proof of Lemma 4.2.10. $\square$

We can now complete the proof of Theorem 4.2.1 (ii). By the definition (4.3) of $\textsc{Run}_\phi$ and Lemma 4.2.10 if suffices to show that V_1-Horn proves the following two formulas.

$$\textsc{Prop}_\phi(\bar{Q}) \to \textsc{Neg}(\hat{a}, \hat{b}, \bar{Q}) \tag{4.13}$$

$$\textsc{Prop}_\phi(\bar{Q}) \ \to \ [\phi \leftrightarrow \exists T_1(\textsc{Sat}(\hat{a}, \hat{b}, \bar{Q}, T_1))] \tag{4.14}$$

That (4.13) is provable follows from the definition (4.5) of $\textsc{Neg}$ and Lemma 4.2.5 (ii).

To show (4.14) is provable we refer to the definition (4.2) of ϕ and show that V_1-Horn proves

$$\textsc{Prop}_\phi(\bar{Q}) \ \to \ \forall x_1 \leq t_1 ... \forall x_k \leq t_k[\psi(\bar{x}, P) \leftrightarrow \textsc{Sat}(\hat{a}, \hat{b}, \bar{Q}, P)] \tag{4.15}$$

Recall (see the proof of Lemma 4.2.5) that $\psi(\bar{x}, P)$ is the conjunction of the clauses $CL_0, ..., CL_{m-1}$. By Lemma 4.2.5 (ii) and the definitions of Ψ_C, Ψ_D, Ψ_V, V_1-Horn proves for $j = 0, ..., m-1$

$$\textsc{Prop}_\phi(\bar{Q}) \ \to \ \forall \bar{x} \leq \bar{t}[CL_j(\bar{x}, P) \leftrightarrow \textsc{ClauseSat}(\langle \bar{x}, j \rangle, P)]$$

This establishes the right-to-left direction of the equivalence in (4.15). To establish the other direction we also need the fact that V_1-Horn proves (assuming $\textsc{Prop}_\phi(\bar{Q})$) that if x is not of the form $\langle x_1, ..., x_k, j \rangle$ then $\Psi_V(x)$ and hence $V(x)$ and hence $\textsc{ClauseSat}(x, P)$.

4.2.6 Counting in V_1-Horn

The results in this subsection are needed to complete the proof of Lemma 4.2.10 (Correctness of $\textsc{HornSat}$).

We define a Σ_1^B-Horn formula $\textsc{NumOnes}(a, y, X)$ which asserts that the number of true values among $X(0), ..., X(a-1)$ is at least y. First we define a formula $\textsc{Count}(a, M, \tilde{M}, X)$ which is Π_1^b-Horn with respect to $M, \tilde{M}$ and which defines complementary arrays $M, \tilde{M}$ so that for $t, y \leq a$, $M(t, y)$ holds iff the number of true values among $X(0), ..., X(t-1)$

is at least y. We give recurrence equations in the style of the definition of PARITY(X) on page 22.

$$\begin{aligned}\textsc{Count}(a, M, \tilde{M}, X) \equiv\ & \forall t \leq a \forall y \leq a \\ & M(t,0) \wedge \neg\tilde{M}(t,0) \wedge \neg M(0,y+1) \wedge \tilde{M}(0,y+1) \\ & \wedge(\neg M(t,y+1) \vee \neg\tilde{M}(t,y+1)) \\ & \wedge(M(t,y) \wedge X(t) \rightarrow M(t+1,y+1)) \\ & \wedge(M(t,y+1) \rightarrow M(t+1,y+1)) \\ & \wedge(\tilde{M}(t,y) \rightarrow \tilde{M}(t+1,y+1)) \\ & \wedge(\tilde{M}(t,y+1) \wedge \neg X(t) \rightarrow \tilde{M}(t+1,y+1)).\end{aligned}$$

Lemma 4.2.12. *V_1-Horn proves*

$$(i)\ \exists M \exists \tilde{M}\, \textsc{Count}(a, M, \tilde{M}, X)$$
$$(ii) \textsc{Count}(a, M, \tilde{M}, X) \rightarrow [t \leq a \rightarrow \forall y \leq a(M(t,y) \leftrightarrow \neg\tilde{M}(t,y))]$$

Proof. Since (i) is a Σ_1^B-Horn formula we may use induction on a. When $a = 0$ we use comprehension to explicitly define M such that $M(0,0)$, $M(1,0)$, $\neg M(0,1)$, and $(M(1,1) \leftrightarrow X(0))$, and similarly for $\tilde{M}$. For the induction step $a \rightarrow a+1$ we use comprehension to define the new values of $M, \tilde{M}$ using the recursion equations and the old values given by the induction hypothesis, in the style of the proof of Lemma 4.2.7 (i).

The proof of (ii) uses the induction scheme applied to $\phi(t)$, where $\phi(t)$ is the RHS. □

The result of Lemma 4.2.12 allows us to use $\neg M$ and $\tilde{M}$ interchangeably, and we shall do this freely in what follows.

Now we give the definition

$$\textsc{NumOnes}(a, y, X) \equiv \exists M \exists \tilde{M}[\textsc{Count}(a, M, \tilde{M}, X) \wedge M(a, y)]$$

Lemma 4.2.13. *V_1-Horn proves the following:*

$$(i)\ \ \textsc{NumOnes}(a, 0, X)$$
$$(ii)\ \textsc{NumOnes}(a, a, X) \rightarrow \forall v < a X(v)$$

Proof. (i) follows immediately from the definitions of NUMONES and COUNT.

To prove (ii) we first show that V_1-Horn proves

$$\textsc{Count}(a, M, \tilde{M}, X) \rightarrow \forall y < a(t < y \rightarrow \neg M(t,y)) \tag{4.16}$$

This follows by induction on t applied to the RHS, using the definition of COUNT.

Next we show that V_1-Horn proves

$$\text{COUNT}(a, M, \tilde{M}, X) \wedge \neg X(v) \rightarrow [v < t \leq a \rightarrow \neg M(t,t)] \tag{4.17}$$

This also follows by induction on t applied to the RHS, using (4.16).

Now (ii) follows from (4.17) by setting $t = a$. □

We introduce the abbreviation

$$X \subseteq_a Y \;\equiv\; \forall y < a(X(y) \rightarrow Y(y))$$

Lemma 4.2.14. *V_1-Horn proves*

$$X \subseteq_a Y \wedge v < a \wedge \neg X(v) \wedge Y(v) \wedge y < a \rightarrow$$
$$[\text{NUMONES}(a, y, X) \rightarrow \text{NUMONES}(a, y+1, Y)]$$

Proof. First we claim that V_1-Horn proves each of the following formulas using induction on t; the second uses the first.

$$X \subseteq_a Y \wedge \text{COUNT}(a, M, \tilde{M}, X) \wedge \text{COUNT}(a, M', \tilde{M}', Y)$$
$$\rightarrow \forall y < a(t \leq a \wedge M(t,y) \rightarrow M'(t,y))$$

$$X \subseteq_a Y \wedge \neg X(v) \wedge Y(v) \wedge \text{COUNT}(a, M, \tilde{M}, X) \wedge \text{COUNT}(a, M', \tilde{M}', Y)$$
$$\rightarrow \; \forall y < a(v < t \leq a \wedge M(t,y) \rightarrow M'(t, y+1))$$

Now the lemma follows from Lemma 4.2.12 and the formula immediately above with $t = a$. □

4.3 Explicit definability theorem for V_1-Horn

Now we are ready to apply the theorem 3.3.13 to V_1-Horn.

Lemma 4.3.1. *The class of formulae Σ_1^B-Horn is strongly closed and constructive.*

Proof. Corollary 4.2.3 shows that Σ_1^B-Horn is closed under Σ_0^B operations. Lemma 4.5.5 shows that the class of functions defined by Σ_1^B-Horn formulae is closed under composition and substitution of a term for a variable. Together, these two facts give a closure under $\mathtt{AC}^0$ reductions, so Property 1 is strongly satisfied by Σ_1^B-Horn.

For the Property 2, we use RUN to define witnesses to the quantified variables in Σ_1^B-Horn formulae. Recall that because of the pairing function, it is sufficient to consider only formulae with one variable. Suppose that the formula is $\phi(\bar{a}, \bar{Y}) \equiv \exists P \forall \bar{x} < \bar{t}\psi(\bar{x}, P, \bar{a}, \bar{Y})$, and let V_1-Horn $\vdash \phi(\bar{a}, \bar{Y})$. Define $\forall i < a$, where a is an upper bound on terms occurring as $P(t)$,

$$P(i) \leftrightarrow \exists R \exists \tilde{R} \text{RUN}_\phi(\bar{a}, \bar{Y}) \wedge T^{[a]}(i).$$

If the formula is satisfiable, the variable T is the satisfying assignment from equation 4.6. By construction, an assignment satisfying TDEF is unique, because the algorithm encoded by TDEF is deterministic. Therefore, $P(i)$ is uniquely defined by this formula. Thus, Σ_1^B-Horn satisfies the constructiveness property and the witnessing function for the existential quantifier in the base case of the witnessing theorem can be defined by an axiom

$$F(\bar{a}, \bar{Y})(i) \leftrightarrow \exists R \exists \tilde{R} \text{RUN}_\phi(\bar{a}, \bar{Y}) \wedge R(0) \wedge T^{[a]}(i).$$

□

Theorem 4.3.2. *The class of Σ_1^B-definable string and number functions of V_1-Horn coincides with the class of polynomial-time functions (FP).*

Proof. Since V_1-Horn satisfies both properties by lemma 4.3.1, theorem 3.3.13 applies. Therefore, the class of Σ_1^B-definable functions of V_1-Horn is P. □

4.4 Finite Axiomatizability

Here we show V_1-Horn is finitely axiomatizable, and that the $\forall\Sigma_1^B$ consequences of V_1-Horn and the $\forall\Sigma_1^b$ consequences of S_2^1 are each finitely axiomatizable.

Since V^0 defines the uniform $\mathtt{AC}^0$ functions, it seems plausible that V_1-Horn could be axiomatized by V^0 together with a formula expressing the comprehension axiom for some predicate which is complete for P under uniform $\mathtt{AC}^0$ reductions. Hence the finite axiomatizability of V_1-Horn should follow from that for V^0, which we already showed in Theorem 3.6.1. In our proof of Theorem 4.4.1 below, that predicate is the Horn satisfiability problem, which is complete for P [GHR95]. [1]

[1] Another approach would be to define a least fixed point function (based on the operator from definition 2.2.2) and use this function in the single instance of the comprehension axiom. In [Grä92],

Theorem 4.4.1. *V_1-Horn is finitely axiomatizable.*

Proof. It suffices to show that Corollary 4.2.4 (i) and (ii) can be proved for any Σ_1^B-Horn formula $\phi(y)$ using finitely many theorems of V_1-Horn as axioms. We first will show how to do this for Theorem 4.2.1 (i) and (ii), and then explain how to modify the proof to get the corollary.

First note that for each Σ_1^B-Horn formula ϕ we can define a version of PROP_ϕ such that (i) and (ii) in Lemma 4.2.5 are theorems of V^0. Thus we include the finite set of axioms for V^0 from Theorem 3.6.1 among the finite axioms for V_1-Horn. The proof of Theorem 4.2.1 depends on Lemma 4.2.5 (which we have established) and some properties of HORNSAT. Since HORNSAT is independent of ϕ, we can take these properties as axioms.

To generalize the proof of Theorem 4.2.1 in order to prove Corollary 4.2.4, we incorporate the variable y in $\phi(y)$ as an argument of each of the arrays $C, D, V, \tilde{C}, \tilde{D}, \tilde{V}$ to define the formula $\text{PROP}_\phi(y)$ in a modified Lemma 4.2.5. Then y is not free in $\text{PROP}_\phi(y)$ (although it could be free in PROP_ϕ). The definition (4.8) of HORNSAT is modified so that the parameter y is incorporated as an argument of each of the arrays $R, \tilde{R}, T, \tilde{T}$. Then Corollary 4.2.4 follows in the same way as Theorem 4.2.1. □

Theorem 4.4.2. *V_1-Horn is axiomatized by its $\forall\Sigma_1^B$ consequences.*

Proof. It suffices to show that each Σ_1^B-Horn comprehension axiom is a consequence of $\forall\Sigma_1^B$ theorems of V_1-Horn. First we show that the second-order quantifiers in Σ_1^B-Horn formulas (3.1) can be bounded. That is, for each Σ_1^B-Horn formula ϕ there is a Σ_1^B formula ϕ^B such that $\forall\Sigma_1^B V_1\text{-Horn} \vdash (\phi \leftrightarrow \phi^B)$. To construct ϕ^B replace each second-order quantifier $\exists P$ in ϕ by a bounded quantifier $\exists P \leq t$, where t is a provable upper bound on all terms u such that $P(u)$ occurs in ϕ. The equivalence of ϕ and ϕ^B requires only ψ-COMP instances for formulas ψ with no second-order quantifiers, and these instances are $\forall\Sigma_1^B$ formulas.

The comprehension axiom for $\phi(z)$ follows from Corollary 4.2.4 (i) and (ii). The Σ_1^B form of (i) we need is

$$\exists R \leq y \exists \tilde{R} \leq y \ \text{RUN}'_{\phi(z)}(y, R, \tilde{R})$$

Grädel mentions that every formula in FO+LFP is equivalent to a formula in normal form $\phi = LFP_{P,\bar{x}}(\exists\bar{y})\psi(P,\bar{x},\bar{y})[\bar{0}]$, where ψ is quantifier-free. From there, he defines a Horn formula encoding the complement of ϕ by $\exists P\forall\bar{x}\forall\bar{y}(\bigwedge_j(\psi_j(P,\bar{x},\bar{y}) \to P(\bar{x})) \wedge \neg P(\bar{0}))$, where ψ_j are terms of the disjunctive normal form of ψ. We use a similar idea in the definition of the system of arithmetic for NL (see chapter 5), however for V_1-Horn we use the algorithm for satisfiability directly.

where $\text{RUN}'_{\phi(z)}$ has suitable bounds on its second-order quantifiers. For (ii) we do not need the clause involving $\tilde{R}$. If we replace ϕ by ϕ^B then a suitable prenex form of the result is $\forall\Sigma_1^B$. □

Corollary 4.4.3. *The $\forall\Sigma_1^B$ consequences of V_1-Horn are finitely axiomatizable.*

Proof. It follows by compactness from Theorems 4.4.2 and 4.4.1. □

4.5 Equivalence of V_1-Horn, P-def and QPV

The first-order theory QPV (called PV1 in [Kra95]) has function symbols for all polynomial-time computable functions, and the axioms include defining equations for these functions (based on Cobham's Theorem) and induction on the length of numbers. The theory has been extensively studied [Coo75, Bus86, CU93, Kra95, Coo98] and shown to robustly capture the notion of "polynomial-time reasoning". Zambella's [Zam96] theory P-def is a second-order version of QPV, and can shown to be equivalent to QPV by the method of RSUV isomorphism (see [Kra95]). Here we show that V_1-Horn is equivalent in power to P-def. This implies that V_1-Horn is equivalent in power to QPV, but is most likely not as powerful as S_2^1 (see Section 1).

4.5.1 Adding function symbols to V_1-Horn

We would like to show that the class of functions Σ_1^B-definable in V_1-Horn coincides with the class of functions Σ_1^B-definable in P-def. However, function symbols in P-def are introduced differently from function symbols in V_1-Horn: the system P-def is essentially an extension of V^0 by all polynomial-time function symbols introduced using a recursion-theoretic characterization, whereas in V_1-Horn function symbols are introduced conservatively by setting their bitgraph (graphs in case of number functions) to be Σ_1^B-Horn formulae.

The class P consists of all relations of the form $R(x_1, ..., x_k, Y_1, ..., Y_m)$ recognizable in time bounded by a polynomial in $(x_1, ..., x_k, |Y_1|, ..., |Y_m|)$. Similarly, the function class FP consists of all functions $F(x_1, ..., x_k, Y_1, ..., Y_m)$ computable in time bounded by a polynomial in $(x_1, ..., x_k, |Y_1|, ..., |Y_m|)$. As described in the section 3.3, if a class is closed under AC^0 reductions then function symbols can be introduced into a corresponding V-Φ by defining them by bitgraphs for string functions and graphs for number functions (see definition 3.3.2. Thus, we introduce function symbols into V_1-Horn by setting

$$F(\bar{x},\bar{Y})(i) \leftrightarrow i < t \wedge B_F(i,\bar{x},\bar{Y}) \qquad f(\bar{x},\bar{Y}) = z \leftrightarrow B_f(z,\bar{x},\bar{Y}),$$

where B_F and B_f are Σ_1^B-Horn.

The following characterization of FP is straightforward.

Lemma 4.5.1. *(i) A string function $F(\bar{x},\bar{Y})$ is in* FP *iff $|F(\bar{x},\bar{Y})|$ is bounded by a polynomial in $(\bar{x},|\bar{Y}|)$ and its bit graph B_F is in* P.
(ii) A number function $f(\bar{x},\bar{Y})$ is in FP *iff $f(\bar{x},\bar{Y}) = |F(\bar{x},\bar{Y})|$ for some string function F in* FP.

We now define a conservative extension V_1-Horn(FP) of V_1-Horn by introducing function symbols for polynomial time functions with defining equations based on the above lemma. By the corollary 4.2.3 to the theorem 4.2.1, the functions in FP are closed under Σ_0^B reductions. Therefore, they are closed under $\mathtt{AC}^0$ reductions (see discussion after definition 3.3.7).

Definition 4.5.2 (Specification of V_1-Horn(FP)). The language $\mathcal{L}_A^2(\mathtt{FP})$ is the language $\mathcal{L}_A^2$ of V_1-Horn extended by new function symbols. We define function symbols, terms, formulas, and Σ_1^B-Horn formulas for V_1-Horn(FP) by simultaneous recursion as follows. In general $\bar{x} = x_1, ..., x_k$ and $\bar{Y} = Y_1, ..., Y_m$.

(i) To every first-order term $\ell(\bar{x},\bar{Y})$ and Σ_1^B-Horn formula $\phi(i,\bar{x},\bar{Y})$ we associate an arity $\langle k,m\rangle$ string function symbol F with defining formulas (renaming ℓ as ℓ_F and ϕ as ϕ_F)

$$|F(\bar{x},\bar{Y})| \leq \ell_F(\bar{x},\bar{Y}) \tag{4.18}$$

$$\forall i < \ell(\bar{x},\bar{Y})[F(\bar{x},\bar{Y})(i) \leftrightarrow \phi_F(i,\bar{x},\bar{Y})] \tag{4.19}$$

To every arity $\langle k,m\rangle$ string function symbol F we associate an arity $\langle k,m\rangle$ number function symbol f with defining formula

$$f(\bar{x},\bar{Y}) = |F(\bar{x},\bar{Y})| \tag{4.20}$$

(ii) First-order variables and 0 and 1 are first-order terms and second-order variables are second-order terms.

(iii) If t_1, t_2 are first-order terms then $t_1 + t_2$ and $t_1 \cdot t_2$ are first-order terms. If T is a second-order term then $|T|$ is a first-order term.

(iv) If $t_1, ..., t_k$ are first-order terms and $T_1, ..., T_m$ are second-order terms, and f and F are arity $\langle k, m\rangle$ number and string function symbols, respectively, then $f(\bar{t}, \bar{T})$ is a first-order term and $F(\bar{t}, \bar{T})$ is a second-order term.

(v) If s, t are first-order terms and T is a second-order term then $s = t, s \leq t$ and $T(t)$ are atomic formulas. Formulas are built from atomic formulas as in V_1-Horn using $\wedge, \vee, \neg$ and the first and second-order quantifiers.

(vi) Σ_1^B-Horn formulas are defined as in Definition 3.1.4, with *term* and *formula* understood in the present context, and with the restriction that no term may include any quantified second-order variable P_i as a proper subpart. (This generalizes the restriction that $|P_i|$ may not appear. However formulas $P_i(t)$ may appear for any term t satisfying this restriction.)

The axioms of V_1-Horn(FP) are the same as for V_1-Horn except that the comprehension scheme is generalized to allow comprehension for all Σ_1^B-Horn formulas of V_1-Horn(FP), and the defining formulas introduced in (i) for all function symbols are included.

We refer to function symbols F and f introduced by (i) as *derived* function symbols, to distinguish them from the original function symbols $0, 1, +, \cdot, |\,|$ of V_1-Horn. In reasoning about V_1-Horn(FP) it is useful to define the *rank* of each function symbol by assigning rank 0 to the original function symbols and in general assigning 1 + the maximum of the ranks of function symbols in ℓ_F and ϕ_F to each function symbol F introduced by (i) above, and 1 + the rank of F for each function symbol f introduced by (i) above.

We claim that (a) every function symbol introduced by (i) represents a polynomial-time function, and (b) each Σ_1^B-Horn formula ϕ of V_1-Horn(FP) represents a relation in P. Claims (a) and (b) are proved simultaneously by induction on the rank of the function symbol introduced in (a), and the maximum of the ranks of the function symbols occurring in ϕ for (b). The base case follows from Theorem 3.1.5, and for the induction step (a) follows from (b) and Lemma 4.5.1. To prove (b), we observe that the proof of the if direction of Theorem 3.1.5 still goes through. In particular, given values for the free variables $\bar{z}, \bar{Y}$ and the quantified variables $\bar{x}$ in (3.2), every first-order term can be evaluated to a number and every second-order term can be evaluated to a string, because the restriction in the definition (vi) of Σ_1^B-Horn insures that no term involves quantified second-order variables P_i.

It is not hard to check that the results in the previous two sections apply to V_1-Horn(FP) as well as to V_1-Horn. This is true in particular to the main theorem on $\textsc{Run}_\phi$.

Theorem 4.5.3. *Theorem 4.2.1 on* RUN_ϕ*, and its corollaries, apply to* V_1*-Horn(*$\texttt{FP}$*). Any derived function symbol occurring in* RUN_ϕ*,* NEG_ϕ*, etc. also occurs in* ϕ*.*

Proof. The formula $\text{RUN}_\phi(R, \tilde{R})$ is constructed from the two formulas PROP_ϕ and HORNSAT. The formula HORNSAT describes the propositional Horn satisfiability algorithm, is independent of ϕ, and is the same in the present context. The formula PROP_ϕ describes the propositional version of ϕ. This does depend on ϕ but it is constructed in the present context exactly as before. □

The following lemma is needed for the proof of the theorem below.

Lemma 4.5.4 (Term Bounding). *(Here all variables are fully indicated.) For each first-order term* $t(\bar{x}, \bar{Y})$ *of* V_1*-Horn(*$\texttt{FP}$*) there is a first-order bounding term* $\ell_t(\bar{x}, \bar{y})$ *of* V_1*-Horn such that*

$$V_1\textit{-Horn}(\texttt{FP}) \ \vdash \ t(\bar{x}, \bar{Y}) \leq \ell_t(\bar{x}, |\bar{Y}|)$$

For each second-order term $T(\bar{x}, \bar{Y})$ *there is a first-order bounding term* $\ell_T(\bar{x}, \bar{y})$ *of* V_1*-Horn such that*

$$V_1\textit{-Horn}(\texttt{FP}) \ \vdash \ |T(\bar{x}, \bar{Y})| \leq \ell_T(\bar{x}, |\bar{Y}|)$$

Proof. The two assertions are proved simultaneously by double induction, first on the highest rank of any function symbol occurring in t or T, and second on the maximum nesting depth of derived function symbols in t and T. □

The next lemma is the key lemma for the proof of the main theorem on this section, theorem 4.5.6.

Lemma 4.5.5. *For every* Σ_1^B*-Horn formula* $\phi'(\bar{x}, \bar{Y})$ *of* V_1*-Horn(*$\texttt{FP}$*) there is a* Σ_1^B*-Horn formula* ϕ *of* V_1*-Horn such that*

$$V_1\textit{-Horn}(\texttt{FP}) \ \vdash \ \phi'(\bar{x}, \bar{Y}) \leftrightarrow \phi(\bar{x}, \bar{Y})$$

Proof. The proof that each such ϕ' can be converted to an appropriate ϕ is carried out by triple induction, first on the highest rank r of any function symbol occurring in ϕ', second on the maximum nesting depth d of derived functions in any term in ϕ' containing a function symbol of rank r, and third on the number of such maximal terms occurring in ϕ'. The base case, $r = 0$, is trivial since we may take $\phi = \phi'$. Now suppose $r > 0$ and let

$$\phi'(\bar{x}, \bar{Y}) \ \equiv \ \exists P_1 ... \exists P_a \forall z_1 < t_1 ... \forall z_b < t_b \psi'(\bar{z}, \bar{P}, \bar{x}, \bar{Y}) \tag{4.21}$$

where ψ' is a quantifier-free Horn formula satisfying the conditions in Definition 3.1.4. We may suppose that none of the quantifier bounding terms t_i contains a function symbol not in V_1-Horn since by the Term Bounding Lemma 4.5.4 we can replace $\forall x_i < t_i$ by its $\forall x_i < \ell_{t_i}$ and add the clause $x_i < t_i$ as a conjunct to ψ'.

