A second-order theory for NL

Stephen Cook and Antonina Kolokolova
University of Toronto
{sacook,kol}@cs.toronto.edu

Abstract

We introduce a second-order theory V-Krom of bounded arithmetic for nondeterministic log space. This system is based on Grädel's characterization of $\mathtt{NL}$ *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 co*$\mathtt{NL}$ *(and hence for* $\mathtt{NL}$*). This theory has the style of the authors' theory V_1-Horn [APAL 124 (2003)] for polynomial time. Both theories use Zambella's elegant second-order syntax, and are axiomatized by a set 2-BASIC of simple formulae, together with a comprehension scheme for either second-order Horn formulae (in the case of V_1-Horn), or second-order Krom (2CNF) formulae (in the case of V-Krom). Our main result for V-Krom is a formalization of the Immerman-Szelepcsényi theorem that* $\mathtt{NL}$ *is closed under complementation. This formalization is necessary to show that the* $\mathtt{NL}$ *functions are Σ_1^B-definable in V-Krom. The only other theory for* $\mathtt{NL}$ *in the literature relies on the Immerman-Szelepcsényi's result rather than proving it.*

1. Introduction

The two most prominent approaches to complexity from 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.

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. So both approaches study 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 relate to the power of the resulting system?

Each of the complexity classes

$$\mathtt{AC}^0 \subset \mathtt{TC}^0 \subseteq \mathtt{NC}^1 \subseteq \mathtt{NL} \subseteq \mathtt{NC} \subseteq \mathtt{P}$$

among others has been associated with one or more theories of bounded arithmetic. In this paper we are concerned with the class $\mathtt{NL}$ (nondeterministic log space). This class is different from all the others, because the proof that $\mathtt{NL}$ is closed under complementation is difficult (see Immerman or Szelepcsényi, [Imm88, Sze88]). Closure under complementation is necessary for $\mathtt{NL}$ to have a nice associated function class, and hence a nice associated theory.

To our knowledge only one other theory has been associated with the class $\mathtt{NL}$, namely the theory S^{Nlog} of [CT92]. This theory is axiomatized by induction over encodings of $\mathtt{NL}$ Turing machines, and the authors of [CT92] state that it is very awkward and hope that it will be a stepping stone to a better system. We believe that the theory V-Krom that we present here is such a better system.

Our theory V-Krom is in the same style as the theory V_1-Horn for deterministic polynomial time presented in our previous work [CK03]. The system V_1-Horn is a second-order theory (with sorts for numbers and finite sets of numbers, or "strings") which is axiomatized by a set 2-BASIC of simple axioms, together with a comprehension axiom for essentially Grädel's [Grä92] SO-Horn formulae. (These formulae capture polynomial time in second-order finite model theory.) Our new system V-Krom has the same language and same set 2-BASIC of simple axioms, but allows comprehension for essentially S0-Krom formulae instead of SO-Horn formulae. In the same paper [Grä92], Grädel showed that these formulae capture the class $\mathtt{NL}$.

We note that the intuitive reason that the two formula classes capture the two complexity classes is that the satisfiability problem for propositional Horn formulae is complete for polynomial time, whereas the satisfiability problem for Krom formulae (2CNF) is complete for coNL (and hence for NL).

In general, in the second-order setting, a relation $R(\bar{x}, \bar{X})$ has natural number variables $\bar{x}$ and string variables $\bar{X}$. This relation is in a given complexity class C if, when inputs $\bar{x}$ are presented in unary notation and inputs $\bar{X}$ are presented as bit strings, an appropriate machine or circuit can determine whether $R(\bar{x}, \bar{X})$ holds using specified resources. In the case of NL, the machine is a nondeterministic Turing machine, and the resource bound in space $O(\log n)$. A string valued function $F(\bar{x}, \bar{X})$ is in the associated complexity class FC if its length $|F(\bar{x}, \bar{X})|$ is bounded by a polynomial in $(\bar{x}, |\bar{X}|)$, and its bitgraph G_F is in C, where $G_F(\bar{x}, \bar{X})$ holds iff the i-th bit of $F(\bar{x}, \bar{X})$ is 1.

