On theories of bounded arithmetic for NC^1

Emil Jeřábek*

Institute of Mathematics of the Academy of Sciences

Žitná 25, 115 67 Praha 1, Czech Republic, email: jerabek@math.cas.cz

December 9, 2008

Abstract

We develop an arithmetical theory VNC^1_* and its variant $\overline{VNC}^1_*$, corresponding to "slightly nonuniform" NC^1. Our theories sit between VNC^1 and VL, and allow evaluation of log-depth bounded fan-in circuits under limited conditions. Propositional translations of $\Sigma^B_0(L_{\overline{VNC}^1_*})$-formulas provable in $\overline{VNC}^1_*$ admit L-uniform polynomial-size Frege proofs.

1 Introduction

In proof complexity, there is a well-known general correspondence between theories of bounded arithmetic, complexity classes, and propositional proof systems (see e.g. [12, 6, 8, 9]). A theory T corresponds to a complexity class C if the provably total computable functions of T are the C-functions. A propositional proof system P corresponds to T if the propositional translations of theorems of T of certain complexity have polynomial-size proofs in P, and T proves a reflection principle for P.

Here we are particularly concerned about theories corresponding to variants of the class NC^1. Several theories corresponding to *uniform* NC^1 (i.e., $ALOGTIME$, U_E-uniform NC^1) and to the Frege propositional proof system have been described in the literature: an equational theory ALV by Clote [5], theories AID and $AID+\Sigma^b_0$-CA by Arai [1], and a second-order theory VNC^1 by Cook and Morioka [7]. (All these theories are more or less equivalent: VNC^1 is $RSUV$-isomorphic to $AID+\Sigma^b_0$-CA, which is in turn a conservative extension of ALV.)

Uniform NC^1 is a robust and well-behaved complexity class, but it is too strict for certain applications, namely those involving circuit evaluation. Nonuniform complexity classes usually consist of languages definable by a family of polynomial-size Boolean circuits satisfying certain requirements (e.g., concerning their depth, fan-in, or available connectives): this holds for example for nonuniform AC^k, NC^k, TC^0, P; in particular, nonuniform NC^1-languages are given by a family of bounded fan-in circuits of logarithmic depth. Typically, the corresponding uniform class consists of languages definable by a sufficiently uniform family of the same kind of circuits, and moreover, the class includes the universal language which

*Supported by grant IAA1019401 of GA AV ČR, and grant 1M0545 of MŠMT ČR.

evaluates circuits of this kind described in a natural way by binary strings. This is not true for NC^1. Even $DLOGTIME$-uniform (i.e., U_D-uniform) families of log-depth circuits define a class (presumably) still larger than uniform NC^1; we can only define uniform NC^1 using circuits by employing the artificial description by so-called extended connection languages of Ruzzo [13]. Likewise, the universal evaluator for log-depth circuits is (presumably) not in NC^1 (even nonuniform).

Consequently, VNC^1 (and friends) do not prove that one can evaluate log-depth circuits, or even a uniformly given (say, definable by a Σ^B_0-formula) sequence of log-depth circuits. There are situations where evaluation of such circuits would be desirable in an NC^1-theory. The particular application we have in mind, and the main motivation for this work, is the paper [11], which aims at formalizing a version of the Ajtai–Komlós–Szemerédi sorting network in bounded arithmetic (under the assumption that we can formalize construction of suitable expander graphs). On the one hand, we need the formalization to proceed in an NC^1-theory, and in particular, in a theory which translates to polynomial-time Frege proofs: the point is that this implies polynomial simulation of the sequent calculus (i.e., Frege) by the monotone sequent calculus MLK, using results of Atserias et al. [2]. On the other hand, the sorting network is essentially a monotone log-depth circuit which we need to evaluate; it is uniformly described, but its extended connection language is not available.

To address these issues, we introduce new theories VNC^1_* and $\overline{VNC}^1_*$, corresponding to a subclass of NC^1 slightly larger than uniform NC^1, which allow evaluation of sufficiently uniform families of log-depth circuits. We work with second-order theories in the spirit of Zambella [14]. The theory VNC^1_* is formulated in the usual language of second-order bounded arithmetic; it includes V^0, and a derivation rule allowing to evaluate a kind of monotone log-depth bounded fan-in circuits described by formulas without second-order parameters which are provably Δ^B_1. The theory $\overline{VNC}^1_*$ has a richer language $L_{\overline{VNC}^1_*}$ including comprehension function symbols for Σ^B_0-formulas, and function symbols for evaluation of monotone log-depth bounded fan-in circuits described by open formulas (in the extended language) without second-order parameters.

In Section 4, we prove basic properties of our new theories: VNC^1_* contains VNC^1 and is contained in VL, $\overline{VNC}^1_*$ is an open theory conservatively extending VNC^1_* (more precisely, it is an extension of VNC^1_* by Σ^B_1-definitions), VNC^1_* is Σ^B_1-axiomatizable, $\exists\Sigma^B_1$-formulas provable in $\overline{VNC}^1_*$ are witnessed by terms in $\overline{VNC}^1_*$ (in particular, provably Δ^B_1-formulas of $\overline{VNC}^1_*$ are equivalent to open formulas), the provably total computable functions of VNC^1_* include uniform NC^1-functions, and are included in L-uniform NC^1-functions, and VNC^1_* extended by the axiom of choice for $\exists\Sigma^B_1$-formulas is $\exists\Sigma^B_1$-conservative over VNC^1_*. To show the latter, we prove a general theorem on conservativity of the axiom of choice over theories meeting certain requirements. In Section 5 we show that propositional translations of $\Sigma^B_0(L_{\overline{VNC}^1_*})$-theorems of $\overline{VNC}^1_*$ have L-uniform polynomial-size Frege proofs.

2 Complexity classes

We recall that a (bounded fan-in) circuit in n inputs is a directed acyclic graph whose nodes are labelled by gate types $\wedge$, $\vee$, $\neg$, or input variables x_i, $i < n$. Input nodes have fan-in 0, $\neg$-gates have fan-in 1, and $\wedge$ and $\vee$-gates have fan-in 2. One node of the circuit is designated as the output node. The circuit computes a Boolean function $f\colon 2^n \to 2$ in the obvious way. The depth of a circuit is the maximal length of a path in the circuit. A formula is a circuit in which all nodes save the output have fan-out 1.

If C is any class of languages, we define FC to be the class of functions $f(\vec{x})$ such that $|f(\vec{x})|$ is at most polynomial in $|\vec{x}|$, and the bit-graph

$$\{\langle \vec{x}, i\rangle \mid \text{the } i\text{th bit of } f(\vec{x}) \text{ is } 1\}$$

is in C. We will sometimes call functions $f \in FC$ just C-functions.

A language L is in *nonuniform* NC^1 if there exists a family $\{C_n \mid n \in \omega\}$ of circuits such that C_n computes the characteristic function of $L \cap 2^n$, and the depth of C_n is $O(\log n)$ (in short, C_n is a log-depth circuit). Equivalently, L is in nonuniform NC^1 if it is computable in a similar way by a family of polynomial-size formulas.

Let U be a complexity class. A language L is in U*-uniform* NC^1 if it is computable by a sequence $\{C_n \mid n \in \omega\}$ of log-depth circuits such that given n in unary, we can compute the description of C_n by a U-function. Since this definition may be sensitive to details of the chosen representation of circuits, we make it more precise using the terminology of Ruzzo [13]. Given a node x in a circuit C, we fix an ordering of its input nodes, and denote by $x(i)$ the ith input of x. The *direct connection language* $L_{DC}(C)$ of a family of circuits $C = \{C_n \mid n \in \omega\}$, where C_n has n inputs, is a set of tuples $\langle n, x, p, y\rangle$, where n is an integer given in unary, x is a binary string identifying a node in a circuit, $p \in \{\varepsilon, 0, 1\}$, and y is either another string denoting a node, or a gate type from $\{x_i, \wedge, \vee, \neg\}$. It is defined by

$$L_{DC}(C) = \{\langle n, x, \varepsilon, t\rangle \mid \text{node } x \text{ in } C_n \text{ is a } t\text{-gate}\} \cup \{\langle n, x, p, y\rangle \mid p \in \{0,1\}, x(p) = y \text{ in } C_n\}.$$

We define U-uniform NC^1 to consist of languages L computable by a family C of log-depth circuits with node labels of length $|x| = O(\log n)$ such that $L_{DC}(C) \in U$.

Fully *uniform* NC^1 is defined as $ALOGTIME$, the languages computable by an alternating Turing machine in $O(\log n)$ steps. Here and below, Turing machines supposed to work in sublinear time do not have the usual input tape. Instead, there is a special index type, and read states. If the machine enters a read state with a, k written on the index tape, where a is a symbol of the input alphabet, and k is a binary integer, it continues in one of two given states according to whether the kth symbol of the input is a.

Uniform NC^1 (presumably) does not coincide with U-uniform NC^1 for any class U. However, we can define it using circuits as follows. We extend the $x(i)$ notation so that if p is a binary string, $x(p)$ is the node we obtain by following the path which starts in x, and moves to the left or right input according to successive bits of p. The *extended connection language*

$L_{EC}(C)$ of a family $C = \{C_n \mid n < \omega\}$ of circuits is defined by

$$L_{EC}(C) = \{\langle n, x, \varepsilon, t\rangle \mid \text{node } x \text{ in } C_n \text{ is a } t\text{-gate}\} \\ \cup \{\langle n, x, p, y\rangle \mid p \in \{0,1\}^*, 0 < |p| \leq \log n, x(p) = y \text{ in } C_n\}.$$

Then a language L is in uniform NC^1 if and only if it is computable by a family C of log-depth circuits such that $L_{EC}(C)$ is computable in $DTIME(O(\log n))$. The class does not change if we allow $L_{EC}(C)$ to be in AC^0 or $ATIME(O(\log n))$. Here, (uniform) AC^0 can be defined as languages computable by an alternating Turing machine in time $O(\log n)$ with $O(1)$ alternations.

Buss [3] has shown that one can evaluate in uniform NC^1 Boolean formulas represented as strings in the usual infix notation. We can define the extended connection language for a single circuit (rather than sequence) in a natural way, and represent it as a polynomial-size string. Log-depth circuits in this representation can be also evaluated in uniform NC^1 (this is implicit in Ruzzo [13]). On the other hand, evaluation of log-depth circuits represented by the direct connection language (or equivalent form) is not known to be possible even in nonuniform NC^1, but it can be done in logarithmic space (L). Regarding the former, we observe the following reduction of a combinatorial problem which is apparently not in nonuniform NC^1:

Proposition 2.1 *The following problem is AC^0-reducible to evaluation of bounded fan-in log-depth circuits (described by L_{DC}). Given a directed graph G on n vertices with bounded out-degree, vertices $x, y \in G$, and a number $d \leq \log n$, determine whether y is reachable from x in at most d steps.*

Proof: Without loss of generality assume that G contains all self-loops. We construct a circuit with $d+1$ layers, where each layer is labeled by nodes of G. Every node u on layer $l+1$ is a disjunction gate, and its inputs are nodes v on layer l such that $u \to v$ is an edge of G. We initialize the bottom layer by assigning 1 to node y, and 0 to all other nodes, and we evaluate the circuit. Then the value of node x on the top layer is 1 iff y is reachable from x in d steps. $\square$

A kind of converse also holds: it can be shown that an algorithm for the problem described in Proposition 2.1 can be used to transform a direct connection language of a log-depth circuit to its extended connection language, which can be evaluated in uniform NC^1.

3 Theories

We will work with second-order (i.e., two-sorted) arithmetical theories as in [14, 9], but for convenience we include the function $|x| = \lceil \log_2(x+1) \rceil$ among the basic symbols. Our theories thus have two sorts of variables: numbers, denoted by lowercase letters, and finite sets or strings, denoted by uppercase letters. The basic language is $L_0 = \langle 0, \mathrm{s}, +, \cdot, |x|, \leq, \in, |X|\rangle$.

