Propositional Translation for $\mathbf{VTC}^0$

Phuong Nguyen

May 15, 2005

1 Two-Sorted First-Order Logic

1.1 Syntax and Semantics

We use the two-sorted syntax of Zambella [13, 14] (see also [5, 4]), which was inspired by Buss's second-order theories defined in [2]. Our language has two sorts of variables: the number variables $x, y, z, \ldots$ whose intended values are natural numbers; and string variables $X, Y, Z, \ldots$, whose intended values are finite sets of natural numbers (which represent binary strings). Our two-sorted vocabulary $\mathcal{L}_A^2$ extends that of Peano Arithmetic:

$$\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =^1, =^2].$$

Here $|\ |$ is a function from strings to numbers, and the intended meaning of $|X|$ is 1 plus the largest element of X. The binary predicate $\in$ denotes set membership. We will use the abbreviation $X(t)$ for $t \in X$. The equality predicates $=^1$ and $=^2$ are for numbers and strings, respectively. We will write $=$ for both $=^1$ and $=^2$; the exact meaning will be clear from the context. The other symbols have their standard meanings.

Number terms are built from the constants 0,1, variables $x, y, z, \ldots$, and length terms $|X|$ using $+$ and $\cdot$. We use $s, t, \ldots$ for number terms. The only *string terms* are string variables $X, Y, Z, \ldots$. The atomic formulas are $\top$, $\bot$, (for True, False), $s = t$, $X = Y$, $s \leq t$, $t \in X$ for any number terms s, t and string variables X, Y. Formulas are built from atomic formulas using $\wedge, \vee, \neg$ and both number and string quantifiers $\exists x, \exists X, \forall x, \forall X$. Bounded number quantifiers are defined as usual, and the bounded string quantifier $\exists X \leq t\ \varphi$ stands for $\exists X (|X| \leq t \wedge \varphi)$ and $\forall X \leq t\ \varphi$ stands for $\forall X (|X| \leq t \supset \varphi)$, where X does not occur in the term t.

A structure for $\mathcal{L}_A^2$ is defined in the same way as a structure for a single-sorted language, except now there are two nonempty domains U_1 and U_2, one for numbers and one for strings. Each symbol of $\mathcal{L}_A^2$ is interpreted in $\langle U_1, U_2 \rangle$ by a relation or function of appropriate type, with $=^1$ and $=^2$ interpreted as true equality on U_1 and U_2, respectively. In the standard structure $\underline{\mathbb{N}}_2$, U_1 is $\mathbb{N}$ and U_2 is the set of finite subsets of $\mathbb{N}$. Each symbol of $\mathcal{L}_A^2$ gets its intended interpretation.

In general we will consider a vocabulary $\mathcal{L}$ which extends $\mathcal{L}_A^2$. A formula is $\Sigma_0^B(\mathcal{L})$ if it has no string quantifiers and all number quantifiers are bounded. A formula is $\Sigma_1^B(\mathcal{L})$ ($\Pi_1^B(\mathcal{L})$, $\Sigma_1^1(\mathcal{L})$, resp.) if it is a $\Sigma_0^B(\mathcal{L})$ formula preceded by a block of quantifiers of the form $\exists X \leq t$ ($\forall X \leq t$, $\exists X$, resp.). If the block contains a single quantifier, the formula is also called single-$\Sigma_1^B(\mathcal{L})$ (single-$\Pi_1^B(\mathcal{L})$, single-$\Sigma_1^1(\mathcal{L})$, resp.). A formula is $g\Sigma_1^B(\mathcal{L})$ (resp. $g\Pi_1^B(\mathcal{L})$) if it is obtained from $\Sigma_0^B(\mathcal{L})$ formulas using the connectives $\wedge$ and $\vee$, bounded number quantifiers and bounded string existential (resp. universal) quantifier ("g" for "general"). A formula is $\exists g\Sigma_1^B(\mathcal{L})$ if it is of the form $\exists \overline{X}\varphi$, where φ is $g\Sigma_1^B(\mathcal{L})$. We will omit $\mathcal{L}$ if it is $\mathcal{L}_A^2$

ulas correspond to (in first-order logic) strict Σ_1^b formulas (i.e., Σ_1^b formulas where no bounded
inside the scope of any sharply bounded quantifier), while $g\Sigma_1^B$ formulas correspond to Σ_1^b
milar for Π_1^B and $g\Pi_1^B$ formulas.