We may replace each occurrence $f(...)$ of a first-order derived function symbol f by its definition $|F(...)|$ without increasing the rank or nesting depth of derived function symbols. Therefore we may assume that no first-order derived function symbol occurs in ϕ'.

Let r be the maximum rank of any function symbol occurring in ϕ', let d be the maximum nesting depth of derived function symbols in terms of rank r, and let T be a second-order term in ϕ' containing a function symbol of rank r and let T have derived nesting depth d. Then T has the form $F(\bar{s}, \bar{S})$ where F is a second-order function symbol, $\bar{s}$ are first-order terms and $\bar{S}$ are second-order terms. There are two cases, depending on how T occurs in ϕ':

Case I: T occurs in a term $|F(\bar{s}, \bar{S})|$.

Case II: T occurs in an atomic formula $F(\bar{s}, \bar{S})(t)$.

For Case I, suppose that $F(\bar{y}, \bar{Z})$ is defined from $\ell_F(\bar{y}, \bar{Z})$ and $\phi_F(\bar{y}, \bar{Z})$ in (i) of Definition 4.5.2. Then according to the axioms of V_1-Horn, $|F(\bar{y}, \bar{Z})|$ is 1 + the largest $j < \ell_F(\bar{y}, \bar{Z})$ such that $\phi_F(j, \bar{y}, \bar{Z})$, or 0 if no such j exists. Therefore

$$V_1\text{-Horn}(\mathtt{FP}) \vdash [i = |F(\bar{y}, \bar{Z})| \ \leftrightarrow \ \Psi(i, \bar{y}, \bar{Z})] \tag{4.22}$$

where $\psi(i, \bar{y}, \bar{Z})$ is the formula

$$\begin{aligned} &i = 0 \wedge \forall j < \ell_F(\bar{y}, \bar{Z}) \neg \phi_F(j, \bar{y}, \bar{Z})) \vee \\ &\quad \exists i' < i[i = i' + 1 \wedge \phi_F(i', \bar{y}, \bar{Z}) \wedge \forall j < \ell_F(\bar{y}, \bar{Z})(i \leq j \supset \neg \phi_F(j, \bar{y}, \bar{Z}))]. \end{aligned}$$

Notice that by definition of *rank*, any function symbol occurring in ϕ_F or ℓ_F has smaller rank than that of F, and therefore rank less than r. Therefore by Corollary 4.2.3 and Theorem 4.5.3, ψ is provably equivalent to a Σ_1^B-Horn formula all of whose derived function symbols have rank less than r, and hence by the induction hypothesis provably equivalent to a Σ_1^B-Horn formula of V_1-Horn. Thus we may assume that $\psi(i, \bar{y}, \bar{Z})$ is a Σ_1^B-Horn formula with no derived function symbol.

Define $\psi'(i, \bar{x}, \bar{z}, \bar{Y}) \equiv \psi(i, \bar{s}, \bar{S})$ where we have indicated all possible free variables of ψ'. Then by (4.22)

$$V_1\text{-Horn}(\texttt{FP}) \ \vdash \ [i = |F(\bar{s}, \bar{S})| \leftrightarrow \psi'(i, \bar{x}, \bar{z}, \bar{Y})] \tag{4.23}$$

The derived nesting depth of terms in $\bar{s}, \bar{S}$ is less than that of $F(\bar{s}, \bar{S})$, and hence by the induction hypothesis we may assume that $\psi'(i, \bar{x}, \bar{z}, \bar{Y})$ is a Σ_1^B-Horn formula with no derived function symbol.

We now apply Corollary 4.2.4 to $\psi'(i, \bar{z})$ (that is, we do not change ψ', only indicate the variables $i, \bar{z}$) to obtain a Σ_1^B-Horn formula $\text{RUN}_{\psi'(i,\bar{z})}(b, \bar{c}, R, \tilde{R}, \bar{x}, \bar{Y})$ satisfying the Corollary. Here b is a bounding variable for i, $\bar{c}$ are bounding variables for $\bar{z}$, and we have indicated the free variables $\bar{x}, \bar{Y}$ which $\text{RUN}_{\psi'(i,\bar{z})}$ inherits from ψ'.

Referring to (4.21), let ψ_i' be ψ' with each occurrence of $|F(\bar{s}, \bar{S})|$ replaced by the variable i. Then by Corollary 4.2.4 and (4.23), noting that $\text{RUN}_{\psi'(i,\bar{z})}$ does not contain any of $i, \bar{z}, \bar{P}$ free,

$$V_1\text{-Horn}(\texttt{FP}) \ \vdash \ [\phi'(\bar{x}, \bar{Y}) \leftrightarrow \phi''(\bar{x}, \bar{Y})]$$

where $\phi''(\bar{x}, \bar{Y})$ is the formula

$$\begin{aligned}&\exists R \exists \tilde{R} \exists \bar{P} \forall \bar{z} < \bar{t} \forall i < \ell_F(\bar{s}, \bar{S}) \\ &\qquad [\text{RUN}_{\psi'(i,\bar{z})}(\ell_F(\bar{s}, \bar{S}), \bar{t}, R, \tilde{R}, \bar{x}, \bar{Y}) \ \wedge \ (\neg R(i, \bar{z}) \vee \psi_i'(\bar{z}, \bar{P}, \bar{x}, \bar{Y})]\end{aligned}$$

Note that ϕ'' can be converted to an equivalent Σ_1^B-Horn formula by first putting it into a suitable prenex form and then putting a copy of the literal $\neg R(i, \bar{z})$ inside every clause of ψ_i' to make the disjunction into a Horn formula. The resulting Σ_1^B-Horn formula has one fewer occurrence of a term of derived depth d containing a function symbol of rank r (since T was removed from ψ' in forming ψ_i' and $\text{RUN}_{\psi'(i,\bar{z})}$ has no derived function symbol). Hence by the induction hypothesis, ϕ'' is provably equivalent to a Σ_1^B-Horn formula with no derived function symbol.

The proof for Case II is similar, but easier. By reasoning as before, we can find a Σ_1^B-Horn formula $\psi'(i, \bar{x}, \bar{z}, \bar{Y})$ with no derived function symbol such that (analogously to (4.23))

$$V_1\text{-Horn}(\texttt{FP}) \ \vdash \ [F(\bar{s}, \bar{S})(i) \leftrightarrow \psi'(i, \bar{x}, \bar{z}, \bar{Y})] \tag{4.24}$$

Again we apply Corollary 4.2.4 to $\psi'(i, \bar{z})$ to obtain a Σ_1^B-Horn formula $\text{RUN}_{\psi'(i,\bar{z})}(b, \bar{c}, R, \tilde{R}, \bar{x}, \bar{Y})$ satisfying the corollary. Again referring to (4.21), let ψ_R' be

ψ' with each positive occurrence of $F(\bar{s},\bar{S})(t)$ replaced by $\neg\tilde{R}(t,\bar{z})$ and each occurrence of $\neg F(\bar{s},\bar{S})(t)$ replaced by $\neg R(t,\bar{z})$. (In this way ψ'_R is Horn with respect to $R,\tilde{R}$ in Definition 3.1.4.) Then by Corollary 4.2.4 and (4.24),

$$V_1\text{-Horn}(\mathtt{FP}) \vdash [\phi'(\bar{x},\bar{Y}) \leftrightarrow \phi''(\bar{x},\bar{Y})]$$

where now $\phi''(\bar{x},\bar{Y})$ is the formula

$$\exists R\exists\tilde{R}\exists\bar{P}\forall\bar{z} < \bar{t}[\textsc{Run}_{\psi'(i,\bar{z})}(\ell_F(\bar{s},\bar{S}),\bar{t},R,\tilde{R},\bar{x},\bar{Y}) \ \wedge\ \psi'_R(\bar{z},\bar{P},\bar{x},\bar{Y})]$$

Again ϕ'' can be converted to an equivalent Σ_1^B-Horn formula by putting it into a suitable prenex form, and hence by the induction hypothesis ϕ'' is provably equivalent to a Σ_1^B-Horn formula with no derived function symbol. □

Using the lemma 4.5.5, we obtain the following result:

Theorem 4.5.6 (Conservativity). *Every theorem of V_1-Horn(*FP*) in the language of V_1-Horn is a theorem of V_1-Horn.*

Proof. It suffices to show that every model M of V_1-Horn has an expansion M' to the language $\mathcal{L}_A^2(\mathtt{FP})$ which is a model of V_1-Horn(FP). To define M' it suffices to specify functions on the universes of M interpreting each function symbol F and f introduced in Definition 4.5.2 (i), in such a way that the defining formulas are satisfied. First note that the value of each first-order function f is uniquely specified by (4.20) as a first-order element of M (assuming that F has been specified). Next note that for each tuple of values for the arguments of F, (4.18,4.19) uniquely specify the value of $F(\bar{x},\bar{Y})$ as a set of first-order elements of M. Further by the theorem, the formula ϕ_F specifying the bit graph of F is equivalent to a Σ_1^B-Horn formula of V_1-Horn, and therefore by Σ_1^B-Horn comprehension this set of elements is realized in M as a second-order object. Finally the comprehension axioms for all Σ_1^B-Horn formulas of V_1-Horn(FP) are satisfied by M', by the theorem. □

4.5.2 Specification of P-def

We present a version of Zambella's [Zam96] P-def which fits our notation and axioms. It is the same in spirit to Zambella's system. The system P-def is obtained from a *Base Theory BT* by introducing function symbols for all functions in FP, based on Cobham's recursion-theoretic characterization of the polynomial-time computable functions.

The Base Theory BT has the language $\mathcal{L}_A^2(=)$, which is $\mathcal{L}_A^2$with second-order equality. System BT has the same terms and formulas as V_1-Horn, except that atomic formulas include equations $X = Y$ between second-order variables. The axioms of BT consist of the axioms B1,...,B13,L1,L2 of V_1-Horn, the axiom E of extensionality (below) and the comprehension scheme for Σ_0^B formulas.

$$\text{E}: \qquad X = Y \leftrightarrow [\ |X| = |Y| \wedge \forall i < |X|(X(i) \leftrightarrow Y(i))] \tag{4.25}$$

As mentioned in Section 3.1, the Σ_0^B formulas represent precisely the $\mathtt{AC}^0$ relations. Analogously to $\mathtt{FP}$, we define $\mathtt{FAC}^0$ to be those polynomially-bounded string and number functions whose bit graphs are $\mathtt{AC}^0$ relations. (The functions in $\mathtt{FAC}^0$ are termed *rudimentary* in [Zam96].) After [Zam96], we define the $\mathtt{R}$-def to be BT augmented with function symbols for functions in $\mathtt{FAC}^0$ and their defining formulas.

More precisely, the language of $\mathtt{R}$-def is $\mathcal{L}_A^2(=)$ augmented with new function symbols, which are defined by simultaneous recursion along with terms, formulas and Σ_0^B formulas, as in Definition 4.5.2 with the following changes. In (i), Σ_1^B-Horn formula is replaced with Σ_B^0 formula. In (v), we now allow $S = T$ as an atomic formula, where S, T are second-order terms. In (vi) we replace the definition of Σ_1^B-Horn formula by that of Σ_0^B formula, which is a bounded formula in the language of $\mathtt{R}$-def with no second-order quantifier.

The axioms of $\mathtt{R}$-def are the axioms B1,...,B13,L1,L2, and E, together with comprehension over the Σ_0^B formulas of $\mathtt{R}$-def and the defining formulas for all derived function symbols.

By an easier version of the proofs of Lemma 4.5.5 and Theorem 4.5.6 we can show that $\mathtt{R}$-def is a conservative extension of the Base Theory BT.

We next name a string function symbol $\textsc{Chop}$ of $\mathtt{R}$-def of arity $\langle 1, 1\rangle$, where $\textsc{Chop}(x, Y)$ is intended to be the initial segment of Y of length at most x. The defining equations of $\textsc{Chop}$ are

$$|\textsc{Chop}(x, Y)| \leq x$$
$$\forall i < x[\textsc{Chop}(x, Y)(i) \leftrightarrow Y(i)]$$

We define $\mathtt{P}$-def to be the extension of $\mathtt{R}$-def obtained by introducing new function symbols and their defining formulas as follows:

To every first-order term $\ell_F(z, \bar{x}, \bar{Y})$ of $\mathtt{P}$-def and function symbols G_F, H_F of $\mathtt{P}$-def of arities $\langle k-1, m\rangle, \langle k, m+1\rangle$ we associate an arity $\langle k, m\rangle$ string function F with defining formulas

$$F(0,\bar{x},\bar{Y}) = \text{CHOP}(\ell_F(0,\bar{x},\bar{Y}), G(\bar{x},\bar{Y})) \tag{4.26}$$

$$F(z+1,\bar{x},\bar{Y}) = \text{CHOP}(\ell_F(z,\bar{x},\bar{Y}), H(z,\bar{x},\bar{Y},F(z,\bar{x},\bar{Y}))). \tag{4.27}$$

In addition we allow new function symbols to be introduced as in (4.18,4.19,4.20), where now ϕ_F is any Σ_0^B formula in the language of P-def.

The axioms for P-def are the same as for R-def, except we include the defining formulas for the new function symbols, and Σ_0^B formulas allow the new function symbols.

We remark that (4.18,4.19,4.20) allow the introduction of a function symbol for the composition of other function symbols. For example, we could take $\phi_F(i,\bar{x},\bar{Y})$ to be $G(H(\bar{x},\bar{Y}))(i)$.

4.5.3 Relating V_1-Horn and P-def

In this subsection we prove that V_1-Horn and P-def have the same power. The proof of the main theorem of this section, theorem 4.5.8, as well as the proof of Lemma 4.5.5, actually show how to translate V_1-Horn(FP) and P-def back and forth in such a way that V_1-Horn is fixed. First, we state a technical lemma.

Lemma 4.5.7. *Let ℓ be a term not involving $|Z|$ and let $\phi(Z)$ be a Σ_1^B-Horn formula. Then there is a Σ_1^B-Horn formula $\psi(Z)$ not involving $|Z|$ such that V_1-Horn proves*

$$|Z| \leq \ell \supset [\phi(Z) \leftrightarrow \psi(Z)]$$

Proof. This argument is similar to Case II in the proof of Lemma 4.5.5. We can define the relation $i = |Z|$ by a Σ_0^B formula $B(i,Z)$ not involving $|Z|$ but using the upper bound ℓ on $|Z|$, so

$$V_1\text{-Horn} \vdash |Z| \leq \ell \supset [i = |Z| \leftrightarrow B(i,Z)] \tag{4.28}$$

Using Corollary 4.1.5 (or Corollary 4.2.3 and the lemma above) we may assume that $B(i,Z)$ is Σ_1^B-Horn. Let $\phi'(Z)$ be the formula

$$\exists R \exists \tilde{R} \forall i < \ell [\text{RUN}_{B(i)}(R,\tilde{R},Z) \wedge (\neg R(i) \vee \phi_i(i,Z))]$$

where $\phi_i(i,Z)$ is obtained from $\phi(Z)$ by replacing each occurrence of $|Z|$ by i. Then by the above Lemma $\phi'(Z)$ does not contain $|Z|$, and by Corollary 4.5.3 and (4.28) V_1-Horn proves

$$|Z| \leq \ell \supset [\phi(Z) \leftrightarrow \phi'(Z)]$$

It remains to show that $\phi'(Z)$ is provably equivalent to a Σ_1^B-Horn formula $\psi(Z)$ which does not introduce an occurrence of $|Z|$. We write $\phi'(Z)$ as $\exists R \exists \tilde{R} \psi(R, \tilde{R}, Z)$ and apply Corollary 4.2.3 to $\psi(R, \tilde{R}, Z)$ to obtain a Σ_1^B-Horn formula ψ' equivalent to ψ in which no terms $|R|, |\tilde{R}|, |Z|$ are introduced and take $\psi(Z)$ to be $\exists R \exists \tilde{R} \psi'(R, \tilde{R}, Z)$. We may assume ψ is Σ_1^B-Horn by replacing any positive occurrence of R in ψ' by $\neg \tilde{R}$ and any positive occurrence of $\tilde{R}$ by $\neg R$. □

Theorem 4.5.8. P*-def is a conservative extension of* V_1*-Horn.*

Proof. First we show that every theorem of V_1-Horn is a theorem of P-def. It suffices to show that every Σ_1^B-Horn-COMP axiom is a theorem of P-def. Since P-def allows the Σ_0^B-COMP axioms, this amounts to showing that P-def proves that each Σ_1^B-Horn formula is equivalent to some Σ_0^B formula in the language of P-def. This can be done by defining function symbols in P-def for witnessing the second-order quantifiers in the Σ_1^B-Horn formula (3.1) and proving them correct. This amounts to describing the Horn satisfiability algorithm in P-def, or more precisely formalizing the proof of Theorem 4.2.1 (describing RUN_ϕ) in P-def. We will not carry out the details here, since as mentioned in the beginning of this section of the power of QPV (and hence P-def) has been well established.

To prove the other direction, we show that every theorem of P-def in the language of V_1-Horn is a theorem of V_1-Horn. First note that using the extensionality axiom E (4.25), every equation $S = T$ between second-order terms is provably equivalent in P-def to a Σ_0^B formula (denoted $E(S = T)$) not involving second-order =. Therefore we may assume that formulas in P-def do not involve such second-order equations.

Now we claim that for every derived function symbol F of P-def there is a function symbol F' of V_1-Horn(FP) which represents the same function, such that V_1-Horn(FP) proves the translation of the defining formula for F. The translation is carried out by replacing each function symbol G in the defining formula by its V_1-Horn(FP) counterpart G', and by replacing each second-order equation $S = T$ by $E(S = T)$. From this property a simple model-theoretic argument shows that for every formula ϕ of P-def, if ϕ is a theorem of P-def then its translation ϕ' is a theorem of V_1-Horn(FP). The lemma follows.

We define the translation of F to F' by induction on the rank of F. If F is introduced in P-def by (4.18,4.19) where ϕ_F is a Σ_0^B formula, then we introduce F' in V_1-Horn(FP) by (4.18,4.19) where $\ell_{F'}$ is ℓ'_F (the translation of ℓ_F into V_1-Horn(FP)) and $\phi_{F'}$ is a Σ_1^B-Horn formula equivalent to ϕ'_F, using Corollary 4.1.5 and Theorem 4.5.3. If f is introduced in P-def by (4.20) then f' is introduced in V_1-Horn(FP) using (4.20) with F' for F.

Now suppose that F is introduced in P-def by (4.26,4.27). The idea is to fix the arguments $(z, \bar{x}, \bar{Y})$ of F and present a formula defining an array $P(i, y)$ (and its negative counterpart $\tilde{P}(i, y)$) giving the i-th bit of $F(y, \bar{x}, \bar{Y})$, $0 \leq i < \ell'_F(y, \bar{x}, \bar{Y})$, $0 \leq y \leq z$, where ℓ'_F is the translation of ℓ_F as a term of V_1-Horn(FP). The formula will recursively define all values of $P(i, y)$ and $\tilde{P}(i, y)$ successively for $y = 0, 1, ..., z$. To give the step from y to $y+1$ we must translate the formula $H(z, \bar{x}, \bar{Y}, Z)(i)$ into one which is "Horn with respect to Z". In what follows we will suppress the variables $\bar{x}, \bar{Y}$.

Applying Lemma 4.5.5, let $\psi(i, y, Z)$ be a Σ_1^B-Horn formula of V_1-Horn equivalent to the formula $H'(y, \textsc{Chop}(\ell'_F(y), Z))(i)$. Next apply Corollary 4.2.4 to obtain the formula $\textsc{Run}_{\psi(i)}(b, R, \tilde{R}, y, Z)$. Now apply Lemma 4.5.7 below to $\textsc{Run}_{\psi(i)}$, using the bound $\ell'_F(y)$ for ℓ to obtain an equivalent Σ_1^B-Horn formula not involving $|Z|$. Further modify this formula by replacing each positive subformula of the form $Z(t)$ by $(t < \ell'_F(y) \wedge \neg\tilde{Z}(t))$ (distribute $\vee$ over $\wedge$ to keep the quantifier-free part in CNF) and each occurrence of the form $\neg Z(t)$ by $(\neg Z(t) \vee \ell'_F(y) < t)$. The result is a formula $\overline{\textsc{Run}}_{\psi(i)}(b, R, \tilde{R}, y, Z, \tilde{Z})$ which is Σ_1^B-Horn with respect to $Z, \tilde{Z}$ whose truth is unchanged if Z is replaced by $\textsc{Chop}(\ell'_F(y), Z)$. Further, defining the hypothesis $\textsc{Hypo}(Z, \tilde{Z})$ to be the formula

$$\textsc{Hypo} \equiv \forall j < \ell'_F(y)(Z(j) \leftrightarrow \neg\tilde{Z}(j))$$

it follows by Corollary 4.2.4 that V_1-Horn(FP) proves

$$\textsc{Hypo} \rightarrow \exists R \exists \tilde{R}\ \overline{\textsc{Run}}_{\psi(i)}(b, R, \tilde{R}, y, Z, \tilde{Z}) \tag{4.29}$$

$$[\textsc{Hypo} \wedge \overline{\textsc{Run}}_{\psi(i)}(b, R, \tilde{R}, y, Z, \tilde{Z})] \rightarrow \forall i < b[(R(i) \leftrightarrow H'(y, \textsc{Chop}(\ell'_F(y), Z))(i)) \wedge (\tilde{R}(i) \leftrightarrow \neg R(i))] \tag{4.30}$$

Referring to (4.18,4.19), we take the defining term $\ell_{F'}(z)$ for $F'(z)$ in V_1-Horn(FP) to be $\ell'_F(z)$, and the bit graph formula $\phi_{F'}(i, z)$ for $F'(z)$ to be a suitable prenex form of

$$\phi_{F'}(i, z) \equiv \exists P \exists \tilde{P}(P(i, z) \wedge \hat{\phi}(z, P, \tilde{P}))$$

where $\hat{\phi}$ is

$$\begin{aligned}\hat{\phi}(z, P, \tilde{P}) \ \equiv \ & \forall j < \ell'_F(0)[(P(j, 0) \leftrightarrow G'()(j)) \wedge (\tilde{P}(j, 0) \leftrightarrow \neg G'()(j))] \wedge \\ & \forall y < z\ \overline{\textsc{Run}}_{\psi(i)}(\ell'_F(y+1), P(*, y+1), \tilde{P}(*, y+1), y, P(*, y), \tilde{P}(*, y))]\end{aligned}$$

where for example the notation $P(*, y+1)$ indicates that each occurrence of the form $R(t)$ in $\overline{\text{RUN}}_{\psi(i)}$ is replaced by $P(t, y+1)$.

It remains to show that the translations of (4.26,4.27) follow in V_1-Horn(FP) from (4.18,4.19). First note that $\text{CHOP}' = \text{CHOP}$, since the defining formulas for CHOP in P-def are also in V_1-Horn(FP). Next note that by (4.18) for F', the RHS's of the translations of (4.26,4.27) can be replaced by the second argument of CHOP in each case; that is by $G'()$ and $H'(z, F'(z))$ respectively. Now (4.26) follows easily from the definition of $\hat{\phi}(0, P\tilde{P})$.

To establish the translation of (4.27) we make a series of Claims.

Claim 1: V_1-Horn(FP) $\vdash\ \hat{\phi}(z, P, \tilde{P}) \to \forall y \leq z\ \text{HYPO}(P(*, y), \tilde{P}(*, y))$

This follows using induction on z and (4.30).

Claim 2: (Uniqueness of P) V_1-Horn(FP) proves

$$[\hat{\phi}(z, P, \tilde{P}) \wedge \hat{\phi}(z, Q, \tilde{Q})] \to \forall y \leq z \forall i < \ell'_F(y)(P(i, y) \leftrightarrow Q(i, y))$$

Again this follows using induction on z and (4.30) and Claim 1.

Claim 3: V_1-Horn(FP) $\vdash\ \hat{\phi}(z, P, \tilde{P}) \to \forall y \leq z \forall i < \ell'_F(y)[P(i, y) \leftrightarrow \phi_{F'}(i, y)]$

The left-to-right direction of the equivalence is immediate from the definition of $\hat{\phi}$. The right-to-left direction requires Claim 2.

Claim 4: V_1-Horn(FP) $\vdash \exists P \exists \tilde{P} \hat{\phi}(z, P, \tilde{P})$

This follows using induction on z, (4.29), and Claim 1.

Claim 5: V_1-Horn(FP) $\vdash\ \forall i < \ell'_F(z)[\phi_{F'}(i, z+1) \leftrightarrow H'(z, F'(z))(i)]$

The left-to-right direction follows from the definition of $\phi_{F'}$, Claim 1, (4.30), and Claim 3. The right-to-left direction uses Claim 4 in addition.

Finally the translation of (4.27) follows immediately from Claim 5.