The theory *BASIC* consists of the axioms

$$
\begin{array}{ll}
x+0=x & x+\mathrm{s}\,y=\mathrm{s}(x+y) \\
x\cdot 0=0 & x\cdot \mathrm{s}\,y=x\cdot y+x \\
\mathrm{s}\,y\le x\to y<x & x\ne 0\to \exists y\, x=\mathrm{s}\,y \\
x\in X\to x<|X| & \mathrm{s}\,x=|X|\to x\in X \\
|0|=0 & x\ne 0\to |x+x|=\mathrm{s}|x| \\
\forall x\,(x\in X\leftrightarrow x\in Y)\to X=Y & |\mathrm{s}(x+x)|=\mathrm{s}|x|
\end{array}
$$

where $x<y$ is an abbreviation for $x\le y\wedge x\ne y$. We also write $X(x)$ for $x\in X$. We define the constants $1=\mathrm{s}\,0$, $2=\mathrm{s}\,\mathrm{s}\,0$, $3=\mathrm{s}\,\mathrm{s}\,\mathrm{s}\,0$, $\dots$, and we will often write $x+1$ for $\mathrm{s}\,x$ (the two expressions being equal by the *BASIC* axioms). We introduce the bounded quantifiers

$$
\begin{aligned}
\exists x\le t\,\varphi &\Leftrightarrow \exists x\,(x\le t\wedge\varphi),\\
\forall x\le t\,\varphi &\Leftrightarrow \forall x\,(x\le t\to\varphi),\\
\exists X\le t\,\varphi &\Leftrightarrow \exists X\,(|X|\le t\wedge\varphi),\\
\forall X\le t\,\varphi &\Leftrightarrow \forall X\,(|X|\le t\to\varphi),
\end{aligned}
$$

where t is a term not involving x or X (respectively), and similarly for strict inequalities. A formula is bounded if it uses only bounded quantifiers. A bounded L_0-formula without set quantifiers is called Σ^B_0 or Π^B_0. Inductively, Σ^B_{i+1} consists of formulas of the form

$$\exists X_1\le t_1\dots\exists X_n\le t_n\,\varphi$$

for $\varphi\in\Pi^B_i$, and Π^B_{i+1} consists of formulas of the form

$$\forall X_1\le t_1\dots\forall X_n\le t_n\,\varphi$$

for $\varphi\in\Sigma^B_i$. A formula is Σ^1_1 if it consists of a block of second-order existential quantifiers followed by a Σ^B_0-formula. A predicate is Σ^B_0-definable in the standard model iff it is computable in AC^0, and for $i>0$, the Σ^B_i-definable (Π^B_i-definable) predicates coincide with the levels Σ^P_i (Π^P_i) of the polynomial hierarchy. Note that we use Σ^B_i and Π^B_i to denote formulas of the basic language L_0 only. If we expand the definition to allow atomic formulas in a richer language L, we will call the corresponding classes $\Sigma^B_i(L)$ and $\Pi^B_i(L)$, respectively.

If Γ is a set of formulas, the Γ-comprehension axiom is the schema

$$(\Gamma\text{-}COMP)\qquad \exists X\le x\,\forall u<x\,(u\in X\leftrightarrow\varphi),$$

where $\varphi\in\Gamma$ has no free occurrence of X. We define the theory V^0 as $BASIC+\Sigma^B_0\text{-}COMP$.

The theory VNC^1 is axiomatized over V^0 by

$$
\begin{aligned}
&\exists Y\le 2a\,\forall x<a\,[(Y(x+a)\leftrightarrow I(x))\\
&\qquad\wedge(Y(x)\leftrightarrow((G(x)\wedge(Y(2x)\vee Y(2x+1)))\vee(\neg G(x)\wedge Y(2x)\wedge Y(2x+1))))].
\end{aligned}
$$

The meaning is that we can evaluate a monotone formula laid out in a balanced binary tree, represented by nonzero numbers below $2a$ so that nodes $x < a$ are conjunction or disjunctions (according to $G(x)$) of nodes $2x$ and $2x+1$, and nodes $x \geq a$ are truth constants given by I.

The theory VL is axiomatized over V^0 by the axiom

$$\forall x < a\, \exists! y < a\, F(x,y) \to \exists P\, ((P)_0 = 0 \wedge \forall v < a\, F((P)_v, (P)_{v+1})),$$

where P encodes a sequence of numbers, and $(P)_v$ is the vth member of the sequence (see [9] for details of the sequence coding). The meaning is that we can iterate a number function, or equivalently, that we can trace a path in a directed graph where each node has out-degree 1.

Let $\varphi(d,x,y)$ be a formula, possibly with other free variables. We put

$$\begin{aligned}
\varphi^*(d,x,y) &\Leftrightarrow \varphi(d,x,y) \wedge (\forall z < y\, \neg\varphi(d,x,z) \vee \forall z > y\, \neg\varphi(d,x,z)),\\
\mathrm{eval}(n,m,\varphi,I,Y) &\Leftrightarrow \forall x < n\, [(Y(0,x) \leftrightarrow I(x))\\
&\qquad \wedge \forall d < m\, (Y(d+1,x) \leftrightarrow ((2 \mid d \wedge \exists y < n\, (\varphi^*(d,x,y) \wedge Y(d,y)))\\
&\qquad\qquad \vee (2 \nmid d \wedge \forall y < n\, (\varphi^*(d,x,y) \to Y(d,y)))))],
\end{aligned}$$

where $Y(d,x)$ stands for $dn+x \in Y$. (By abuse of notation, we include φ among the arguments of eval to indicate the dependence of eval on φ, even though φ is a formula, not a variable. Note that free variables of eval include parameters of φ, i.e., its free variables other than d, x, y.) The meaning of eval is that Y is the evaluation of a bounded fan-in monotone circuit described by φ on input I. The circuit consists of $m+1$ layers, each with n nodes. Nodes on layer 0 are truth constants given by I. Layers $d > 0$ consist of alternating disjunction (odd d) and conjunction (even d) gates. Gates on level d can only use nodes on level $d-1$ as inputs. The formula $\varphi(d,x,y)$ means that node x on level $d+1$ uses node y on level d as input. The formula φ^* is actually employed instead of φ to force each gate to have at most two inputs.

We define VNC^1_* to be the closure of V^0 under the derivation rule

$$(\Delta^B_1\text{-}SCV) \qquad \frac{\varphi \leftrightarrow \neg\varphi'}{\exists Y \leq (|m|+1)n\ \mathrm{eval}(n,|m|,\varphi,I,Y)},$$

where φ and φ' are Σ^B_1-formulas with no free set variables. (A Σ^B_1-formula provably equivalent to a Π^B_1-formula in a theory T will be called a $\Delta^B_1(T)$-formula.)

The language $L_{\overline{VNC}^1_*}$ contains L_0, and a function symbol $C_\varphi(n,\vec{x},\vec{X})$ for each Σ^B_0-formula $\varphi(u,\vec{x},\vec{X})$ (with all free variables indicated). Moreover, it is closed under the following rule: for each open $L_{\overline{VNC}^1_*}$-formula $\varphi(\vec{p},d,x,y)$ without free set variables, we include a function symbol $Y_\varphi(\vec{p},n,m,I)$. We will usually denote $C_\varphi(n,\vec{x},\vec{X})$ by $\{u < n \mid \varphi(u,\vec{x},\vec{X})\}$.

$\overline{VNC}^1_*$ is a theory in $L_{\overline{VNC}^1_*}$ consisting of the axioms of $BASIC$, the axiom

$$(\Sigma^B_0\text{-}\overline{COMP}) \qquad u \in C_\varphi(n,\vec{x},\vec{X}) \leftrightarrow u < n \wedge \varphi(u,\vec{x},\vec{X})$$

for each Σ^B_0-formula $\varphi(u,\vec{x},\vec{X})$, and the axiom

$$(Open\text{-}\overline{SCV}) \qquad |Y_\varphi(\vec{p},n,m,I)| \leq (|m|+1)n \wedge \mathrm{eval}(n,|m|,\varphi,I,Y_\varphi(\vec{p},n,m,I))$$

for each open $L_{\overline{VNC}^1_*}$-formula $\varphi(\vec{p},d,x,y)$.

Notice that $\overline{VNC}^1_*$ contains V^0.

4 Properties of VNC^1_* and $\overline{VNC}^1_*$

The Δ^B_1-SCV and *Open*-$\overline{SCV}$ axioms provide evaluation of a certain type of circuits, but they were designed to be formally simple rather than feature-rich. We will introduce a more elaborate setting for convenient evaluation of log-depth circuits.

We will describe circuits using the following data:

- Numbers k, m, and s, where k is the number of inputs, m is the number of layers, and s is the size of each layer.
- A function $T\colon m \times s \to \{\ulcorner\vee\urcorner, \ulcorner\wedge\urcorner, \ulcorner\neg\urcorner\} \cup \{\ulcorner x_i \urcorner \mid i < k\}$ indicating the type of each node, where we put e.g. $\ulcorner\vee\urcorner = 0$, $\ulcorner\wedge\urcorner = 1$, $\ulcorner\neg\urcorner = 2$, and $\ulcorner x_i\urcorner = i+3$, and we represent T by its graph, i.e., as a set $T \leq ms(k+3)$.
- A formula $\varphi(d, x, d', x')$ (possibly with other parameters) which states that node x' on layer d' is an input of gate x on layer d.

In order for a circuit to be well-formed, we demand that any gate uses only nodes on lower layers as inputs (but not necessarily from the adjacent layer), and all nodes have the correct number of inputs: 1 for negation nodes, 0 for input nodes, and at most 2 for conjunction and disjunction gates. Notice that we allow $\wedge$ and $\vee$ gates with no inputs, which compute the truth constants $\perp$ and $\top$, or with one input, which act as the identity function. The formula

$$
\begin{aligned}
\mathrm{Circ}(k, m, s, T, \varphi) \Leftrightarrow{} & \forall d < m \, \forall x < s \, \exists! p < k + 3 \, T(d, x, p) \\
& \wedge \forall d, d' < m \, \forall x, x' < s \, (\varphi(d, x, d', x') \to d' < d) \\
& \wedge \forall d, d_0, d_1, d_2 < m \, \forall x, x_0, x_1, x_2 < s \\
& \qquad \Big(\bigwedge_{i<3} \varphi(d, x, d_i, x_i) \to \bigvee_{i<j} (d_i = d_j \wedge x_i = x_j)\Big) \\
& \wedge \forall d, d_0, d_1 < m \, \forall x, x_0, x_1 < s \\
& \qquad \Big(T(d, x, \ulcorner\neg\urcorner) \wedge \bigwedge_{i<2} \varphi(d, x, d_i, x_i) \to d_0 = d_1 \wedge x_0 = x_1\Big) \\
& \wedge \forall d < m \, \forall x < s \, (T(d, x, \ulcorner\neg\urcorner) \to \exists d' < m \, \exists x' < s \, \varphi(d, x, d', x')) \\
& \wedge \forall d, d' < m \, \forall x, x' < s \, \forall i < k \, (T(d, x, \ulcorner x_i\urcorner) \to \neg\varphi(d, x, d', x')).
\end{aligned}
$$

formalizes these requirements. The formula

$$
\begin{aligned}
\mathrm{Eval}(k, m, s, T, \varphi, I, Y) \Leftrightarrow{} & \forall d < m \, \forall x < s \, \Big(Y(d, x) \leftrightarrow \\
& (T(d, x, \ulcorner\vee\urcorner) \wedge \exists d' < m \, \exists x' < s \, (\varphi(d, x, d', x') \wedge Y(d', x'))) \\
& \vee (T(d, x, \ulcorner\wedge\urcorner) \wedge \forall d' < m \, \forall x' < s \, (\varphi(d, x, d', x') \to Y(d', x'))) \\
& \vee (T(d, x, \ulcorner\neg\urcorner) \wedge \exists d' < m \, \exists x' < s \, (\varphi(d, x, d', x') \wedge \neg Y(d', x'))) \\
& \vee \exists i < k \, (T(d, x, \ulcorner x_i\urcorner) \wedge I(i))\Big)
\end{aligned}
$$

states that Y is an evaluation of the circuit described by k, m, s, T, φ on input $I \leq k$.

Remark 4.1 Note that any Σ_0^B-formula φ is equivalent in $\overline{VNC}^1_*$ to an open formula, e.g., $0 \in \{u < 1 \mid \varphi\}$ (where u is not free in φ). We will prove later (Corollary 4.7) that the same also holds for $\Sigma_0^B(L_{\overline{VNC}^1_*})$-formulas.

Theorem 4.2

(*i*) *If φ is a $\Delta_1^B(VNC^1_*)$-formula without free set variables, then VNC^1_* proves*

$$\mathrm{Circ}(k, |m|, s, T, \varphi) \to \exists! Y \leq |m|s\ \mathrm{Eval}(k, |m|, s, T, \varphi, I, Y).$$

(*ii*) *If φ is an open $L_{\overline{VNC}^1_*}$-formula without free set variables, then there exists an $L_{\overline{VNC}^1_*}$-term Y such that $\overline{VNC}^1_*$ proves*

$$\mathrm{Circ}(k, |m|, s, T, \varphi) \to \mathrm{Eval}(k, |m|, s, T, \varphi, I, Y(\vec{p}, k, m, s, T, I)),$$

where $\vec{p}$ are the parameters of φ.