When we consider a vocabulary $\mathcal{L}$, we require that the bounding terms $\vec{t}$ to be $\mathcal{L}_A^2$ terms. In par-
) and $\mathbf{\Pi}_i^B(\mathcal{L})$ are defined similarly to $\mathbf{\Sigma}_i^B$ and $\mathbf{\Pi}_i^B$, with the additional requirement
of all the quantifiers are $\mathcal{L}_A^2$ terms.

xiomatized by the set of axioms 2-**BASIC** (Figure 1) and the $\mathbf{\Sigma}_0^B$-**COMP** axiom
^{B} is the set of all formula of the form

$$\exists X < a \forall z < a,\ X(z) \leftrightarrow \varphi(z) \tag{1}$$

not containing X.

B1. $x + 1 \neq 0$	**B7.** $(x \leq y \wedge y \leq x) \supset x = y$
B2. $x + 1 = y + 1 \supset x = y$	**B8.** $x \leq x + y$
B3. $x + 0 = x$	**B9.** $0 \leq x$
B4. $x + (y+1) = (x+y) + 1$	**B10.** $x \leq y \vee y \leq x$
B5. $x \cdot 0 = 0$	**B11.** $x \leq y \leftrightarrow x < y + 1$
B6. $x \cdot (y+1) = (x \cdot y) + x$	**B12.** $x \neq 0 \supset \exists y \leq x(y + 1 = x)$
L1. $X(y) \supset y < \lvert X \rvert$	**L2.** $y + 1 = \lvert X \rvert \supset X(y)$
SE. $[\lvert X \rvert = \lvert Y \rvert \wedge \forall i < \lvert X \rvert (X(i) \leftrightarrow Y(i))] \ \supset\ X = Y$	

Figure 1: 2-**BASIC**

It has been shown [4, 5] that $\mathbf{V}^0$ characterizes $\mathbf{AC}^0$ in the sense that the $\mathbf{AC}^0$ functions are precisely the $\mathbf{\Sigma}_1^B$-definable functions of $\mathbf{V}^0$. An important $\mathbf{AC}^0$ function is $Row(z, X)$ (also $X^{[z]}$) which is defined as

$$|Row(z, X)| \leq |X| \wedge \forall x < |X|\, Row(z, X)(x) \leftrightarrow X(z, x)$$

Using Row we can code any finitely many strings into one string. Also, it is $\mathbf{\Sigma}_0^B$-definable in $\mathbf{V}^0$, and $\mathbf{V}^0(Row)$ proves $\mathbf{\Sigma}_0^B(Row)$-**COMP**. Indeed, every $\mathbf{\Sigma}_0^B(Row)$-formula is provably equivalent in $\mathbf{V}^0(Row)$ to a $\mathbf{\Sigma}_0^B$ formula.

The theory $\mathbf{VTC}^0$ defined below characterizes $\mathbf{TC}^0$ in the same way [10, 11]. Consider the function $numones(x, X)$ which is the number of elements of X that are $< x$ (thus the number of elements of X is $numones(|X|, X)$). The axiom *NUMONES* states the existence of a counting array Y for any given string X, i.e., $Y(x, y) \leftrightarrow numones(x, X) = y$.

$$\begin{aligned} NUMONES \equiv & \forall X \exists Y,\ \forall z \leq |X| \exists ! y \leq |X| Y(z, y)\ \wedge\ Y(0,0)\ \wedge \\ & \forall z < |X| \forall y \leq |X|,\ Y(z, y) \supset [(X(z) \supset Y(z+1, y+1)) \wedge (\neg X(z) \supset Y(z+1, y))]. \end{aligned} \tag{2}$$

Notice that Y has length bounded by $1 + \langle |X|, |X| \rangle$.