This completes the proof of Theorem 4.5.8. □

Corollary 4.5.9. *The $\forall\Sigma_1^b$ consequences of S_2^1 are finitely axiomatizable.*

Proof. This is a direct consequence of theorem 4.4.2. Since V^1 is $\forall\Sigma_1^B$ conservative over P-def [Zam96], it follows from Theorem 4.5.8 that the $\forall\Sigma_1^B$ consequences of V^1 and of V_1-Horn are the same, and hence are finitely axiomatizable. The corollary is equivalent to asserting that the $\forall\Sigma_1^B$ consequences of V^1 are finitely axiomatizable, by the RSUV isomorphism. $\square$

Chapter 5

V-Krom: a system of arithmetic for NL

In this chapter we describe a second-order theory V-Krom of bounded arithmetic for nondeterministic log space. This system is based on Grädel's characterization of NL by second-order Krom formulae with only universal first-order quantifiers, which in turn is motivated by the result that the decision problem for 2-CNF satisfiability is complete for coNL (and hence for NL).

Our main result for V-Krom is a formalization of the Immerman-Szelepcsényi theorem that NL is closed under complementation. This formalization is necessary for the proof of the strict closure property, which is then used to show that the NL functions are Σ_1^B-definable in V-Krom.

To our knowledge only one other theory associated with the class NL was published, namely the theory S^{Nlog} of [CT92]. This is a second-order theory axiomatized by induction over encodings of NL Turing machines, and even the authors of [CT92] state that it is awkward. Besides, Clote and Takeuti rely on Immerman-Szelepcsényi theorem for their witnessing proof, whereas we can formalize the Immerman-Szelepcsényi proof, as given in [Imm99], explicitly in our system, thus proving the strong closure property. Additionally, we show that Σ_1^B-Krom is constructive by formalizing Σ_1^B-Krom satisfiability proof using transitive closure. Together these properties imply that the class of Σ_1^B-definable functions of V-Krom is precisely the NL functions, that is string functions having NL relations as their bitgraphs (and number functions having NL relations as their graphs).

Recently another system of arithmetic for NL was suggested by Nguyen. His system VNL consists of V^0 together with an axiom stating an existence of paths in directed graphs. In the last part of this chapter we show that our V-Krom is equivalent to VNL.

5.1 System V-Krom.

Analogously to Σ_1^B-Horn as a translation of $SO\exists$-Horn to the bounded arithmetic setting, we define Σ_1^B-Krom as a translation of $SO\exists$-Krom from the definition 2.2.6. This is another example of a restricted Σ_1^B class of formulae.

Definition 5.1.1. A formula is Σ_1^B-Krom if it is of the form

$$\exists P_1 ... \exists P_k \forall x_1 < t_1(\bar{a}) ... \forall x_m < t_m(\bar{a}, x_1, \dots x_{m-1}) \psi(\bar{x}, \bar{P}, n, \bar{a}, \bar{Y}), \tag{5.1}$$

where ψ is Krom with respect to $P_1, \dots, P_k$.

That is, a Σ_1^B-Krom formula is essentially a 2-CNF if we only consider $P_i(t)$ as significant literals, and P_i may only occur as a P-literal. We define V-Krom to be V-Φ with $\Phi = \Sigma_1^B$-Krom.

Definition 5.1.2. The theory V-Krom is the theory over $\mathcal{L}_A^2$ axiomatized by 2-BASIC axioms together with a comprehension scheme over Σ_1^B-Krom formulae.

5.2 V-Krom extends V^0.

Existential first-order quantifiers are not allowed in a Σ_1^B-Krom formula. That is, a Σ_0^B formula is not automatically a Σ_1^B-Krom formula, though Σ_1^B-Krom clearly has much more expressive power. In this section, we develop a construction which allows us to convert Σ_0^B formulae to Σ_1^B-Krom. This is a similar result to Theorem 4.1.4 for V_1-Horn, but the construction for V-Krom is quite different.

Theorem 5.2.1. *For every Σ_0^B formula ψ there is a Σ_1^B-Krom formula ψ^* such that*

$$V\text{-}Krom \vdash \psi \leftrightarrow \psi^*$$

Proof. Assume that ψ is in the prenex form. The idea behind the proof of Theorem 5.2.1 is that ψ^* begins with $\exists S$, where S is a multi-dimensional array with one dimension per each alternation of quantifier in ψ. For every dimension corresponding to existential, the first element is set to false, and the last element to true. The clauses encode a pass through the array from the first to last element, with a property that false values can only become true values during this pass if a witness to the existential quantifier was found.

Base case 1: Let $\psi = \exists z < n\phi(z)$. Set ψ' to

$$\exists S \forall z < n \neg S(0) \wedge S(n) \wedge (\neg\phi(z) \wedge \neg S(z) \rightarrow \neg S(z+1))$$

Suppose ψ is satisfied. Let z_0 be the minimal witness. Then $S(z) = \begin{cases} \bot & z \leq z_0 \\ \top & z > z_0 \end{cases}$ satisfies ψ'. The comprehension scheme guarantees the existence of such S.

Now, let S be a witness for ψ'. Take minimal z such that $S(z) = \top$; it cannot be 0 since $S(0) = \bot$. Then there exists z_0 such that $z_0 + 1 = z$. Since z is minimal, $S(z_0)$ is false. Therefore, to satisfy the last clause, $\phi(z_0)$ must hold, so z_0 is the witness for $\exists z$.

Base case 2: Let $\psi = \exists z < n \forall u < n\phi(z, u)$. Set ψ' to

$$\begin{aligned}\exists S \forall z < n \forall u < n \neg S(0) \wedge S(n) \\ \wedge(\neg\phi(z, u) \wedge \neg S(z) \rightarrow \neg S(z+1))\end{aligned}$$

Suppose ψ is satisfied. Let z_0 be a witness to $\exists z$. Define $S(z) = \begin{cases} \bot & z \leq z_0 \\ \top & z > z_0 \end{cases}$. Here, $\neg S(0)$ and $S(n)$ are satisfied trivially. For $z < z_0$, if $S(z+1) = \bot$, the last clause is always satisfied. For $z > z_0$, since $S(z) = \top$, the LHS of the last clause is falsified, allowing $S(z+1) = \top$. Now, if $z = z_0$, then for all u $\phi(z_0, u)$ holds, falsifying LHS for all u and allowing $S(z_0 + 1)$ to be $\top$.

Now, let ψ' be true, with S a witness. Since $S(0)$ is false and $S(n)$ is true, there is a value z_0 such that $\neg S(z_0)$ and $\forall z > z_0$ $S(z)$ (induction on $Y(i) \leftrightarrow \neg S(n-i)$). By assumption, $S(z_0)$ is false and $S(z_0+1)$ is true, so to satisfy the last clause for all u $\phi(z, u)$ is true.

Induction step: Let ϕ be a Σ_0^B formula starting with $\exists z$, with k blocks of $\exists\forall$ quantifiers and with x, y as free variables. Assume that ϕ is equivalent to a Σ_1^B-Krom formula

$$\phi'(x, y) \equiv \exists S \forall z < n \forall \bar{u} < \bar{n} \neg S(0, \bar{0}) \wedge S(n, \bar{0}) \wedge \phi_k$$

Here, ϕ_k consists of the set of formulae for a corresponding base case (without the $S(0)$

and $S(n)$), together with clauses of the form

$$(\neg S(\bar{v}, z, u, n, \bar{0}) \to \neg S(\bar{v}, z, n, n, \bar{0}))$$
$$(\neg S(\bar{v}, z, n, n, \bar{0}) \to \neg S(\bar{v}, z+1, u, 0, \bar{0}))$$

for each additional $\exists z \forall u$ block.

Then the following formula is equivalent to
$\psi \equiv \exists x < n \forall y < n \phi(x, y)$:

$$\begin{aligned} \psi' \equiv \exists S' \forall x < n \forall y < n \forall z < n \forall \bar{u} < \bar{n} \\ \neg S'(0,0,0,\bar{0}) \wedge S'(n,0,0,\bar{0}) \wedge \phi'_k \\ \wedge(\neg S'(x,y,n,\bar{0}) \to \neg S'(x,n,n,\bar{0})) \\ \wedge(\neg S'(x,n,n,\bar{0}) \to \neg S'(x+1,y,0,\bar{0})) \end{aligned}$$

where ϕ'_k is ϕ_k with every occurrence of $S(\bar{v})$ changed to $S'(x, y, \bar{v})$.

Suppose that ψ is true, that is, there exists x that witnesses the outermost existential quantifier. Let x_0 be the smallest witness, and $S_{x',y'}(z, \bar{u})$ a witness of ϕ' with $x = x', y = y'$. We define S' to be:

$$S'(x,y,z,\bar{u}) \equiv \begin{cases} \perp & x < x_0 \\ S_{x_0,y}(z,\bar{u}) & x = x_0, y < n \\ \top & x > x_0 \text{ or } x = x_0, y = n \end{cases}$$

$S'(x_0, 0, 0, \bar{0})$ is false because either $x_0 = 0$ and $S'(0,0,0,\bar{0})$, or $S'(x_0-1, n, n, \bar{0}) = \perp$ since x_0 is the smallest witness. So there is nothing forcing $S'(x, y, z, \bar{u})$ to be true for $x < x_0$; we can safely set all of these values to $\perp$. If $S'(x_0, n, n, \bar{0})$ is true, we can set $S'(x, y, z, \bar{u}) = \top$ for any $x > x_0$ by the last clause of the formula. Now since x_0 is a witness, for all y there exists $S_{x_0,y}$ satisfying the formula from the induction hypothesis, so we can use these witnesses to construct $S'(x_0, y, z, \bar{u})$ for all values of $y < n, z \leq n, \bar{u} \leq \bar{n}$. By induction hypothesis, $S_{x_0,y}(n, \bar{0})$ is true for any y. Therefore, $S'(x_0, n, n, \bar{0}) = \top$ satisfies the clause $\neg S'(x, y, n, \bar{0}) \to \neg S'(x, n, n, \bar{0})$.

Now assume that ψ' is true, that is, there exists S' that witnesses the second-order existential quantifier. Take the smallest value x_0 such that $S'(x_0, n, n, \bar{0}) = \top$. Such value exists because $S'(n, 0, 0, \bar{0}) = \top$ by the last clause follows from $S'(n-1, n, n, \bar{0})$. We know that $S'(x_0, 0, 0, \bar{0}) = \perp$ by minimality of x_0. Therefore, $S'(x_0, y, n, \bar{0}) = \top$ for all y. Now we can use $S'(x_0, y)$ to witness ϕ' on x_0, y. By induction hypothesis, that gives witnesses to $\exists z$, so $\phi(x_0, y)$ holds for all y. □

We can choose $S'(x, n, 0, \bar{0})$ rather than $S'(x, n, n, \bar{0})$ to be the flag for the $\forall$ quantifier, but the intuitive meaning of $S'(x, n, n, \bar{0})$ is "for all y $S'(x, y, n, \bar{0})$ holds, where $S'(x, y, n, \bar{0})$ is $S(n, \bar{0})$, the "result" variable of the inner subformula, with x and y introduced by replacement.

It is easy to generalize the construction to allow different bounds on different first-order variables, as well as blocks of quantifiers rather than one of each type alternating.

From Theorem 5.2.1, the following corollary is easy.

Corollary 5.2.2. *Comprehension over Σ_0^B formulae is a theorem of V-Krom. Thus, V-Krom is an extension of V^0, so Theorem 3.2.10 holds for V-Krom.*

With the help of Corollary 5.2.2, V-Krom proves induction on both Σ_0^B and Σ_1^B-Krom formulae. By using the comprehension scheme for both formula classes we can justify induction over $\Sigma_0^B(\Sigma_1^B$-Krom) formulae, and in fact over formulae built by nesting Σ_1^B-Krom formulae with bounded quantifiers and the Boolean connectives. This idea is used implicitly in later sections.

5.3 V-Krom(TrCl)

In this section we show how to introduce the transitive closure operator into V-Krom, which we then use to prove Immerman-Szelepcsényi theorem. We show that V-Krom can formalize the proof given in [Imm99], sections 9.2–9.5.

5.3.1 Definitions

Recall the transitive closure operator from example 2.2.1. In this section we show how to "translate" it into the bounded arithmetic setting by adding a defining axiom for it into our theory. Since the defining axiom is a (negated) Σ_1^B-Krom formula, the resulting theory has the same power as the original V-Krom.

We wish to define the transitive closure of a relation given by a formula $\phi(x, y)$ (which may contain free variables besides x, y) on the domain $\{0, 1, ..., n-1\}$ of n elements. Any relation $R(x, y)$ that contains this transitive closure must satisfy conditions of reflexivity and ϕ-step transitivity on the domain above. The following formula *Cond* encodes these conditions:

$$\begin{aligned}&Cond(\phi, R, n) \equiv \\ &\quad \forall x, y, z < n(R(x, x) \wedge (\phi(x, y) \wedge R(y, z) \to R(x, z)))\end{aligned}$$

We will write just $Cond(\phi, R)$ when n is clear from the context.

Now we define the transitive closure relation $TrCl\phi$ to be the intersection of all relations R satisfying $Cond(\phi, R)$.

$$TrCl_{x,y}\phi(x, y)[a, b, n] \leftrightarrow \forall R(Cond(\phi, R, n) \to R(a, b)) \qquad \text{(AxTC)}$$

Remark 5.3.1. It is important that the negation of the RHS of (AxTC) is equivalent to a Σ_1^B-Krom formula if ϕ is quantifier-free. This is because when $Cond(\phi, R, n)$ is put in conjunctive normal form, each clause has at most two occurrences of R. Note that an alternative definition of $TrCl$ would be to change the condition $Cond(\phi, R)$ to a condition $Cond'(\phi, R)$, where $Cond'(\phi, R)$ asserts that R is reflexive, transitive, and $\phi(x, y) \supset R(x, y)$. However then the negation of the RHS of (AxTC) would not be a Σ_1-Krom formula because the transitivity clause in $Cond'$ requires three occurrences of R. Our use of $Cond$ instead of $Cond'$ makes the proof of transitivity of $TrCl$ just a little harder (see Lemma 5.3.4). However, whenever a pair (a, b) is not in the transitive closure of ϕ, the comprehension axiom immediately gives an existence of a relation R containing all of the transitive closure over ϕ, but not containing (a, b).

We want to extend the vocabulary of V-Krom by including instances of TrCl as defined above.

Definition 5.3.2. The class $\Sigma_0^B(\text{TrCl})$ is defined inductively as follows:

(i) Every quantifier-free formula of V-Krom is in $\Sigma_0^B(\text{TrCl})$

(ii) If ϕ is in $\Sigma_0^B(\text{TrCl})$, then so is $TrCl_{x,y}\phi(x, y)[a, b, n]$

(iii) Every Σ_0^B combination of formulae in $\Sigma_0^B(\text{TrCl})$ is in $\Sigma_0^B(\text{TrCl})$

The class $\Sigma_0^B(\text{TrCl}^+)$ is defined in the same way, except in (iii) we allow only Σ_0^B combinations with positive occurrences of $\Sigma_0^B(\text{TrCl})$ formulae.

The system V-Krom(TrCl) is V-Krom augmented with the class $\Sigma_0^B(\text{TrCl})$ of formulae, and has (AxTC) for each ϕ in $\Sigma_0^B(\text{TrCl})$.

Since the only new axioms in V-Krom(TrCl) are definitions of new relations, it is a conservative extension of V-Krom.

Theorem 5.3.3. *V-Krom(TrCl) proves the induction axiom and the comprehension axiom for every formula in Σ_0^B(TrCl).*

Proof. The essential point is that the negation of the RHS of (AxTC) is equivalent to a Σ_1^B-Krom formula if ϕ is quantifier-free (see Remark 5.3.1). The theorem follows by induction on the depth of nesting of $TrCl$ formulae. □

In the axiom of transitive closure (AxTC), n is a bound on the first-order variables, and the transitive closure relation $TrCl(a, b)$ is false unless $a, b < n$. In the special case $n = 0$, $TrCl(a, b)$ is always false, and when $n = 1$, $TrCl(a, b)$ holds iff $a = b = 0$.

5.3.2 Properties of transitive closure

First we show that V-Krom proves the transitivity of the transitive closure relation.

Lemma 5.3.4. *Let $TrCl(x, y)$ stand for $TrCl_{u,v}\phi(u, v)[x, y]$. Then for all Σ_0^B(TrCl) formulae ϕ, V-Krom(TrCl) proves*

$$TrCl(x, y) \wedge TrCl(y, z) \longrightarrow TrCl(x, z)$$

Proof. Reasoning in V-Krom(TrCl), fix x, y, z and assume $TrCl(x, y)$ and $TrCl(y, z)$. Referring to (AxTC), let R be any relation satisfying $Cond(\phi, R)$. It suffices to show $R(x, z)$.

Define R' by the condition

$$R'(a, b) \leftrightarrow (b = y \wedge R(a, z)) \vee (b \neq y \wedge R(a, b))$$

Note that R' can be defined in V-Krom(TrCl) by comprehension. Using the facts $Cond(\phi, R)$ and $R(y, z)$ (because $TrCl(y, z)$) it is easy to show $Cond(\phi, R')$. Therefore $R'(x, y)$ (because $TrCl(x, y)$), and hence $R(x, z)$ (by definition of R'). □

The definition of transitive closure is robust enough in that adding ϕ-edges from the left or from the right gives the same answer. That is, suppose that instead of $Cond$, we define AxTC using $Cond^r$ of the form

$$Cond^r(\phi, R, n) \equiv \forall x, y, z < n(R(x, x) \wedge (R(x, y) \wedge \phi(y, z) \rightarrow R(x, z))$$

Define $TrCl^r$ by

$$TrCl^r(a,b) \leftrightarrow \forall R(Cond^r(\phi, R, n) \to R(a,b))$$

Lemma 5.3.5. *V-Krom proves*

$$TrCl^r(a,b) \leftrightarrow TrCl_{u,v}\phi(u,v)[a,b,n]$$

Proof. By an argument similar to the proof of Lemma 5.3.4, V-Krom proves transitivity of $TrCl^r$. Therefore V-Krom proves $Cond(\phi, TrCl^r)$, from which the right-to-left direction follows. The left-to-right direction follows by symmetry. □

5.4 Normal form of TrCl

In this section we formalize the proof from [EF95, Imm99] of the theorem stating, informally, that any bounded formula with only positive occurrences of transitive closure operator can be converted into a formula with only one, outermost occurrence of TrCl. Moreover, the bounds of this transitive closure operator can be arbitrary (under some restrictions). This is the most technical result needed for the proof of closure of Σ_1^B-Krom under complementation.

In the following result, the notation $[\bar{0}, \bar{n}]$ stands for $[s,t]$, where s and t are term coding the tuples $\bar{0}$ and $\bar{n}$, respectively using the pairing function 3.2.5.

Theorem 5.4.1. *Any $\Sigma_0^B(TrCl^+)$ formula ϕ is equivalent to $TrCl_{\bar{x},\bar{x}'}\psi(\bar{x},\bar{x}')[\bar{0},\bar{n}]$, where ψ is quantifier-free. Here, $\bar{n}$ and the number of variables in the vectors $\bar{x}, \bar{x}', \bar{0}, \bar{n}$ depend on the structure of ϕ. Moreover, V-Krom(TrCl) proves this equivalence.*

The proof is by structural induction on ϕ, and formalizes in V-Krom(TrCl) the arguments in [EF95, Imm99], using results in the previous subsections. For every boolean connective (except negation) and quantifier, an equivalence between the original and constructed formula is shown by expanding the definitions of transitive closure via $AxTC$, negating both sides, and constructing assignments for the variables under second-order existential quantifiers for one side from the other. Since the negation of $AxTC$ for a quantifier-free ϕ is Σ_1^B-Krom, the existence of such witnesses is guaranteed by Σ_1^B-Krom comprehension axioms.

Proof. The proof is by structural induction on ϕ. The base case is when ϕ is a quantifier-free formula. The induction hypothesis is that ϕ consists of an outermost connective over one or two formulae already in the form $TrCl_{\bar{x},\bar{x}'}\psi(\bar{x},\bar{x}')[\bar{0},\bar{n}]$. There would be extra "flag" variables added at every step of induction into the outermost transitive closure with limits $0, n_k$. That is, in the resulting formula the transitive closure is taken over a tuple of variables with limits $\{\bar{0},\bar{n}\}$. In the axiom of transitive closure we will assume that x, y, z, a, b are all single variables. We can convert tuples into single variables by using a pairing function, and adjusting the bound accordingly.

Base case: Let ϕ be a quantifier-free formula without occurrences of TrCl. Take x, x' not occurring in ϕ. Then, $\phi \equiv TrCl_{x,x'}\phi(x,x')[0,1]$. That is, $n_1 = 1$.

Proof. Suppose that ϕ evaluates to $\top$ on its free variables. Then the relation $\phi(x,x')$ is true for any $x, x' \leq n_1$, in particular for $0, 1$. Therefore, for all R in AxTC, $R(0,1)$ holds. This happens in the AxTC when $R(1,1) \wedge (\phi(0,1) \wedge R(1,1) \rightarrow R(0,1))$.

Now suppose that that ϕ evaluates to $\perp$. Consider minimal R satisfying the axiom; this will be $R(x,x') \equiv x = x'$, where $x, y \leq n$. Its existence is guaranteed by the comprehension axiom and it satisfies the hypothesis of AxTC. Since $\phi(x,x')$ is false for any $x \neq x'$, that clause will always hold, for any R containing the equivalence relation. Since $0 \neq 1$, $(0,1) \notin R$ and thus not in the transitive closure of ϕ. □

Adjusting upper bound: Given $\phi \equiv TrCl_{\bar{u},\bar{u}'}\psi(\bar{u},\bar{u}')[\bar{0},\bar{n}]$, we would like to change bounds on $\bar{u},\bar{u}'$ to $\bar{n}'$, where for each $n_i \in \bar{n}$, $n_i' \in \bar{n}'$, $n_i \leq n_i'$. Then $\phi' = TrCl_{\bar{u},\bar{u}'}\psi'(\bar{u},\bar{u}')[\bar{0},\bar{n}']$, where

$$\psi' \equiv (\bar{u} < \bar{n} \wedge \psi(\bar{u},\bar{u}')) \vee (\bar{u} = \bar{n} \wedge \bar{u}' = \bar{n}')$$

Proof. The idea is to run $\psi(\bar{u},\bar{u}')$ until we reach $\bar{n}$, and then make an additional step. By negating both sides of the axiom of transitive closure representation of $\phi \leftrightarrow \phi'$ we get

$$\begin{aligned} &\exists R\ Cond(\psi, R, \bar{n}) \wedge \neg R(\bar{0},\bar{n}) \\ \Leftrightarrow &\exists Q\ Cond(\psi', Q, \bar{n}') \wedge \neg Q(\bar{0},\bar{n}') \end{aligned}$$

Given Q, we can define R as $R(\bar{u},\bar{u}') \Leftrightarrow (\bar{u} < \bar{n} \wedge \bar{u}' \leq \bar{n} \wedge Q(\bar{u},\bar{u}')) \vee (\bar{u} = \bar{n} \wedge \bar{u}' = \bar{n}')$. The existence of such R is given by comprehension. It is easy to check that it satisfies the condition of AxTC(ϕ) and does not contain $(\bar{0},\bar{n})$. Now to construct Q from R set

$Q(\bar{u}, \bar{u}') \Leftrightarrow (R(\bar{u}, \bar{u}') \wedge \bar{u} < \bar{n} \wedge \bar{u}' \leq \bar{n})$ Again, Q satisfies the condition of AxTC(ϕ') and does not contain $(\bar{0}, \bar{n}')$. □

Disjunction: Let $\phi \equiv \phi_1 \vee \phi_2$, where $\phi_i \equiv TrCl_{\bar{x_i}, \bar{x_i}'} \psi_i(\bar{x_i}, \bar{x_i}')[\bar{0}, \bar{n}]$ for $i = \{1, 2\}$. Introduce new variables v, v'. If one of the formulae has less variables, all vectors of variables are padded with dummy variables. By the previous case, we can assume that the bound $\bar{n}$ is the same in both formulae. Let u, u' denote $\bar{x_i}, \bar{x_i}'$, possibly padded with dummy variables. Let $n_k = 1$ again. Now, $\phi \equiv TrCl_{uv, u'v'} \psi(uv, u'v')[\bar{0}0, \bar{n}n_k]$, where

$$\begin{aligned}\psi(uv, u'v') \equiv & (v = v' = 0 \wedge \psi_1(u, u')) \\ & \vee(u = u' = \bar{0} \wedge v = 0 \wedge v' = 1) \\ & \vee(u = u' = \bar{n} \wedge v = 0 \wedge v' = 1) \\ & \vee(v = v' = 1 \wedge \psi_2(u, u'))\end{aligned}$$

Proof. We need to prove that

$$AxTC(\phi) \Leftrightarrow AxTC(\phi_1) \vee AxTC(\phi_2)$$

Let x, y, z encode pairs of the form (u, v). By negating both sides of the previous statement, obtain

$$\begin{aligned}V\text{-Krom} \vdash & \exists R \; Cond(\phi, R, n) \wedge \neg R(\langle \bar{0}, 0\rangle, \langle \bar{n}, 1\rangle)) \\ & \Leftrightarrow (\exists R_1 \; Cond(\phi_1, R_2, n) \wedge \neg R_1(\bar{0}, \bar{n}))) \\ & \quad \wedge (\exists R_2 \; Cond(\phi_2, R_2, n) \wedge \neg R_2(\bar{0}, \bar{n})))\end{aligned}$$

Suppose we have R_1 and R_2 witnessing ψ_1' and ψ_2'. By binary comprehension, the following formula defines R from R_1 and R_2:

$$\begin{aligned}R(\langle u, v\rangle, \langle u', v'\rangle) \Leftrightarrow & (\langle u, v\rangle = \langle u', v'\rangle) \\ & \vee(v = 0 \wedge v' = 0 \wedge R_1(u, u')) \\ & \vee(v = 1 \wedge v' = 1 \wedge R_2(u, u')) \\ & \vee(u = \bar{0} \wedge v = 0 \wedge u' = \bar{0} \wedge v' = 1) \\ & \vee(u = \bar{n} \wedge v = 0 \wedge u' = \bar{n} \wedge v' = 1)\end{aligned}$$

Such R satisfies the condition of AxTC(ϕ) and does not contain $(\langle \bar{0}, 0\rangle, \langle \bar{n}, 1\rangle)$. Since R is defined by an open formula, first-order reasoning is sufficient to prove its correctness.