However, this class FC is not closed under composition unless C is closed under complementation, assuming that the function which interchanges 0 and 1 is in the class. This is why the function class FNL did not become interesting before the Immerman-Szelepcsényi theorem.

Our main result for V-Krom is to show that V-Krom proves the Immerman-Szelepcsényi theorem. (This was not shown for the theory S^{Nlog} of [CT92].) We do this by showing how to formalize the proof given in [Imm99]. After this, we prove that V-Krom "captures" FNL in the standard sense that a function is Σ_1^B-definable in V-Krom iff it is in FNL.

2. System V-Krom.

The system V-Krom defined in this work belongs to a family of second-order systems with syntax similar to that of Zambella's Σ_i^b-comp ([Zam96]). The language of V-Krom is $\mathcal{L}_A^2 = \{0, 1, +, \cdot, |\,|; <, =, \in\}$, which is 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 lowercase letters, called *number variables*) are natural numbers; second-order objects (denoted by upper-case letters, called *string variables*) 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.

Bounded number quantifiers are defined in the usual way. A bounded string quantifier $\exists X < t\phi$ stands for $\exists X(|X| < t \wedge \phi)$, and $\forall X < t\phi$ stands for $\forall X(|X| < t \supset \phi)$. We use Σ_0^B to denote the set of formulae with all number quantifiers bounded and no string quantifiers, and Σ_1^B denotes the class of formulae which begin with zero or more bounded existential string quantifiers followed by a Σ_0^B formula. The classes Σ_i^B and Π_i^B are defined similarly, where in all cases string quantifiers must be bounded and appear in front of the formula.

The system $\mathbf{V}^i, i \geq 0$ is axiomatized by the 2-BASIC axioms together with a comprehension scheme for Σ_i^B formulae. For $i \geq 1$, $\mathbf{V}^i$ is RSUV isomorphic to the first-order theory S_2^i. The system V^0 corresponds to the complexity class uniform AC^0.

Similarly to V_1-Horn from [CK03], the system V-Krom is defined as 2-BASIC axioms plus the comprehension scheme over a version of $SO\exists$-Krom formulae over $\mathcal{L}_A^2$. By Grädel's result, $SO\exists$-Krom formulae capture NL in the finite model theory setting (in presence of order).

Definition 2.1. A quantifier-free formula $\phi(\bar{P}, \bar{X}, \bar{x})$ is Krom with respect to $\bar{P}$ if ϕ is a CNF formula in which each occurrence of each P_i is as a *P-literal* $P_i(t)$ or $\neg P_i(t)$ in a clause, where t is a term (ϕ may not involve any term of the form $|P_i|$.) Further, each clause may contain at most two P-literals, although it may contain any number of other literals.

That is, a 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.

Definition 2.2. A formula is Σ_1-Krom if it is of the form:

$$\exists \bar{P} \forall \bar{x} < \bar{n} \phi(\bar{P}, \bar{x}, \bar{A}, \bar{a})$$

where $\bar{A}$ and $\bar{a}$ are free second- and first order variables, $\bar{n}$ are terms not involving $\bar{x}$ or $\bar{P}$ and ϕ is Krom with respect to $\bar{P}$.

Let $\phi(i, \bar{a}, \bar{X})$ be a Σ_1-Krom formula with first-order free variables i and $\bar{a}$ and second-order free variables $\bar{X}$. Then a comprehension axiom for ϕ and a variable b is

$$\exists Z \forall i < b(Z(i) \leftrightarrow \phi(i, \bar{a}, \bar{X})). \qquad (\Sigma_1\text{-Krom-comp})$$

Note that b could be used to bound the length of Z without changing the meaning. Since the only way of introducing second-order variables into the system is by applying the comprehension axiom, every second-order variable could be bounded by a polynomial in first-order variables.

Definition 2.3. The theory V-Krom is the theory over $\mathcal{L}_A^2$ axiomatized by 2-BASIC axioms together with a comprehension scheme Σ_1-Krom-comp over Σ_1-Krom formulae.