Definition 1.1 **($\mathbf{VTC}^0$).** *The theory* $\mathbf{VTC}^0$ *has vocabulary* $\mathcal{L}_A^2$*, and is axiomatized by* $\mathbf{V}^0$ *together with NUMONES.*

Proposition 1.2. *The function numones is* $\mathbf{\Sigma}_1^B$*-definable in* $\mathbf{VTC}^0$*.*

Proof. It is easy to see that *numones* can be defined by the following $\mathbf{\Sigma}_1$-formula:

$$\begin{aligned} numones(x, X) = y \leftrightarrow & \exists Y,\ \forall z \leq x \exists ! y \leq x Y(z, y)\ \wedge\ Y(0,0)\ \wedge \\ & \forall z < x \forall y \leq x,\ Y(z, y) \supset [(X(z) \supset Y(z+1, y+1)) \wedge (\neg X(z) \supset Y(z+1, y))]. \end{aligned} \tag{3}$$

Using *NUMONES* (for existence) and $\mathbf{\Sigma}_0^B$-**IND** (for uniqueness), it is straightforward that $\mathbf{VTC}^0(numones)$ proves $\forall x \forall X \exists ! y\, numones(x, X) = y$. $\square$

Let $\mathbf{VTC}^0(\mathit{numones})$ denote the extension of $\mathbf{VTC}^0$ obtained by adding $\mathbf{\Sigma}_1^B$-defining axiom (3) for *numones*. It follows from the above proposition that $\mathbf{VTC}^0(\mathit{numones})$ is a conservative extension of $\mathbf{VTC}^0$, because any model of $\mathbf{VTC}^0$ can be expanded to a model of $\mathbf{VTC}^0(\mathit{numones})$.

Lemma 1.3. *The theory* $\mathbf{VTC}^0(\mathit{numones})$ *can be equivalently axiomatized by* 2-**BASIC**, $\mathbf{\Sigma}_1^B(\mathit{numones})$-**COMP** *and the following axioms:*

$$\mathit{numones}(X,0) = 0 \tag{4}$$

$$X(z) \supset \mathit{numones}(X, z+1) = \mathit{numones}(X,z) + 1 \tag{5}$$

$$\neg X(z) \supset \mathit{numones}(X, z+1) = \mathit{numones}(X,z). \tag{6}$$

Proof. For one direction, we prove that $\mathbf{VTC}^0(\mathit{numones})$ proves $\mathbf{\Sigma}_1^B(\mathit{numones})$-**COMP** and the axioms (4), (5) and (6). It is easy to see that (4), (5) and (6) are provable in $\mathbf{VTC}^0(\mathit{numones})$. For $\mathbf{\Sigma}_1^B(\mathit{numones})$-**COMP**, it is shown [12] that $\overline{\mathbf{VTC}}^0$ is conservative over $\mathbf{VTC}^0(\mathit{Row}, \mathit{numones})$, and that $\overline{\mathbf{VTC}}^0$ proves $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FTC}^0})$-**COMP**. Since $\mathbf{VTC}^0(\mathit{Row}, \mathit{numones})$ is a conservative extension of $\mathbf{VTC}^0(\mathit{numones})$, it follows that the latter proves $\mathbf{\Sigma}_1^B(\mathit{numones})$-**COMP**.

For the other direction, it suffices to show that $\mathbf{VTC}^0(\mathit{numones})$ proves *NUMONES*. The string Y in (2) is proved to exist in $\mathbf{VTC}^0(\mathit{numones})$ by $\mathbf{\Sigma}_0^B(\mathit{numones})$-**COMP**:

$$\forall z \le |X| \forall y \le |X|\, Y(z,y) \leftrightarrow \mathit{numones}(z, X) = y$$

The correctness of Y is proved in $\mathbf{VTC}^0(\mathit{numones})$ using the properties of *numones* (4), (5) and (6). □

2 Propositional Translation

We will now discuss the connection between our theory $\mathbf{VTC}^0(\mathit{numones})$ and the *propositional threshold logic* **PTK** [3] (a member of the so–called $\mathbf{TC}^0$–**Frege** family [1]). We will show that each $\mathbf{\Sigma}_0^B(\mathit{numones})$ theorem of $\overline{\mathbf{VTC}}^0$ translates into a family of tautologies which have short **PTK** proofs, where the depths of the propositional fomulas are bounded by some constant. This is done by translating the $\mathbf{VTC}^0(\mathit{numones})$-proofs into **PTK**-proofs. First, we recall the definition of **PTK**.