For the opposite direction, define R_1 and R_2 from R by

$$R_1(u,u') \Leftrightarrow R(\langle u,0\rangle,\langle u',0\rangle),\ R_2(u,u') \Leftrightarrow (R(\langle u,1\rangle,\langle u',1\rangle)$$

Again, the proof of correctness is a simple proof by cases. $\square$

Conjunction: Now let $\phi \equiv \phi_1 \wedge \phi_2$, where $\phi_i \equiv TrCl_{\bar{x_i},\bar{x_i}'}\psi_i(\bar{x_i},\bar{x_i}')[\bar{c},\bar{d}]$ for $i = \{1,2\}$. As before, introduce new variables u,u',v,v', set $n_k = 1$. Now, ϕ becomes $TrCl_{uv,u'v'}\psi(uv,u'v')[\bar{0}0,\bar{n}1]$, where

$$\begin{aligned}\psi(uv,u'v') \equiv &(v = v' = 0 \wedge \psi_1(u,u'))\\ &\vee (v = v' = 1 \wedge \psi_2(u,u'))\\ &\vee (u = \bar{n} \wedge v = 0 \wedge u' = 0 \wedge v' = 1)\end{aligned}$$

Proof. The proof is similar to the previous case. We define R from R_1, R_2 by

$$\begin{aligned}R(\langle u,v\rangle,\langle u',v'\rangle) \Leftrightarrow &(\langle u,v\rangle = \langle u',v'\rangle)\\ &\vee (v = 0 \wedge v' = 0 \wedge R_1(u,u'))\\ &\vee (v = 1 \wedge v' = 1 \wedge R_2(u,u'))\\ &\vee (u = \bar{n} \wedge v = 0 \wedge u' = \bar{n} \wedge v' = 0)\end{aligned}$$

For the opposite direction, we need to show that if there exists R then there is at least one of R_1 or R_2. Define them as in the previous case. If $(\langle\bar{0},0\rangle,\langle\bar{n},0\rangle) \in R$, then R_1 does not satisfy the negation of $AxTC(\phi_1)$. But then R_2 satisfies the negation of $AxTC(\phi_2)$, and we need the disjunction of them to hold. $\square$

Existential quantifier: We want to show that

$$\exists w \leq n TrCl_{\bar{u},\bar{u}'}\psi(\bar{u},\bar{u}';w)[\bar{0},\bar{n}] \equiv TrCl_{\bar{u}v,\bar{u}'v'}\psi'(\bar{u}v,\bar{u}'v')[\bar{0}0,\bar{n}n],$$

where ψ' is

$$\begin{aligned}\psi' = &(\bar{u} = \bar{0} \wedge v' = v+1) \vee (\psi(\bar{u},\bar{u}';v) \wedge v' = v)\\ &\vee (\bar{u} = \bar{n} \wedge \bar{u}' = \bar{n} \wedge v' = v+1)\end{aligned}$$

By negating both sides of $\exists w \leq n AxTC(\phi) \Leftrightarrow AxTC(\phi')$ we obtain

$$\begin{aligned}&\forall w \leq n \exists Q\ Cond(\psi,Q,n+1) \wedge \neg Q(\bar{0},\bar{n}))\\ &\quad\Leftrightarrow \exists R\ Cond(\psi',R,n+1) \wedge \neg R(\langle\bar{0},0\rangle,\langle\bar{n},n\rangle))\end{aligned}$$

Proof. Let $Q(\bar{u}, \bar{u}', w)$ be an array of Q_w obtained by exchanging the order of $\forall w \leq n$ and $\exists Q$ quantifiers (V-Krom proves replacement for Σ_1^B-Krom formulae).

Now, for the $\Rightarrow$ direction, define

$$\begin{aligned} R(\bar{u}v, \bar{u}'v') \equiv & (v = v' \wedge Q(\bar{u}, \bar{u}', v)) \\ & \wedge (\bar{u} = \bar{0} \wedge v \leq v' \wedge Q(\bar{u}, \bar{u}', v')) \\ & \wedge (\bar{u}' = \bar{n} \wedge v \geq v' \wedge Q(\bar{u}, \bar{u}', v)) \end{aligned}$$

For the $\Leftarrow$ direction, define $Q(\bar{u}, \bar{u}', w) \equiv R(\bar{u}w, \bar{u}'w)$. □

Universal quantifier: We want to show that

$$\forall w \leq n TrCl_{\bar{u},\bar{u}'}\psi(\bar{u}, \bar{u}'; w)[\bar{0}, \bar{n}] \equiv TrCl_{\bar{u}v,\bar{u}'v'}\psi'(\bar{u}v, \bar{u}'v')[\bar{0}0, \bar{n}n],$$

where ψ' is

$$\psi' = (\bar{u} < n \wedge \psi(\bar{u}, \bar{u}'; v) \wedge v' = v) \vee (\bar{u} = \bar{n} \wedge \bar{u}' = \bar{0} \wedge v' = v + 1)$$

By negating both sides of $\forall w \leq n AxTC(\phi) \Leftrightarrow AxTC(\phi')$ we obtain

$$\begin{aligned} & \exists w \leq n \exists Q\ Cond(\psi, Q, n+1) \wedge \neg Q(\bar{0}, \bar{n})) \\ \Leftrightarrow & \exists R\ Cond(\psi', R, n+1) \wedge \neg R(\langle \bar{0}, 0 \rangle, \langle \bar{n}, n \rangle)) \end{aligned}$$

Proof. To define the pair w_0 and the corresponding Q from R, set $w_0 \equiv min_w \neg R(\bar{0}0, \bar{n}w)$. Since $\neg R(\bar{0}0, \bar{n}n)$, $w_0 \leq n$. Now let $Q_{w_0}(\bar{u}, \bar{u}') \Leftrightarrow R(\bar{u}w_0, \bar{u}'w_0)$. Since by minimality of w_0 either $w_0 = 0$ or $R(\bar{0}0, \bar{n}w_0 - 1)$, $R(\bar{0}0, \bar{0}w)$. If $Q_{w_0}(\bar{0}, \bar{n})$, then by condition of transitive closure $R(\bar{0}0, \bar{n}w_0)$, contradicting the definition of w_0. Therefore, Q_{w_0} satisfies the condition of transitive closure of ϕ, and does not contain $(\bar{0}, \bar{n})$.

For the other direction, set

$$\begin{aligned} R(\bar{u}v, \bar{u}'v) \equiv & (\bar{u} = \bar{u}' \wedge v = v') \\ & \vee (v = v' \wedge v = w_0 \wedge Q_{w_0}(\bar{u}, \bar{u}')) \\ & \vee (v \leq v' \wedge v' < w_0) \vee (w_0 < v \wedge v \leq v') \\ & \vee (v < v' \wedge v' = w_0 \wedge Q_{w_0}(\bar{0}, \bar{u}')) \\ & \vee (v = w_0 \wedge v < v' \wedge Q_{w_0}(\bar{u}, \bar{n})) \end{aligned}$$

□

Nested TrCl: Now let ϕ be of the form $TrCl_{\bar{v},\bar{w}}(TrCl_{\bar{u},\bar{u}'}\psi(\bar{u},\bar{u}';\bar{v},\bar{w})[\bar{0},\bar{n}])[\bar{s},\bar{t},\bar{n}']$ Here $\bar{n}'$ is a bound on $\bar{v}$ and $\bar{w}$ that is a bound on the $\forall$ quantifiers in AxTC. This is the most complicated subcase. Then

$$\phi' = TrCl_{\bar{u},\bar{v},\bar{w},f,\bar{u}',\bar{v}',\bar{w}',f'}\psi'(\bar{u},\bar{v},\bar{w},f,\bar{u}',\bar{v}',\bar{w}',f')[\bar{0},\bar{0},\bar{0},0,\bar{n},\bar{n}',\bar{n}',2],$$

with $\psi' \equiv$

$$\begin{aligned}
&(\bar{v} = \bar{0} \wedge \bar{w} = \bar{0} \wedge \bar{u} = \bar{0} \wedge \bar{v}' = \bar{s} \wedge \bar{u}' = \bar{0} \wedge f = 0 \wedge f' = 1) \\
&\vee(\bar{u} \neq \bar{n} \wedge \bar{v} = \bar{v}' \wedge \bar{w} = \bar{w}' \wedge \psi(\bar{u},\bar{u}';\bar{v},\bar{w}) \wedge f' = f = 1) \\
&\vee(\bar{u} = \bar{n} \wedge \bar{w} \neq \bar{t} \wedge \bar{v}' = \bar{w} \wedge \bar{u}' = \bar{0} \wedge f' = f = 1) \\
&\vee(\bar{u} = \bar{n} \wedge \bar{w} = \bar{t} \wedge \bar{v}' = \bar{n}' \wedge \bar{w}' = \bar{n}' \wedge \bar{u}' = \bar{n} \wedge f = 1 \wedge f' = 2)
\end{aligned}$$

Proof. Opening the three TrCl by AxTC, call the quantified SO variable in the outer transitive closure of the original formula P, in the inner TrCl Q and in the new formula R. Negate, as before, both sides of the equivalence.

For the $\Rightarrow$ direction, given R we define P as

$$P(\bar{v},\bar{w}) \leftrightarrow (\bar{v} = \bar{w}) \vee \exists \bar{v}', \bar{w}' \leq n' R(\bar{0}\bar{v}\bar{w}'1, \bar{n}\bar{v}'\bar{w}1)$$

Since $\neg R(\bar{0}\bar{0}\bar{0}0, \bar{n}\bar{n}'\bar{n}'2)$ implies $\forall \bar{v}, \bar{w} \leq n' \neg R(\bar{0}\bar{s}\bar{w}1, \bar{n}\bar{v}\bar{t}1)$, so by construction $\neg P(\bar{s},\bar{t})$. Now we need to show that P satisfies the condition of AxTC. That is, for all $\bar{x},\bar{y},\bar{z} \leq \bar{n}'$, $P(\bar{y},\bar{z}) \wedge \neg P(\bar{x},\bar{z}) \rightarrow \neg\phi(\bar{x},\bar{y})$. By construction, this corresponds to $R(\bar{0}\bar{y}\bar{w}'1, \bar{n}\bar{v}'\bar{z}1)$ for some $\bar{v}', \bar{w}' \leq \bar{n}'$, and $\neg R(\bar{0}\bar{x}\bar{w}''1, \bar{n}\bar{v}''\bar{z}1)$ for all $\bar{v}'', \bar{w}'' \leq \bar{n}'$. It cannot be the case that $R(\bar{0}\bar{x}\bar{y}1, \bar{n}\bar{x}\bar{y}1)$, since otherwise by lemma 5.3.4 $R(\bar{0}\bar{x}\bar{y}1, \bar{n}\bar{y}\bar{z}1)$. The lemma 5.3.4 can be applied since ψ' is quantifier-free. Define $Q(\bar{u},\bar{u}') \leftrightarrow R(\bar{u}\bar{x}\bar{y}1, \bar{u}'\bar{x}\bar{y}1)$. This Q is the witness for the $\exists Q$ in the formula

$$\exists Q \; Cond(\psi(\bar{x},\bar{y}), Q, n+1) \wedge Q(\bar{0},\bar{n}),$$

where $\bar{x},\bar{y}$ are parameters of ψ. For the $\Leftarrow$ direction, note that $\neg P(\bar{v},\bar{w}) \rightarrow \neg\phi(\bar{v},\bar{w})$, so for all $(\bar{v},\bar{w}) \notin P$ there exists Q_{vw} containing transitive closure of $\psi(\bar{u},\bar{u}';\bar{v},\bar{w})$ such that

$\neg Q_{vw}(\bar{0}, \bar{n})$. Now we can define $R(\bar{u}, \bar{v}, \bar{w}, f, \bar{u}', \bar{v}', \bar{w}', f')$ by the formula

$$
\begin{array}{rl}
\text{Reflexivity:} & (\bar{u} = \bar{u}' \wedge \bar{v} = \bar{v}' \wedge \bar{w} = \bar{w}' \wedge f = f') \\
\text{Same internal TrCl path:} & \vee(\bar{v} = \bar{v}' \wedge \bar{w} = \bar{w}' \wedge f' = f = 1 \\
& \wedge (P(\bar{v}, \bar{w}) \vee (\neg P(\bar{v}, \bar{w}) \wedge Q_{vw}(\bar{u}, \bar{u}')))) \\
\text{Different internal TrCl paths:} & \vee((\bar{v} \neq \bar{v}' \vee \bar{w} \neq \bar{w}') \wedge f' = f = 1 \\
& \wedge (P(\bar{v}, \bar{w}) \vee (\neg P(\bar{v}, \bar{w}) \wedge Q_{vw}(\bar{u}, \bar{n}))) \\
& \wedge P(\bar{w}, \bar{v}') \\
& \wedge (P(\bar{v}', \bar{w}') \vee (\neg P(\bar{v}', \bar{w}') \wedge Q_{v'w'}(\bar{0}, \bar{u}')))) \\
\text{Initialization:} & \vee(\bar{v} = \bar{w} = \bar{0} \wedge \bar{u} = \bar{0} \wedge f = 0 \wedge f' = 1 \\
& \wedge P(\bar{s}, \bar{v}') \wedge (P(\bar{v}', \bar{w}') \vee (\neg P(\bar{v}', \bar{w}') \wedge Q_{v'w'}(\bar{0}, \bar{u}')))) \\
\text{Last step:} & \vee(P(\bar{v}, \bar{w}) \vee (\neg P(\bar{v}, \bar{w}) \wedge Q_{vw}(\bar{u}, \bar{n}))) \wedge P(\bar{w}, \bar{t}) \\
& \wedge \bar{u}' = \bar{n} \wedge \bar{v}' = \bar{w}' = \bar{n}' \wedge f = 1 \wedge f' = 2)
\end{array}
$$

It is clear that $\neg R(\bar{0}, \bar{0}, \bar{0}0, \bar{n}, \bar{n}', \bar{n}'2)$, because R does not hold on $f = 0, f' = 2$ for any setting of the other variables. Now we need to show that it satisfies the condition of the axiom of transitive closure.

The first line of the definition guarantees that R is reflexive. The second line defines the case when both $\bar{u}$ and $\bar{u}'$ are on in the same internal transitive closure path. Here we first use an expression $(P(\bar{v}, \bar{w}) \vee (\neg P(\bar{v}, \bar{w}) \wedge Q_{vw}(\bar{u}, \bar{u}')))$. The intuition behind it is that if $\neg P(\bar{v}, \bar{w})$ holds, then we are guaranteed the existence of a Q_{vw} containing transitive closure of ψ with parameters $\bar{v}, \bar{w}$ such that Q_{vw} does not contain $(\bar{0}, \bar{n})$. Now suppose that $P(\bar{v}, \bar{w})$ holds. It can happen in two cases: either when $\phi(\bar{v}, \bar{w})$ holds, or when $\exists \bar{v}' \phi(\bar{v}, \bar{v}') \wedge P(\bar{v}', \bar{w})$. In the first case, we know that for all Q_{vw}, $Q_{vw}(\bar{0}, \bar{n})$ holds. So we take Q_{vw} that contains all pairs $(\bar{u}, \bar{u}')$. Such Q_{vw} definitely satisfies the condition of transitive closure and contains $(\bar{0}, \bar{n})$. Since we cannot differentiate between the two cases when $P(\bar{v}, \bar{w})$ holds, and R is allowed to be much bigger than the transitive closure, as long as it satisfies the condition of transitive closure on all the additional points, we can set $Q_{vw}(\bar{u}, \bar{u}')$ to be true in all cases when $P(\bar{v}, \bar{w})$ holds. Envisioning the graph over which we take the outer transitive closure, when there is an edge between two points if the inner transitive closure contains $(\bar{0}, \bar{n})$, we add directed edges from v to w iff there is a path from v to w. It will not affect reachability for any pair of points. And again we take Q_{vw} to contain all $(\bar{u}, \bar{u}')$.

In the case of $\bar{u}$ and $\bar{u}'$ occurring in different instances of the inner transitive closure we split the "path" from $(\bar{u}, \bar{v}, \bar{w})$ to $(\bar{u}', \bar{v}', \bar{w}')$ into three steps. First, using trick described above, we check whether we can get from $\bar{u}$ to the end of the transitive closure (assuming that we can if Q_{vw} is defined to contain all pairs). Then we test if there is a path in the outer transitive closure from $\bar{w}$ to $\bar{v}'$: that corresponds to the truth value of $P(\bar{w}, \bar{v}')$. Lastly, we verify that there is an inner transitive closure path from $\bar{0}$ to $\bar{u}'$ with $\bar{v}', \bar{w}'$ as parameters.

The last two parts of the definition of R handle the cases of the first and last step of transitive closure. We say that there is a path from $(\bar{0}, \bar{0}, \bar{0})$ to $(\bar{u}', \bar{v}', \bar{w}')$ if there is a path from $\bar{s}$ to $\bar{v}'$ in the outer transitive closure, and from $\bar{0}$ to $\bar{u}'$ in the inner TrCl on $\bar{v}', \bar{w}'$. The last step is handled similarly. We need to make f-steps in these two cases to avoid the case when there is a path from $(\bar{0}, \bar{0}, \bar{0})$ to $(\bar{n}, \bar{n}', \bar{n}')$ not involving $\bar{s}$ or $\bar{t}$. □

□

5.5 Relating Σ_1^B-Krom and $\Sigma_0^B(TrCl^+)$

By the results of the previous sections, a bounded formula with positive occurrences of the transitive closure operator can be converted into a formula with a single outermost occurrence of TrCl, and then to a negated Σ_1^B-Krom formula by the axioms of transitive closure. This section shows how to convert an arbitrary Σ_1^B-Krom formula to negation of a $\Sigma_0^B(TrCl^+)$ formula; by appealing to Theorem 5.4.1 it is equivalent to a negated transitive closure of a quantifier-free formula.

5.5.1 $SO\exists$-Krom unsatisfiability algorithm

To achieve this goal we formalize the SO Krom satisfiability algorithm [Kro67], and represent it as negated transitive closure formulae. Using a pairing function, we may assume that we only have one second-order variable. Let ϕ^* be the following Σ_1^B-Krom formula:

$$\phi^* \equiv \exists P \forall x_1 < n_1 \ldots \forall x_k < n_k \psi(P, \bar{x}), \tag{5.2}$$
$$\text{where } \psi(P, \bar{x}) \equiv \bigwedge^{m} (L_j(t_j(\bar{x})) \vee L'_j(t'_j(\bar{x})) \vee \phi_j(\bar{x})).$$

Here, L_j and L'_j are P or $\neg P$, and ϕ_j are quantifier-free and contain no occurrence of P.

The algorithm below reduces the truth of this formula (given values for the free variables) to reachability in a directed graph. Step 1 reduces truth to the satisfiability of a propositional CNF formula A with at most two literals per clause, and Steps 2 and 3 construct a directed graph G whose nodes are literals in the formula, such that A is unsatisfiable iff G has a directed cycle containing some variable and its negation.

Step 1: Since Σ_1^B-Krom is restricted Σ_1^B, the first step is the same as described in the section 3.1.4. That is, convert a Σ_1^B-Krom formula to propositional 2-CNF by making a conjunction of $n_1 \cdot \ldots \cdot n_k$ copies of the formula, one for each $\langle x_1 \ldots x_k \rangle$, and evaluate the terms in each copy on a corresponding value of $\langle x_1 \ldots x_k \rangle$. If a clause evaluates to true due to $\phi_j(\bar{x})$ becoming true, delete the clause. If ϕ_j evaluates to false, then if there are no quantified second-order variables in this formula, the whole formula is false. Otherwise delete ϕ_j from the clause, evaluate $t_j(\bar{x})$ and $t'_j(\bar{x})$ and assign propositional variables to them as follows:

Assign a different propositional variable p_i to every value of a term on a tuple of first-order variables, and make an occurrence of it negated if the corresponding literal was $\neg P$. There are as many variables as there are possible values of t_j's on $\bar{x}$, at most $2m \cdot n_1 \cdot \ldots \cdot n_k$. If two different terms evaluate to the same value on possibly different tuples, they get mapped to the same propositional variable.

Step 2: Now we construct a graph of the resulting propositional formula. The vertices of the graph are the propositional variables and their negations. For every clause $(p_j \vee p'_j)$ create edges $\neg p_j \to p'_j$ and $\neg p'_j \to p_j$.

Step 3: For every propositional variable p_i, check whether both paths from p_i to $\neg p_i$ and from $\neg p_i$ to p_i are in the graph. If there exists p_i for which there are both such paths, then the original formula is unsatisfiable, otherwise satisfiable.

If there is no variable with both paths in the graph, construct the satisfying assignment by repeating the following procedure: pick a variable p_i to which no value has been assigned yet. We know that $p_i \not\to \neg p_i$ or $\neg p_i \not\to p_i$. In the first case, set p_i to true, otherwise set $\neg p_i$ to true; set the opposite literal to false. Now set to true all literals reachable from the literal we set to true (p_i or $\neg p_i$).

Example 5.5.1. Consider the following Σ_1^B-Krom formula

$$\exists P \forall x < 2 \forall y < 1 (x = y \vee P(x) \vee \neg P(y)) \wedge (\neg P(y+1) \vee y > 3) \wedge P(0)$$

After Step 1, the formula becomes

$$\neg P(1) \wedge P(0) \wedge (P(1) \vee \neg P(0)) \wedge P(0),$$

which corresponds to an unsatisfiable propositional formula $\neg p_1 \wedge p_0 \wedge (p_1 \vee \neg p_0)$, omitting the repeating clause (p_0). The formula becomes a graph with four vertices $p_0, p_1, \neg p_0$ and $\neg p_1$, and edges $e_1 = (p_1 \to \neg p_1)$, $e_2 = (\neg p_0 \to p_0)$, $e_3 = (\neg p_1 \to \neg p_0)$ and $e_4 = (p_0 \to p_1)$. This graph contains a cycle

$$p_0 \to_{e_4} p_1 \to_{e_1} \neg p_1 \to_{e_3} \neg p_0 \to_{e_2} p_0.$$

5.5.2 Construction

Here is how we construct a formula equivalent to ϕ^* from (5.2), with occurrences of transitive closure and no second-order quantifiers. If a clause c_j is of the form

$$c_j \equiv (L_j(t_j(\bar{x})) \vee L'_j(t'_j(\bar{x})) \vee \phi_j(\bar{x})),$$

where L_j and L'_j are positive or negative second-order atoms, it translates into two clauses corresponding to the two implications $(\neg L_j \to L'_j)$ and $(\neg L'_j \to L_j)$. There are five pieces of information about each clause: values of $t_j(\bar{x})$ and $t'_j(\bar{x})$, whether L_j and L'_j are P or $\neg P$, and the value of $\phi_j(\bar{x})$. There is a step of transitive closure on the translation of the original clause if one of the two implications $(\neg L_j \to L'_j), (\neg L'_j \to L_j)$ holds.

Introduce for every clause constants z_j, z'_j depending only on the structure of c_j to encode whether L_j, L'_j have negation: ($z_j = 0$ iff $L_j = \neg P$, and $z'_j = 0$ iff $L'_j = \neg P$). Let $\langle u, s\rangle, \langle v, s'\rangle$ be variables used in the transitive closure: a step is $\langle u, s\rangle \to \langle v, s'\rangle$, where u, v correspond to $P(u), P(v)$, and s, s' to the negation parameters. For example, $\langle u, 1\rangle \to \langle v, 1\rangle$ means that the implication $(P(u) \to P(v))$ must hold in order for some clause to be satisfied. Now a translation C_j of c_j becomes

$$\begin{gathered}(\neg\phi_j(\bar{x}) \wedge t_j(\bar{x}) = u \wedge \neg z_j = s \wedge t'_j(\bar{x}) = v \wedge z'_j = s') \\ \vee(\neg\phi_j(\bar{x}) \wedge t'_j(\bar{x}) = u \wedge \neg z'_j = s \wedge t_j(\bar{x}) = v \wedge z_j = s').\end{gathered}$$

The nodes of the graph of the propositional formula are the values of all terms on all tuples of $\bar{x}$. We need to find a value $i < t$, where $t = \max_j(t_j(\bar{n}), t'_j(\bar{n}))$, such that there are chains of implications from $\langle i, 0\rangle$ to $\langle i, 1\rangle$ and from $\langle i, 1\rangle$ to $\langle i, 0\rangle$, corresponding to chains of implications from $\neg p_i$ to p_i and from p_i to $\neg p_i$. Let

$$\psi'(u, s, v, s') \equiv \exists \bar{x} < \bar{n} \bigvee_{j=1}^{m} C_j(\bar{x}).$$

The following formula is equivalent to the negation of ϕ^* from (5.2):

$$\exists i < t(TrCl_{us,vs'}\psi'(u,s,v,s')[i0,i1] \wedge TrCl_{us,vs'}\psi'(u,s,v,s')[i1,i0]). \tag{NegKrom}$$

5.5.3 Proof of correctness

Theorem 5.5.1. *Let $\phi^*(\bar{X},\bar{y})$ be a Σ_1^B-Krom formula. Then there exists a quantifier-free formula ϕ and tuples $\bar{0},\bar{n}$ such that*

$$V\text{-}Krom(TrCl) \vdash \phi^*(\bar{X},\bar{y}) \leftrightarrow \neg TrCl_{\bar{x},\bar{x}'}\phi(\bar{x},\bar{x}')[\bar{0},\bar{n}]$$

Proof. By Theorem 5.4.1 (normal form theorem) it suffices to prove equivalence between ϕ^* in (5.2) and the negation of (NegKrom).