Proof: Uniqueness of Y can be proved by straightforward Σ_0^B-induction, the problem is to show its existence. We will reduce evaluation of the circuit to another circuit in the simplified framework of eval, which can be evaluated using the axioms Δ_1^B-*SCV* or *Open*-$\overline{SCV}$. We subject the circuit to the following transformations:

- The input layer of the new circuit will consist of bits $I(j)$ of the original input string I, their negations $\neg I(j)$, and bits $T(d, x, p)$ of T.
- We introduce a dual node $x^\neg$ to each node x in the circuit, in order to allow making the new circuit monotone.
- We replicate each node on all layers to overcome the restriction that each gate may only use nodes of its immediately preceding layer as inputs in the new circuit.
- If x is a node with possible inputs y_0, y_1, we include in the new circuit the following gadgets (omitting the mention of layers for simplicity):

$$\begin{aligned} x = {} & \bigvee_{j<k} (T(x, \ulcorner x_j \urcorner) \wedge I(j)) \vee (T(x, \ulcorner \neg \urcorner) \wedge y_0^\neg) \\ & \vee (T(x, \ulcorner \wedge \urcorner) \wedge y_0 \wedge y_1) \vee (T(x, \ulcorner \vee \urcorner) \wedge (y_0 \vee y_1)), \\ x^\neg = {} & \bigvee_{j<k} (T(x, \ulcorner x_j \urcorner) \wedge \neg I(j)) \vee (T(x, \ulcorner \neg \urcorner) \wedge y_0) \\ & \vee (T(x, \ulcorner \vee \urcorner) \wedge y_0^\neg \wedge y_1^\neg) \vee (T(x, \ulcorner \wedge \urcorner) \wedge (y_0^\neg \vee y_1^\neg)). \end{aligned}$$

  More precisely, we put $O(|k|)$ layers to the bottom of the circuit which compute the disjunctions $\bigvee_{j<k}(T(x, \ulcorner x_j \urcorner) \wedge (\neg) I(j))$ arranged in a balanced binary tree, and we replace each node in the original circuit with the constant-size remaining part of its gadget.

- We introduce padding to shift the nodes so that odd layers consist of disjunctions, and even layers of conjunctions.

We proceed with the formal details to verify that we can arrange the result in such a way that the wires of the new circuit are described by a Δ^B_1-formula or an open $L_{\overline{VNC}^1_*}$-formula without set parameters, as required by the axioms.

Our new circuit will have $m'+1 := 2+2|k|+6|m|$ layers, each containing $n' := 2k+(5k+7)|m|s$ nodes.

Nodes $i(x) := x < k$ on each layer represent the input bits $I(x)$, nodes $i^\neg(x) := k+x$ give $\neg I(x)$, and nodes $t(d,x,p) := 2k+(ds+x)(k+3)+p$ give $T(d,x,p)$ for $d < |m|$, $x < s$, $p < k+3$. Nodes

$$o(\varepsilon,d,x,u) := 2k+(k+3)|m|s+((\varepsilon|m|+d)s+x)(2k-1)+u$$

for $\varepsilon < 2$, $d < |m|$, $x < s$, and $u < 2k-1$ are used to compute $\bigvee_{j<k}(T(d,x,\ulcorner x_j\urcorner)\wedge I^\varepsilon(j))$, where $I^0 = I$, $I^1 = \neg I$. Finally, nodes

$$n(\varepsilon,d,x,u) := 2k+(5k+1)|m|s+((\varepsilon|m|+d)s+x)3+u$$

for $\varepsilon < 2$, $d < |m|$, $x < s$, $u < 3$ represent node x (if $\varepsilon = 0$) or $x^\neg$ (if $\varepsilon = 1$) on layer d in the original circuit, as well as its associated gadget.

The layers are laid out as follows. Layer 0 is the input layer, initialized to

$$I' = \{i(x) \mid I(x)\} \cup \{i^\neg(x) \mid \neg I(x)\} \cup \{t(d,x,p) \mid T(d,x,p)\}.$$

Layer 1 is a copy of layer 0 (as we need conjunctions at the bottom of our new circuit, and odd layers are disjunctions). Layers 2 to $2|k|+1$ are used to compute $\bigvee_{j<k}(T(d,x,\ulcorner x_j\urcorner)\wedge I^\varepsilon(j))$ in node $o(\varepsilon,d,x,0)$. On layer 2, we put $T(d,x,\ulcorner x_j\urcorner)\wedge I^\varepsilon(j)$ to node $o(\varepsilon,d,x,k-1+j)$. Odd layers 3 to $2|k|+1$ then consist of disjunctions arranged in a balanced binary tree, where the children of node $o(\varepsilon,d,x,u)$, $u < k-1$, are $o(\varepsilon,d,x,2u+1)$ and $o(\varepsilon,d,x,2u+2)$. Even layers 4 to $2|k|$ copy the previous layer. The remaining layers $2|k|+2$ to $2|k|+1+6|m|$ do the main simulation of the original circuit. Let $l(d,u) = 2|k|+2+6d+u$ for $d < |m|$, $u \le 5$. Node x on layer d of the original circuit is simulated by node $n(0,d,x,0)$ on layers $l(d',5)$ for all $d' \ge d$, and its negation $x^\neg$ is in node $n(1,d,x,0)$. They are also replicated on the next layer $l(d'+1,0)$ as $n(\varepsilon,d,x,2)$. Other nodes $n(\varepsilon,d,x,u)$, $u \le 2$, on layers $l(d',v)$, $v \le 4$, are parts of the gadget need to compute x or $x^\neg$.

The wires of the new circuit are thus described by the following formula, where we abbreviate the modulo operation by %:

$$\begin{aligned}
&\varphi'(d,x,y) \Leftrightarrow \\
&\quad (x < o(0,0,0,0) \wedge y = x) \\
&\quad \vee (o(0,0,0,0) \le x < n(0,0,0,0) \wedge (d = 0 \vee 2 \nmid d \ne 1 \vee d \ge 2|k|+1) \wedge y = x) \\
&\quad \vee (o(0,0,0,0) \le x < n(0,0,0,0) \wedge d = 1 \wedge (x - o(0,0,0,0)) \,\%\, (2k-1) \ge k-1 \\
&\qquad \wedge (y = (x - o(0,0,0,0)) \,\%\, (2k-1) - (k-1) + \lfloor x/((2k-1)|m|s)\rfloor k \\
&\qquad\quad \vee y = 2k + (\lfloor x/(2k-1)\rfloor \,\%\, |m|s)(k+3) \\
&\qquad\qquad\qquad + (x - o(0,0,0,0)) \,\%\, (2k-1) - (k-1) + 3)) \\
&\quad \vee (o(0,0,0,0) \le x < n(0,0,0,0) \wedge 0 < d \le 2|k| \wedge 2 \mid d \wedge y = x
\end{aligned}$$

$$
\begin{aligned}
&\qquad \wedge (x - o(0,0,0,0)) \,\%\, (2k-1) \geq k-1 \\
&\vee (o(0,0,0,0) \leq x < n(0,0,0,0) \wedge 0 < d \leq 2|k| \wedge 2 \mid d \\
&\qquad \wedge (x - o(0,0,0,0)) \,\%\, (2k-1) < k-1 \\
&\qquad \wedge 1 \leq y - x - ((x - o(0,0,0,0)) \,\%\, (2k-1) - (k-1)) \leq 2) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = x \\
&\qquad \wedge \langle (x - n(0,0,0,0)) \,\%\, 3, (d - 2|k|) \,\%\, 6 \rangle \\
&\qquad\qquad \in \{\langle 0,1\rangle, \langle 1,1\rangle, \langle 0,2\rangle, \langle 1,2\rangle, \langle 2,2\rangle, \langle 0,3\rangle, \langle 1,3\rangle, \langle 0,4\rangle, \langle 1,4\rangle, \langle 0,5\rangle\}) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = x+1 \\
&\qquad \wedge \langle (x - n(0,0,0,0)) \,\%\, 3, (d - 2|k|) \,\%\, 6 \rangle \in \{\langle 1,3\rangle, \langle 0,5\rangle\}) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = x-2 \\
&\qquad \wedge (x - n(0,0,0,0)) \,\%\, 3 = 2 \wedge 6 \mid (d - 2|k|)) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = 2k + (k+3)|m|s + \lfloor (x - n(0,0,0,0))/3 \rfloor (2k-1) \\
&\qquad \wedge 3 \mid (x - n(0,0,0,0)) \wedge (d - 2|k|) \,\%\, 6 = 3) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = 2k + (\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)(k+3) + \ulcorner\neg\urcorner \\
&\qquad \wedge 3 \mid (x - n(0,0,0,0)) \wedge (d - 2|k|) \,\%\, 6 = 2) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = 2k + (\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)(k+3) + \ulcorner\wedge\urcorner \\
&\qquad \wedge (x - n(0,0,0,0)) \,\%\, 3 = 1 + \lfloor (x - n(0,0,0,0))/3|m|s \rfloor \wedge (d - 2|k|) \,\%\, 6 = 2) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y = 2k + (\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)(k+3) + \ulcorner\vee\urcorner \\
&\qquad \wedge (x - n(0,0,0,0)) \,\%\, 3 = 2 - \lfloor (x - n(0,0,0,0))/3|m|s \rfloor \wedge (d - 2|k|) \,\%\, 6 = 2) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y \geq n(0,0,0,0) \wedge 6 \mid (d - 2|k|) \\
&\qquad \wedge 3 \mid (x - n(0,0,0,0)) \wedge 3 \mid (y - n(0,0,0,0)) \\
&\qquad \wedge (x \geq n(1,0,0,0) \nleftrightarrow y \geq n(1,0,0,0)) \\
&\qquad \wedge \varphi^+(\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s, \lfloor (y - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y \geq n(0,0,0,0) \wedge 6 \mid (d - 2|k|) \\
&\qquad \wedge (x - n(0,0,0,0)) \,\%\, 3 = 1 \wedge 3 \mid (y - n(0,0,0,0)) \\
&\qquad \wedge (x \geq n(1,0,0,0) \leftrightarrow y \geq n(1,0,0,0)) \\
&\qquad \wedge \varphi^+(\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s, \lfloor (y - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)) \\
&\vee (x \geq n(0,0,0,0) \wedge d \geq 2|k|+1 \wedge y \geq n(0,0,0,0) \wedge (d - 2|k|) \,\%\, 6 = 1 \\
&\qquad \wedge (x - n(0,0,0,0)) \,\%\, 3 = 2 \wedge (y - n(0,0,0,0)) \,\%\, 3 = 2 \\
&\qquad \wedge (x \geq n(1,0,0,0) \leftrightarrow y \geq n(1,0,0,0)) \\
&\qquad \wedge \varphi^+(\lfloor (x - n(0,0,0,0))/3 \rfloor \,\%\, |m|s, \lfloor (y - n(0,0,0,0))/3 \rfloor \,\%\, |m|s)),
\end{aligned}
$$

where

$$\varphi^+(u, v) \Leftrightarrow \varphi(\lfloor u/s \rfloor, u \,\%\, s, \lfloor v/s \rfloor, v \,\%\, s).$$

Note that integer division and % are Σ_0^B-definable. It is thus easy to see that $\varphi' \in \Delta_1^B(VNC_*^1)$ if $\varphi \in \Delta_1^B(VNC_*^1)$, and, using Remark 4.1, that φ' is equivalent to an open $L_{\overline{VNC_*^1}}$-formula if φ is. By Δ_1^B-*SCV* or *Open*-$\overline{SCV}$, there exists Y' such that $\mathrm{eval}(n', m', \varphi', I', Y')$. It is tedious, but completely straightforward, to verify that parts of Y' correspond to an evaluation of the original circuit as described above, hence $\mathrm{Eval}(k, |m|, s, T, \varphi, I, Y)$, where

$$Y = \{\langle d, x\rangle \mid Y'(m', n(0, d, x, 5))\}.$$

In the case of $\overline{VNC}^1_*$, we can compute I' from I and T by a comprehension function symbol, compute Y' using the $Y_{\varphi'}$ function, and compute Y from Y' by another comprehension function, hence Y is given by a term in the original data. □

Corollary 4.3 *VNC^1_* and $\overline{VNC}^1_*$ contain VNC^1.* □

Definition 4.4 Let Γ be a set of formulas. A set function $F(X_0, \dots, X_{c-1})$ is *computable by a family of Γ-definable shallow circuits* (*computable by Γ-circuits* for short) if there are L_0-terms $s(n)$, $m(n)$, and $o(n)$, a Σ^B_0-formula $\tau(n, d, x, p)$, and a Γ-formula $\varphi(n, d, x, d', x')$, such that

- $s(n) \geq cn$, $s(n) \geq o(n)$, $m(n) > 0$
- $\mathrm{Circ}(cn, |m(n)|, s(n), T(n), \varphi)$, where $T(n) = \{(s(n)d + x)(cn + 3) + p \mid \tau(n, d, x, p)\}$,
- if $\vec{X}$ are sets such that $|X_i| \leq n$, $I = \{in + u \mid i < c, u \in X_i\}$, and

  $$\mathrm{Eval}(cn, |m(n)|, s(n), T(n), \varphi, I, Y),$$

  then

  $$(*) \qquad F(\vec{X}) = \{u < o(n) \mid Y(|m(n)| - 1, u)\}.$$

A function $F(\vec{u}, \vec{X})$ or $f(\vec{u}, \vec{X})$ with number inputs and/or output is computable by Γ-circuits, if the same holds for the set function $F'(\vec{U}, \vec{X})$ which we obtain by representing every number x by the set $\{u \mid u < x\}$. A predicate $\psi(\vec{u}, \vec{X})$ is computable by Γ-circuits if its characteristic function

$$\chi_\psi(\vec{u}, \vec{X}) = \begin{cases} \{0\} & \text{if } \psi(\vec{u}, \vec{X}), \\ \varnothing & \text{if } \neg\psi(\vec{u}, \vec{X}) \end{cases}$$

is. In other words, if we can fix $o(n) = 1$ in the above definition, and replace $(*)$ with

$$\psi(\vec{u}, \vec{X}) \leftrightarrow Y(|m(n)| - 1, 0).$$

The next lemma is a key technical result needed to show various properties of VNC^1_* and $\overline{VNC}^1_*$, e.g., that $\overline{VNC}^1_*$ is a conservative extension of VNC^1_*.