2.1 PTK

The system **PTK** is introduced in [3]. It extends **Frege** systems by the *threshold connectives* Th_k^n $(1 \le k \le n)$, where $\mathsf{Th}_k^n(\phi_1, \ldots, \phi_n)$ is true ($\top$) if and only if there are at least k true ϕ_i's. In **PTK** $\wedge$ and $\vee$ become superficial, but we will still use them for readability. We will drop n, since the number of the arguments $\phi_1, \ldots, \phi_n$ will be clear from the context. Also, $\mathsf{Th}_k(\phi_1, \ldots, \phi_n)$ is syntactically $\bot$ if $k > n$, and $\top$ if $k = 0$. Axioms of **PTK** include

$$\longrightarrow \top \qquad\qquad \bot \longrightarrow \qquad\qquad \varphi \longrightarrow \varphi \tag{7}$$

for any **PTK** formula φ. The rules of **PTK** consist of the *structural*, *logical* and *cut* rules. The structural rules include the "standard" rules: weakening, contraction and exchange rules. The logical rules are presented in Figure 2.[1]

Buss and Clote [3] also introduce the system **PTK**$'$, which is p–equivalent to **PTK**. The rules of **PTK**$'$ are the same as that of **PTK**, except for the Th_k introduction rules, which are given in Figure 3.

[1] We slightly modify the rules Th_k-**right** and Th_k-**left** of [3]. It is easy to see that the modified system p-simulates the original one presented in [3]. The reverse direction can be derived from the proof of Theorem 2.2.

$$\frac{\Lambda \longrightarrow \varphi, \Gamma}{\neg\varphi, \Lambda \longrightarrow \Gamma}\ \neg\textbf{-left} \qquad \frac{\varphi, \Lambda \longrightarrow \Gamma}{\Lambda \longrightarrow \neg\varphi, \Gamma}\ \neg\textbf{-right}$$

$$\frac{\varphi_1, \ldots, \varphi_n, \Lambda \longrightarrow \Gamma}{\mathsf{Th}_n(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \wedge\textbf{-left} \qquad \frac{\Lambda \longrightarrow \varphi_1, \Gamma \quad \ldots \quad \Lambda \longrightarrow \varphi_n, \Gamma}{\Lambda \longrightarrow \mathsf{Th}_n(\varphi_1, \ldots, \varphi_n), \Gamma}\ \wedge\textbf{-right}$$

$$\frac{\varphi_1, \Lambda \longrightarrow \Gamma \quad \ldots \quad \varphi_n, \Lambda \longrightarrow \Gamma}{\mathsf{Th}_1(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \vee\textbf{-left} \qquad \frac{\Lambda \longrightarrow \varphi_1, \ldots, \varphi_n, \Gamma}{\Lambda \longrightarrow \mathsf{Th}_1(\varphi_1, \ldots, \varphi_n), \Gamma}\ \vee\textbf{-right}$$

$$\frac{\mathsf{Th}_k(\varphi_2, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma \qquad \varphi_1, \mathsf{Th}_{k-1}(\varphi_2, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}{\mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \mathsf{Th}_k\textbf{-left}$$

$$\frac{\Lambda \longrightarrow \varphi_1, \mathsf{Th}_k(\varphi_2, \ldots, \varphi_n), \Gamma \qquad \Lambda \longrightarrow \mathsf{Th}_{k-1}(\varphi_2, \ldots, \varphi_n), \Gamma}{\Lambda \longrightarrow \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Gamma}\ \mathsf{Th}_k\textbf{-right}$$

Figure 2: Introduction rules of **PTK**

$$\frac{\mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}{\mathsf{Th}_{k+\ell}(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \mathsf{Th}'_k\textbf{-left } 1 \qquad \frac{\mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}{\mathsf{Th}_{k+\ell}(\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_\ell), \Lambda \longrightarrow \Gamma}\ \mathsf{Th}'_k\textbf{-left } 2$$

$$\frac{\neg\psi_1, \ldots, \neg\psi_m, \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda \longrightarrow \Gamma}{\neg\psi_1, \ldots, \neg\psi_m, \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m), \Lambda \longrightarrow \Gamma}\ \mathsf{Th}'_k\textbf{-left } 3$$

$$\frac{\Lambda \longrightarrow \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Gamma}{\Lambda \longrightarrow \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m), \Gamma}\ \mathsf{Th}'_k\textbf{-right } 1$$

$$\frac{\Lambda \longrightarrow \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Gamma \qquad \Lambda \longrightarrow \mathsf{Th}_\ell(\psi_1, \ldots, \psi_m), \Gamma}{\Lambda \longrightarrow \mathsf{Th}_{k+\ell}(\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m), \Gamma}\ \mathsf{Th}'_k\textbf{-right } 2$$

Figure 3: Th_k–introduction rules of $\mathbf{PTK'}$

Both **PTK** and $\mathbf{PTK'}$ are sound and complete, but **PTK** enjoys cut elimination while this is unknown for $\mathbf{PTK'}$. Also $\mathbf{PTK'}$ seems at first sight more powerful than **PTK**. Nevertheless, they are p–equivalent.[2] We show in Theorem 2.2 that **PTK** p-simulates $\mathbf{PTK'}$. This also verifies that **PTK** is *p*-equivalent to its original form defined in [3]. The proof is interesting, but it is independed of the rest of the paper.