Let $\phi^* = \exists P\forall\bar{x} < \bar{n}\psi(P,\bar{x})$ be the formula (5.2). We need to prove the equivalence

$$\begin{aligned}&\exists P\forall\bar{x} < \bar{n}\psi(P,\bar{x}) \Leftrightarrow\\ &\forall i < t\exists Q[Cond(\psi',Q,\langle t,2\rangle) \wedge (\neg Q(i0,i1) \vee \neg Q(i1,i0))]\end{aligned} \tag{5.3}$$

where

$$\begin{aligned}&Cond(\psi',Q,\langle t,2\rangle) \equiv \forall u,v,w < t\forall s,s',s'' < 2\\ &Q(us,us) \wedge (\psi'(us,vs') \wedge Q(vs',ws'') \rightarrow Q(us,ws''))\end{aligned}$$

First, note that ψ' does not depend on i. The second part is equivalent to

$$\exists Q\ Cond(\psi',Q,\langle t,2\rangle) \wedge \forall i < t(\neg Q(i0,i1) \vee \neg Q(i1,i0))$$

The easy direction of the proof is to show that given a satisfying assignment P to the original formula we can construct Q. We define Q such that $Q(ui,vj)$ holds iff the variable corresponding to ui implies the variable corresponding to vj, under the truth assignment P. Explicitly, we define Q by cases: $Q(u0,v0) \Leftrightarrow (P(v) \rightarrow P(u))$, $Q(u0,v1) \Leftrightarrow (\neg P(u) \rightarrow \neg P(v))$, $Q(u1,v0) \Leftrightarrow (P(u) \rightarrow \neg P(v))$, $Q(u1,v1) \Leftrightarrow (P(u) \rightarrow P(v))$.

It is clear that for Q defined in this fashion $\neg Q(i1,i0) \vee \neg Q(i0,i1)$ for all i, because exactly one of them will be $\top \rightarrow \bot$. If $P(i)$ holds, then $Q(i1,i0)$ is false, otherwise $Q(i0,i1)$ fails. Also, this definition trivially satisfies reflexivity.

To show that Q satisfies step-transitivity, consider $\neg\psi'(us, vs') \vee \neg Q(vs', ws'') \vee Q(us, ws'')$. Suppose that $Q(vs', ws'')$ and $\neg Q(us, ws'')$ hold. In case of $s = s' = s'' = 1$, that corresponds to $P(v) \to P(w)$ and $\neg(P(u) \to P(w))$. That can happen only when $P(u) = \top$, and $P(w) = \bot$. Then $P(v) = \bot$ by $Q(v1, w1)$. It remains to be shown that $\psi'(u1, v1)$ fails. Suppose there exists $\bar{x} < \bar{n}$ and C_j such that $C_j(\bar{x}, u, 1, v, 1)$ holds. The original clause corresponding to C_j is $(\neg P(u) \vee P(v) \vee \phi(\bar{x}))$. Since C_j holds, $\neg\phi(\bar{x})$, and since $P(u) = \top$ and $P(v) = \bot$, this clause is not satisfied by P, contradicting the assumption that P is a satisfying assignment. The cases for other values of s, s', s'' are similar.

The more complicated direction is to construct a satisfying assignment P given Q. Let

$$\begin{aligned} Force(i, s) \equiv Q(i\neg s, is) \vee (\exists j < t \\ Q(j0, j1) \wedge Q(j1, is) \vee Q(j1, j0) \wedge Q(j0, is)) \end{aligned}$$

$Force(i, 1)$ holds if $P(i)$ is directly forced to $\top$, that is, if either $(\neg P(i) \to P(i))$ or $(L(j) \to P(i))$ and $L(j) = \top$, where L is either P or $\neg P$. $Force(i, 0)$ means $P(i) = \bot$. Let $UnForced(i) \equiv \neg Force(i, 0) \wedge \neg Force(i, 1)$.

The hard case is when nothing is forcing $P(i)$ to be $\top$ or $\bot$ except consistency with already assigned values. The idea here is to set the minimal of every set of unassigned variables to $\top$ and make sure that we account for all variables forced to some values by this decision. Since Q contains transitive closure, for all variables i forced by $P(j) = s$ to $P(i) = s'$, $Q(js, is')$. So, we say that i is assigned s if

$$\begin{aligned} Assign(i, s) \equiv \exists j \leq t \forall k \leq t UnForced(j) \\ \wedge (UnForced(k) \to k \geq j) \wedge Q(j1, is). \end{aligned}$$

Now P is defined as follows:

$$P(i) \Leftrightarrow (Force(i, 1) \vee UnForced(i) \wedge Assign(i, 1))$$

Suppose for the sake of contradiction that P is not a satisfying assignment, that is, there exists an assignment $\bar{x}_0$ to $\bar{x}$ and a clause $(L_j(t_j(\bar{x}_0)) \vee L'_j(t'_j(\bar{x}_0)) \vee \phi_j(\bar{x}_0))$ that evaluates to $\bot$ under P. The proof proceeds by cases: L_j and L'_j can be negated literals or not, and in each combination of negations the cases depend on the reason why L_j and L'_j are set to false (forced vs. assigned $P(i)$). $\square$

5.6 The Immerman-Szelepcsényi theorem

In this section we show how to formalize Immerman's proof of closure under complementation of $FO + TC$ (first-order with transitive closure logic). This formalization is split into several subsection. First, we state the theorem and outline the main idea behind the proof. Then, we develop some techniques useful for the proof, namely show how to count in V-Krom. The subsequent section presents the construction, and the section after that contains the correctness proof.

Theorem 5.6.1. *For any $\Sigma_0^B(TrCl^+)$ formula ϕ there is a $\Sigma_0^B(TrCl^+)$ formula ϕ' such that*

$$V\text{-}Krom(TrCl) \vdash \phi \leftrightarrow \neg\phi'$$

Thus, by theorem 5.5.1 and $AxTC$, for any Σ_1^B-Krom formula ϕ^* there exists a Σ_1^B-Krom formula $\phi^{*\prime}$ such that V-Krom $\vdash \phi^* \leftrightarrow \neg\phi^{*\prime}$.

By the Normal Form Theorem (Theorem 5.4.1), it is sufficient to consider formulae ϕ of the form $TrCl_{u,v}\psi(u,v)[0,x]$. Given ϕ of this form, in the proof of Theorem 5.6.1 we construct a formula $\phi' \equiv NegTrCl(\psi, x, n)$ with only positive occurrences of transitive closure operator such that

$$V\text{-Krom} \vdash \neg TrCl_{u,v}\psi(u,v)[0,x] \leftrightarrow NegTrCl(\psi, x, n).$$

The remainder of this subsection contains a brief outline of the proof of Theorem 5.6.1. The actual proof is postponed until subsection 5.6.4.

We associate with the pair ψ, n a graph with n vertices numbered 0 through n-1, and with an edge u, v whenever $\psi(u,v)$ holds. The question becomes the reachability of a vertex numbered x from the vertex numbered 0.

The main idea of Immerman's construction is counting, for every distance $d < n$, the exact number of vertices reachable from 0 in d steps, as well as counting the number of vertices other than x reachable from 0 in d steps. If the two numbers are the same, then x is not reachable from 0 in d steps, and if $d = n - 1$, then x is not reachable from 0 at all, so $(0, x)$ is not in the transitive closure of ψ. In the subsequent formulae, v, v' correspond to the vertices of the graph, c and c' are the values of the counter, and n_d is the number of vertices reachable from 0 in d steps.

The two main formulae used in the construction are $DIST(x,d)$ and $NDIST(x,d;m)$, stating, respectively, that x is reachable from 0 in d steps for $DIST$ and that there are at least m vertices reachable from 0 in d steps not including x for $NDIST$. The final formula $NegTrCl(\psi, x, n)$ states, essentially, that there is some number k of vertices reachable from 0 and the number of vertices reachable from 0 not including x is at least k. The bulk of the proof is an induction argument, showing that for every distance d there is a unique number n_d such that there are exactly n_d vertices reachable from 0 in d steps.

5.6.1 Counting in V-Krom

Since the construction is based on counting, we introduce a notion of "counters" to formalize Immerman's proof.

Definition 5.6.2. A *counter* (transitive closure counter) is a formula of the form $\textsc{Cnt}(vc, v'c') \equiv (c' = c + 1 \wedge \phi(v, v', c) \vee c' = c \wedge \tilde{\phi}(v, v', c))$, where ϕ and $\tilde{\phi}$ are $\Sigma_0^B(\mathrm{TrCl}^+)$. A counter is *fair* if c and c' are not free variables of ϕ and $\tilde{\phi}$. A fair counter is *linear* if, additionally, $\wedge(v' = v + 1)$ is either a part of the counter formula, or the part of both ϕ and $\tilde{\phi}$. In the first case, ϕ and $\tilde{\phi}$ only take one argument, usually v'. A counter is *exact* if $\tilde{\phi} \leftrightarrow \neg\phi$; otherwise a counter is *sloppy*.

Usually we are interested in the value of transitive closure over a counter, with the ranges on vertices and on counter variables as bounds. $TrCl_{vc,v'c'}\textsc{Cnt}(vc, v'c')[yd, ze]$ means that there exists a ϕ-path from y to z of length at least $e - d$. The "at least" part of this statement is due to overlapping ϕ and $\tilde{\phi}$ steps: if there are k steps on which both ϕ and $\tilde{\phi}$ hold, then $TrCl_{vc,v'c'}\textsc{Cnt}(vc, v'c')[yd, ze]$ holds for k consecutive values of e. Since for fair counters the actual values of counter variables do not matter (only the difference does), most counters start at $v = 0, c = 1$ or $c = 0$ and go to $v = n$, with the second boundary value of c being the object of interest.

The simplest counter in Immerman's construction is $\alpha \equiv [(\psi(v, v') \vee v = v') \wedge c' = c + 1]$, with $\phi_\alpha \equiv (\psi(v, v') \vee v = v')$ and $\tilde{\phi}_\alpha \equiv \perp$. It is used to define $DIST(x, d) \equiv TrCl_{vc,v'c'}\alpha(vc, v'c')[00, xd]$. The meaning is that there is a ψ-path from 0 to x of length at most d. The counter α is fair, but not linear and not exact.

All formulae under transitive closure in the Immerman's construction (α, β, γ and δ) are counters. Of them, δ is the only unfair counter, and β and γ are linear, where β is sloppy, and γ can be shown to be exact. The following lemmas are the bulk of the proof:

Lemma 5.6.3. *Let* $\mathrm{LCNT}(vc, v'c')$ *be an* exact *linear counter. Then*

$$V\text{-}Krom \vdash \forall y \leq n \exists! z \leq n TrCl_{vc,v'c'} \mathrm{LCNT}(vc, v'c')[01, yz]$$

Proof. We prove this statement by induction on y. The only two cases to consider for the induction step are whether $\phi(y+1)$ or $\tilde{\phi}(y+1)$ holds; in either case the value of z is clear. □

Lemma 5.6.4. *Let* $\mathrm{LCNT}_1(vc, v'c')$ *and* $\mathrm{LCNT}_2(vc, v'c')$ *be two linear counters with* $\forall v \leq n \phi_1(v) \to \phi_2(v)$ *and* $\tilde{\phi}_2(v) \to \tilde{\phi}_1(v)$*, and let* LCNT_2 *be exact. Then, provably in* V*-Krom,* LCNT_1 *cannot count to a larger value than* LCNT_2*. Moreover, if for some* $v < y$ $\phi_2(v+1) \wedge \neg\phi_1(v+1)$*,*

$$TrCl_{vc,v'c'} \mathrm{LCNT}_2(vc, v'c')[01, yd] \to \neg TrCl_{vc,v'c'} \mathrm{LCNT}_1(vc, v'c')[01, yd];$$

otherwise (that is, if $\forall v < y(\phi_2(v+1) \to \phi_1(v+1))$*),*

$$TrCl_{vc,v'c'} \mathrm{LCNT}_2(vc, v'c')[01, yd] \to TrCl_{vc,v'c'} \mathrm{LCNT}_1(vc, v'c')[01, yd]$$

Proof. The proof is by induction on y. We omit details. □

5.6.2 Properties of the distance predicate

Since the Immerman's construction relies mainly on the properties of the $DIST$ function and its relationship with the transitive closure operator, we prove the basic properties of $DIST$ before proceeding to the correctness proof for the construction. The main lemma in this proof is lemma 5.6.10 stating that there is a path from x to y in a graph if and only if there is a path from x to y of length at most $n-1$. To simplify the proofs, we will use a more general version of $DIST$.

Definition 5.6.5. We define a distance predicate $Dist_\phi(x, y, d)$ which holds on (x, y, d) iff there is a path from x to y of length at most d.

$$Dist_\phi(x, y, d) \equiv TrCl_{vc,v'c'} \psi(vc, v'c')[x0, yd],$$

where $\psi(vc, v'c') \equiv ((\phi(v, v') \vee v' = v) \wedge c' = c+1)$.

Note that $Dist_\phi$ is a fair counter, with the first formula being ψ and the second $\perp$. Now $DIST$ used in the construction below is just

$$DIST(x, d) \equiv Dist_\psi(0, x, d).$$

Lemma 5.6.6. *If there is a path from x to z of length $\leq d+1 < n$, then either there is a successor y to x such that $\phi(x,y)$ and a path of length $\leq d$ from y to z, or a path of length $\leq d$ from x to z.*

$$V\text{-}Krom \vdash \exists y < nDist_\phi(y,z,d) \wedge (\phi(x,y) \vee x = y) \leftrightarrow Dist_\phi(x,z,d+1)$$

Proof. Open transitive closure by AxTC. After easy adjustments,

$$\begin{aligned} &\exists R\ Cond(\psi, R, (n,n)) \wedge \neg R(x0, z\ (d+1)) \rightarrow \\ &\exists Q\ Cond(\psi, Q, (n,n)) \wedge \\ &\qquad\qquad \forall y < n \neg Q(x0, yd) \vee (\neg(\phi(y,z,d) \wedge c' = d+1) \end{aligned}$$

Take $Q = R$. Suppose there is $y < n$ such that $Q(x0, yd) \wedge \phi(y,z,d) \wedge c' = d+1$. Then by transitivity $Q(x0, z\ d+1)$ holds, contradicting the assumption.

For the other direction, take $R(vc, v'c') = Q(vc, v'c') \wedge (c \leq c' \wedge (c' \leq d \vee d+1 \leq c))$ It is easy to check that this R satisfies $Cond((\phi \wedge c' = c+1), R, (n,n))$ and does not contain $(x0, z(d+1))$. $\square$

We can state a version of Lemma 5.6.6 in which the edge is added at the end of a path, rather than at the beginning.

Lemma 5.6.7.

$$V\text{-}Krom \vdash \exists y < nDist_\phi(x,y,d) \wedge (\phi(y,z) \vee y = z) \leftrightarrow Dist_\phi(x,z,d+1)$$

Proof. The proof is very similar to the proof of Lemma 5.6.6, except we use the the right-side axiom of transitive closure $TrCl^r$, and refer to Lemma 5.3.5. $\square$

Note that the way the proof of Lemma 5.6.6 is structured relies on the fact that $d+1 < n$. However, to prove Lemma 5.6.10 below, we should be able to handle the case of adding an edge to a path that is of length $n-1$ already: that is, to prove that if there is a path of length (at least) $n-1$ from x to y and an edge from y to z, then there is a path from x to z. But a path from x to z cannot contain more than $n-1$ edges, because there are only n nodes. The distance function would fail to count such step because of the restriction $c' < n$. So we need to prove that if there is a path from x to y at all, then there is a path of length $n-1$.

Lemma 5.6.8. *Let x and y be arbitrary vertices of a graph. If the shortest ϕ-path from x to y is of length $n-1$, then every vertex v occurs on that path. That is,*

$$V\text{-}Krom \vdash Dist_\phi(x,y,n-1) \wedge \neg Dist_\phi(x,y,n-2) \rightarrow (\forall z < n(z \neq y \rightarrow Dist_\phi(x,z,n-2)))$$

Proof. The proof is by induction on the size n of a graph. The main idea of the proof is that the only situation in which a path from x to y is of length $n-1$ is when the graph is "layered", with one vertex per layer, x in the first layer, y in the last layer, and in each layer all out-edges are either to previous layer, or to the single next layer.

The base cases $n = 0$, $n = 1$ and $n = 2$ are trivial. For the induction step, assume that the claim holds for graphs on n vertices. We consider now a graph on $n+1$ vertices, with vertices labeled $\{0, \ldots, n\}$. Note that relabeling the vertices does not change the properties of the graph, and the claim is stated for every pair (x, y) of vertices. That is, let ϕ' be the same as ϕ except in the corresponding graph the vertices y and n are interchanged. Then, V-Krom $\vdash Dist_\phi(x,y,d) \leftrightarrow Dist_{\phi'}(x,n,d)$..

Now by Lemma 5.6.7 $\exists z < n Dist_{\phi'}(x,z,n-1) \wedge \phi'(z,n)$, since by assumption $\neg Dist_{\phi'}(x,n,n-1)$ and thus $x \neq n$. By the induction hypothesis, $\forall u < n(u \neq z \rightarrow Dist_{\phi'}(x,z,n-2))$. We know that $\phi'(u,n)$ does not hold for any $u \neq z$: in that case, by Lemma 5.6.7, $Dist_{\phi'}(x,n,n-1)$ would hold. Therefore, since $Dist_{\phi'}(x,n,n)$, $\phi'(z,n)$ holds. Thus, every vertex v occurs on the path from x to n, since every vertex $u < n$ occurs on the path from x to z and vertex z occurs on the path from x to n. Relabeling n and y back, we get $\forall z < n+1(z \neq y \rightarrow Dist_\phi(x,z,n))$, concluding the proof. □

Corollary 5.6.9. *Extending lemma 5.6.6,*

$$V\text{-}Krom \vdash \exists y < n Dist_\phi(y,z,d) \wedge \phi(x,y) \leftrightarrow Dist_\phi(x,z,\min\{d+1,n-1\})$$

Proof. For the case of $d+1 < n$, the proof follows from lemma 5.6.6. If the minimal such d is $n-1$, then the statement follows from the lemma 5.6.8: since z is on the path from x to y, $\exists d' < n Dist_\phi(x,z,d')$. And since $Dist_\phi(x,z,d') \rightarrow Dist_\phi(x,z,d'+d'')$ for any d'' such that $d''+d' < n$, $Dist_\phi(x,z,n-1)$ holds. □

Finally, we establish the relation between transitive closure over ϕ and $Dist_\phi$.

Lemma 5.6.10. *If there is a ϕ-path from x to y, then there is a ϕ-path from x to y of length $\leq n-1$. That is,*

$$TrCl_{u,v}\phi(u,v)[x,y,n] \leftrightarrow Dist_\phi(x,y,n-1)$$

Proof. As usual, negate both sides and open them by AxTC. Let $R(v, v')$ be the relation satisfying $Cond(\phi, R, n)$ and not containing (x, y) Define $Q(v, c, v', c') \equiv c < c' \wedge R(v, v')$. Then $Cond(\psi, Q, \langle n, n \rangle)$ and $\neg Q(x, 0, y, n-1)$.

For the other direction, define $R(v, v') \equiv Q(v, 0, v', n-1)$. For the transitivity proof, note that either $\exists d < n-1 Q(v, 0, v', d)$ or $Q(v, 0, v', n-1)$ and $\neg Q(v, 0, v', n-2)$. In the first case, the adding one step just increments d; in the second case, we need to appeal to corollary 5.6.9. □

5.6.3 Immerman's construction

The body of the proof of Immerman's theorem is by induction on the number of steps d of the outermost counter (that is, on the length of paths starting at 0). The formula γ defining the value of n_d for every step is a linear counter with $\phi_\gamma \equiv DIST(v', d+1)$ and

$$\tilde{\phi}_\gamma \equiv \forall z < n(NDIST(z, d; m) \vee (z \neq v' \wedge \neg\psi(z, v'))).$$

Intuitively, γ increments its counter variable c for every v reachable in $d+1$ steps and does not increment the counter for unreachable (in $d+1$ steps) vertices, under the assumption that there are at least m vertices reachable in d steps. The induction statement is that for a step d, γ is an exact counter giving a unique value n_d and $\forall x < n(NDIST(x, d; n_d) \leftrightarrow \neg DIST(x, d))$. The first part is proven by using Lemma 5.6.3 with LCNT $= \gamma$; the second part by applying Lemma 5.6.4 with LCNT2 $= \gamma$ and LCNT1 being the counter formula of $NDIST$, β, with $\phi_\beta \equiv DIST(v', d) \wedge v \neq x$ and $\tilde{\phi}_\beta \equiv \top$.

For $d = n-1$ this statement implies that if there are $k = n_{n-1}$ vertices reachable from 0 and by the formula $NDIST(x, n-1; n_{n-1})$ the vertex x is not one of them, then $\neg DIST(x, n-1)$. The proof is completed by showing that $DIST(x, n-1) \leftrightarrow TrCl_{u,v}\psi(u, v)[0, x]$.

The following is a formalization in V-Krom of the construction from [Imm99]. First, note that by theorem 5.4.1 we can assume that $\phi \equiv TrCl_{\bar{u},\bar{v}}\psi(\bar{u}, \bar{v})[\bar{0}, \bar{n}]$ for some $\bar{n}$ and quantifier-free ψ , and that all variables in $\bar{u}, \bar{v}$ are bounded by $\bar{n}$. We will write, for notational simplicity, $\bar{u}, \bar{v}, \bar{n}$ as single variables. Also, all variables are bounded by n in the proof; in case of $\bar{u}$ and $\bar{v}$ being vectors, take u and v be the result of a pairing function applied to $\bar{u}$ and $\bar{v}$, and take n to be $\langle \bar{n} \rangle$.

As mentioned before, we are constructing a formula $NegTrCl(\psi, x, n)$ with only positive occurrences of the transitive closure operator such that

$$V\text{-Krom} \vdash \neg TrCl_{u,v}\psi(u,v)[0,x] \leftrightarrow NegTrCl(\psi,x,n).$$

The following two functions encode the notions of reachability and unreachability:

$$DIST(x,d) \equiv TrCl_{vc,v'c'}\alpha(vc,v'c')[00,xd]$$

where $\alpha \equiv (\psi(v,v') \vee v = v') \wedge c = c+1$, and

$$NDIST(x,d;m) \equiv TrCl_{vc,v'c'}\beta(vc,v'c';d)[01,nm]$$

where

$$\begin{aligned}\beta(vc,v'c';d,x) \equiv (&0 \neq x \wedge v' = v+1\\ &\wedge(c' = c \vee (c' = c+1 \wedge DIST(v',d) \wedge v' \neq x)))\end{aligned}$$

$DIST(x,d)$ states that there is a path from 0 to x of length at most d, and $NDIST(x,d;m)$ states that there exist at least m vertices other than x reachable from 0 in d steps. Now define the function γ that is used to count the exact number of vertices

$$\begin{aligned}&\gamma(v,c,v',c';d,m) \equiv (v' = v+1\\ &\quad\wedge((c' = c+1 \wedge DIST(v',d+1)) \vee (c' = c\\ &\qquad\wedge \forall z < n(NDIST(z,d;m) \vee (z \neq v' \wedge \neg\psi(z,v'))))))\end{aligned}$$

and

$$\delta(d,m,d',m') \equiv (d' = d+1 \wedge TrCl_{vc,v'c'}\gamma(vc,v'c';d,m)[01,nm']$$

So, γ counts the total number of vertices reachable in $d+1$ step, given the total number m of vertices reachable in d steps. The role of δ is just the transition from step d to step $d+1$. Finally,

$$\begin{aligned}&\neg TrCl_{u,v}\psi(u,v)[0,x] \equiv \exists k < n\\ &\quad(TrCl_{dm,d'm'}\delta(dm,d'm')[01,n-1,k] \wedge NDIST(x,n-1;k))\end{aligned} \tag{NegTC}$$

5.6.4 Proof of correctness of the construction

We want to prove the statement NegTC. For that, we will show that for every d there exists n_d that is the number of vertices reachable from 0 in d steps. Moreover, for that n_d it is true for every x that $NDIST(x,d;n_d)$ holds iff $\neg DIST(x,d)$. More formally, the bulk of the proof is the following induction statement:

Lemma 5.6.11 (Correctness lemma). *V-Krom proves that $\forall d \leq n-1$,*

1) $\exists n_d \leq n TrCl_{me,m'e'}\delta(me, m'e')[01, dn_d]$
2) $TrCl_{me,m'e'}\delta(me, m'e')[01, dn_d] \rightarrow \forall x < n(NDIST(x, d; n_d) \leftrightarrow \neg DIST(x, d))$

Given the result of the lemma 5.6.11, for $d = n-1$ we get that

$$\exists n_{n-1} \leq n TrCl_{me,m'e'}\delta(me, m'e')[01, (n-1)n_{n-1}]$$

and for that n_{n-1} the equivalence $\neg DIST(z, n-1) \leftrightarrow NDIST(z, n-1; n_{n-1})$ holds $\forall z < n$. Now, setting $k = n_{n-1}$, and showing, by the lemma 5.6.10 below, that $DIST(z, n-1) \leftrightarrow TrCl_{u,v}\psi(u, v)[0, z, n]$, we obtain the statement (NegTC).