Lemma 4.5 *Let $\alpha(\vec{X}, \vec{x})$ be a $L_{\overline{VNC}^1_*}$-term, or a $\Sigma^B_0(L_{\overline{VNC}^1_*})$-formula. Then α is provably in $\overline{VNC}^1_*$ computable by $Open(L_{\overline{VNC}^1_*})$-circuits, and provably in VNC^1_* computable by $\Delta^B_1(VNC^1_*)$-circuits in such a way that VNC^1_* proves the defining axiom of α.*

Moreover, the graph of α or of its characteristic function is definable in VNC^1_* by a Σ^B_1-formula $\exists Y \leq t\, \vartheta(\vec{X}, \vec{x}, \varepsilon, Y)$ with $\vartheta \in \Sigma^B_0$, and provably in VNC^1_*, we can compute some Y satisfying the formula from $\vec{x}, \vec{X}$ by $\Delta^B_1(VNC^1_*)$-circuits.

Proof: We proceed by induction on the complexity of α (defined in such a way that the complexity of C_φ and Y_φ is larger than that of φ, and in the case of Y_φ, also φ^*).We will show two cases, and leave the rest to the reader.

Let α be the formula $\exists x_c \le t(\vec{X},\vec{x})\,\beta(\vec{X},\vec{x},x_c)$, and fix a polynomial $p(n)$ such that $t(\vec{X},\vec{x}) < p(n)$ whenever $|\vec{X}|, \vec{x} \le n$. By the induction hypothesis, we can compute the formula $\alpha' = x_c \le t(\vec{X},\vec{x}) \wedge \beta(\vec{X},\vec{x},x_c)$ by $Open(\overline{VNC}^1_*)$-circuits or $\Delta^B_1(VNC^1_*)$-circuits described by s', m', τ', and φ'. We construct circuits for α by taking $p(n)$ copies of the circuit for α', fixing the value of x_c to the representation of i in the ith copy, and computing the disjunction of the outputs (arranged in a binary tree, as in the proof of Theorem 4.2). To be exact, we put $s(n) = p(n)s'(n)$ (assuming $s'(n) \ge 2$), $m(n) = 4m'(n)p(n)$ (so that $|m'(n)| \ge |m(n)| + |p(n)| + 1$),

$$
\begin{aligned}
\tau(n,d,x,p) \Leftrightarrow{}& (d \ge |m'| \wedge p = \ulcorner\vee\urcorner)\\
&\vee (d < |m'| \wedge \tau'(n,d,x \,\%\, s',p) \wedge \neg\exists j < n\, p = \ulcorner x_{cn+j}\urcorner)\\
&\vee (d < |m'| \wedge \exists j < n\,(\tau'(n,d,x \,\%\, s',\ulcorner x_{cn+j}\urcorner)\\
&\qquad\qquad \wedge ((j < \lfloor x/s'\rfloor \wedge p = \ulcorner\wedge\urcorner) \vee (j \ge \lfloor x/s'\rfloor \wedge p = \ulcorner\vee\urcorner)))),\\
\varphi(n,d,x,d',x') \Leftrightarrow{}& (d < |m'| \wedge \varphi'(n,d,x \,\%\, s',d',x' \,\%\, s') \wedge \lfloor x/s'\rfloor = \lfloor x'/s'\rfloor)\\
&\vee (d = |m'| \wedge d' = d-1 \wedge p-1 \le x < 2p-1 \wedge x' = (x-(p-1))s')\\
&\vee (d > |m'| \wedge d' = d-1 \wedge x \ge p-1 \wedge x' = x)\\
&\vee (d > |m'| \wedge d' = d-1 \wedge x < p-1 \wedge 1 \le x' - 2x \le 2),
\end{aligned}
$$

where we write m', s', p for $m'(n)$, $s'(n)$, $p(n)$. Clearly, τ is Σ^B_0, φ is $Open(L_{\overline{VNC}^1_*})$ or $\Delta^B_1(VNC^1_*)$ as appropriate, and it is easy to see that the circuit defined by s, m, τ, φ computes α.

Let $\exists Y \le u\,\vartheta(\vec{X},\vec{x},x_c,\varepsilon,Y)$ be a Σ^B_1-definition of the graph $\chi_{\alpha'}(\vec{X},\vec{x},x_c) = \varepsilon$ of the characteristic function of α', such that Y is computable from $\vec{X},\vec{x},x_c$ by $\Delta^B_1(VNC^1_*)$-circuits. Consider the Σ^B_1-formula

$$
\begin{aligned}
(*)\quad |\varepsilon| \le 1 \wedge \exists Z \le up(n)\,(\forall x_c < p(n)\,(\vartheta(\vec{X},\vec{x},x_c,\varnothing,Z^{[x_c]}) \vee \vartheta(\vec{X},\vec{x},x_c,\{0\},Z^{[x_c]}))\\
\wedge (0 \in \varepsilon \leftrightarrow \exists x_c < p(n)\,\vartheta(\vec{X},\vec{x},x_c,\{0\},Z^{[x_c]}))),
\end{aligned}
$$

where $n = \sum_i |X_i| + \sum_i x_i$, and $Z^{[x]}$ denotes $\{y < u \mid xu+y \in Z\}$. We take $p(n)$ parallel copies of the circuit computing Y, and wire the x_c inputs in the ith copy to the representation of i, as above in the construction of the circuit for α. The resulting circuit computes Z satisfying

$$\forall x_c < p(n)\,(\vartheta(\vec{X},\vec{x},x_c,\varnothing,Z^{[x_c]}) \vee \vartheta(\vec{X},\vec{x},x_c,\{0\},Z^{[x_c]}))$$

from $\vec{X},\vec{x}$. Given Z, it is easy to see that $(*)$ is equivalent to $\chi_\alpha(\vec{X},\vec{x}) = \varepsilon$.

Let us turn to the case $\alpha = Y_\psi(\vec{p}(\vec{X},\vec{x}), s(\vec{X},\vec{x}), m(\vec{X},\vec{x}), I(\vec{X},\vec{x}))$, where $\psi(\vec{p},d,x,y)$ is an open $\overline{VNC}^1_*$-formula. By the induction hypothesis, we can compute the terms $\vec{p}$, s, m, and I by suitable circuits. Let $q(n)$ be a polynomial such that $\vec{p}(\vec{X},\vec{x}), s(\vec{X},\vec{x}), m(\vec{X},\vec{x}), |I(\vec{X},\vec{x})| < q(n)$ whenever $|\vec{X}|, \vec{x} \le n$. We construct circuits computing α as follows:

- We compute $s(\vec{X},\vec{x}), m(\vec{X},\vec{x}), I(\vec{X},\vec{x}), \vec{p}(\vec{X},\vec{x})$ using their respective circuits. We denote the jth bit of the result by s_j, m_j, i_j, p^r_j.

- For every $\vec{p}, s, m < q(n)$, we evaluate in parallel the eval-style circuit defined by s, $|m|$, and $\psi^*(\vec{p}, \cdot, \cdot, \cdot)$ on input I. That is, we take the circuit with $|m| + 1$ layers, each of size s. The bottom layer is initialized to the first s bits i_j, and the other layers are alternating disjunctions and conjunctions, where yth node on dth layer is an input to xth node on $(d+1)$st layer iff $\psi^*(\vec{p}, d, x, y)$. We denote the value of the xth node on dth layer by $v_{\vec{p},s,m,d,x}$.

- For each $\vec{p}, s, m < q(n)$, we compute in parallel the "selector" $h_{\vec{p},s,m}$ which states that $\bigwedge_r (p^r(\vec{X}, \vec{x}) = p^r) \wedge s(\vec{X}, \vec{x}) = s \wedge m(\vec{X}, \vec{x}) = m$. This can be done as
$$h_{\vec{p},s,m} = \bigwedge_r (p^r_{p^r-1} \wedge \neg p^r_{p^r}) \wedge s_{s-1} \wedge \neg s_s \wedge m_{m-1} \wedge \neg m_m,$$
where we omit the conjuncts with index -1 (i.e., treat them as $\top$).

- We compute in parallel the output bits
$$o_{d,x} = \bigvee_{\vec{p},s,m<q(n)} (h_{\vec{p},s,m} \wedge v_{\vec{p},s,m,d,x}).$$

We spare the reader the formal definitions of the τ and φ formulas describing the circuit, and leave it to their imagination to verify that τ is Σ^B_0, and φ is a Boolean combination of Σ^B_0 formulas and ψ^* (substituted with Σ^B_0-definable functions like division with remainder). By the induction hypothesis, ψ^* is equivalent to a $\Delta^B_1(\mathit{VNC}^1_*)$- and $\mathit{Open}(L_{\overline{\mathit{VNC}^1_*}})$-formula, therefore so is φ. It is easy to see that the circuit indeed computes α.

Let ϑ be Σ^B_0-formula such that the graph $\chi_\psi(\vec{p}, s, m) = \varepsilon$ of the characteristic function of ψ is equivalent to $\exists W \le t\, \vartheta(\vec{p}, d, x, y, \varepsilon, W)$, and W is computable by $\Delta^B_1(\mathit{VNC}^1_*)$-circuits. Consider the formula

$$(**) \qquad \exists Z \le (q(n))^3 t\, \exists I, \vec{p}, s, m \le q(n) \Big(\mathrm{eval}(s, |m|, \xi, I, Y)$$
$$\wedge \bigwedge_r p^r(\vec{X}, \vec{x}) = p^r \wedge s(\vec{X}, \vec{x}) = s \wedge m(\vec{X}, \vec{x}) = m \wedge I(\vec{X}, \vec{x}) = I$$
$$\wedge \forall d < |m|\, \forall x, y < s\, (\vartheta(\vec{p}, d, x, y, \varnothing, Z^{[d,x,y]}) \vee \vartheta(\vec{p}, d, x, y, \{0\}, Z^{[d,x,y]}))\Big),$$

where
$$\xi(d, x, y) \Leftrightarrow \vartheta(\vec{p}, d, x, y, \{0\}, Z^{[d,x,y]}),$$
$n = \sum_i |X_i| + \sum_i x_i$, and $Z^{[d,x,y]}$ denotes $\{u < t \mid ((dq(n) + x)q(n) + y)t + u \in Z\}$. If we replace $p^r(\vec{X}, \vec{x})$, $s(\vec{X}, \vec{x})$, and $m(\vec{X}, \vec{x})$ with their Σ^B_1-definitions which exist by the induction hypothesis and prenex the second-order existential quantifiers, we obtain a Σ^B_1-formula, which we can further normalize to the form with only one second-order quantifier using a pairing function. Given $\vec{X}, \vec{x}$, we can compute a witness to this formula by $\Delta^B_1(\mathit{VNC}^1_*)$-circuits as follows. We compute (using the induction hypothesis) the values of $\vec{p}$, s, m, and I, and witnesses to the second-order quantifiers used in their graphs. Then we take the circuit computing W, and evaluate in parallel its q^3 copies for all fixed values $d, x, y < q(n)$ to obtain a Z such that
$$\forall d < |m|\, \forall x, y < s\, (\vartheta(\vec{p}, d, x, y, \varnothing, Z^{[d,x,y]}) \vee \vartheta(\vec{p}, d, x, y, \{0\}, Z^{[d,x,y]})).$$

Given such Z, we have $\xi(d,x,y) \leftrightarrow \psi(d,x,y)$, hence $\mathrm{eval}(s,|m|,\xi,I,Y)$ is valid for $Y = Y_\psi(\vec{p},s,m,I)$, and only for this Y. Thus, $(**)$ defines the graph of α, and witnesses for its second-order quantifiers can be computed by $\Delta^B_1(\mathit{VNC}^1_*)$-circuits. $\square$

Corollary 4.6 *$\overline{\mathit{VNC}}^1_*$ is contained in an extension of VNC^1_* by Σ^B_1-definitions. In particular, $\overline{\mathit{VNC}}^1_*$ is conservative over VNC^1_*.* $\square$

Corollary 4.7 *Every $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-formula is in $\overline{\mathit{VNC}}^1_*$ equivalent to an open formula.* $\square$

Corollary 4.8 *$\overline{\mathit{VNC}}^1_*$ proves $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-COMP, and $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-IND. Moreover, there are comprehension terms $F(a,\vec{x},\vec{X}) = \{u < a \mid \varphi(u,\vec{x},\vec{X})\}$ for $\Sigma^B_0(\overline{\mathit{VNC}}^1_*)$-formulas φ.*