First we formally define the size and depth of a propositional threshold formula.

Definition 2.1. *If φ is $\perp$, $\top$ or a propositional variable, then size$(\varphi) = 1$, depth$(\varphi) = 0$. If $\varphi \equiv \neg\psi$ then size$(\varphi) = 1 +$ size(ψ) and depth$(\varphi) = 1 +$ depth(ψ). If $\varphi \equiv \mathsf{Th}_k(\varphi_1, \ldots, \varphi_n)$ $(1 \leq k \leq n)$, then size$(\varphi) = n + k + 1 + \sum$ size(φ_i), and depth$(\varphi) = 1 + max\{$depth$(\varphi_i)\}$. The size (resp. depth) of a sequent is the total size (resp. maximal depth) of the formulas in the sequent.*

Theorem 2.2. **PTK** *p-simulates* $\mathbf{PTK'}$.

Proof. We will show that the rules of $\mathbf{PTK'}$ can be derived in **PTK** using polynomial size proofs. First, the rules Th'_k**-left** 1, Th'_k**-left** 2 and Th'_k**-right** 1 can be derived using the following tautologies

$$\mathsf{Th}_k(p_1, \ldots, p_m) \longrightarrow \mathsf{Th}_{k'}(p_1, \ldots, p_{m'}), \tag{8}$$

[2]And both are *p*-equivalent to *FC* [8], where *FC* is the system that extends **Frege** proof systems by new connectives $C_{n,k}$ which count exactly the number of true arguments. A drawback of these connectives is that they are not distributive over either $\wedge$ or $\vee$.

where $1 \leq k \leq m \leq n$, $1 \leq k' \leq m' \leq n$, $k' \leq k$ and $m' - k' \geq m - k$. In the next lemma, we will show that the above tautologies can be derived in **PTK** by short proofs. This is shown by *dynamic programming technique*, i.e., we will show that there are polynomial size **PTK**–proofs that contain all such tautologies. Same technique can be used to derive the other rules of **PTK**$'$. □

Lemma 2.3. *There is a polynomial* $\mathbf{p}(n)$ *so that for* $n \in \mathbb{N}, n \geq 1$, *there is a* **PTK** *proof* π_n *of size* $\leq \mathbf{p}(n)$ *that contains all sequents of the form* (8).

(Observe that we state the lemma using propositional variables p_i's, but it is straightforward to replace them with formulas φ_i's.)

The Lemma is proved by induction on n. First we consider some simple cases.

Lemma 2.4. *The following tautologies have short proofs in* **PTK**:
a) $\mathsf{Th}_{k+1}(p_1, \ldots, p_{m'}) \longrightarrow \mathsf{Th}_1(p_1, \ldots, p_m)$, *for* $m \geq 1, m' \leq m + k$.
b) $\mathsf{Th}_n(p_1, \ldots, p_n) \longrightarrow \mathsf{Th}_k(p_1, \ldots, p_m)$, *for* $1 \leq k \leq m \leq n$.

Proof. a) Consider the case where $m' = m + k$. We will prove

$$\mathsf{Th}_{k+1}(p_1, \ldots, p_{m+k}) \longrightarrow \mathsf{Th}_1(p_1, \ldots, p_m)$$

in **PTK**. (The case where $m' < m + k$ is similar.) Reasoning backward.