Proof of lemma 5.6.11. First, assume that $n > 1$. The cases of $n = 0$ and $n = 1$ are trivial. Now the proof is by induction on d.

Base case. For $d = 0$, 0 is the only vertex reachable from 0 in 0 steps, so $n_0 = 1$. By reflexivity of transitive closure, $TrCl_{me,m'e'}\delta(me, m'e')[01, 01]$ holds. Since any δ-step requires $d' = d+1$, $n_0 = 1$ is unique.

Now, for $x = 0$, $DIST(x, d)$ by reflexivity and $\neg NDIST(0, d; m)$ for any d, m since for any $v, c, v'c'$ $\neg\beta(vc, v'c'; d, 0)$. So let $x \neq 0$. Then, in the case of $d = 0$, $\neg DIST(x, d)$. Because of that, for all $v \leq n-1$, $\beta(v1, (v+1)1)$ holds. Therefore, $NDIST(x, 0; 1)$ and thus $NDIST(x, 0; 1) \leftrightarrow \neg DIST(x, 0)$.

Induction step. Suppose that

$$\exists n_d \leq n TrCl_{me,m'e'}\delta(me, m'e')[01, dn_d],$$

and

$$TrCl_{me,m'e'}\delta(me, m'e')[01, dn_d] \rightarrow \forall x < n(NDIST(x, d; n_d) \leftrightarrow \neg DIST(x, d))$$

Then V-Krom proves the following:

1) $\exists n_{d+1} \leq n TrCl_{me,m'e'}\delta(me, m'e')[01, d+1\, n_{d+1}]$
2) $TrCl_{me,m'e'}\delta(me, m'e')[01, (d+1)n_{d+1}] \rightarrow$
$$\forall x \leq n(NDIST(x, (d+1); n_{d+1}) \leftrightarrow \neg DIST(x, d+1))$$

We start by showing the first part, that is, the existence of n_{d+1}. By the lemma 5.6.6, setting $y = n_d$ and $z = n_{d+1}$, it is sufficient to show that $TrCl_{vc,v'c'}\gamma(vc, v'c'; d, n_d)[01, nn_{d+1}]$ holds.

It is clear that γ is a linear counter. We want to prove that γ is an exact counter. For that, we need to show that $\neg DIST(v, d+1) \leftrightarrow \forall z < n(NDIST(z, d; n_d) \vee (z \neq v' \wedge \neg\psi(z, v')))$. By induction hypothesis, $NDIST(z, d; n_d) \leftrightarrow \neg DIST(z, d)$. By the contrapositive of lemma 5.6.6, $\neg DIST(v, d+1) \leftrightarrow \forall z < n(\neg DIST(z, d) \vee (z \neq v' \wedge \neg\psi(z, v')))$. Therefore, γ is an exact counter, so we can apply lemma 5.6.3 which gives both existence and uniqueness of n_{d+1}.

Now we need to show that for that n_{d+1}, $\forall x < nNDIST(x, d{+}1; n_{d+1}) \leftrightarrow \neg DIST(x, d{+}1)$. As noted in the base case, the case $x = 0$ is trivial. Let $x \neq 0$. First, note that $\phi_\beta \rightarrow \phi_\gamma$ (since β has an additional restriction of $v \neq x$), and $\psi_\gamma \rightarrow \psi_\beta$ (since $\psi_\beta \equiv \top$). As shown before, γ is an exact counter, and β is a sloppy linear counter.

Suppose that $DIST(x, d{+}1)$ holds. Then $\gamma((x{-}1)0, x1)$ holds, but $\beta((x{-}1)0, x1)$ does not. Then, by lemma 5.6.4, $TrCl_{vc,v'c'}\gamma(vc, v'c')[01, nn_{d+1}] \rightarrow \neg TrCl_{vc,v'c'}\beta(vc, v'c')[01, nn_{d+1}]$. In this case, $NDIST(x, d+1; n_{d+1})$ is false.

Now suppose that $DIST(x, d+1)$ does not hold. Then $\forall v < n\phi_\gamma(v) \leftrightarrow \phi_\beta(v)$. Then, again by lemma 5.6.4, $TrCl_{vc,v'c'}\gamma(vc, v'c')[01, nn_{d+1}] \rightarrow TrCl_{vc,v'c'}\beta(vc, v'c')[01, nn_{d+1}]$, so $NDIST(x, d+1; n_{d+1})$ is true. $\square$

Proof of theorem 5.6.1. Now that we have the lemma 5.6.11, the proof of the theorem 5.6.1 follows easily. By the first part of lemma 5.6.11, for every d $\delta(dn_d, (d{+}1)n_{d+1})$ holds. By induction on d and properties of transitive closure, $\exists n_{n-1}TrCl_{dm,d'm'}\delta(dm, d'm')[01, (n-1)\ n_{n-1}]$. Since by definition of δ n_{n-1} satisfies the corresponding $TrCl_{vc,v'c'}\gamma(vc, v'c')[01, nn_{n-1}]$, by the second part of lemma 5.6.11 $NDIST(x, n-1; n_{n-1}) \leftrightarrow \neg DIST(x, n-1)$. Since by lemma 5.6.10 with $x = 0$, $DIST(x, n-1) \leftrightarrow TrCl_{u,v}\psi(u, v)[0, v]$, $NDIST(x, n-1; n_{n-1}) \leftrightarrow \neg TrCl_{u,v}\psi(u, v)[0, v]$, completing the proof. $\square$

5.7 Definability in V-Krom

In this section we use Definability Theorem 3.3.13 to prove that V-Krom indeed captures NL tightly.

Theorem 5.7.1. *A predicate $R(\bar{x}, \bar{Y})$ is Δ_1^B-definable in V-Krom iff it is in* NL.

We define the function class FNL associated with NL according to definition 3.3.1. These functions can be defined by Σ_1^B-Krom formulae following the definition 3.3.5.

Lemma 5.7.2. *Let ϕ be a Σ_0^B formula with possible occurrences of string and number function symbols from the definition 3.3.5 with $\Phi = \Sigma_1^B$-Krom. Then there exists a Σ_1^B-Krom formula with no occurrences of function symbols that is provably in V-Krom equivalent to ϕ.*

Proof. Structural induction on ϕ, using Theorems 5.4.1, 5.5.1, and 5.6.1. □

Definition 5.7.3. A string function $F(\bar{x}, \bar{Y})$ is Σ_1^B-definable in V-Krom iff there is a Σ_1^B formula ϕ such that

$$Z = F(\bar{x}, \bar{Y}) \leftrightarrow \phi(\bar{x}, \bar{Y}, Z)$$

and

$$V\text{-Krom} \vdash \forall \bar{x} \forall \bar{Y} \exists ! Z \phi(\bar{x}, \bar{Y}, Z)$$

Similarly for number functions.

Lemma 5.7.4. *The class of formulae Σ_1^B-Krom is strongly closed and constructive.*

Proof. First we argue that Σ_1^B-Krom is strongly closed. By the theorem 5.4.1, every formula with nested occurrences of the transitive closure (not necessarily positive) can be converted to a formula with a single outermost occurrence of the transitive closure. By Lemma 5.7.2 above, V-Krom $\vdash \Sigma_0^B(\Sigma_1^B\text{-Krom}) \equiv \Sigma_1^B$-Krom. Therefore, FNL is closed under composition and $\mathtt{AC}^0$ reductions.

For the constructiveness property, we again refer to the transitive closure operator. Define a transitive closure function $TC_\phi(\bar{X}, \bar{y}, n)(a, b)$ by setting its bitgraph to be AxTC. The existence and uniqueness of the graph of this function is proven by comprehension over the negation of AxTC, using the fact that it is Δ_1^B-definable in V-Krom by Theorem 5.6.1.

Suppose that V-Krom $\vdash \exists P \forall \bar{x} < \bar{t} \psi(\bar{x}, P, \bar{a}, \bar{Y})$. Recall the construction from Section 5.5.3 of a satisfying assignment P from the transitive closure witness Q. We use exactly the same construction, replacing Q with the actual transitive closure defined above, that is, using TC_ψ and $TC_{\tilde{\psi}}$ respectively instead of Q and $\neg Q$ in the formula (5.3). By Lemma 5.7.2, a Σ_0^B formula with occurrences of TC is equivalent to a Σ_1^B-Krom formula, which in turn defines an FNL function $F_{wc}(a, b, y, \bar{a}, \bar{Y})$, which is a witnessing function for P. □

Theorem 5.7.5. *A function (string or number) is Σ_1^B-definable in V-Krom iff it is in FNL.*

Proof. This result follows immediately from Corollary 5.7.4 and Theorem 3.3.13. □

5.8 V-Krom is finitely axiomatizable.

Since it is possible to encode Σ_1^B-Krom satisfiability as a Σ_1^B-Krom formula, we can show that V-Krom is finitely axiomatizable in a similar fashion to the proof that V_1-Horn is finitely axiomatizable.

By theorem 3.6.1, V^0, is finitely axiomatizable. Since the Σ_0^B comprehension scheme is provable in V-Krom, V-Krom can be viewed as V^0 extended by the Σ_1^B-Krom comprehension axiom scheme. If we can show that finitely many occurrences of Σ_1^B-Krom comprehension are sufficient, we prove that V-Krom is finitely axiomatizable.

In proving Theorem 5.5.1 we showed that every Σ_1^B-Krom formula $\phi^*(\bar{X}, y, \bar{a})$ is provably equivalent to a negated transitive closure. This is done by showing that ϕ^* is provably equivalent to the negation of the formula (NegKrom), which involves the transitive closure of a formula $\psi'(u, s, v, s')$. Inspection of the latter argument shows that this equivalence can be proved in V^0. Notice that ψ' is a Σ_0^B formula, and has free variable parameters $y, \bar{a}, \bar{X}$, which we will indicate by writing $\psi'(u, s, v, s', y, \bar{a}, \bar{X})$. We can use Σ_0^B comprehension to define a string variable $E(u, s, v, s', y)$, which for fixed $\bar{X}$ and $\bar{a}$ a codes the values of ψ'. Thus

$$V^0 \vdash \exists E \forall u, v < t \forall s, s' < 2 \forall y < b[E(u, s, v, s', y) \leftrightarrow \psi'(u, s, v, s', y, \bar{a}, \bar{X}).]$$

The proof of Theorem 5.5.1 shows that $\phi^*(\bar{X}, y, \bar{a})$ is equivalent to the RHS of (5.3), and this is provable in V^0. Let $\Psi(y, E)$ be the result of replacing each occurrence of ψ' in the RHS of (5.3) by E. Then it suffices to add the following single comprehension axiom for Ψ to V^0 to get V-Krom :

$$\exists Z \leq b \forall y < b(Z(y) \leftrightarrow \Psi(y, E)).$$

This is because the comprehension axiom for $\phi^*(\bar{X}, y, \bar{a})$ follows from this one comprehension axiom by reasoning in V^0, and this axiom is the same for every Σ_1^B-Krom formula ϕ^*.

5.9 Equivalence with Nguyen's VNL

5.9.1 Definition of VNL

The following material is based on [NKC04]. In that paper, Nguyen defines a several systems of arithmetic by adding variants of a reachability axiom to V^0. Let the following relation express reachability:

Definition 5.9.1 (LC)**.**

$$\begin{aligned} LC(a,E,Z) \equiv & Z(0,0) \wedge \forall i < a \neg Z(0,i) \\ & \wedge \forall k, i < a,\ Z(k+1,i) \leftrightarrow [Z(k,i) \vee \exists j < a,\ E(j,i) \wedge Z(k,j)]. \end{aligned}$$

The relation LC states that $Z(k,i)$ is true iff there is a path from 0 to i of length at most k. Thus, LC plays a role very similar to the $DIST$ function, except the calculation is exact.

Now define a system of arithmetic VNL by adding an axiom stating existence of Z unconditionally satisfying LC :

$$\mathbf{VNL} \equiv V^0 + \exists Z < \langle a,a\rangle LC(a,E,Z).$$

5.10 Equivalence between VNL and V-Krom

Here we show that V-Krom is equivalent to VNL. Since V-Krom extends V^0, it is sufficient to show that VNL proves Krom comprehension scheme and that V-Krom proves $\exists Z < \langle a,a\rangle LC(a,E,Z)$.

The main idea behind the proofs is that the $DIST_\phi(x,d)$ function, stating that there is a path from 0 to x in a n-node graph with ϕ defining the edge relations, is definable (and its properties provable) in V-Krom. $DIST$ is very similar to LC, so $DIST$ can be used to define LC in V-Krom. For the other direction, we show how to define transitive closure relation in VNL, and then refer to the results in [CK04] for the proof that transitive closure relation can be used to prove Krom comprehension.

Lemma 5.10.1. *V-Krom proves* $\exists Z < \langle a,a\rangle LC(a,E,Z)$.

Proof. Following Immerman's proof of closure of TC^+ under complementation and its formalization in [CK04], define

$$DIST(x,d,E,n) \equiv TrCl_{vc,v'c'}((E(v,v') \vee v = v') \wedge c' = c+1)[00, xd; n].$$

That is, the pair $(00, xd)$ is in the transitive closure of the graph of E on the first n vertices iff $DIST(x, d, E, n)$ holds. We will omit E and n when it is clear from the context. To simplify the notation, let $\alpha(v, c, v', c', E) \equiv ((E(v, v') \vee v = v') \wedge c' = c + 1)$, so $DIST$ is a transitive closure over α.

To show that $\exists Z\langle a, a\rangle LC(a, E, Z)$, define Z as $\forall i, k < aZ(k, i) \leftrightarrow DIST(i, k, E, a)$. Since V-Krom proves comprehension over $\Sigma_0^B(\Sigma_1^B\text{-}Krom)$ formulae, it proves the existence of such Z. It remains to show that this Z satisfies $LC(a, E, Z)$. We will show it by induction on k.

Base case: $Z(0, i) \equiv DIST(i, 0, E, a)$. By reflexivity of transitive closure, the only i for which $DIST(i, 0, E, a)$ holds is 0. Therefore, $Z(0, 0)$ and $\forall i < a \neg Z(0, i + 1)$.

Induction step: $\forall i < a, Z(k + 1, i) \leftrightarrow (Z(k, i) \vee \exists j < aE(j, i) \wedge Z(k, j))$. The $\leftarrow$ direction is simple: $DIST(i, k) \implies DIST(i, k + 1)$ because of the $v = v'$ part of the definition of $DIST$, and $\exists j < aE(j, i) \wedge DIST(j, k))$ implies $DIST(i, k + 1)$ by setting $v = j, v' = j$ and referring to a lemma in [CK04] that if there is a path of length d to j and an edge $E(j, i)$, then there is a path of length $d + 1$ to i.

For the other direction, recall that transitive closure in V-Krom is defined by the formula

$$TrCl_{x,y}\phi(x, y)[a, b, n] \leftrightarrow \forall R(Cond(\phi, R, n) \to R(a, b)),$$

where transitivity condition $Cond$ is

$$Cond(\phi, R, n) \equiv \forall x, y, z < n(R(x, x) \wedge (\phi(x, y) \wedge R(y, z) \to R(x, z))).$$

The statement that we need to prove is, writing out $DIST$ according to this definition and negating both sides,

$$\begin{aligned}&\exists Cond(R, \alpha, \langle a, a\rangle) \wedge \neg R(0, 0, i, k) \wedge (\forall j < a \neg E(j, i) \vee \neg R(0, 0, j, k))\\ &\quad \to \exists Q Cond(Q, \alpha, \langle a, a\rangle) \wedge \neg Q(0, 0, i, k + 1).\end{aligned}$$

Construct $Q(v, c, v', c')$ to coincide with R on all $c' \leq k$, $v, v', c < a$. For $c' > k$, set $Q(v, c, v', c') \equiv \top$ for all $v, c, v'c'$ except $v' = i, c' = k + 1$. This definition of Q satisfies $Cond(Q, \alpha, \langle a, a\rangle)$, and does not contain $Q(0, 0, i, k + 1)$ by construction. $\square$

Since V-Krom is axiomatizable by the finite set of axioms of V^0 together with the comprehension axiom of the form $\exists Z \forall x, y < b(Z(y) \leftrightarrow \Psi(y, E))$, where Ψ encodes transitive closure over graph E representing a formula ϕ, it is sufficient to show that VNL proves this single comprehension axiom. Moreover, Ψ is based on graph reachability in E, which simplifies the proof.

Lemma 5.10.2. $VNL \vdash \exists Z \forall y < b(Z(y) \leftrightarrow \Psi(y, E))$.

Proof. The finite axiomatizability proof of V-Krom describes how to replace comprehension over a Krom formula ϕ with a transitive closure over a graph E encoding ϕ. So the only part that needs to be proven is that $\exists Z < \langle n, n\rangle LC(n, E, Z)$ can be used to define $TrCl_{x,y}E(x, y)[a, b, n]$.

First, note that VNL proves uniqueness of Z satisfying $LC(n, E, Z)$. The proof is by induction on indices of Z. It remains to be shown how to construct a relation R satisfying the axiom of transitive closure. The relation LC is only concerned with paths starting at 0, whereas for the transitive closure we need paths starting at arbitrary vertices. Given a graph E, define a family of graphs E_i, for $i = 0$ to $n - 1$ as follows:

$$E_i = \{(0, i + 1)\} \cup \{(x + 1, y + 1) | E(x, y)\}$$

That is, all indices are increased by 1, and an edge added from 0 to the vertex corresponding i from which we want to find a path. Clearly, there is a path from i to any j in the original graph E iff there is a path from 0 to $j + 1$ in E_i. VNL proves existence of $Z_1 \ldots Z_n$ satisfying $LC(a, E_i, Z_i)$ for all $i < n$.

Now define $\forall i, j < nR(i, j) \leftrightarrow Z_i(n, j + 1)$. The existence of such R is given by Σ_0^B comprehension. The reflexivity condition $R(i, i)$ holds: either $i = 0$, then $Z_i(0, 0)$ for any i, or $Z_i(1, i+1)$ holds by $Z_i(0, 0) \wedge E_i(0, i+1)$ and so is $Z_i(n+1, i+1)$ by induction. For the transitivity condition, suppose that $R(x, y) \wedge E(y, z)$. Then $Z_x(n + 1, y)$, and there exists $k < n + 1$ such that $Z_x(k + 1, y)$. Consider minimal such k. If $k < n$, then setting $j = y$ we get $Z_x(k + 1, z)$, and thus $Z_x(n + 1, z)$, implying $R(x, z)$.

Suppose now that $k = n$. Then we want to show that there exists $k' < n$ such that $Z_x(k', z)$. That is, if the only path from 0 to y in E_x has length n, then every vertex including z occurs on this path. This proof is by induction on n.

Lemma 5.10.3. *VNL proves that if the shortest path from* 0 *to* j *in* E *is of length* $n-1$, *then every vertex* i *occurs on the path from* 0 *to* j.

Proof. This lemma is similar to Lemma 5.6.8. However, the proof is carried out in VNL rather than in V-Krom, and so is slightly different.

The proof is by induction on n. The base case is $n = 1$. Then there is just one vertex $i = j = 0$, and every vertex is on the path. If $n = 2$, then there is a path of length 0 to

0, so this path is of length $0 < n-1$. There is a path of length $n-1=1$ from vertex 0 to vertex 1, and both vertices occur on this path.

Now suppose that the statement is true for some n. That is, of every graph E of size n, every vertex $j < n$, if the shortest path from 0 to j is of length $n-1$, then every vertex $i < n$ occurs on that path. More precisely, a shortest path from 0 to j is of length $n-1$ iff the smallest k for which $Z(k,j)$ holds is $k = n$. Consider a graph E' on $n+1$ vertices in which the shortest path from 0 to j is of length n. Without loss of generality, we can relabel vertices so that $j = n$: j cannot be 0 and otherwise exchanging j^{th} and n^{th} vertices does not change the properties of the graph. It follows from the induction hypothesis that there is just one i such that there is an edge (i,j) in E', and that i is such that the shortest path from 0 to i in E' is of length $n-1$. If there were two edges leading into j, then one would come from the vertex on the path to i, by induction hypothesis, which would result in a shorter path to j. So $E' = E \cup (i,n)$, where vertex n is not in E. Now every vertex of E, including i, is on the path from 0 to i by induction hypothesis, and so is on the path to j, since the only path to j leads through i. $\square$

Now by the lemma 5.10.3 vertex z is on the path from 0 to y, and thus there is a path from 0 to z in E_x. Therefore, there is a path from x to z in E, and $R(x,z)$ holds. Thus, R is transitive.

It remains to show that R is minimal. Suppose that there exists Q such that $Cond(Q,E,n)$ and for some $x,y < n$, $R(x,y)$ but $\neg Q(x,y)$. Consider Z_x. As stated before, $Z_x(n+1,y)$ holds iff there is a path from x to y in E. Consider the smallest k such that $Z_x(k+1,i+1)$ for some i, but not $Q(x,i)$. Since $Z_x(k+1,i+1)$ holds, either $Z(k,i+1)$ holds or $\exists j < n(Z(k,j+1) \wedge E(j+1,i+1))$. The first case is not possible by the minimality of k, and the second case violates the transitivity condition: by the minimality of k, $Q(x,j)$ should hold, but then $Q(x,j) \wedge E(j,i)$ would give $Q(x,i)$. Therefore, R is the transitive closure of E. $\square$

Chapter 6

A weakly closed system: symmetric logspace

The complexity class Symmetric Logspace ($\mathtt{SL}$) is not as well-known as the classes considered before. This class was first mentioned by Jones, Lien and Laaser [JLL76] as "a class of problems between $\mathtt{L}$ and $\mathtt{NL}$". In that paper, they show the equivalences between undirected graph accessibility, non-bipartiteness, unsatisfiability of 2CNF with exclusive or $\oplus$ instead of $\vee$, and several other problems. Some of their equivalences we formalize later in this section. They did not use the term "symmetric logspace": the term "symmetric computation" and the definition of $\mathtt{SL}$ via that notion first appeared in the paper by Lewis and Papadimitriou [LP82]. Schaefer [Sch78] indirectly mentions both $\mathtt{SL}$ and co-$\mathtt{SL}$: two of his classes are "problems logspace-reducible to undirected graph reachability" and "problems reducible to testing bipartiteness". The proof that $\mathtt{SL}$ is closed under complementation was discovered fairly recently by Nisan and Ta-Shma [NTS95], and needed techniques different from inductive counting used in proofs that classes like $\mathtt{NL}$ and $\mathtt{SAC}^1$ are closed under complementation. In the same paper where he defined $SO\exists$-Krom and $SO\exists$-Horn, Grädel gave a descriptive complexity characterization of $\mathtt{SL}$ by symmetric $SO\exists$-Krom formulae over successor structures.

After the examples of V_1-Horn and V-Krom, it would seem that creating a system of arithmetic for the class $\mathtt{SL}$ is straightforward: this is one of Schaefer's classes, it is closed under complementation, and it has a characterization very similar to $SO\exists$-Krom. However, it seems that the proof of the closure of $\mathtt{SL}$ under complementation as presented in Nisan and Ta-Shma's work [NTS95], uses techniques inherently not formalizable in such weak system: it (indirectly) relies on properties of expander graphs.

Therefore, Property 1 only holds weakly for a class of formulae capturing SL. However, we can still prove the weak version of the definability theorem (theorem 3.3.13).

Similarly to the definitions 3.1.4 and 5.1.1, we define symmetric Σ_1^B-Krom formulae as follows.