3. Basic properties of V-Krom

Many of the basic properties of V-Krom are proved in the same way they are for V_1-Horn, so in this section we frequently refer to V_1-Horn as presented in [CK03]. Since

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:$x \cdot (y+1)=(x \cdot y)+x$
B7: $0 \leq x$	B9: $x \leq y \wedge y \leq z \to x \leq z$
B8: $x \leq x+y$	B10: $(x \leq y \wedge y \leq x) \to x=y$
B11: $x \leq y \vee y \leq x$	B12: $x \leq y \leftrightarrow x < y+1$
B13: $x \neq 0 \to \exists y(y+1=x)$	
L1: $X(y) \to y < \lvert X \rvert$	L2: $y+1=\lvert X \rvert \to X(y)$

Table 1. The 2-BASIC **axioms**

the length function $|X|$ tells us that a finite nonempty set has a maximum element, it is easy to show that V-Krom proves the LNP for finite sets, as well as the induction axiom

$$X(0) \wedge \forall y < z(X(y) \to X(y+1)) \to X(z)$$

From this and Σ_1-Krom comprehension it follows that V-Krom proves the induction scheme and LNP for Σ_1-Krom formulae, and in particular for open formulae. Using a standard pairing function V-Krom can code a tuple $\bar{x}$ of numbers as a single number, and many of our definitions and results implicitly assume this coding. Also V-Krom proves the replacement scheme for Σ_1-Krom formulae, just as V_1-Horn proves replacement for Σ_1^B-Horn formulae.

3.1. Simulating Σ_0^B formulae.

Existential first-order quantifiers are not allowed in a Σ_1-Krom formula. That is, a Σ_0^B formula is not automatically a Σ_1-Krom formula, though Σ_1-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-Krom. A similar result holds in V_1-Horn, but the construction for V-Krom is quite different.

Theorem 3.1. *For every Σ_0^B formula ψ there is a Σ_1-Krom formula ψ^* such that*

$$V\text{-}Krom \vdash \psi \leftrightarrow \psi^*$$

From this theorem, the following corollary is easy.

Corollary 3.2. *Comprehension over Σ_0^B formulae is a theorem of V-Krom. Thus, V-Krom is an extension of V^0 and proves the Σ_0^B induction axioms.*

The idea behind the proof of Theorem 3.1 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. □

With the help of Corollary 3.2, V-Krom proves induction on both Σ_0^B and Σ_1-Krom formulae. By using the comprehension scheme for both formula classes we can justify induction over $\Sigma_0^B(\Sigma_1$-Krom) formulae, and in fact over formulae built by nesting Σ_1-Krom formulae with bounded quantifiers and the Boolean connectives. This idea is used implicitly in later sections.

4. V-Krom(TrCl)

In this section we show how to introduce the transitive closure operator into V-Krom, and use it to prove the Immerman-Szelepcsényi theorem. We show that V-Krom can formalize the proof given in [Imm99], sections 9.2–9.5.

4.1. Definitions

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:

$$Cond(\phi,R,n) \equiv$$
$$\forall x,y,z < n(R(x,x) \wedge (\phi(x,y) \wedge R(y,z) \to R(x,z)))$$

We will write just $Cond(\phi,R)$ when n is clear from the context.

Remark 4.1. It is important for the proof of Theorem 4.3 that the negation of the RHS of (AxTC) is equivalent to a Σ_1-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 4.4).

Now we define the transitive closure relation $TrCl\phi$ to be the intersection of all relations R satisfying $Cond(\phi,R)$.

$$TrCl_{x,y}\phi(x,y)[a,b,n] \leftrightarrow \forall R(Cond(\phi,R,n) \to R(a,b)) \qquad \text{(AxTC)}$$

We want to extend the vocabulary of V-Krom by including instances of TrCl as defined above.

Definition 4.2. The class Σ_0^B(TrCl) is defined inductively as follows:

(i) Every quantifier-free formula of V-Krom is in Σ_0^B(TrCl)

(ii) If ϕ is in Σ_0^B(TrCl), then so is $TrCl_{x,y}\phi(x,y)[a,b,n]$

(iii) Every Σ_0^B combination of formulae in Σ_0^B(TrCl) is in Σ_0^B(TrCl)

The class Σ_0^B(TrCl$^+$) is defined in the same way, except in (iii) we allow only Σ_0^B combinations with positive occurrences of Σ_0^B(TrCl) formulae.