Proof: Induction follows from comprehension. Let $\varphi(u,\vec{x},\vec{X})$ be a $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-formula, and take a sufficiently large n. By Lemma 4.5, φ is computable by an $\mathit{Open}(L_{\overline{\mathit{VNC}}^1_*})$-circuit. We take a parallel copies of the circuit as in the proof of Lemma 4.5, and wire the output of the ith circuit to the ith new output bit. We evaluate the circuit on the input which sets $\vec{x}$ and $\vec{X}$ in each copy to the value of the respective parameters, and sets u to the representation of i in the ith copy. Then the output of the new circuit is $\{u < a \mid \varphi\}$. The circuit is described by an open formula, hence its value is computable by an $L_{\overline{\mathit{VNC}}^1_*}$-term using Theorem 4.2. $\square$

Theorem 4.9 *$\overline{\mathit{VNC}}^1_*$ is an open theory.*

Proof: For any $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-formula φ, let $\overline{\varphi}$ be an open formula equivalent to φ in $\overline{\mathit{VNC}}^1_*$ by Corollary 4.7. We may assume that $\varphi = \overline{\varphi}$ if φ is already open. Let T be the set of formulas which contains

$$\overline{\varphi \vee \psi} \leftrightarrow \overline{\varphi} \vee \overline{\psi},$$

and similarly for other Boolean connectives, and the formulas

$$\overline{\varphi}(x) \wedge x \le t \to \overline{\exists x \le t\, \varphi(x)},$$
$$\overline{\exists x \le t\, \varphi(x)} \to \overline{\varphi}(|S|) \wedge |S| \le t,$$

where S is a term (with the same free variables as $\exists x \le t\,\varphi$) such that $\overline{\mathit{VNC}}^1_*$ proves $S = \{x < t \mid \varphi(x+1)\}$. Notice that $\{x < t \mid \varphi(x+1)\}$ is computable by an $L_{\overline{\mathit{VNC}}^1_*}$-term by Corollary 4.8.

Clearly, T is an open subtheory of $\overline{\mathit{VNC}}^1_*$, and every $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-formula is in T equivalent to an open formula. As $\overline{\mathit{VNC}}^1_*$ is $\Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$-axiomatized, it is equivalent to an open extension of T. $\square$

Theorem 4.10 *If $\overline{\mathit{VNC}}^1_*$ proves $\exists Y\, \varphi(\vec{x},\vec{X},Y)$, where φ is a Σ^1_1-formula, then there exists an $L_{\overline{\mathit{VNC}}^1_*}$-term F such that $\overline{\mathit{VNC}}^1_*$ proves $\varphi(\vec{x},\vec{X},F(\vec{x},\vec{X}))$.*

Proof: Write $\varphi = \exists \vec{Z}\, \vartheta(\vec{x},\vec{X},Y,\vec{Z})$ with $\vartheta \in \Sigma^B_0(L_{\overline{\mathit{VNC}}^1_*})$. By Corollary 4.7, ϑ is equivalent to an open formula. By Theorem 4.9 and Herbrand's theorem, there exist terms F_r, G^j_r such that $\overline{\mathit{VNC}}^1_*$ proves

$$\vartheta(\vec{x},\vec{X},F_0(\vec{x},\vec{X}),\vec{G}_0(\vec{x},\vec{X})) \vee \dots \vee \vartheta(\vec{x},\vec{X},F_c(\vec{x},\vec{X}),\vec{G}_c(\vec{x},\vec{X}))$$

for some c. Put

$$\alpha_r \Leftrightarrow \vartheta(\vec{x}, \vec{X}, F_r(\vec{x}, \vec{X}), \vec{G}_r(\vec{x}, \vec{X})) \wedge \bigwedge_{s<r} \neg\vartheta(\vec{x}, \vec{X}, F_s(\vec{x}, \vec{X}), \vec{G}_s(\vec{x}, \vec{X})),$$

and let p be a polynomial such that $|F_r|, |G_r^j| \le p(\vec{x}, |\vec{X}|)$. By Corollary 4.8, there are terms F and G^j such that $\overline{VNC^1_*}$ proves

$$F(\vec{x}, \vec{X}) = \Big\{u < p(\vec{x}, |\vec{X}|) \Bigm| \bigvee_r (\alpha_r \wedge u \in F_r(\vec{x}, \vec{X}))\Big\},$$
$$G^j(\vec{x}, \vec{X}) = \Big\{u < p(\vec{x}, |\vec{X}|) \Bigm| \bigvee_r (\alpha_r \wedge u \in G_r^j(\vec{x}, \vec{X}))\Big\},$$

Clearly, $\overline{VNC^1_*}$ proves

$$\bigvee_r \alpha_r,$$
$$\alpha_r \to F(\vec{x}, \vec{X}) = F_r(\vec{x}, \vec{X}),$$
$$\alpha_r \to G^j(\vec{x}, \vec{X}) = G_r^j(\vec{x}, \vec{X}),$$

hence also

$$\vartheta(\vec{x}, \vec{X}, F(\vec{x}, \vec{X}), \vec{G}(\vec{x}, \vec{X})),$$

which implies $\varphi(\vec{x}, \vec{X}, F(\vec{x}, \vec{X}))$. □

Corollary 4.11 *Every $\Delta^B_1(\overline{VNC^1_*})$-formula is in $\overline{VNC^1_*}$ equivalent to an open formula.*

Proof: Given $\varphi \in \Delta^B_1(\overline{VNC^1_*})$ (or even $\Delta^1_1(\overline{VNC^1_*})$), we apply Theorem 4.10 to the formula $\exists Y\,(0 \in Y \leftrightarrow \varphi)$. We obtain a term F such that the open formula $0 \in F(\vec{x}, \vec{X})$ is equivalent to φ. □

Corollary 4.12 *$\overline{VNC^1_*}$ contains VNC^1_*, thus VNC^1_* is the L_0-fragment of $\overline{VNC^1_*}$.*

Proof: By Corollary 4.11, $\overline{VNC^1_*}$ is closed under Δ^B_1-SCV. □

Corollary 4.13 *VNC^1_* is Σ^B_1-axiomatizable.*

Proof: We can take axioms stating the totality of Σ^B_1-definitions of $L_{\overline{VNC^1_*}}$-functions by Corollary 4.6, and a translation of an open axiom system for $\overline{VNC^1_*}$ to L_0, which exists by Theorem 4.9. The resulting theory exhausts VNC^1_* by Corollary 4.12.

Alternatively, assume that a Σ^B_1-formula $\varphi = \exists \vec{Z} \le t\,\vartheta(\vec{p}, d, x, y, \vec{Z})$ is equivalent to a Π^B_1-formula $\neg\exists \vec{Z} \le t\,\lambda(\vec{p}, d, x, y, \vec{Z})$. Then φ is equivalent to an open $\overline{VNC^1_*}$-formula by Corollary 4.11, hence by the proof of Lemma 4.5, VNC^1_* proves

$$(*)\quad \exists Y \le (|m|+1)n\,\exists Z \le |m|n^2t\,[\mathrm{eval}(n, |m|, \xi, I, Y)$$
$$\wedge \forall d < |m|\,\forall x, y < n\,(\vartheta(\vec{p}, d, x, y, Z^{[d,x,y]}) \vee \lambda(\vec{p}, d, x, y, Z^{[d,x,y]}))],$$

where

$$\xi(d, x, y) \Leftrightarrow \vartheta(\vec{p}, d, x, y, Z^{[d,x,y]}).$$

Clearly, $(*)$ is a Σ_1^B-formula, and it implies

$$\exists Y \leq (|m| + 1)n \ \mathrm{eval}(n, |m|, \varphi, I, Y)$$

over V^0, hence we can axiomatize VNC^1_* by $(*)$ for all such φ over V^0. $\square$

Theorem 4.14 *VNC^1_* is contained in VL.*

Proof: We need to show that VL is closed under the Δ_1^B-SCV rule. If $\varphi \in \Delta_1^B(VL)$, then φ is, provably in VL, log-space computable, hence VL proves comprehension for φ (see [9]). It thus suffices to show that VL proves

$$\forall n, m, E, I \, \exists Y \ \mathrm{eval}(n, |m|, E, I, Y).$$

We will prove this by formalizing in VL the standard log-space algorithm for evaluation of log-depth circuits.

Fix $d_0 \leq |m|$ and $x_0 < n$, we will describe how to evaluate the node x_0 on layer d_0 of the circuit. The idea of the algorithm is to make a depth-first traversal of the circuit, evaluating the nodes along the way, and taking short cuts when we have enough information to determine the value of a particular node. The states of the algorithm will be described by numbers below some a, and we will define the graph of the transition function $F\colon a \to a$ of the algorithm; computation of the algorithm will then be simulated by iterating F using the VL axiom. The states of the algorithm will have the following form:

(i) $\langle \circ, b \rangle$, where $b < 2$. This is the final state, b is the result of the computation.

(ii) $\langle \downarrow, s, x \rangle$, where $x < n$, $0 < s < 2^{|m|+1}$. We have just descended one layer down the circuit. The path from $\langle d_0, x_0 \rangle$ to the current node is recorded by a sequence encoded by s: if the binary expansion of s is $1s_0 \dots s_{k-1}$, then s_i is 0 (1) if we have descended to the left-most (right-most, resp.) child at the ith branching (i.e., at ith layer below the top). The current node is node x on layer $d_0 - k = d_0 - |s| + 1$.

(iii) $\langle \uparrow, s, b, t, i, x \rangle$, where $0 < s < 2^{|m|+1}$, $b < 2$, $t < 2$, $i < |s|$, $x < n$. We have ascended up from a child node. Again, s describes the path to the current node. b is the computed value of the child, and t is 0 if the child was the left-most child, or 1 otherwise. In this situation, we do not know the number of the node we are in, as it cannot be uniquely inferred from the child node; we can however recover it from the sequence s. We compute the node number in a loop with $|s| - 1$ steps, we use i as the loop counter, and x to keep track of the node number. We will obtain the current node number in x when $i = |s| - 1$.

We pick sufficiently large a so that all states above are encoded by a number below a. The function F is Σ_0^B-defined by

$$
\begin{aligned}
F(\langle \circ, b\rangle) &= \langle \circ, b\rangle \\
F(\langle \downarrow, s, x\rangle) &= \begin{cases}
\langle \circ, I(x)\rangle & d_0 = 0 \\
\langle \uparrow, \lfloor s/2\rfloor, I(x), s \,\%\, 2, 0, x_0\rangle & |s| - 1 = d_0 > 0 \\
\langle \uparrow, \lfloor s/2\rfloor, (d_0 - |s|) \,\%\, 2, s \,\%\, 2, 0, x_0\rangle & |s| - 1 < d_0, \\
& \forall y < n\, \neg E(d_0 - |s|, x, y) \\
\langle \downarrow, 2s, l(d_0 - |s| + 1, x)\rangle & |s| - 1 < d_0, \\
& \exists y < n\, E(d_0 - |s|, x, y)
\end{cases} \\
F(\langle \uparrow, s, b, t, i, x\rangle) &= \begin{cases}
\langle \uparrow, s, b, t, i+1, l(d_0 - i, x)\rangle & i < |s| - 1, s_i = 0 \\
\langle \uparrow, s, b, t, i+1, r(d_0 - i, x)\rangle & i < |s| - 1, s_i = 1 \\
\langle \circ, b\rangle & i = |s| - 1 = 0, \\
& t = 1 \text{ or } d_0 - |s| \not\equiv b \pmod 2 \\
\langle \uparrow, \lfloor s/2\rfloor, b, s \,\%\, 2, 0, x_0\rangle & i = |s| - 1 > 0, \\
& t = 1 \text{ or } d_0 - |s| \not\equiv b \pmod 2 \\
\langle \downarrow, 2s+1, r(d_0 - |s| + 1, x)\rangle & i = |s| - 1 > 0, \\
& t = 0, d_0 - |s| \equiv b \pmod 2
\end{cases}
\end{aligned}
$$

where

$$
\begin{aligned}
l(d, x) &= \min\{y < n \mid E(d-1, x, y)\}, \\
r(d, x) &= \max\{y < n \mid E(d-1, x, y)\},
\end{aligned}
$$

and F is defined arbitrarily on other numbers below a. By the VL axiom, there exists a sequence P such that $(P)_0 = \langle \downarrow, 1, x_0\rangle$ and $(P)_{v+1} = F((P)_v)$ for all $v < a$. We leave to the reader to verify that P determines a correct partial evaluation of the original circuit, in particular, $(P)_a = \langle \circ, b\rangle$, where b is the value of node x_0 on layer d_0.