1.	$\mathsf{Th}_{k+1}(p_1, \ldots, p_{m+k}) \longrightarrow \mathsf{Th}_1(p_1, \ldots, p_m)$	
2.	$\mathsf{Th}_{k+1}(p_1, \ldots, p_{m+k}) \longrightarrow p_1, \ldots, p_m$	from 1, using $\vee$-**right**
3 (i).	$\mathsf{Th}_{k+1}(p_2, \ldots, p_{m+k}) \longrightarrow p_1, \ldots, p_m$	from 2, using Th_{k+1}-**left**
3 (ii).	$p_1, \mathsf{Th}_k(p_2, \ldots, p_{m+k}) \longrightarrow p_1, \ldots, p_m$	from 2, using Th_{k+1}-**left**

The sequent 3.(ii) comes from axioms using structural rules only. Repeatedly applying the Th_{k+1}-**left** rule on 3.(i) we obtain the following sequent

$$\mathsf{Th}_{k+1}(p_m, \ldots, p_{m+k}) \longrightarrow p_1, \ldots, p_m.$$

This is derivable using $\wedge$-**left** rule and the structural rules.

Part b) is proved similarly. □

Proof of Lemma 2.3. Note that the proof of Lemma 2.4 already shows that the sequents there have **PTK** proofs of sizes bounded by some polynomial in n. The current Lemma is proved by induction on n. The base case is straightforward. For the induction step, We will construct the proof π_n from π_{n-1}. It will be evident from the construction that in general, the size of π_n is bounded by some polynomial $\mathbf{p}(n)$.

Consider the sequent (8) for the case when $k' = 1$. Lemma 2.4(a) shows that the sequent (8) has polynomial-size **PTK** proof, we can simply add it to π_{n-1}. Similarly, suppose $k = n$, then $m = n$, and the sequent (8) also has polynomial-size **PTK** proof according to Lemma 2.4(b).

Now suppose that $1 < k' \leq k < n$. If both $m' < n, m < n$, then the sequent (8) already exists in π_{n-1}. Suppose that either $m = n$ or $m' = n$. Reasoning backward. Since both $k' < n$ and $k < n$, we can apply the appropriate rule (either Th_k-**left** or Th_k-**right**) to obtain the sequent (8) from the existing sequents in π_{n-1}. □

We are interested in subsystem of **PTK** where the cut formulas are restricted to certain formulas.

Definition 2.5. *For each* $d \in \mathbb{N}, d \geq 0$, *a* d-**PTK** *proof is a* **PTK** *proof where every cut formula has depth* $\leq d$.

The standard completeness argument for **PTK** actually shows that 0-**PTK** is complete for threshold formulas.

The technique from [7] shows that treelike **PTK** p-simulate daglike **PTK**, with a constant increase in the depth.

Theorem 2.6. *For $d \geq 1$, treelike $(d+3)$-**PTK** p–simulates daglike d-**PTK**.*

Proof. The idea is to avoid reusing sequents by converting each initial segment of a proof into a single formula (of higher depth) which carries the same information. In particular, for each sequent

$$\mathcal{S} = \varphi_1, \ldots, \varphi_m \longrightarrow \psi_1, \ldots, \psi_n$$

in a **PTK** proof, let $\hat{\mathcal{S}}$ be the formula that expresses the meaning of the sequent $\mathcal{S}$:

$$\hat{\mathcal{S}} \equiv \mathsf{Th}_1(\neg\varphi_1, \ldots, \neg\varphi_m, \psi_1, \ldots, \psi_n).$$

We will prove that for a **PTK** proof π,

$$\pi = \mathcal{S}_1, \ldots, \mathcal{S}_n,$$

there is a treelike **PTK** proof of

$$\longrightarrow \mathsf{Th}_n(\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n).$$

The proof of this claim is straightforward by induction, using the next lemma. Note that the above formula has depth $d+3$ if π is of depth d.

Now, we can derive $\mathcal{S}_n$ from $\longrightarrow \hat{\mathcal{S}}_n$, which is derived from $\longrightarrow \mathsf{Th}_n(\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n)$, by short treelike **PTK** proof. □

Lemma 2.7. *Let Λ be a list of m formulas ($m \geq 0$), and $\dfrac{\mathcal{S}_1 \ldots \mathcal{S}_n}{\mathcal{S}}$ be any instance of **PTK** rules ($n \geq 1$). Then the following tautology has short treelike **PTK** proof:*
a) $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n \longrightarrow \hat{\mathcal{S}}$,
b) $\mathsf{Th}_{n+m}(\Lambda, \hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n) \longrightarrow \mathsf{Th}_{n+m+1}(\Lambda, \hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n, \hat{\mathcal{S}})$.