The system V-Krom(TrCl) is V-Krom augmented with the class Σ_0^B(TrCl) of formulae, and has (AxTC) for each ϕ in Σ_0^B(TrCl).

Since the only new axioms in V-Krom(TrCl) are definitions of new relations, it is a conservative extension of V-Krom.

Theorem 4.3. *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-Krom formula if ϕ is quantifier-free (see Remark 4.1). The theorem follows by induction on the depth of nesting of $TrCl$ formulae, using the discussion following the proof of Corollary 3.2. □

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 $a = b = 0$.

4.2. Properties of transitive closure

In this section we make frequent tacit use of Theorem 4.3.

First we show that V-Krom proves the transitivity of the transitive closure relation.

Lemma 4.4. *Let $TrCl(x,y)$ stand for $TrCl_{u,v}\phi[x,y]$. 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

$$TrCl^r(a,b) \leftrightarrow \forall R(Cond^r(\phi,R,n) \rightarrow R(a,b))$$

Lemma 4.5. *V-Krom proves*

$$TrCl^r(a,b) \leftrightarrow TrCl_{u,v}\phi[a,b,n]$$

Proof. By an argument similar to the proof of Lemma 4.4, 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. □

4.3. 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-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 standard tupling functions.

Theorem 4.6. *Any Σ_0^B(TrCl$^+$) formula ϕ is equivalent to $TrCl_{\bar{x},\bar{x}'}\psi[\bar{0},\bar{n}]$, 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.*

Proof. (Sketch) 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-Krom, the existence of such witnesses is guaranteed by Σ_1-Krom comprehension axioms. □

4.4. Relating Σ_1-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-Krom formula by the axioms of transitive closure. This section shows how to convert an arbitrary Σ_1-Krom formula to negation of a $\Sigma_0^B(TrCl)$ formula; by appealing to Theorem 4.6 it is equivalent to a negated transitive closure of a quantifier-free formula.

4.4.1 $SO\exists$-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-Krom formula:

$$\Phi \equiv \exists P \forall x_1 < n_1 \dots \forall x_k < n_k \psi(P, \bar{x}), \tag{1}$$

$$\text{where } \psi(P, \bar{x}) \equiv \bigwedge^{m} (L_j(t_j(\bar{x})) \vee L'_j(t'_j(\bar{x})) \vee \phi_j(\bar{x})).$$

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: Convert a $SO\exists$-Krom formula to propositional 2-CNF. Make a conjunction of $n_1 \cdot \ldots \cdot n_k$ copies of the formula, one for each $\langle x_1 \dots x_k \rangle$, and evaluate the terms in each copy on a corresponding value of $\langle x_1 \dots x_k \rangle$. 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 $\neg p_i$).

4.4.2 Construction

Here is how we construct a formula equivalent to Φ from (1), 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

$$(\neg\phi_j(\bar{x}) \wedge t_j(\bar{x}) = u \wedge \neg z_j = s \wedge t'_j(\bar{x}) = v \wedge z'_j = s')$$
$$\vee(\neg\phi_j(\bar{x}) \wedge t'_j(\bar{x}) = u \wedge \neg z'_j = s \wedge t_j(\bar{x}) = v \wedge z_j = s')$$

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 (1):

$$\exists i < t(TrCl_{us,vs'}\psi'(u, s, v, s')[i0, i1] \wedge TrCl_{us,vs'}\psi'(u, s, v, s')[i1, i0]) \quad \text{(NegKrom)}$$

4.4.3 Proof of correctness

Theorem 4.7. *Let $\Phi(\bar{X}, \bar{y})$ be a Σ_1-Krom formula. Then there exists a quantifier-free formula ϕ and tuples $\bar{0}, \bar{n}$ such that*

$$V\text{-}Krom(TrCl) \vdash \Phi(\bar{X}, \bar{y}) \leftrightarrow \neg TrCl_{\bar{x},\bar{x}'}\phi[\bar{0}, \bar{n}]$$

Proof. By Theorem 4.6 (normal form theorem) it suffices to prove equivalence between Φ in (1) and the negation of (NegKrom).