In order to evaluate the whole circuit at once, we take a copy of the above algorithm for every $d_0 \le |m|$ and $x_0 < n$, and "concatenate" them in such a way that a final state $\langle \circ, b\rangle$ of node $\langle d_0, x_0\rangle$ is followed by the initial state $\langle \downarrow, 1, x_0'\rangle$ of the next node $\langle d_0', x_0'\rangle$. We leave the details to the reader. □

Definition 4.15 A function $F(\vec{x}, \vec{X})$ is a *provably total computable function* of a theory $T \supseteq V^0$, if there exists a Σ_1^1-formula $\varphi(\vec{x}, \vec{X}, Y)$ which defines the graph of F in the standard model such that

$$T \vdash \exists! Y\, \varphi(\vec{x}, \vec{X}, Y).$$

Complexity classes like NC^1 can be adapted to the second-order setting in a straightforward way: we represent sets by binary strings, and we write numbers in unary (i.e., as in Definition 4.4).

Corollary 4.16 *The provably total computable functions of VNC^1_* and $\overline{VNC}^1_*$ include the uniform NC^1-functions, and are contained in the L-uniform NC^1-functions.*

Proof: Uniform NC^1-functions are provably total already in VNC^1. On the other hand, assume that $F(\vec{x}, \vec{X})$ is provably total in $\overline{VNC}^1_*$. By Theorem 4.10, F is definable by an $L_{\overline{VNC}^1_*}$-term, hence it is computable by $\Delta^B_1(VNC^1_*)$-circuits using Lemma 4.5. As $VNC^1_* \subseteq VL$, the formula φ defining the circuits as in Definition 4.4 must be in $\Delta^B_1(VL) = L$. The description of the circuits by the formulas φ and τ is a notational variant of the direct connection language, hence F is in L-uniform FNC^1. □

The theory V^i extended by the *axiom of choice*

$$\forall x < a\, \exists X \leq b\, \varphi(x, X) \to \exists Z\, \forall x < a\, \varphi(x, Z^{[x]})$$

for Σ^B_{i+1}-formulas φ is $\forall\exists\Sigma^B_{i+1}$-conservative over V^i (Zambella [14]). We will prove that the axiom of choice for Σ^B_1-formulas can be similarly $\forall\exists\Sigma^B_1$-conservatively added to VNC^1_*. We will in fact show that the same holds for a version of the axiom of choice without the bound on X.

Definition 4.17 Let Γ be a set of formulas. The *unbounded axiom of choice* is the schema

$$(\Gamma\text{-}AC) \qquad \forall x < a\, \exists X\, \varphi(x, X) \to \exists Z\, \forall x < a\, \varphi(x, Z^{[x]}),$$

where $\varphi \in \Gamma$ may have other parameters, and $Z^{[x]}$ denotes $\{u \mid \langle x, u\rangle \in Z\}$, where $\langle\cdot,\cdot\rangle$ is a pairing function. A theory T is closed under the *unbounded choice rule* Γ-CR, if

$$T \vdash \exists X\, \varphi(x, X) \Rightarrow T \vdash \exists Z\, \forall x < a\, \varphi(x, Z^{[x]}),$$

where $\varphi \in \Gamma$ may have other parameters.

It is easy to see that Σ^B_0-AC is equivalent to $\exists\Sigma^B_1$-AC, and similarly for CR.

Theorem 4.18 *Let T be a $\forall\exists\forall\Pi^B_1$-axiomatized extension of V^0 closed under Σ^B_0-CR. Then $T + \exists\Sigma^B_1$-AC is a $\forall\exists\Sigma^B_1$-conservative extension of T.*

Proof:

Claim 1 *Let $\mathcal{M} \vDash T$, $a \in M$, and φ a Σ^B_0-formula with parameters from M. Then there exists a model $\mathcal{N} \vDash T$ such that $\mathcal{M} \preceq_{\exists\Sigma^B_1} \mathcal{N}$, and $\mathcal{N}$ satisfies*

$$\exists Z\, \forall x < a\, \varphi(x, Z^{[x]})$$

or

$$\exists x < a\, \forall X\, \neg\varphi(x, X).$$

Proof: Let $\mathcal{M}_M$ be the expansion of $\mathcal{M}$ by constants for all elements of M. If

$$T + \mathrm{Th}_{\forall\Pi^B_1}(\mathcal{M}_M) + \exists x < a\, \forall X\, \neg\varphi(x, X)$$

is consistent, then any its model $\mathcal{N}$ satisfies the conclusion. Otherwise there is a sentence $\psi = \forall X\, \vartheta(X)$, where $\vartheta \in \Sigma^B_0$ has parameters from M, such that $\mathcal{M} \vDash \psi$, and

$$T \vdash \psi \to \forall x < a\, \exists X\, \varphi(x, X).$$

We can rewrite it as

$$T \vdash \exists X\, (\vartheta(X) \wedge x < a \to \varphi(x, X)),$$

hence

$$T \vdash \exists Z\, \forall x < a\, (\vartheta(Z^{[x]}) \to \varphi(x, Z^{[x]}))$$

by Σ^B_0-CR, which implies

$$\mathcal{M} \vDash \exists Z\, \forall x < a\, \varphi(x, Z^{[x]}).$$

Thus we may take $\mathcal{N} = \mathcal{M}$. □ (Claim 1)

Claim 2 *Any model of T has an $\exists\Sigma^B_1$-elementary extension to a model of $T + \Sigma^B_0$-AC.*

Proof: Let $\mathcal{M}_0 \vDash T$. We enumerate all pairs of an element $a \in M_0$ and a formula $\varphi \in \Sigma^B_0$ with parameters from M_0 as $\langle a_\alpha, \varphi_\alpha \rangle$ for $\alpha < \varkappa$, where $\varkappa$ is a cardinal. We construct an $\exists\Sigma^B_1$-elementary chain of models $\mathcal{N}_\alpha \vDash T$, $\alpha \le \varkappa$, where $\mathcal{N}_0 = \mathcal{M}_0$, $\mathcal{N}_{\alpha+1}$ is obtained from $\mathcal{N}_\alpha$ by an application of Claim 1 using $a = a_\alpha$, $\varphi = \varphi_\alpha$, and $\mathcal{N}_\lambda = \bigcup_{\alpha<\lambda} \mathcal{N}_\alpha$ for limit λ. Notice that validity of T is preserved by unions of $\exists\Sigma^B_1$-elementary chains, as T is $\forall\exists\forall\Pi^B_1$-axiomatized. Then $\mathcal{M}_1 := \mathcal{N}_\varkappa$ is an $\exists\Sigma^B_1$-elementary extension of $\mathcal{M}_0$, $\mathcal{M}_1 \vDash T$, and

$$\mathcal{M}_1 \vDash \forall x < a\, \exists X\, \varphi(x, X) \to \exists Z\, \forall x < a\, \varphi(x, Z^{[x]})$$

for all $a \in M_0$, and $\varphi \in \Sigma^B_0$ with parameters from M_0. We continue in the same way to construct a chain $\mathcal{M}_0 \preceq_{\exists\Sigma^B_1} \mathcal{M}_1 \preceq_{\exists\Sigma^B_1} \mathcal{M}_2 \preceq_{\exists\Sigma^B_1} \dots$, whose union is a model of $T + \Sigma^B_0$-AC. □ (Claim 2)

Assume that $T + \exists\Sigma^B_1$-$AC = T + \Sigma^B_0$-AC proves a $\forall\exists\Sigma^B_1$-formula α, and let $\mathcal{M}$ be any model of T. Take an $\exists\Sigma^B_1$-elementary extension $\mathcal{N} \vDash T + \Sigma^B_0$-$AC$ of $\mathcal{M}$ by Claim 2. Then $\mathcal{N} \vDash \alpha$, hence $\mathcal{M} \vDash \alpha$. □

Corollary 4.19 *$VNC^1_* + \exists\Sigma^B_1$-$AC$ is a $\forall\exists\Sigma^B_1$-conservative extension of VNC^1_*.*

Proof: In view of Theorem 4.18 and Corollary 4.13, it suffices to show that VNC^1_* is closed under Σ^B_0-CR. Let

$$VNC^1_* \vdash \exists X\, \varphi(x, X, \vec{a}, \vec{A}),$$

where $\varphi \in \Sigma^B_0$ with all free variables shown. By Corollary 4.12 and Theorem 4.10, there exists an $L_{\overline{VNC}^1_*}$-term F such that

$$\overline{VNC}^1_* \vdash \varphi(x, F(x, \vec{a}, \vec{A}), \vec{a}, \vec{A}).$$

By Corollary 4.8, there exists an $L_{\overline{VNC}^1_*}$-term G such that $\overline{VNC}^1_*$ proves

$$G(a, \vec{a}, \vec{A}) = \{\langle x, y \rangle \mid x < a, y \in F(x, \vec{a}, \vec{A})\}.$$

Then

$$\overline{VNC}^1_* \vdash \forall x < a\, \varphi(x, G(a, \vec{a}, \vec{A})^{[x]}, \vec{a}, \vec{A}),$$

hence

$$VNC^1_* \vdash \exists Z\, \forall x < a\, \varphi(x, Z^{[x]}, \vec{a}, \vec{A})$$

by Corollary 4.6. □

5 Propositional translation

We will define a propositional formula

$$[\![\varphi(x_1, \dots, x_r, X_1, \dots, X_s)]\!]_{n_1,\dots,n_r,m_1,\dots,m_s}(p_{1,0}, \dots, p_{1,m_1-1}, \dots, p_{s,0}, \dots, p_{s,m_s-1})$$

for each $\Sigma^B_0(L_{\overline{VNC}^1_*})$-formula $\varphi(\vec{x}, \vec{X})$, and natural numbers $\vec{n}, \vec{m}$. Let $X_1, \dots, X_s$ be sets such that $|X_i| \le m_i$, and let $\tilde{X}_i$ denote the propositional valuation which assigns the value 1 to $p_{i,k}$ iff $k \in X_i$. Then the translation is defined in such a way that

$$(1) \qquad [\![\varphi]\!]_{\vec{n},\vec{m}}(\tilde{X}_1, \dots, \tilde{X}_s) = 1 \Leftrightarrow \mathbb{N} \vDash \varphi(\vec{n}, \vec{X}).$$

If $T(\vec{x}, \vec{X})$ is a set $L_{\overline{VNC}^1_*}$-term, we define a bounding term $b_T(\vec{n}, \vec{m})$, that is a number L_0-term such that $|T(\vec{n}, \vec{X})| \le b_T(\vec{n}, \vec{m})$ whenever $|X_i| \le m_i$ for each i, and we define propositional formulas $[\![T]\!]^k_{\vec{n},\vec{m}}$ for $k < b_T(\vec{n}, \vec{m})$ so that

$$(2) \qquad [\![T]\!]^k_{\vec{n},\vec{m}}(\tilde{X}_1, \dots, \tilde{X}_s) = 1 \Leftrightarrow \mathbb{N} \vDash k \in T(\vec{n}, \vec{X}).$$

Finally, if $t(\vec{x}, \vec{X})$ is a number $L_{\overline{VNC}^1_*}$-term, we define a bounding L_0-term b_t such that $t(\vec{n}, \vec{X}) \le b_t(\vec{n}, \vec{m})$ whenever $|X_i| \le m_i$ for all i, and we introduce propositional formulas $[\![t]\!]^k_{\vec{n},\vec{m}}$ for $k \le b_t(\vec{n}, \vec{m})$ so that

$$(3) \qquad [\![t]\!]^k_{\vec{n},\vec{m}}(\tilde{X}_1, \dots, \tilde{X}_s) = 1 \Leftrightarrow \mathbb{N} \vDash t(\vec{n}, \vec{X}) = k.$$

The bounding terms are defined inductively as follows:

$$\begin{aligned}
b_{x_i}(\vec{n}, \vec{m}) &= n_i, \\
b_{X_j}(\vec{n}, \vec{m}) &= m_j, \\
b_{f(t_1,\dots,t_r)}(\vec{n}, \vec{m}) &= f(b_{t_1}(\vec{n}, \vec{m}), \dots, b_{t_r}(\vec{n}, \vec{m})), \qquad f \in \{0, \mathrm{s}, +, \cdot, |x|\}, \\
b_{|T|}(\vec{n}, \vec{m}) &= b_T(\vec{n}, \vec{m}), \\
b_{C_\varphi(s,\vec{t},\vec{T})}(\vec{n}, \vec{m}) &= b_s(\vec{n}, \vec{m}), \\
b_{Y_\varphi(\vec{t},s,u,T)}(\vec{n}, \vec{m}) &= (|b_u(\vec{n}, \vec{m})| + 1) b_s(\vec{n}, \vec{m}).
\end{aligned}$$