Definition 6.0.4. Following Grädel ([Grä91]), we define Σ_1^B-SymKrom formulae to be restricted Σ_1^B of the form

$$\exists P_1 ... \exists P_k \forall x_1 < b_1(\bar{a}) ... \forall x_m < b_m(\bar{a}, x_1, \dots x_{m-1}) \psi(\bar{x}, \bar{P}, n, \bar{a}, \bar{Y}), \tag{6.1}$$

where ψ has clauses of the form $(\phi_j \to (L_j(t_j) \oplus L'_j(t'_j))$ (that is,$(\phi_j \to \neg(L_j(t_j) \leftrightarrow L'_j(t'_j)))$ or $(\phi_j \to L_j(t))$ or (ϕ_j), where ϕ_j is quantifier-free and L_j are of the form P_i or $\neg P_i$. As before, second-order quantified variables P_i can only occur as P-literals (as $P_i(t)$ or $\neg P_i(t)$). In particular, they cannot be arguments to functions.

Note that if a clause has one negated literal, it can be assumed to be the first literal; if both literals are negated, then both can be replaced with positive literals without changing the value of the clause.

Now define a system of arithmetic V-SymKrom to be V-Φ with $\Phi \equiv \Sigma_1^B$-SymKrom.

6.1 Symmetric transitive closure

Following techniques from chapter 5, we introduce a symmetric transitive closure operator STC into V-SymKrom. A symmetric transitive closure operator is a transitive closure operator on undirected graphs: that is, $STC_{x,y}\phi(\bar{a}, \bar{Y})[a, b]$ holds iff a and b are connected in an *undirected* graph with ϕ labeling the edges. We introduce STC into V-SymKrom using a the same defining axiom as AxTC, except $Cond$ is defined differently. So,

$$STC_{x,y}\phi(x, y, \bar{a}, \bar{Y})[a, b, n] \leftrightarrow \forall R(CondS(\phi, R, n) \to R(a, b)), \tag{AxSTC}$$

where

$$CondS(\phi, R, n) \equiv$$
$$\forall x, y, z < n(R(x, x) \wedge (\phi(x, y) \to (R(y, z) \leftrightarrow R(x, z))))$$

Note that if ϕ is quantifier-free except for bounded existential first-order quantifiers, then the negation of $STC_{x,y}\phi(x, y)[a, b, n]$ is equivalent to a Σ_1^B-SymKrom formula. Therefore, V-SymKrom proves induction on Σ_0^B combinations of STC functions.

By the same reasoning as for V-Krom, the function STC defined in this manner is reflexive, transitive and robust against adding the edge on the left versus on the right (that is, conditions with $\phi(x, y) \to (R(x, z) \leftrightarrow R(y, z))$ and $\phi(y, z) \to (R(x, z) \leftrightarrow R(x, y))$ are equivalent).

The following lemma is specific to symmetric transitive closure. It states that symmetric transitive closure is indeed symmetric: it is a transitive closure over an underlying undirected graph.

Lemma 6.1.1.

$$V\text{-}SymKrom \vdash STC_{u,v}\phi[a, b] \leftrightarrow STC_{u,v}\phi[b, a].$$

Proof outline. We think of $\phi(u, v)$ as giving an edge relation for a graph $E(u, v)$ on n vertices. Using this notion, the proof is by induction on the number of edges in a path from a to b. For the base case, consider an edge (a, b). Then $(\phi(a, b) \to (R(a, a) \leftrightarrow R(b, a)))$. Since by reflexivity $R(a, a)$ holds, so does $R(b, a)$.

Now suppose that for some c such that $\phi(a, c)$ and there is a path from c to b of length k. We want to show that $R(b, a)$ holds. Here we refer to transitivity of R. By the base case, $R(c, a)$ holds, and by induction hypothesis $R(b, c)$ holds. Together, by transitivity, they give $R(b, a)$.

A more formal proof uses counters similar to the ones described in section 5.6.1. It is quite technical, so we omit it here. □

6.2 Simulating Σ_0^B formulae

As before, we need to show that $V^0 \subseteq V$-SymKrom, and every Σ_0^B-definable function is Σ_1^B-SymKrom-definable. The main goal is to show how to simulate existential first-order quantifiers by using existential second-order quantifiers.

The following theorem is a V-SymKrom analog of theorems 5.2.1 and 4.1.4.

Theorem 6.2.1. *For every Σ_0^B formula ϕ there is a Σ_1^B-SymKrom formula ϕ^* such that*

$$V\text{-}SymKrom \vdash \phi \leftrightarrow \phi^*$$

By Theorem 3.2.10, we immediately get the following corollary.

Corollary 6.2.2. *V-SymKrom proves induction and comprehension over Σ_0^B (Σ_1^B-SymKrom) formulae.*

Proof of theorem 6.2.1. The proof is somewhat similar to the proof of theorem 5.2.1, however the underlying graph structure is different. Since we already defined the STC operator, we will appeal to it instead of writing out the formulae from scratch, as we did in the V-Krom case.

Assume that ϕ is in prenex form. We give a formula defining the edges of a graph on vertices $\{0, \ldots, n\}$, in which the start vertex 0 and the target vertex n are in *different* connected components iff the original formula is *true* on its free variables. For simplicity, let n be a bound for every variable, and every alternation of quantifiers contains exactly one quantified variable. Otherwise, we could talk about tuples of variables witnessing or contradicting an existential or universal quantifier. The proof proceeds by induction on the number of bounded quantifier alternations.

Base case.

In the base case we consider a quantifier-free formula with either one quantifier or a pair of alternating quantifiers. We will show how to construct a formula ψ^* defining a graph such that $\phi \equiv \neg STC_{x,y}\psi^*(x,y)[0,n,n+1]$.

Single existential quantifier: Suppose that $\phi = \exists z < n\psi(z)$, where ψ is quantifier-free, and fixing the free variables. Consider a graph $E(x,y)$ with vertices $\{0, \ldots, n\}$ in which the only edges present are of the form $(x, x+1)$. Now since $\psi(z)$ is quantifier-free, the negation of it is also quantifier-free and thus can be used in STC leaving $AxSTC$ a negated Σ_1^B-SymKrom formula. That is, a graph $E(x,y)$ corresponding to the formula is defined by $E(x,y) \leftrightarrow (\neg\psi(x) \wedge y = x+1)$. A formula defining this graph is, respectively, $\psi^*(x,y) \equiv (\neg\psi(x) \wedge y = x+1)$. Now, $\phi \equiv \neg STC_{x,y}\psi^*(x,y)[0,n,n+1]$.

Suppose that $\psi(z_0)$ holds for some z_0. Now, R that satisfies $CondS(R, \psi^*, n+1)$, but does not contain $(0,n)$ is

$$R(x,y) \equiv \begin{cases} \top & x \le z_0 \wedge y \le z_0 \vee y \ge z_0+1 \wedge x \ge z_0+1 \\ \bot & \text{otherwise} \end{cases}$$

That is, R contains all pairs of vertices on either the part of the path before z_0, or on the part of the path after z_0+1. If the edge (z_0, z_0+1) is missing, then R contains the symmetric transitive closure of ψ^*.

Single universal quantifier: For the induction step, we need to construct graphs corresponding to a universally quantified formulae such as $\forall u < n\psi(u)$. If in the case of existential quantifier the graph was disconnected if at least one edge was missing, now we want a graph that is disconnected only if all edges are missing.

We represent a universal quantifier by the following graph E which will have a path from s to t iff there is a counterexample u. Define $E(s,u) \leftrightarrow E(u,t) \leftrightarrow \neg\psi(u)$. That is, $\psi^*(x,y) \equiv (x = s \wedge \neg\psi(y) \vee y = t \wedge \neg\psi(x))$. For the base case, we can set $s = n$ and $t = n+1$. Note that if $\forall u < n\psi(u)$ holds, then E has no edges.

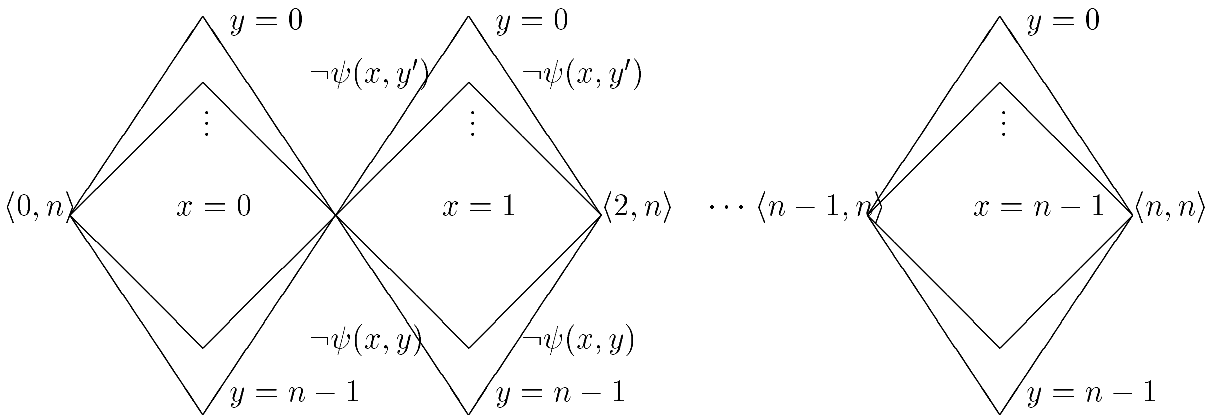
Suppose that $\forall u < n\psi(u)$. Set $R(x,y) \equiv x = y$. This R satisfies $\neg R(n, n+1)$ trivially. Now, $CondS(R, \psi^*, n+2)$ holds because none of $\psi(u)$ holds and thus there are no edges forced in the graph E.

General case for a universal quantifier: The base case $\phi = \exists z < n \forall u < n\psi(z,u)$ can be treated in a way similar to the case of single existential quantifier: since $\forall$ is innermost, it is sufficient to set $\psi^*(x,y) \equiv \exists u < n(\neg\psi(x,u) \wedge y = x+1)$.

Here we present a more general way of representing the case $\phi = \exists z < n\forall u < n\psi(z,u)$. It illustrates the construction used in the induction step proof. Now we are combining the single existential and single universal quantifier cases to obtain

$$\psi^*(\langle x,y\rangle, \langle x',y'\rangle) \equiv ((y = n \wedge x' = x \wedge \neg\psi(x,y')) \vee y' = n \wedge x' = x+1 \wedge \neg\psi(x,y')).$$

That is, there are vertices of two kinds: the x-vertices ("existential vertices") and the y-vertices ("universal vertices"). The vertices of the first type are encoded by tuples $\langle x,n\rangle$. The vertices of the second type are encoded by tuples $\langle x,y\rangle$, $y < n$. For every counterexample $\neg\psi(x,y)$ there are two edges of the form $(x,n) \to (x,y) \to (x+1,n)$. That is, there is a path from (x,n) to $(x+1,n)$ iff there was a counterexample y. The second-order variable R satisfying $CondS$ and not containing $\langle 0,n\rangle, \langle n,n\rangle$ is constructed as before, by considering all vertices before and after a witness x for which there is no y connected to (x,n) and $(x+1,n)$. The following picture describes this construction.



The correctness argument for this case is a combination of arguments for the single existential and single universal cases.

Induction step.

As in the base case, we are constructing a graph in which the starting and target vertices are in different connected components iff the formula is true. Consider the formula of the form $\phi \equiv \exists x < n \forall y < n \phi_k(x, y)$, where now ϕ_k contains k alternations of first-order quantifiers. Let $E(u, w)$ be a graph encoding $\phi_k(x, y)$. To construct a graph $E'(u', w')$ representing ϕ we take a graph for a $\exists x < n \forall y < n$ formula, and replace every edge in it by a graph denoted by $E(u, w)$. The resulting E' has two additional elements x and y in every tuple denoting a vertex. The start vertex will be of the form $\langle 0, n, 0, n, \ldots, 0, n \rangle$ and the end vertex will be $\bar{n}$. The construction of the corresponding R is similar to the base cases.

$\square$

6.3 Constructiveness of Σ_1^B-SymKrom

In this section we will show how to encode a witness to a Σ_1^B-SymKrom formula by using STC. This is very similar to section 5.5 in chapter 5.

Let ϕ^* be a Σ_1^B-SymKrom formula. We would like to construct a formula ϕ such that ϕ does not have second-order quantifiers, but has occurrences of STC, and V-SymKrom $\vdash \phi \leftrightarrow \phi^*$.

6.3.1 SymKrom satisfiability algorithm

The Σ_1^B-SymKrom algorithm follows the outline in section 3.1.4. After obtaining a propositional symmetric 2SAT formula, we construct a corresponding graph, which is bipartite iff the original formula was satisfiable. Let E_ϕ be the graph corresponding to the formula. The vertices of this graph are literals of the propositional formula: that is, there are two literals p_{t_i} and $\neg p_{t_i}$ for every value of a term t_i occurring as $P(t_i)$ or $\neg P(t_i)$, and thus two corresponding vertices v_{t_i} and u_{t_i}. Additionally, there are two special vertices $\top$ and $\bot$. The two types of edges of E_ϕ are edges between a literal and its negation (in particular, and edge $\top \to \bot$), and edges between two literals occurring in a same clause. If a clause contains only one literal, then if that literal is positive then there is an edge $(v_{t_i}, \bot)$, and if negative $(v_{t_i}, \top)$.

Let E be a graph. Then a Σ_1^B-SymKrom formula stating its bipartiteness is

$$\exists R \forall x < n \forall y < n (E(x,y) \to (\neg R(x) \leftrightarrow R(y))) \qquad \text{(Bipartiteness)}$$

This formula is still equivalent to Σ_1^B-SymKrom if $E(x,y)$ is replaced by a prenex Σ_0^B formula with only existential bounded quantifiers.

Consider a following Σ_1^B-SymKrom formula $\phi^*(\bar{a}, \bar{Y})$. To simplify the notation, we will omit free variables, although all terms and ϕ_j have $\bar{a}, \bar{Y}$ as parameters in addition to $\bar{x}$.

$$\phi^* \equiv \exists P \forall x_1 < n_1 \ldots \forall x_k < n_k \psi(P, \bar{x}), \tag{6.2}$$
$$\text{where } \psi(P, \bar{x}) \equiv \bigwedge^m (L_j(t_j(\bar{x})) \oplus L'_j(t'_j(\bar{x})) \vee \neg\phi_j(\bar{x})).$$

Now we construct a formula ϕ' defining a graph for ϕ^*, such that the graph is bipartite iff ϕ^* holds. Every vertex is encoded by a pair (v, s), where $v = t_i$ for a term occurring in a P-literal, and s is 0 if v corresponds to a negative literal, and 1 if v corresponds to a positive literal.

Let

$$c_j = (L_j(t_j(\bar{x})) \oplus L'_j(t'_j(\bar{x})) \vee \neg\phi_j(\bar{x})).$$

There are two cases. The first is when both literals are positive or negative, and the second is when one is positive and one is negative. If the literals have the same sign, then the translation is

$$C_j(\bar{x}) = (\phi_j(\bar{x}) \wedge s = s' \wedge t_j(\bar{x}) = u \wedge t'_j(\bar{x}) = v)$$

If the literals have the opposite sign, then the translation is

$$C_j(\bar{x}) = (\phi_j(\bar{x}) \wedge s \neq s' \wedge t_j(\bar{x}) = u \wedge t'_j(\bar{x}) = v)$$

If there is a single literal in the clause, then the clause is $(\phi_j(\bar{x}) \wedge s = s' \wedge t_j(\bar{x}) = u \wedge v = b)$ or $(\phi_j(\bar{x}) \wedge s \neq s' \wedge t_j(\bar{x}) = u \wedge v = b)$, depending whether that literal is positive or negative. Here, b is a special vertex with the value equal to $1 + \max_j\{t_j, t'_j\}$. The interpretation is $(b, 0) = \perp$ and $(b, 1) = \top$.

Additionally, we need to enforce the edges between literals and their negations. We do it by adding the following clause

$$(v = u \wedge s \neq s')$$

Now a formula ϕ' encoding the edge relation graph corresponding to ϕ^* is

$$\phi'(u, s, v, s') \equiv (u = v \wedge s \neq s') \vee \exists \bar{x} < \bar{n} \bigvee_j C_j(\bar{x})$$

This ϕ' holds on (u, s, v, s') iff there is an edge from (u, s) to (v, s'), where u and v are literals and s, s' can be 0 or 1 depending whether the corresponding literal was negated.

Finally, the formula ϕ^* holds if

$$\exists R \forall u, v \leq b \forall s, s' < 2(\phi'(u, s, v, s') \rightarrow (\neg R(\langle u, s\rangle) \leftrightarrow R(\langle v, s'\rangle)))$$

With this R, we can define $P(i)$ as follows

$$P(i) \leftrightarrow R(i, 0) \wedge R(b, 0) \vee R(i, 1) \wedge R(b, 1) \tag{6.3}$$

That is, $P(i)$ is true if i is on the same side of the bipartite graph as the special symbol $\top$.

6.3.2 Relation between transitive closure and bipartiteness

We would like to express the satisfiability problem for ϕ^* using STC rather than just bipartiteness. For that, we describe how to relate the notions of reachability and bipartiteness. For simplicity, we describe the constructions for an arbitrary graph E on n vertices.

First we describe how to use STC to test bipartiteness. For $x, y < n$ and $t, t' < 2$ define $E'(x, t, y, t') \equiv (E(x, y) \wedge t \neq t')$. That is, if there is an edge (x, y) in E, then there are edges $((x,0),(y,1))$ and $((x,1)(y,0))$ in E'. A path in E corresponds to a path of the same length in E', however the path in E' also keeps track of its parity by alternating between "even side" $(x,0)$ and "odd side" $(x,1)$ of E'.

Recall that a graph is bipartite iff it does not contain an odd cycle. Therefore, E is bipartite iff there is no $x < n$ such that there is a path from $(x,0)$ to $(x,1)$ in E'.

Lemma 6.3.1. *Let E be a graph. Then*

$$V\textit{-SymKrom} \vdash (\exists R \forall x, y < n(E(x,y) \to (R(x) \oplus R(y))))$$
$$\leftrightarrow \neg(\exists w < n STC_{(u,t)(v,t')} E'[(w,0)(w,1),\langle n,2\rangle])$$

Proof. We can rephrase the condition of the lemma as follows, using the fact that intersection of several R satisfying $Cond$ satisfies $Cond$ as well.

$$\begin{aligned} V\text{-Krom} \vdash & \exists R \forall x, y < n(E(x,y) \to \neg R(x) \leftrightarrow R(y)) \\ \leftrightarrow & \exists Q CondS(Q, E', \langle n,2\rangle) \wedge \forall w < n \neg Q((w,0),(w,1)) \end{aligned} \tag{6.4}$$

Suppose that R satisfies the bipartiteness condition in equation 6.4 above. Define

$$Q((x,t)(y,t')) \leftrightarrow ((x,t) = (y,t')) \vee (t \neq t' \wedge (\neg R(x) \leftrightarrow R(y))) \vee (t = t' \wedge R(x) \leftrightarrow R(y)).$$

Call $x_1 \dots x_k$ the vertices of E in R, and $y_1 \dots y_{n-k}$ the vertices of E not in R. Then Q consists of all edges of the form $((x_i,0),(y_j,1))((x_i,1),(y_j,0)), ((x_i,t),(x_j,t)), ((y_i,t),(y_j,t))$. Therefore, Q contains every edge of E and does not contain $((x,0)(x,1))$.

It remains to show that Q is transitive. Suppose that $E'((x,t),(y,t'))$ holds. Then by construction $(t \neq t')$ and $\neg R(x) \leftrightarrow R(y)$. Suppose that $Q((y,t'),(z,t''))$ holds. If $t' = t''$ then either $y = z$, in which case $Q((x,t),(z,t''))$ holds trivially, or $R(y) \leftrightarrow R(z)$. Then, $\neg R(x) \leftrightarrow R(z)$ and $t \neq t''$, so $Q((x,t),(z,t''))$ holds by construction. If $t' \neq t''$, then $(\neg R(y) \leftrightarrow R(z))$. In this case, $t'' \neq t$ and $R(x) \equiv R(z)$, therefore $Q((x,t),(z,t''))$ holds.

Now suppose that $Q((y,t')(z,t''))$ does not hold. Again, first let $t' = t''$. Then $R(y) \neq R(z)$, and $R(z) = R(x)$. But in this case $Q(x,t,z,t'')$ does not hold since $t \neq t''$ but $R(x) \leftrightarrow R(z)$. Now let $t' \neq t''$, so $t'' = t$. Now, either $R(y) = R(z)$ or $y = z$. In the first case, $R(z) \neq R(x)$ so $Q(x,t,y,t'')$ does not hold. The second case contradicts the assumption $\forall y < n \neg Q(y,0,y,1)$ (note that $Q(x,0,y,1) \leftrightarrow Q(x,1,y,0)$ by construction of E).Therefore, Q satisfies the negated STC above.

To show the other direction, let Q satisfy the second formula. The way $R(x)$ is defined closely resembles the equation 5.5.3.

$$R(x) \leftrightarrow (\exists y < n \forall z < n \forall t, t' < 2(y \geq x) \wedge (z > y \to \neg Q(z,t,x,t')) \wedge Q(y,1,x,1)) \tag{6.5}$$

That is, if x is the maximal element in its connected component (so $x = y$) then it is placed in R; otherwise, x is in R if it is on the same side as the largest element in its connected component.

The proof of the correctness of this construction is straightforward. It follows from the construction of E' that $Q(z, 1, x, 1)$ if there is an even length path from z to x, and $Q(z, 1, x, 0)$ if there is an odd length path (the proof is by induction on the number of elements in a connected component: every time we add an edge, we jump to the other side). Note that both cannot be true without contradicting the condition $\forall w < n \neg Q(w, 0, w, 1)$.

Let u be the largest element in the same connected component as x, that is, $\forall z > u \forall t, t' < 2 \neg Q(z, t, x, t')$. Then every element in the same component as u will have its value set with respect to u; therefore, there is a unique splitting for each connected component fixing the position of u. Also, the way different connected components are split is independent from each other. So by setting every largest element in every connected component to be in R, we obtain a unique splitting of E into two sets of vertices $R(x)$ and $\neg R(x)$, satisfying bipartiteness. $\square$

Now we are ready to state a STC version of the Σ_1^B-SymKrom satisfiability. As in the Σ_1^B-Krom case, a formula is unsatisfiable if for some $j < a$ both $(j, 0)$ and $(j, 1)$ are forced to the same value. That is, there exists an odd cycle containing $(j, 0)$ and $(j, 1)$. To find, using reachability, an odd cycle in a graph we make two copies of every vertex to be on the "even" and "odd" side of the bipartite graph. So if there were vertices (j, s) and (k, s') in the original graph, then there are vertices $(j, s, 0), (j, s, 1), (k, s', 0), (k, s', 1)$ in the new graph, and edges $(j, s, 0)((k, s', 1)$ and $(j, s, 1), (k, s', 0)$ for an edge $(j, s)(k, s')$ in the original graph. Let ϕ'' be a version of ϕ with this doubling of edges. The following formula encodes the unsatisfiability of ϕ^*:

$$\exists j < a \exists s < 2\ STC_{(u,s,t),(v,s',t')} \phi''[(j, s, 0), (j, s, 1), (a, 2, 2)]. \tag{6.6}$$

Let Q be the variable satisfying the negation of STC in equation 6.6, that is, $Cond(\phi'', Q, \langle a, 2, 2\rangle) \wedge \forall j < a \forall s, t < 2 \neg Q((j, s, 0), (j, s, 1))$. Using equations 6.5 and 6.3, we obtain the following unique definition of P by Q:

$$\begin{aligned} P(x) \equiv & (\exists y < n \exists s < 2 \forall z < n \forall s', s'' < 2 \forall t, t' < 2 \\ & (y \geq x) \wedge ((z > y \vee (z = y \wedge s' < s)) \rightarrow \neg Q(z, s', t, x, s'', t')) \wedge Q(y, 1, 1, x, 1, 1)) \end{aligned} \tag{6.7}$$

Note that in this case we can replace Q by the function STC, giving the witnessing for Σ_1^B-SymKrom satisfiability problem.

Lemma 6.3.2. *Let $F_{STC}(E,n)$ be a string function returning an array Z where $Z(u,v)$ holds iff (u,v) is in the transitive closure of E on the first n vertices. Let ϕ^* be a Σ_1^B-SymKrom formula with a single second-order variable P and let $E_{\phi''}$ be the edge relation graph of ϕ''. Then the following $\Sigma_0^B(F\mathtt{SL})$ function computes $P(x)$, if P exists:*

$$(\exists y < n \exists s < 2 \forall z < n \forall s', s'' < 2 \forall t, t' < 2 (y \geq x) \wedge ((z > y \vee (z = y \wedge s' < s)) \tag{6.8}$$
$$\rightarrow \neg F_{STC}(E_{\phi''}, \langle a,2,2\rangle)(\langle z,s',t\rangle, \langle x,s'',t'\rangle)) \wedge F_{STC}(E_{\phi''}, \langle a,2,2\rangle)(\langle y,1,1\rangle, \langle x,1,1\rangle))$$

The proof of lemma 6.3.2 is immediate from the definition of STC, minimality of Q and equation 6.7.