Let $\Phi = \exists P \forall \bar{x} < \bar{n}\psi(P, \bar{x})$ be the formula (1). We need to prove the equivalence

$$\exists P \forall \bar{x} < \bar{n}\psi(P, \bar{x}) \Leftrightarrow \forall i < t \exists Q[Cond(\psi', Q, \langle t, 2\rangle) \wedge (\neg Q(i0, i1) \vee \neg Q(i1, i0))] \quad (2)$$

where

$$Cond(\psi', Q, \langle t, 2\rangle) \equiv \forall u, v, w < t \forall s, s', s'' < 2 \; Q(us, us) \wedge (\psi'(us, vs') \wedge Q(vs', ws'') \rightarrow Q(us, ws''))$$

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 $P(v) \rightarrow P(w)$ and $\neg(P(u) \rightarrow P(w))$. 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

$$Force(i, s) \equiv Q(i\neg s, is) \vee (\exists j < t \; Q(j0, j1) \wedge Q(j1, is) \vee Q(j1, j0) \wedge Q(j0, is))$$

$Force(i, 1)$ holds if $P(i)$ is directly forced to $\top$, that is, if either $(\neg P(i) \rightarrow P(i))$ or $(L(j) \rightarrow P(i))$ 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

$$Assign(i, s) \equiv \exists j \leq t \forall k \leq t UnForced(j) \wedge (UnForced(k) \rightarrow k \geq j) \wedge Q(j1, is).$$

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 $P(i)$). □

4.5. Immerman-Szelepcsényi's construction

Now we can formalize Immerman's construction.

Theorem 4.8. *For any Σ_0^B(TrCl$^+$) formula ϕ there is a Σ_0^B(TrCl$^+$) formula ϕ' such that*

$$V\text{-}Krom(TrCl) \vdash \phi \leftrightarrow \neg\phi'$$

Thus, by theorem 4.7 and $AxTC$, for any Σ_1-Krom formula Φ there exists a Σ_1-Krom formula Φ' such that V-Krom $\vdash \Phi \leftrightarrow \neg\Phi'$.

We would like to construct a formula $NegTrCl(\psi, x, n)$ with only positive occurrences of transitive closure operator such that

$$V\text{-Krom} \vdash \neg TrCl_{u,v}\psi[0, x] \leftrightarrow NegTrCl(\psi, x, n).$$

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 showing, inductively, 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.

Since the construction is based on counting, we we introduce a notion of "counters" to formalize Immerman's proof.

Definition 4.9. A *counter* (transitive closure counter) is a formula of the form $\text{CNT}(vc, v'c') \equiv (c' = c+1 \wedge \phi(v, v', c) \vee c' = c \wedge \tilde{\phi}(v, v', c))$, where ϕ and $\tilde{\phi}$ are Σ_0^B(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. $TrCl_{vc,v'c'}\text{CNT}[yd, ze]$ 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 $TrCl_{vc,v'c'}\text{CNT}[yd, ze]$ 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 $DIST(x, d) \equiv TrCl_{vc,v'c'}\alpha[00, xd]$. 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 4.10. *Let* $\text{LCNT}(vc, v'c')$ *be an* exact *linear counter. Then*

$$V\text{-}Krom \vdash \forall y \leq n \exists! z \leq n TrCl_{vc,v'c'}\text{LCNT}[01, yz]$$

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 4.11. *Let* $\text{LCNT}_1(vc, v'c')$ *and* $\text{LCNT}_2(vc, v'c')$ *be two linear counters with* $\forall v \leq n\phi_1(v) \rightarrow \phi_2(v)$ *and* $\tilde{\phi}_2(v) \rightarrow \tilde{\phi}_1(v)$*, 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 $v < y$ $\phi_2(v+1) \wedge \neg\phi_1(v+1)$,*

$$TrCl_{vc,v'c'}\text{LCNT}_2[01, yd] \rightarrow \neg TrCl_{vc,v'c'}\text{LCNT}_1[01, yd];$$

otherwise (that is, if $\forall v < y(\phi_2(v + 1) \rightarrow \phi_1(v + 1))$,

$$TrCl_{vc,v'c'}\text{LCNT}_2[01, yd] \rightarrow TrCl_{vc,v'c'}\text{LCNT}_1[01, yd]$$

Proof. The proof is by induction on y. We omit details. □

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 4.10 with LCNT $= \gamma$; the second part by applying Lemma 4.11 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 $DIST(x,n-1) \leftrightarrow TrCl_{u,v}\psi[0,x]$.