The translations $[\![\varphi]\!]_{\vec{n},\vec{m}}$, $[\![T]\!]^k_{\vec{n},\vec{m}}$, $[\![t]\!]^k_{\vec{n},\vec{m}}$ are defined by simultaneous induction on complexity, along with formulas $\{\!\!\{R\}\!\!\}_{\vec{n},\vec{m}}$, $\{\!\!\{F\}\!\!\}^k_{\vec{n},\vec{m}}$, $\{\!\!\{f\}\!\!\}^k_{\vec{n},\vec{m}}$ for predicates R (including equality), set function symbols F, and number function symbols f. (The formulas $\{\!\!\{\alpha\}\!\!\}$ are minor variants of $[\![\alpha]\!]$, cf. Lemma 5.1 (v). We introduce them to make the definition of $[\![\alpha(\vec{t}, \vec{T})]\!]$ below uniform, so that we do not have to treat specially the case where $\vec{t}, \vec{T}$ are simple variables.) Let us denote

$$I(\varphi) = \begin{cases} \top & \text{if } \varphi \text{ holds,} \\ \bot & \text{otherwise.} \end{cases}$$

If α is a predicate or function symbol, we put

$$\begin{aligned} \llbracket \alpha(t_1, \dots, t_r, T_1, \dots, T_s) \rrbracket^k_{\vec{n},\vec{m}} = & \bigvee_{\substack{k_1 \le b_{t_1}(\vec{n},\vec{m}) \\ \cdots \\ k_r \le b_{t_r}(\vec{n},\vec{m})}} \Big(\bigwedge_{i=1}^{r} \llbracket t_i \rrbracket^{k_i}_{\vec{n},\vec{m}} \\ & \wedge \{\!\{\alpha\}\!\}^k_{\vec{k}, b_{T_1}(\vec{n},\vec{m}), \dots, b_{T_s}(\vec{n},\vec{m})} \big(\llbracket T_1 \rrbracket^0_{\vec{n},\vec{m}}, \dots, \llbracket T_1 \rrbracket^{b_{T_1}-1}_{\vec{n},\vec{m}}, \dots, \llbracket T_s \rrbracket^0_{\vec{n},\vec{m}}, \dots, \llbracket T_s \rrbracket^{b_{T_s}-1}_{\vec{n},\vec{m}} \big) \Big), \end{aligned}$$

where the superscript k is omitted if α is a predicate. We further define

$$\begin{aligned} \llbracket x_i \rrbracket^k_{\vec{n},\vec{m}} &= I(k = n_i), \\ \llbracket X_j \rrbracket^k_{\vec{n},\vec{m}} &= p_{j,k}, \\ \llbracket \varphi \circ \psi \rrbracket_{\vec{n},\vec{m}} &= \llbracket \varphi \rrbracket_{\vec{n},\vec{m}} \circ \llbracket \psi \rrbracket_{\vec{n},\vec{m}}, \qquad \circ \in \{\wedge, \vee, \neg\}, \\ \llbracket \exists x \le t\, \varphi \rrbracket_{\vec{n},\vec{m}} &= \bigvee_{k \le b_t(\vec{n},\vec{m})} \llbracket x \le t \wedge \varphi \rrbracket_{k,\vec{n},\vec{m}}, \\ \llbracket \forall x \le t\, \varphi \rrbracket_{\vec{n},\vec{m}} &= \bigwedge_{k \le b_t(\vec{n},\vec{m})} \llbracket x \le t \to \varphi \rrbracket_{k,\vec{n},\vec{m}}, \\ \{\!\{R\}\!\}_{n,n'} &= I(n \mathrel{R} n'), \qquad R \in \{\le, =\}, \\ \{\!\{\in\}\!\}_{n,m}(\vec{p}) &= \begin{cases} p_n & \text{if } n < m, \\ \bot & \text{otherwise,} \end{cases} \\ \{\!\{X = Y\}\!\}_{m,m'}(\vec{p}, \vec{q}) &= \bigwedge_{j < \min(m,m')} (p_j \leftrightarrow q_j) \wedge \bigwedge_{j=m'}^{m-1} \neg p_j \wedge \bigwedge_{j=m}^{m'-1} \neg q_j, \\ \{\!\{f\}\!\}^k_{\vec{n}} &= I(f(\vec{n}) = k), \qquad f \in \{0, \mathrm{s}, +, \cdot, |x|\}, \\ \{\!\{|X|\}\!\}^k_m(\vec{p}) &= \begin{cases} p_{k-1} \wedge \bigwedge_{j=k}^{m-1} \neg p_j & \text{if } k > 0, \\ \bigwedge_{j<m} \neg p_j & \text{otherwise,} \end{cases} \\ \{\!\{C_{\varphi(u,\vec{x},\vec{X})}\}\!\}^k_{n,\vec{n},\vec{m}}(\vec{p}) &= I(k < n) \wedge \llbracket \varphi \rrbracket_{k,\vec{n},\vec{m}}(\vec{p}). \end{aligned}$$

It remains to define $\{\!\{Y_\varphi(\vec{p}, n, r, I)\}\!\}^k_{\vec{p},n,r,m}$ for an open $L_{\overline{VNC^1_*}}$-formula $\varphi(\vec{p}, d, x, y)$. We fix $\vec{p}, n, m, r$, and we write $\{\!\{Y_\varphi\}\!\}^{d,x}$ for $\{\!\{Y_\varphi\}\!\}^{dn+x}_{\vec{p},n,r,m}$, where $x < n$. As φ has no free set variables, $\llbracket \varphi \rrbracket_{\vec{p},d,x,y}$ is a Boolean sentence with a definite truth value. We may thus define the relations

$$\begin{aligned} e(d, x, y) &\Leftrightarrow \llbracket \varphi \rrbracket_{\vec{p},d,x,y} = 1, \\ e^*(d, x, y) &\Leftrightarrow e(d, x, y) \wedge \Big(\bigwedge_{z<y} \neg e(d, x, z) \vee \bigwedge_{z=y+1}^{n-1} \neg e(d, x, z) \Big) \end{aligned}$$

for $d < |m|$, $x, y < n$. By induction on $d < |m|$, we define

$$\{\!\{Y_\varphi\}\!\}^{0,x} = \begin{cases} p_x & \text{if } x < r, \\ \bot & \text{otherwise,} \end{cases}$$

$$\{\!\{Y_\varphi\}\!\}^{d+1,x} = \begin{cases} \bigvee\limits_{e^*(d,x,y)} \{\!\{Y_\varphi\}\!\}^{d,y} & \text{if } d \text{ is even,} \\ \bigwedge\limits_{e^*(d,x,y)} \{\!\{Y_\varphi\}\!\}^{d,y} & \text{if } d \text{ is odd.} \end{cases}$$

We also put $\{\!\{Y_\varphi\}\!\}^{d,x} = \bot$ for $d > |m|$. Notice that the definition of e^* ensures that there are at most two y such that $e^*(d, x, y)$ for any given d, x, hence the conjunctions and disjunctions in the definition of $\{\!\{Y_\varphi\}\!\}^{d+1,x}$ are at most binary. As the formulas have depth $d \le |m|$, they are of size $O(m)$. It follows by induction on complexity that the formulas $[\![\alpha]\!]^{(k)}_{\vec{n},\vec{m}}$ for any fixed formula or term α have size $poly(\vec{n}, \vec{m})$ and logarithmic depth. In fact, $[\![\alpha]\!]^{(k)}_{\vec{n},\vec{m}}$ is constructible in logarithmic space given $\vec{n}, \vec{m}, k$ in unary (the most difficult case is again Y_φ: we observe that we can evaluate the logspace-constructible Boolean sentence $[\![\varphi]\!]$ needed to define e in logarithmic space). It is also straightforward to show (1), (2), (3) by induction on complexity.

We recall that a *Frege system* is a propositional proof system given by a finite set F of rules of the form

$$\frac{\varphi_1, \dots, \varphi_n}{\varphi}$$

which is sound and implicationally complete. An F-proof of a formula φ is a sequence of propositional formulas ending with φ such that every formula is derived from previous formulas by an instance of an F-rule. By a well-known theorem of Cook and Reckhow [10], all Frege systems are polynomially equivalent, hence the choice of the basic rules does not matter (often one takes Modus Ponens and a list of axioms). Frege systems are also polynomially equivalent to the propositional version of Gentzen's sequent calculus LK, which is easier to work with in some contexts.

Lemma 5.1

(*i*) *If τ, σ are terms, then $b_{\tau(\vec{x},\vec{X},\sigma(\vec{x},\vec{X}))}(\vec{n}, \vec{m}) = b_\tau(\vec{n}, \vec{m}, b_\sigma(\vec{n}, \vec{m}))$.*

(*ii*) *If $\alpha(\vec{x}, \vec{X}, Y)$ is a formula or term, and $T(\vec{x}, \vec{X})$ is a set term, then*

$$[\![\alpha(\vec{x}, \vec{X}, T(\vec{x}, \vec{X}))]\!]^{(k)}_{\vec{n},\vec{m}} = [\![\alpha]\!]^{(k)}_{\vec{n},\vec{m},b_T(\vec{n},\vec{m})}\big([\![T]\!]^0_{\vec{n},\vec{m}}, \dots, [\![T]\!]^{b_T(\vec{n},\vec{m})-1}_{\vec{n},\vec{m}}\big),$$

where k is present only if α is a term, and on the right-hand side the formulas are substituted for the variables corresponding to Y.

(*iii*) *If $t(\vec{x}, \vec{X})$ is a number term, there are size $poly(\vec{n}, \vec{m})$ log-space constructible Frege proofs of the formulas*

$$\bigvee_{k \le b_t(\vec{n},\vec{m})} [\![t]\!]^k_{\vec{n},\vec{m}},$$

$$\bigwedge_{k < l \le b_t(\vec{n},\vec{m})} \big([\![t]\!]^k_{\vec{n},\vec{m}} \to \neg [\![t]\!]^l_{\vec{n},\vec{m}}\big).$$

(iv) *If $\alpha(y,\vec{x},\vec{X})$ is a formula or term, and $t(\vec{x},\vec{X})$ is a number term, then there are size poly$(\vec{n},\vec{m})$ log-space constructible Frege proofs of the formulas*

$$[\![\alpha(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{(k)}_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{r\le b_t(\vec{n},\vec{m})} \big([\![t]\!]^r_{\vec{n},\vec{m}} \wedge [\![\alpha]\!]^{(k)}_{r,\vec{n},\vec{m}}\big),$$

where k is present only if α is a term, and we put $[\![\alpha]\!]^k_{r,\vec{n},\vec{m}} = \bot$ if α is a number term and $k > b_\alpha(\vec{n},\vec{m})$, or if α is a set term and $k \ge b_\alpha(\vec{n},\vec{m})$.

(v) *If $\alpha(\vec{x},\vec{X})$ is a predicate or function symbol, there are size poly$(\vec{n},\vec{m})$ log-space constructible Frege proofs of*

$$\{\!\{\alpha\}\!\}^{(k)}_{\vec{n},\vec{m}} \leftrightarrow [\![\alpha]\!]^{(k)}_{\vec{n},\vec{m}}.$$

Proof: By straightforward induction on complexity. For example, we will show the proof of the step for $\alpha = \beta(\vec{t},\vec{T})$ in (iv), where β is a predicate or function symbol. Let $r \le b_t(\vec{n},\vec{m})$. By the induction hypothesis, we can construct proofs of

$$[\![t_i(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{k_i}_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{s\le b_t(\vec{n},\vec{m})} \big([\![t]\!]^s_{\vec{n},\vec{m}} \wedge [\![t_i]\!]^{k_i}_{s,\vec{n},\vec{m}}\big),$$

hence we construct proofs of

$$[\![t]\!]^r_{\vec{n},\vec{m}} \to \big([\![t_i(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{k_i}_{\vec{n},\vec{m}} \leftrightarrow [\![t_i]\!]^{k_i}_{r,\vec{n},\vec{m}}\big)$$

using (iii). Similarly, we can construct proofs of

$$[\![t]\!]^r_{\vec{n},\vec{m}} \to \big([\![T_i(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{j}_{\vec{n},\vec{m}} \leftrightarrow [\![T_i]\!]^{j}_{r,\vec{n},\vec{m}}\big).$$

Using the definition of $[\![\beta(\vec{t},\vec{T})]\!]$ and (i), we infer

$$[\![t]\!]^r_{\vec{n},\vec{m}} \to \Big[[\![\alpha(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{(k)}_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{\substack{k_1\le b_{t_1}(b_t(\vec{n},\vec{m}),\vec{n},\vec{m})\\ \cdots}} \Big(\bigwedge_i [\![t_i]\!]^{k_i}_{r,\vec{n},\vec{m}} \\ \wedge \{\!\{\beta\}\!\}^{(k)}_{\vec{k},b_{T_1}(b_t(\vec{n},\vec{m}),\vec{n},\vec{m}),\dots}([\![T_1]\!]^0_{r,\vec{n},\vec{m}},\dots)\Big)\Big].$$