Proof. Since Λ contains exactly m formulas, it is straightforward to derive the sequent in part b) from the sequent in part a) using the $\wedge$-**right** and $\wedge$-**left** rules.

For part a), we need to check all the rules of **PTK**. Most cases are considered in [7]. Here we consider a rule in **PTK**, i.e., the Th_k-**right** rule. Suppose that

$$\frac{\mathcal{S}_1 \qquad \mathcal{S}_2}{\mathcal{S}} = \frac{\mathsf{Th}_k(\varphi_2, \ldots, \varphi_n), \Delta \longrightarrow \Gamma \qquad \varphi_1, \mathsf{Th}_{k-1}(\varphi_2, \ldots, \varphi_n), \Delta \longrightarrow \Gamma}{\mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Delta \longrightarrow \Gamma}$$

By definition,

$$\hat{\mathcal{S}}_1 \equiv \mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_2, \ldots, \varphi_n), \Lambda)$$
$$\hat{\mathcal{S}}_2 \equiv \mathsf{Th}_1(\neg\varphi_1, \neg\mathsf{Th}_{k-1}(\varphi_2, \ldots, \varphi_n), \Lambda)$$
$$\hat{\mathcal{S}} \equiv \mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda)$$

Reasoning backward (ignore sequents which can be obtained from axioms just by structural rules).

$$\begin{array}{lr}
1.\mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda),\mathsf{Th}_1(\neg\varphi_1,\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\Lambda),\longrightarrow \mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\Lambda) & \\
2.\mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda),\mathsf{Th}_1(\neg\varphi_1,\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\Lambda),\longrightarrow \neg\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\Lambda & (1.\ \vee\textbf{-right}) \\
3.\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\mathsf{Th}_1(\neg\mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda),\mathsf{Th}_1(\neg\varphi_1,\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\Lambda),\longrightarrow \Lambda & (2.\ \neg\textbf{-right}) \\
4.\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\neg\mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\mathsf{Th}_1(\neg\varphi_1,\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\Lambda),\longrightarrow \Lambda & (3.\ \vee\textbf{-left}) \\
5.\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\mathsf{Th}_1(\neg\varphi_1,\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\Lambda),\longrightarrow \mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda & (4.\ \neg\textbf{-left}) \\
6.\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\neg\varphi_1,\longrightarrow \mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda & (5.\ \vee\textbf{-left}) \\
7.\mathsf{Th}_k(\varphi_1,\ldots,\varphi_n),\neg\mathsf{Th}_{k-1}(\varphi_2,\ldots,\varphi_n),\longrightarrow \mathsf{Th}_k(\varphi_2,\ldots,\varphi_n),\Lambda & (5.\ \vee\textbf{-left})
\end{array}$$

Now, 6. and 7. are easily obtained from the $\neg$-**left** and Th_k-**left** rules. □

Corollary 2.8. *Treelike* **PTK** *p–simulates daglike* **PTK**.

2.2 Translating $\Sigma_0^B(numones)$ Formulas

First we recall the translation of a $\Sigma_0^B(\mathcal{L}_A^2)$-formula $\varphi(\vec{X})$ [4]: $\varphi(\vec{X})$ translates into a family of propositional formulas $\{\varphi(\vec{X})[\vec{n}] : \vec{n} \geq 0\}$, so that for all $\vec{n} \in \mathbb{N}$, $\varphi(\vec{X})[\vec{n}]$ is valid iff $\underline{\mathbb{N}}_2 \models \forall\vec{X}(|\vec{X}| = \vec{n} \supset \varphi(X))$. For each string variable X we use the free propositional variables $p_0^X, p_1^X, \ldots$ to represent the bits of X. Let $val(t)$ be the numerical value of a closed term t. The translation is defined inductively. For the base case, for example,

$$(s = t)[n] =_{\rm def} \begin{cases} \top & \text{if } val(s(\underline{n})) = val(t(\underline{n})) \\ \bot & \text{otherwise} \end{cases}$$

and

$$X(t)[n] =_{\rm def} \begin{cases} p_j^X & \text{if } j < n-1 \\ \top & \text{if } j = n-1 \\ \bot & \text{if } j > n-1 \end{cases}$$

where j is the value of t when $|X| = n$. For the induction step we apply the obvious translation of the (first-order) connectives and number quantifiers.