5. Definability in V-Krom

In this section the goal is to prove that V-Krom indeed captures NL tightly.

Definition 5.1. A predicate $R(\bar{x},\bar{Y})$ is Δ_1^B-definable in a second-order system of arithmetic V if there are Σ_1^B formulas ϕ and ψ such that R satisfies

$$R(\bar{x},\bar{Y}) \leftrightarrow \phi(\bar{x},\bar{Y}))$$

and

$$V \vdash (\phi(\bar{x},\bar{Y}) \leftrightarrow \neg\psi(\bar{x},\bar{Y})$$

V captures a complexity class C if the Δ_1^B-definable predicates of V are exactly the predicates of C.

Theorem 5.2. *A predicate $R(\bar{x},\bar{Y})$ is Δ_1^B-definable in V-Krom iff it is in* NL*.*

Proof. By Grädel's theorem, every co-NL predicate (and by Immerman-Szelepcsényi every NL predicate) is definable by a Σ_1-Krom formula. From Theorem 4.8 and the fact that Σ_1-Krom formulae are also Σ_1^B formulae, it follows that every NL predicate is Δ_1^B-definable in V-Krom. The converse follows from Theorem 5.7 (witnessing) below. □

We define the function class FNL associated with NL in the standard way for the second-order setting (see [Coo02, Coo04]): number functions are defined from NL predicates using bounded minimization, and string functions must be polybounded and have NL bit graphs. The following definition provides a way of introducing function symbols for FNL functions in a theory. It makes sense because the NL predicates are precisely those definable by Σ_1-Krom formulae.

Definition 5.3. A number function $f : \mathbb{N}^k \times (\{0,1\}^*)^l \to \mathbb{N}$ is NL-definable iff there is a formula $\phi \in \Sigma_1$-Krom and a polynomial p such that f has defining axiom

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

A string function $F : \mathbb{N}^k \times (\{0,1\}^*)^l \to \{0,1\}^*$ is NL-definable iff there is a formula $\phi \in \Sigma_1$-Krom and a polynomial p such that F has defining axiom

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

Lemma 5.4. *Let ϕ be a Σ_0^B formula with possible occurrences of string and number function symbols from the definition 5.3. Then there exists a Σ_1-Krom formula with no occurrences of function symbols that is provably in V-Krom equivalent to ϕ.*

Proof. Structural induction on ϕ, using Theorems 4.6, 4.7, and 4.8. □

Definition 5.5. 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.

Theorem 5.6. *A function (string or number) is Σ_1^B-definable in V-Krom iff it is in FNL.*

Proof. NL number functions are Σ_1^B-definable because V-Krom proves bounded minimization for Σ_1-Krom formulae, and NL string functions are Σ_1^B-definable because V-Krom proves Σ_1-Krom comprehension. The converse follows from the following witnessing theorem. □

Theorem 5.7 (Witnessing theorem for V-Krom). *If V-Krom $\vdash \exists Z B(\bar{x},\bar{Y},Z)$, where $B \in \Sigma_1^B$, then there is a string function $F(\bar{x},\bar{Y}) \in FNL$ such that*

$$V\textit{-Krom}, AX(F) \vdash B(\bar{x},\bar{Y},F(\bar{x},\bar{Y})),$$

where $AX(F)$ is a defining axiom for F.

The proof is based on a cut-elimination argument using the method pioneered by Buss [Bus86] (see [Coo02] for a second-order version). The idea is to put proof of the Σ_1^B formula into a normal form in which every formula is Σ_1^B. Unfortunately the Σ_1-Krom comprehension axioms are Σ_2^B formulae, so we need a modified system in which the comprehension axioms are indeed Σ_1^B formulae.

5.1. Σ_1^B-axiomatized version of V-Krom

Here we give a system that has comprehension formulae which are "general Σ_1^B". That is, they have a prenex form in which bounded number quantifiers precede a Σ_1^B formula. The main thing is that they can be easily witnessed in NL.