It is easy to see that there are short proofs of

$$\{\!\{\beta\}\!\}^{(k)}_{\vec{k},\vec{v}}(\vec{p}) \leftrightarrow \{\!\{\beta\}\!\}^{(k)}_{\vec{k},\vec{u}}(\vec{p},\vec{\bot})$$

for any $\vec{u} \ge \vec{v}$. Using the fact that $b_{T_j}(r,\vec{n},\vec{m}) \le b_{T_j}(b_t(\vec{n},\vec{m}),\vec{n},\vec{m})$, and the definition of $[\![t_i]\!]^j$ or $[\![T_i]\!]^j$ as $\bot$ for too large j, we obtain a proof of

$$[\![t]\!]^r_{\vec{n},\vec{m}} \to \Big[[\![\alpha(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{(k)}_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{\substack{k_1\le b_{t_1}(r,\vec{n},\vec{m})\\ \cdots}} \Big(\bigwedge_i [\![t_i]\!]^{k_i}_{r,\vec{n},\vec{m}} \\ \wedge \{\!\{\beta\}\!\}^{(k)}_{\vec{k},b_{T_1}(r,\vec{n},\vec{m}),\dots}([\![T_1]\!]^0_{r,\vec{n},\vec{m}},\dots)\Big)\Big],$$

hence

$$[\![t]\!]^r_{\vec{n},\vec{m}} \to \big([\![\alpha(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{(k)}_{\vec{n},\vec{m}} \leftrightarrow [\![\alpha]\!]^{(k)}_{r,\vec{n},\vec{m}}\big)$$

by the definition of $[\![\beta(\vec{t},\vec{T})]\!]$. We get the required

$$[\![\alpha(t(\vec{x},\vec{X}),\vec{x},\vec{X})]\!]^{(k)}_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{r\le b_t(\vec{n},\vec{m})} \big([\![t]\!]^r_{\vec{n},\vec{m}} \wedge [\![\alpha]\!]^{(k)}_{r,\vec{n},\vec{m}}\big)$$

using (iii). □

Theorem 5.2 *Let $\varphi(\vec{x},\vec{X})$ be a $\Sigma^B_0(L_{\overline{VNC}^1_*})$-formula provable in $\overline{VNC}^1_*$. Then the formulas $[\![\varphi]\!]_{\vec{n},\vec{m}}$ have Frege proofs of size poly$(\vec{n},\vec{m})$ constructible in logarithmic space.*

Proof: It will be more convenient to work with sequent calculus, which is p-equivalent to Frege systems. The sequent $\vdash \varphi$ has an LK-proof π using substitution instances of axioms of $\overline{VNC}^1_*$ and equality axioms as extra initial sequents. We may reformulate the extensionality axiom as

$$\forall x < |X|\,(x \in X \to x \in Y) \wedge \forall x < |Y|\,(x \in Y \to x \in X) \to X = Y,$$

hence all the initial sequents are $\Sigma^B_0(L_{\overline{VNC}^1_*})$. Using the free-cut elimination theorem [4], we may thus assume that all formulas in π are $\Sigma^B_0(L_{\overline{VNC}^1_*})$. We will show by induction on the length of the proof that for every sequent $\Gamma \vdash \Delta$ in π, the sequents $[\![\Gamma]\!]_{\vec{n},\vec{m}} \vdash [\![\Delta]\!]_{\vec{n},\vec{m}}$ have propositional LK-proofs constructible in logarithmic space, where $[\![\Gamma]\!]_{\vec{n},\vec{m}}$ denotes $\{[\![\psi]\!]_{\vec{n},\vec{m}} \mid \psi \in \Gamma\}$ for any set of formulas Γ.

The induction steps for the cut rule, propositional rules, and structural rules is trivial, we simply use the induction hypothesis and apply the same rule.

If the last rule in the proof is the $\forall$-right rule, it must have the form

$$\frac{\Gamma \vdash y \le t \to \psi(y), \Delta}{\Gamma \vdash \forall x \le t\, \psi(x), \Delta}$$

as the conclusion is $\Sigma^B_0(L_{\overline{VNC}^1_*})$. By the induction hypothesis we can construct proofs of

$$[\![\Gamma]\!]_{\vec{n},\vec{m}} \vdash [\![y \le t \to \psi(y)]\!]_{r,\vec{n},\vec{m}}, [\![\Delta]\!]_{\vec{n},\vec{m}}$$

for every $r \le b_t(\vec{n},\vec{m})$, from which we derive

$$[\![\Gamma]\!]_{\vec{n},\vec{m}} \vdash \bigwedge_{r\le b_t(\vec{n},\vec{m})} [\![y \le t \to \psi(y)]\!]_{r,\vec{n},\vec{m}}, [\![\Delta]\!]_{\vec{n},\vec{m}}$$

using the $\wedge$-right rule. The case of $\exists$-left is similar.

If the last rule in the proof is the $\exists$-right rule, it must have the form

$$\frac{\Gamma \vdash s \le t \wedge \psi(s), \Delta}{\Gamma \vdash \exists x \le t\, \psi(x), \Delta}$$

where s is a term. By the induction hypothesis we can construct a proof of

$$[\![\Gamma]\!]_{\vec{n},\vec{m}} \vdash [\![s \le t \wedge \psi(s)]\!]_{\vec{n},\vec{m}}, [\![\Delta]\!]_{\vec{n},\vec{m}}.$$

By Lemma 5.1 (iv), there are short Frege proofs of

$$[\![s \le t \wedge \psi(s)]\!]_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{r \le b_s(\vec{n},\vec{m})} ([\![s]\!]^r_{\vec{n},\vec{m}} \wedge [\![x \le t \wedge \psi(x)]\!]_{r,\vec{n},\vec{m}}).$$

Moreover, we can construct Frege proofs of $\neg[\![x \le t \wedge \psi(x)]\!]_{r,\vec{n},\vec{m}}$ for all $b_t(\vec{n},\vec{m}) < r \le b_s(\vec{n},\vec{m})$, hence we can construct a proof of the sequent

$$[\![s \le t \wedge \psi(s)]\!]_{\vec{n},\vec{m}} \vdash \bigvee_{r \le b_t(\vec{n},\vec{m})} [\![x \le t \wedge \psi(x)]\!]_{\vec{r},\vec{n},\vec{m}}.$$

We derive

$$[\![\Gamma]\!]_{\vec{n},\vec{m}} \vdash \bigvee_{r \le b_t(\vec{n},\vec{m})} [\![x \le t \wedge \psi(x)]\!]_{\vec{r},\vec{n},\vec{m}}, [\![\Delta]\!]_{\vec{n},\vec{m}}$$

by a cut. The case of the $\forall$-left rule is analogous.

It remains to construct proofs of propositional translations of substitution instances of axioms of $\overline{VNC}^1_*$ and equality axioms. If $\psi' = \psi(\vec{t},\vec{T})$ is an instance of an axiom ψ, then there are short Frege proofs of

$$(*) \qquad [\![\psi']\!]_{\vec{n},\vec{m}} \leftrightarrow \bigvee_{\substack{k_1 \le b_{t_1}(\vec{n},\vec{m})\\ \cdots}} \Big(\bigwedge_i [\![t_i]\!]^{k_i}_{\vec{n},\vec{m}} \wedge [\![\psi]\!]_{\vec{k},b_{T_1}(\vec{n},\vec{m}),\ldots}([\![T_1]\!]^0_{\vec{n},\vec{m}},\ldots)\Big)$$

by Lemma 5.1 (ii,iv). If we can construct short proofs of $[\![\psi]\!]$, we can substitute the formulas $[\![T_i]\!]^j_{\vec{n},\vec{m}}$ in the proof (incurring a polynomial blow-up) and combine it with Lemma 5.1 (iii) to obtain the right-hand side of $(*)$. It thus suffices to construct translations of the base form of the axioms.

Axioms of *BASIC* and equality axioms for L_0 are provable in V^0, hence their translations have log-space constructible proofs already in bounded-depth Frege [9].

The Σ^B_0-$\overline{COMP}$ axiom translates to

$$[\![u \in C_\psi(v,\vec{x},\vec{X})]\!]_{k,l,\vec{n},\vec{m}} \leftrightarrow [\![u < v]\!]_{k,l} \wedge [\![\psi(u,\vec{x},\vec{X})]\!]_{k,\vec{n},\vec{m}},$$

which can be proven equivalent to the tautology

$$I(k < l) \wedge [\![\psi]\!]_{k,\vec{n},\vec{m}} \leftrightarrow I(k < l) \wedge [\![\psi]\!]_{k,\vec{n},\vec{m}}$$

by Lemma 5.1 (v) and the definition of $\{\!\{C_\psi\}\!\}$.

Consider an instance

$$|Y_\psi(\vec{p},n,r,I)| \le (|r|+1)n \wedge \mathrm{eval}(n,|r|,\psi,I,Y_\psi(\vec{p},n,r,I))$$

of *Open*-$\overline{SCV}$. We can prove

$$[\![|Y_\psi(\vec{p},n,r,I)| \le (|r|+1)n]\!]_{\vec{p},n,r,m}$$

easily using Lemma 5.1 (iii) and $b_{Y_\psi} = (|r|+1)n$. Using the notation from the definition of $\{\!\{Y_\psi\}\!\}$, we can construct short proofs of

$$[\![dn + x \in Y_\psi(\vec{p},n,r,I)]\!]_{d,x,\vec{p},n,r,m} \leftrightarrow \{\!\{Y_\psi\}\!\}^{d,x}$$

using Lemma 5.1 (v). As there are short proofs evaluating the Boolean sentences $[\![2 \mid d]\!]_d$ and $[\![\psi^*(\vec{p}, d, x, y)]\!]_{\vec{p},d,x,y}$ to $I(2 \mid d)$ and $I(e^*(d, x, y))$, we can construct short proofs of

$$\{\!\{Y_\psi\}\!\}^{d+1,x} \leftrightarrow \Big(\Big([\![2 \mid d]\!]_d \wedge \bigvee_{y<n} ([\![\psi^*]\!]_{\vec{p},d,x,y} \wedge \{\!\{Y_\psi\}\!\}^{d,y})\Big) \\ \vee \Big([\![2 \nmid d]\!]_d \wedge \bigwedge_{y<n} ([\![\psi^*]\!]_{\vec{p},d,x,y} \to \{\!\{Y_\psi\}\!\}^{d,y})\Big)\Big)$$

for $d < |r|$ and $x < n$, using the definition of $\{\!\{Y_\psi\}\!\}^{d+1,x}$. Similarly, we construct proofs of

$$\{\!\{Y_\psi\}\!\}^{0,x} \leftrightarrow [\![x \in I]\!]_{x,m}.$$

Putting it all together, we obtain a proof of

$$[\![\mathrm{eval}(n, |r|, \psi, I, Y_\psi(\vec{p}, n, r, I))]\!]_{\vec{p},n,r,m}.$$

Translation of the equality axioms for C_ψ and Y_ψ is easy and left to the reader. (As a matter of fact, one can show that these axioms are redundant in $\overline{VNC}^1_*$.) $\square$

6 Acknowledgement

I am grateful to Phuong Nguyen for enlightening discussions on VNC^1.

References

[1] Toshiyasu Arai, *A bounded arithmetic AID for Frege systems*, Annals of Pure and Applied Logic 103 (2000), pp. 155–199.

[2] Albert Atserias, Nicola Galesi, and Pavel Pudlák, *Monotone simulations of non-monotone proofs*, Journal of Computer and System Sciences 65 (2002), no. 4, pp. 626–638.

[3] Samuel R. Buss, *The Boolean formula value problem is in ALOGTIME*, in: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 123–131.

[4] __________, *An introduction to proof theory*, in: Handbook of Proof Theory (S. R. Buss, ed.), Studies in Logic and the Foundations of Mathematics vol. 137, Elsevier, Amsterdam, 1998, pp. 1–78.

[5] Peter Clote, *ALOGTIME and a conjecture of S.A. Cook*, Annals of Mathematics and Artificial Intelligence 6 (1992), no. 1–3, pp. 57–106.

[6] Peter Clote and Gaisi Takeuti, *Bounded arithmetic for NC, ALogTIME, L and NL*, Annals of Pure and Applied Logic 56 (1992), pp. 73–117.

[7] Stephen Cook and Tsuyoshi Morioka, *Quantified propositional calculus and a second-order theory for* $\mathbf{NC}^1$, Archive for Mathematical Logic 44 (2005), no. 6, pp. 711–749.

[8] Stephen A. Cook, *Theories for complexity classes and their propositional translations*, in: Complexity of computations and proofs (J. Krajíček, ed.), Quaderni di Matematica vol. 13, Seconda Universita di Napoli, 2004, pp. 175–227.

[9] Stephen A. Cook and Phuong Nguyen, *Logical foundations of proof complexity*, book in preparation, `http://www.cs.toronto.edu/~sacook/homepage/book/`.

[10] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic 44 (1979), no. 1, pp. 36–50.

[11] Emil Jeřábek, *A sorting network in bounded arithmetic*, preprint, 2008.

[12] Jan Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications vol. 60, Cambridge University Press, 1995.

[13] Walter L. Ruzzo, *On uniform circuit complexity*, Journal of Computer and System Sciences 22 (1981), no. 3, pp. 365–383.

[14] Domenico Zambella, *Notes on polynomially bounded arithmetic*, Journal of Symbolic Logic 61 (1996), no. 3, pp. 942–966.