6.3.3 A Σ_1^B predicate equivalent to STC: reachability.

At this time we cannot prove that $\mathtt{SL}$ is strongly closed under complementation. The only known proof of closure of $\mathtt{SL}$ under complementation is due to Nisan and Ta-Shma [NTS95]. This result came less than 10 years ago, long after the proof of closure of $\mathtt{NL}$ and $\mathtt{SAC^1}$ under complementation was known. The idea of the proof is to estimate the number of connected components in a graph by two measures: by the number of vertices with the largest index in their connected component (upper bound) and by the number of leaves in a lexicographically first spanning forest (lower bound). The construction produces two vectors, a vector LI which contains 0 for every i which is largest in its component, and a vector LFF, which contains a 0 for every pair (u,v) such that there is an edge (u,v) in the lexicographically first spanning forest. If there are k connected components in a graph, then there are $n-k$ leaves in the lexicographically first spanning forest, so if LI contains k zeroes and LFF contains $n-k$ zeroes, then both estimates are correct.

Both of these concepts are easily formalizable in V-SymKrom. The problem arises at the stage of comparison of the number of 0's in two vectors. For that, a *monotone* formula computing a function similar to $NUMONES$ is needed.

The original Nisan and Ta-Shma's approach refers to the existence of sorting networks in $\mathtt{NC^1}$. However, the constructive proof of the existence of such sorting networks is very complex. It was first discovered by Ajtai, Komlos and Szemeredi [AKS83], and their construction was later simplified by Paterson [Pat90]. But even the simplified construction relies on the properties of expander graphs. In order to talk about expander graphs we need formalization of algebraic properties, such as "second eigenvalue". That we do not know how to do in less than polynomial-time reasoning (see [SC02] for discussion of formalizing algebraic properties).

Another approach would be to use a monotone $\mathtt{TC}^0$ formula for MAJORITY, given by Valiant in [Val84]. This result has a fairly short and elegant proof, but the proof is based on probabilistic reasoning and is non-constructive. Again, we do not know how to formalize probabilistic reasoning in our class of systems. Therefore, the current proofs of closure of $\mathtt{SL}$, and thus Σ_1^B-SymKrom, under complementation are not formalizable in V-SymKrom.

However, if we can prove the second property, that is the constructiveness, then we can give a weaker form of the witnessing theorem, where witnessing functions are in the $\mathtt{AC}^0$ closure of $F\mathtt{SL}$. For that, we need to give a Σ_1^B formula defining a predicate which is provably equivalent to the negation of one of the complete problems for $\mathtt{coSL}$. Note that it is sufficient to make that predicate Σ_1^B, it is not necessary to give a Σ_1^B-SymKrom predicate. The main concerns are witnessing and proving equivalence to the negation of a Σ_1^B-SymKrom predicate.

The negated Σ_1^B-SymKrom statement which we will use states the existence of an odd cycle, or, equivalently, a path in a doubled graph. Therefore, it is sufficient to consider the problem of representing existence of a path between two given vertices in a graph as a Σ_1^B formula and proving its equivalence to a transitive closure formula.

Before describing the Σ_1^B formula for the existence of a path, we will give some intuition about the witnessing proof. Recall the definition 5.6.2 of counters from the proof of Immerman's theorem for $\mathtt{NL}$. More specifically, we will use a variant of the $Dist$ function, but now on an undirected graph.

Definition 6.3.3. Let ϕ be a formula defining an edge relation of a graph. Let

$$UDist_\phi(x,y,d) \equiv STC_{(u,c),(v,c')}\alpha[(x,0),(y,d),(n,n)],$$

where $\alpha(u,c,v,c') \equiv (c' = c+1 \wedge (\phi(u,v) \vee u = v))$. For simplicity, we assume that ϕ is represented by the corresponding graph E, and write $UDist(x,y,d)$ in that case.

By that definition, $UDist_\phi$ holds iff there is a path from x to y of length at most d in a graph with edge relation defined by ϕ. If ϕ is a Σ_0^B formula with only bounded existential quantifiers, then STC is negated Σ_1^B-SymKrom. The definition of α is the same as in the NL case.

We can prove analogs of lemmas in section 5.6.2 for symmetric transitive closure. In particular, we will use the following analog of Lemma 5.6.10:

Lemma 6.3.4. *If there is a path in E from x to y, then there is a path from x to y of length $\leq n-1$. That is,*

$$STC_{u,v}E(u,v)[x,y,n] \leftrightarrow UDist(x,y,n-1)$$

Now we are ready to describe undirected reachability. To make the presentation cleaner, assume that the graph has $n+1$ vertices (that is, the longest path in the graph is of length at most n). We will first say how to witness the variable R in our Σ_1^B predicate $\textsc{Reach}(E,n,a,b)$, and then give the predicate itself.

Let variable $R(x,i)$ be true if there is a path (with possible loops and vertex repetitions) starting from a in which the i^{th} vertex is x. Define $R(x,i)$ by the following formula, for $x \leq n, i \leq n$:

$$R(x,i) \equiv UDist(a,x,i) \tag{6.9}$$

The following Σ_0^B formula, which is implicitly based on this definition of $R(x,i)$, is defined to correspond to the R from equation 6.9:

$$\begin{aligned}\textsc{ReachCond}(R,E,n+1,a) \equiv & \forall x \leq n \forall i \leq n (R(x,0) \leftrightarrow x = a) \\ & \wedge (R(x,i+1) \leftrightarrow (\exists y \leq n R(y,i) \wedge (E(y,x) \vee y = x)))\end{aligned} \tag{6.10}$$

The following lemma states that $UDist$ witnesses $\exists R$ in $\exists R\textsc{ReachCond}(R,E,n+1,a)$.

Lemma 6.3.5. *V-SymKrom proves*

$$\textsc{ReachCond}(R,E,a,n+1) \implies \forall x \leq n \forall i \leq n(R(x,i) \leftrightarrow UDist(a,x,i))$$

Proof idea. The idea of the proof is to verify, by induction on i, that R defined by the equation 6.9 satisfies each of the conditions of the formula 6.10. We can use induction because R is $\Sigma_0^B(\Sigma_1^B$-SymKrom), and by theorem 3.2.10 we can do induction and comprehension on Σ_0^B-combinations of Σ_1^B-SymKrom formulae. All the conditions of equation 6.10 that need checking are Σ_0^B. $\square$

Now, the following Σ_1^B formula states that there is a path from a to b in E:

$$\textsc{Reach}(E, n+1, a, b) \equiv \exists R \textsc{ReachCond}(R, E, n+1, a) \wedge R(b, n).$$

Now we want to show that $\textsc{Reach}$ and STC are equivalent.

Theorem 6.3.6. *Let E be a graph on $n+1$ vertices, and let a, b be two arbitrary vertices in E. Then*

$$V\text{-}SymKrom \vdash \textsc{Reach}(E, n+1, a, b) \leftrightarrow STC_{u,v}E(u, v)[a, b, n+1].$$

Proof. The proof of this theorem is immediate from lemma 6.3.4, lemma 6.3.5, and the following claim 6.3.7.

Claim 6.3.7. *The statement $\exists R ReachCond(R, E, n+1, a)$ is a theorem of V-SymKrom.*

Proof idea. The idea of the proof is to use comprehension over equation 6.10 with definition of R via $UDist$ substituted into the formula under the existential second-order quantifier. □

□

Now we can use $\textsc{Reach}$ predicate to Σ_1^B-define the condition that a Σ_1^B-SymKrom formula ϕ^* does not have a satisfying assignment.

Theorem 6.3.8. *Let ϕ^* be a Σ_1^B-SymKrom formula. Then there is a Σ_1^B predicate equivalent to the negation of ϕ^*. Moreover, the existential quantifier in that predicate can be witnessed in $\Sigma_0^B(\Sigma_1^B$-SymKrom).*

Proof. Consider the equation 6.6. It can be converted to an equation with a single outermost occurrence of transitive closure. The idea is to take $2n$ copies of the graph corresponding to ϕ'', add two new vertices $(n, 0, 0, 0)$ and $(n, 0, 0, 1)$, and connect, in the copy $\langle j, s\rangle$, $(n, 0, 0, 0)$ to $(j, j, s, 0)$ and $(n, 0, 0, 1)$ to $(j, j, s, 1)$. The idea behind this construction is similar to the treatment of the universal quantifier in the proof of theorem 6.2.1. Now replace ϕ'' with ϕ_c which is equivalent to ϕ'' on each copy of the graph and has additional edges from $(n, 0, 0, 0)$ and $(n, 0, 0, 1)$, as described above. The formula encoding satisfiability for ϕ^* now becomes $STC_{\langle j,u,s,t\rangle,\langle j',v,s',t'\rangle}\phi_c[\langle n, 0, 0, 0\rangle, \langle n, 0, 0, 1\rangle, \langle n+1, n, 2, 2\rangle]$.

Given this formula, a Σ_1^B formula equivalent to it is $\textsc{Reach}(\phi_c, \langle n+1, n, 2, 2\rangle, \langle n, 0, 0, 0\rangle, \langle n, 0, 0, 1\rangle)$. By construction, $\textsc{Reach}$ holds iff ϕ^* is not satisfiable. □

6.4 A weak definability theorem for V-SymKrom and finite axiomatizability.

Let $F\mathtt{SL}$ be the class of functions with Σ_1^B-SymKrom formulae as defining axioms, following the definition 3.3.5. For this system we have to use the weak version of theorem 3.3.13. That is, the definability theorem for V-SymKrom is stated as follows:

Theorem 6.4.1. *Provably in V-SymKrom, every $F\mathtt{SL}$ function is Σ_1^B-definable in V-SymKrom and every function Σ_1^B-definable in V-SymKrom is in $\mathtt{AC}^0(F\mathtt{SL})$. Since $\mathtt{SL}$ is closed under complementation, $\mathtt{AC}^0(F\mathtt{SL}) = F\mathtt{SL}$, so the class of Σ_1^B-definable functions of V-SymKrom coincides with $F\mathtt{SL}$. However, the latter statement is not known to be provable in V-SymKrom.*

Proof. We prove the theorem using only constructiveness property. Σ_1^B-SymKrom is constructive by the results in the previous section: we can either witness the existence of a path in a graph or a transitive closure which does not contain a given pair.

Lemma 3.3.14 applied to V-SymKrom states that every $F\mathtt{SL}$ function is Σ_1^B-definable in V-SymKrom. For the other direction, apply the weak version of theorem 3.4.2 to conclude that the class of witnessing functions of V-SymKrom is $\Sigma_0^B(F\mathtt{SL})$.

By Nisan and Ta-Shma's result [NTS95], $\mathtt{SL}$ is closed under complementation. Therefore, the class of Σ_1^B-definable functions of V-SymKrom is indeed $F\mathtt{SL}$, however V-SymKrom does not prove it. □

As in the case of V-Krom, we can show that V-SymKrom is finitely axiomatizable.

Theorem 6.4.2. *V-SymKrom is finitely axiomatizable.*

Proof. The proof of this theorem is very similar to the proof of finite axiomatizability of V-Krom. Since by theorem 6.2.1 Σ_0^B comprehension is provable in V-SymKrom, we know that V-SymKrom extends V^0. By theorem 3.6.1, V^0 is finitely axiomatizable. Now, let $\phi^*(i, \bar{a}, \bar{Y})$ be a Σ_1^B-SymKrom formula and let ϕ'' be the same as used in the equation 6.6. Then by comprehension for any value of the free variables $\bar{a}, \bar{Y}$ of ϕ^* there exists a graph E with ϕ'' as its edge relation.

Consider the following formula (which is a negation of formula 6.6):

$$\Psi(i, E) \equiv \exists Q\ CondS(E, Q, \langle a, 2, 2\rangle) \wedge \forall j < a \forall s < 2 \neg Q(\langle j, s, 0\rangle, \langle j, s, 1\rangle).$$

Comprehension over $\Psi(i, E)$ plus V^0 proves all of V-SymKrom. □

Chapter 7

Conclusion

In this work we presented a general framework for constructing systems of arithmetic with a predefined power. The setback is that the requirements on the class of formulae which can serve as a basis for such systems are quite strict: the closure under $\mathtt{AC}^0$ relations is already a very powerful requirement, and the descriptive capture requirement makes constructing such systems even more complicated. So the main object of interest is to pinpoint as precisely as possible which properties on the complexity classes and the corresponding classes of formulae are necessary for such construction of systems of bounded arithmetic, in particular in which cases a version of constructiveness holds. Does it only work for Schaefer's classes (see Theorem 2.2.8)? Can the construction be based on fixed point logics directly, rather than on a respective second-order restricted formulae? Can the constructiveness property be rephrased to apply to a more general setting?

A separate question arises from the Symmetric Logspace example. Is it intrinsically hard to prove closure of $\mathtt{SL}$ under complementation using $\mathtt{SL}$ concepts? Such a result, formulated as an independence statement for V-SymKrom, would be very interesting, although quite unlikely. This touches on a deep question in complexity theory: under which conditions it is possible to prove properties of complexity classes using only concepts within these classes? We showed that closure under complementation can be proven within the class for $\mathtt{AC}^0$, $\mathtt{P}$ and $\mathtt{NL}$, but $\mathtt{SL}$ is a possible example for which this might not be the case. What is so different about $\mathtt{SL}$, then? Or is it the case that a simpler proof exists of existence of monotone formula for MAJORITY below $\mathtt{SL}$, and thus of closure of $\mathtt{SL}$ under complementation?

An attempt to build a system of arithmetic not based on one of Schaefer's classes is a current project concerning a class $\mathtt{LOGCFL}$. $\mathtt{LOGCFL}$ is a class of languages logspace-reducible to context-free languages. It was first studied in [Sud78]; there, an equivalent characterization via non-deterministic pushdown automata with a logspace-bounded auxiliary tape was given. LOGCFL contains NL and is contained in AC^1; that is,

$$\mathtt{AC^0} \subseteq \mathtt{NC^1} \subseteq \mathtt{L} \subseteq \mathtt{SL} \subseteq \mathtt{NL} \subseteq \mathtt{LOGCFL} \subseteq AC^1 \subseteq NC \subseteq \mathtt{P}.$$

One of the cleanest characterizations of LOGCFL is a uniform $\mathtt{SAC^1}$: a class of languages recognized by semi-unbounded fan-in depths $O(\log n)$ circuits. This class is halfway between $\mathtt{NC^1}$ and AC^1: the AND gates in $\mathtt{SAC^1}$ circuits have fan-in 2, and OR gates have unbounded fan-in. The other characterization of LOGCFL, due to Ruzzo, [Ruz80], is alternating logspace Turing machines with polynomial tree size. The relation between $\mathtt{SAC^1}$ characterization and Ruzzo's ATMs with bounded tree size was later shown by Venkateswaran [Ven87]. In 1989, Borodin et al. extended Immerman's inductive counting technique to show that $\mathtt{SAC^1}$ is closed under complementation [BCD$^+$89].

An important result came in 1998: Gottlob, Leone and Scarcello [GLS01] showed that, together with properties of semi-unboundedness and polynomial tree size, the property of acyclicity (or, equivalently, bounded tree-width) is a characteristic trait of LOGCFL. They gave a first natural satisfiability-like problem that is complete for LOGCFL: acyclic conjunctive boolean queries.

It seems that in the satisfiability community there is more emphasis on creating faster algorithms for (subclasses of) satisfiability than proving completeness for smaller classes. There is some work on acyclic satisfiability (for example, by Szeider or by Makovsky), but they do not prove any completeness results and their definitions are somewhat different from GLS (although related).

We would like to define a Grädel-style second-order version of acyclic restricted $SO\exists$. The idea is to somehow restrict allowed tuples of first-order variables, together with an acyclicity condition on the formula, so that the resulting propositional formula is acyclic. One approach is to treat clauses as implications, and require that the terms on one side of the implication are strictly smaller than on the other side. For example, in a clause $(P(x) \to Q(y))$ the only allowed values of first-order variables would be $x < y$. This might be an interesting result in its own right.

Then we can try to build a V-Φ system based on such "acyclic Σ_1^B" formulae. The next step would be to prove that LOGCFL is closed under complementation. One approach is to formalize the inductive counting proof from [BCD$^+$89]. In any case, we can try to prove constructiveness and apply Definability theorem.

Alternatively, it might be interesting to build a system of arithmetic based on a natural second-order version of Schaefer's 4th class of formulae, that is, "symmetric CNF" formulae, which are defined like symmetric Krom (see definition 6.0.4), except the number of literals in a clause is not restricted.

Yet another direction would be to consider complexity classes larger than NP. Even though some of them are not known to be closed under complementation, others are; it would be interesting to see if a version of Theorem 3.3.13 applies to classes like PSPACE or EXP. One approach here would be to use third-order theories; another to enrich the language with # and restate the results in that framework.

A traditional direction of research in bounded arithmetic is to augment known systems of arithmetic with axioms for combinatorial principles, and then study which other principles can be proven in that stronger theory. Examples of such an approach are presented in the work of Kerry Ojakian [Oja04], Neil Thapen, Michael Soltys [TS], in which they formalize mathematical notions such as Ramsey Theorem and linear algebra by adding combinatorial principles like pigeonhole principle to their systems of arithmetic. It would be interesting to see how our approach relates to that framework.

Bibliography

[AKS83] M. Ajtai, J. Komlos, and E. Szemeredi. An O(n log n) sorting network. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 1–9. Association for Computing Machinery, 1983.

[Ats02] Albert Atserias. *Fixed-point logics, descriptive complexity and random satisfiability.* PhD thesis, UCSC, 2002.

[BCD+89] A. Borodin, S. A. Cook, P. W. Dymond, W. L. Ruzzo, and M. Tompa. Two applications of inductive counting for complementation problems. *SIAM J. Comput.*, 18(3):559–578, 1989.

[BIS90] D. M. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274 – 306, 1990.

[Bus86] S. Buss. *Bounded Arithmetic.* Bibliopolis, Naples, 1986.

[Bus95] S. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Annals of Pure and Applied Logic*, 75:67–77, 1995.

[CK01] S.A. Cook and A. Kolokolova. A second-order system for polynomial-time reasoning based on Grädel's theorem. In *Proceedings of the Sixteenth annual IEEE symposium on Logic in Computer Science*, pages 177–186, 2001.

[CK03] S.A. Cook and A. Kolokolova. A second-order system for polytime reasoning based on Grädel's theorem. *Annals of Pure and Applied Logic*, 124:193–231, 2003.

[CK04] S.A. Cook and A. Kolokolova. Bounded arithmetic of NL. In *Proceedings of the Nineteenth annual IEEE symposium on Logic in Computer Science*, pages 398–407, 2004.

[Coo75] S. A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 83 –97, 1975.

[Coo98] S. A. Cook. Relating the provable collapse of P to NC^1 and the power of logical theories. *DIMACS series in Discrete mathematics and theoretical computer science*, 39:73–91, 1998.

[Coo02] S. A. Cook. CSC 2429S: Proof Complexity and Bounded Arithmetic. Course notes, URL: "http://www.cs.toronto.edu/~sacook/csc2429h", Spring 1998-2002.

[Coo04] S. Cook. Theories for complexity classes and their propositional translations. *submitted*, pages 1–36, 2004.

[CT86] P. Clote and G. Takeuti. Exponential time and bounded arithmetic. In *Proceedings of the Conference on Structure in Complexity Theory*, pages 125 – 143. Springer Verlag, 1986.

[CT92] P. Clote and G. Takeuti. Bounded arithmetic for NC, ALOGTIME, L and NL. *Annals of Pure and Applied Logic*, 56:73 – 117, 1992.

[CT95] P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In *Feasible Mathematics*, volume II. Birkhuser Inc., 1995.

[CU93] S. A. Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103 – 200, 1993.

[DDLW98] A. Dawar, K. Doets, S. Lindell, and S. Weinstein. Elementary properties of finite ranks, 1998.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite model theory.* Springer Verlag, 1995.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of computation, SIAM-AMC proceedings*, 7:43–73, 1974.

[GHR95] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation.* Oxford University Press, 1995.

[GLS01] Georg Gottlob, Nicola Leone, and Francesco Scarcello. The complexity of acyclic conjunctive queries. *JACM*, 48(3):431–498, 2001.

[Grä91] E. Grädel. The Expressive Power of Second Order Horn Logic. In *Proceedings of 8th Symposium on Theoretical Aspects of Computer Science STACS '91, Hamburg 1991*, volume 480 of *LNCS*, pages 466–477. Springer-Verlag, 1991.

[Grä92] E. Grädel. Capturing Complexity Classes by Fragments of Second Order Logic. *Theoretical Computer Science*, 101:35–57, 1992.

[HP93] Petr Hájek and Pavel Pudlák. *Metamathematics of first-order arithmetic.* Springer Verlag, 1993.

[Imm82] N. Immerman. Relational queries computable in polytime. In *14th ACM Symp.on Theory of Computing, Springer Verlag (Heidelberg, FRG and NewYork NY, USA)-Verlag*, pages 147 –152, 1982.

[Imm83] Neil Immerman. Languages that capture complexity classes. In *15th ACM STOC symposium*, pages 347–354, 1983.

[Imm87] Neil Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16(4):760–778, 1987.

[Imm99] N. Immerman. *Descriptive complexity.* Springer Verlag, New York, 1999.

[JLL76] Neil D. Jones, Y. Edmund Lien, and William T. Laaser. New problems complete for nondeterministic log space. *Mathematical Systems Theory*, 10:1–17, 1976.

[KPT91] J. Krajiček, P. Pudlák, and G. Takeuti. Bounded arithmetic and the polynomial time hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.

[Kra95] J. Krajiček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, New York, USA, 1995.

[Kro67] M. R. Krom. The decision problem for a class of first-order formulas in which all disjunctions are binary. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 13(1):15–20, 1967.

[Lib04] L. Libkin. *Elements of Finite Model Theory.* Springer Verlag, 2004.

[Liv82] A.B. Livchak. Languages for polynomial-time queries. In *Computer-based modeling and optimization of heat-power and electrochemical objects*, page 41, 1982.

[LP82] Harry R. Lewis and Christos H. Papadimiriou. Symmetric space-bounded computation. *Theoretical Computer Science*, 19:161–187, 1982.

[Min73] G.E. Mints. Quantifier-free and one-quantifier systems. *Journal of Soviet Mathematics*, 1:71–84, 1973.

[NC04] Phuong Nguyen and Stephen Cook. VTC^0: a second-order theory for TC^0. In *Proceedings of the Nineteenth annual IEEE symposium on Logic in Computer Science*, pages 378–387, 2004.

[NKC04] P. Nguyen, A. Kolokolova, and S. Cook. Two-sorted theories for L, SL, NL and P based on graphs accessibility problems. *in progress*, 2004.

[NTS95] Noam Nisan and Amnon Ta-Shma. Symmetric logspace is closed under complement. In *Proc. 27th Ann. ACM Symp. on Theory of Computing (STOC'95)*, pages 140–146, 1995.

[Oja04] Kerry Ojakian. *Combinatorics in bounded arithmetic.* PhD thesis, CMU, 2004.

[Par68] C. Parsons. On a number-theoretic choice schema and its relation to induction. In *Intuitionism and Proof Theory: Proceedings of the summer conference at Buffalo N. Y.*, pages 459–473, 1968.

[Par71] R. Parikh. Existence and feasibility of arithmetic. *Journal of Symbolic Logic*, 36:494 –508, 1971.

[Pat90] M. S. Paterson. Improved sorting networks with o(logn) depth. *Algorithmica*, 5:75–92, 1990.

[PW81a] J. Paris and A.J. Wilkie. Δ_0 sets and induction. In *Proceedings of the Jadwisin Logic Conference*, pages 237–48. Leeds University Press, 1981.

[PW81b] J. Paris and A.J. Wilkie. Models of arithmetic and rudimentary sets. *Bull. Soc. Mathem. Belg., Ser. B*, 33:157 – 69, 1981.

[Raz93] A. Razborov. An equivalence between second-order bounded domain bounded arithmetic and first-order bounded arithmetic. In P. Clote and J. Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 247–277. Clarendon Press, Oxford, 1993.

[Ruz80] Walter L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Science*, 21:218–235, 1980.

[Saz80] V. Sazonov. Polynomial computability and recursivity in finite domains. *Elektronische Informationsverarbeitung und Kybernetik*, 16:319 – 323, 1980.

[SC02] Michael Soltys and Stephen Cook. The proof complexity of linear algebra. In *17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)*, pages 335–344, 2002.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[Sto77] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Sud78] I. H. Sudborough. On the tape complexity of deterministic context-free languages. *J. ACM*, 25(3):405–414, 1978.

[Tak93] G. Takeuti. RSUV isomorphism. In P. Clote and J. Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 364–386. Clarendon Press, Oxford, 1993.

[Tra50] B. Trahtenbrot. The impossibility of an algorithm for the decision problem for finite domains. *Doklady Academii Nauk SSSR*, 70:569–572, 1950. In Russian.

[TS] Neil Thapen and Michael Soltys. Weak theories of linear algebra.

[Val84] L.G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5:363–366, 1984.

[Var82] Moshe Y. Vardi. The complexity of relational query language. In *14th ACM Symp.on Theory of Computing, Springer Verlag (Heidelberg, FRG and NewYork NY, USA)-Verlag*, 1982.

[Ven87] H. Venkateswaran. Properties that characterize LOGCFL. In *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 141–150. ACM Press, 1987.

[Zam96] D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.

[Zam97] D. Zambella. End extensions of models of linearly bounded arithmetic. *Annals of Pure and Applied Logic*, 88(2-3):263–277, 1997.