By Theorem 4.8 we know that for every Σ_1-Krom formula ϕ there is a Σ_1-Krom formula $\tilde{\phi}$ such that V-Krom $\vdash$ $\phi \leftrightarrow \neg\tilde{\phi}$. Now we can replace a negated occurrence of ϕ in the comprehension axiom of V-Krom by $\tilde{\phi}$.

Definition 5.8. The system $\tilde{V}$-Krom consists of axioms 2-BASIC, together with sequents $\phi, \tilde{\phi} \longrightarrow$ and $\longrightarrow \phi, \tilde{\phi}$ for every $\phi \in \Sigma_1$-Krom, and a comprehension scheme

$$\exists Z \forall y < b((Z(y) \to \phi(y)) \wedge (\neg Z(y) \to \tilde{\phi}(y))). \qquad (\Sigma_1\text{-Krom-comp}')$$

Lemma 5.9. *The systems V-Krom and $\tilde{V}$-Krom have the same theorems.*

Proof. Since V-Krom proves $\phi \leftrightarrow \neg\tilde{\phi}$, it suffices to observe that the revised scheme (Σ_1-Krom-comp$'$) is equivalent to the original scheme (Σ_1-Krom-comp) under the assumption $\phi \leftrightarrow \tilde{\phi}$. □

Lemma 5.10. *The scheme (Σ_1-Krom-comp$'$) is equivalent in $\tilde{V}$-Krom to a Σ_1^B formula.*

Proof. Consider the subformula of Σ_1-Krom-comp$'$ with Z as a free variable. Now it is a Σ_1-Krom formula, preceded by a universal first-order quantifier. Let $\phi \equiv \exists \bar{P}' \forall \bar{x} \le t(\bar{a}, \bar{X}) \psi(y, \bar{x}, \bar{P}', \bar{a}, \bar{X})$ and $\tilde{\phi} \equiv \exists \bar{Q}' \forall \bar{x}' \le t'(\bar{a}, \bar{X}) \bar{\tilde{\psi}}(y, \bar{x}', \bar{Q}', \bar{a}, \bar{X})$; assume without loss of generality that $t = t'$. Putting the subformula under $\exists Z \forall y < b$ in prenex form, and encoding, using pairing function, vectors of second-order variables as single variables, get

$$\exists P' \exists Q' \forall \bar{x}, \bar{x}' \le t(\bar{a}, \bar{X})(Z(y) \to \psi(y, \bar{x}, P')) \wedge(\neg Z(y) \to \tilde{\psi}(y, \bar{x}', Q')).$$

Applying replacement, obtain

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

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). □

5.2. Proof of the witnessing theorem.

Since V-Krom and $\tilde{V}$-Krom are equivalent theories, to prove Theorem 5.7 if suffices to prove the statement with $\tilde{V}$-Krom replacing V-Krom. The proof follows the same steps as the proof of V^0 witnessing theorem in [Coo02]. The only difference is in proving the base case, the case of comprehension axiom.

Lemma 5.11. *The string quantifiers in Σ_1-Krom-comp$'$ can be witnessed by* `NL` *functions.*

Proof. By Lemma 5.10 and using pairing function to combine several second-order variables into one, Σ_1-Krom-comp$'$ is equivalent to the following formula (omitting the free variables):

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

It is very easy to witness Z: simply use a function defined by the bit graph of ϕ. To witness P and Q we again appeal to transitive closure.

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 (that is, there exists $Z'(a, b) \leftrightarrow AxTC(a, b)$ and $Z(a, b) \leftrightarrow \neg Z'(a, b)$ for $a, b < n$). Now for all $i \le n$ $P^{[i]}$ and $Q^{[i]}$ can be defined by the construction from Section 4.4.3, using TC_ψ and $TC_{\tilde{\psi}}$ respectively instead of Q and $\neg Q$ in the formula (2). By Lemma 5.4, a Σ_0^B formula with occurrences of TC is equivalent to a Σ_1-Krom formula, which in turn defines an NL function, which is a witnessing function for $P^{[y]}$ and $Q^{[y]}$. From there we obtain functions $F_{wc}(a, b, y)$ and $F_{\widetilde{wc}}(a, b, y)$ (with free variables of ϕ and $\tilde{\phi}$), which witness P and Q, respectively. □

The proof of the witnessing theorem here uses proof-theoretic techniques similar to those of Buss' original proof of S_2^1 witnessing theorem from [Bus86]. We use Gentzen-style sequent calculus system, extended by second-order quantifier introduction rules; such system is anchored if the cut rule can only apply to axioms (logical or non-logical).

We start by considering an anchored LK^2-$\tilde{V}$-Krom proof of $\longrightarrow \exists Z B(\bar{a}, \bar{X}, Z)$. Since it is anchored, the cut rule is only applied to formulae in axioms of $\tilde{V}$-Krom. The last formula in the proof is Σ_1^B, so every formula in the proof that is Σ_1^B as well.

The most interesting case is the base case. Suppose that the sequent is an axiom of $\tilde{V}$-Krom. If it only involves open axioms B1-B13, L1,L2, then no witnessing function is necessary. If it is an instance of comprehension scheme, the three quantifiers are witnessed according to Lemma 5.11.

The remaining cases are $\longrightarrow \phi, \tilde{\phi}$ and $\phi, \tilde{\phi} \longrightarrow$. The second case does not need witnessing, and the first case can again be witnessed by construction from Lemma 5.11, omitting y.

The rest of the proof of witnessing theorem is exactly the same as in the case of V^0 from [Coo02].

6. V-Krom is finitely axiomatizable.

Since it is possible to encode Σ_1-Krom satisfiability as a Σ_1-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.

We know that V^0, axiomatized by 2-BASIC with comprehension scheme over Σ_0^B formulae, is finitely axiomatizable (see [CK03] for the proof). 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-Krom comprehension axiom scheme. If we can show that finitely many occurrences of Σ_1-Krom comprehension are sufficient, we prove that V-Krom is finitely axiomatizable.

In proving Theorem 4.7 we showed that every Σ_1-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 4.7 shows that $\Phi(\bar{X}, y, \bar{a})$ is equivalent to the RHS of (2), and this is provable in V^0. Let $\Psi(y, E)$ be the result of replacing each occurrence of ψ' in the RHS of (2) by E. Then it suffices to add the following single comprehension axiom for Ψ to V^0 to get V-Krom.

$$\exists Z \forall y < b(Z(y) \leftrightarrow \Psi(y, E))$$

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-Krom formula Φ.

7. Future work

Another natural way of representing NL is to define a system of arithmetic by augmenting V^0 by adding a string function $TC(E, n)$, together with axioms defining it as the transitive closure of the edge relation E restricted to the nodes $\{0, ..., n-1\}$. This could be made a universal theory by adding $\mathtt{AC}^0$ functions, similar to the way that V^0 is made into a universal theory $\overline{V^0}$ in [Coo04]. The resulting theory should be a universal conservative extension of V-Krom.

A more interesting direction, however, is to extend this Σ_1^B-definability result to classes of formulae other than $SO\exists$-Krom, and thus to other complexity classes. Suppose that a class of formulae is (provably) closed under $\mathtt{AC}^0$ reductions and its descriptive complexity and complexity of satisfiability coincide. Construct a system of arithmetic by adding comprehension over that class of formulae to V^0, just as V-Krom is V^0 with comprehension over Σ_1-Krom formulae. Then Σ_1^B-definability properties of this system should be similar to V-Krom: namely, descriptive complexity and complexity of Σ_1^B-definable predicates in that system should be the same.

References

[Bus86] S. Buss. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[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.

[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.

[CT92] P. Clote and G. Takeuti. Bounded arithmetic for NC, ALOGTIME, L and NL. *Annals of Pure and Applied Logic*, 56:73 – 117, 1992.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Springer Verlag, 1995.

[Grä92] E. Grädel. Capturing Complexity Classes by Fragments of Second Order Logic. *Theoretical Computer Science*, 101:35–57, 1992.

[Imm88] Immerman. Nondeterministic space is closed under complementation. In *SCT: Annual Conference on Structure in Complexity Theory*, 1988.

[Imm99] N. Immerman. *Descriptive complexity*. Springer Verlag, New York, 1999.

[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.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

[Zam96] D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.