Now consider an atomic formula $\varphi(\vec{X}) \equiv s = t$ that contains *numones*. Translation of this function requires the threshold connectives in **PTK**. Let $numones(X_i, t_i)$ (for $i = 0, \ldots, \ell$) be all occurrences of *numones* in φ (possibly with repetitions). Fix $\vec{n}$. Let S be the set of all tuples $(k_0, \ldots, k_\ell)$ (where $k_i \leq \min\{|X_i|, t_i\}$) that make $\varphi(\vec{X})$ true when $|\vec{X}| = \vec{n}$ and $numones(X_j, t_j) = k_j$. Then,

$$(s = t)[\vec{n}] =_{\rm def} \bigvee_{\vec{k} \in S} \bigwedge_{j=0}^{\ell} [\mathsf{Th}_{k_j}(\vec{p}^{Y_j}) \wedge \neg\mathsf{Th}_{k_j+1}(\vec{p}^{Y_j})],$$

where $\mathsf{Th}_0(\ldots) =_{\rm syn} \top$ and $\mathsf{Th}_k(p_1, \ldots, p_n) =_{\rm syn} \bot$ if $k > n$. Similar for an atomic formula $s \leq t$. We translate an atomic formula of the form $X(t)$ by noting that it is equivalent to $\exists z < |X|(z = t \wedge X(z))$. Also, to translate any atomic formula of the form $\alpha = \beta$, we translate the LHS of **SE** (Figure 1).

In general, $\varphi(\vec{x}, \vec{X})[\vec{m}; \vec{n}]$ is defined to be $\varphi(\underline{\vec{m}}, \vec{X})[\vec{n}]$. The following lemma is straightforward.

Lemma 2.9. *For each simple* $\Sigma_0^B(numones)$ *formula* $\varphi(\vec{x}, \vec{Y})$, *there is a constant* d *and a polynomial* $\mathbf{p}$ *so that for all sequences* $\vec{m}, \vec{n}$, *the propositional formula* $\varphi(\vec{x}, \vec{Y})[\vec{m}; \vec{n}]$ *has depth* d *and size bounded by* $\mathbf{p}(\vec{n})$.

The remaining of this section is devoted to the proof of the following theorem.

Theorem 2.10 (Propositional Translation Theorem for $\mathbf{VTC}^0$). *For each $\mathbf{\Sigma}_0^B(numones)$ theorem $\varphi(\vec{X})$ of $\mathbf{VTC}^0(numones)$, there is a constant d and a polynomial $\mathbf{p}$ such that the family of tautologies $\{\varphi(\vec{X})[\vec{n}] : \vec{n} \in \mathbb{N}\}$ has d-**PTK** proofs of sizes bounded by $\mathbf{p}(\vec{n})$.*

First we prove:

Lemma 2.11. *Let $\vec{p}$ denote $p_0, \ldots, p_{m-1}$. The following sequents have polynomial-size* **cut**-*free* **PTK** *proofs:*

$$p_m \longrightarrow \bigvee_{0\le k\le m} [\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m)] \tag{9}$$

$$\longrightarrow \bigvee_{0\le k\le m} [\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p})] \tag{10}$$

Proof. For the sequent (9), we need to derive

$$p_m \longrightarrow \mathsf{Th}_0(\vec{p}) \wedge \neg\mathsf{Th}_2(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m).$$

Reasoning backward (ignore sequents that can be obtained by structural rules from the axioms).

1. $p_m \longrightarrow \mathsf{Th}_0(\vec{p}) \wedge \neg\mathsf{Th}_2(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$
2. $p_m \longrightarrow \neg\mathsf{Th}_2(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$ — 1. $\wedge$-**right**
3. $p_m, \mathsf{Th}_2(\vec{p}, p_m) \longrightarrow \mathsf{Th}_1(\vec{p}) \wedge \neg\mathsf{Th}_3(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$ — 2. $\neg$-**right**
4. $p_m, \mathsf{Th}_1(\vec{p}) \longrightarrow \mathsf{Th}_1(\vec{p}) \wedge \neg\mathsf{Th}_3(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$ — 3. Th_2-**left**
5. $p_m, \mathsf{Th}_1(\vec{p}) \longrightarrow \neg\mathsf{Th}_3(\vec{p}, p_m), \ldots, \mathsf{Th}_m(\vec{p}) \wedge \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$ — 4. $\wedge$-**right**

...

$3m+2$. $p_m, \mathsf{Th}_1(\vec{p}), \ldots, \mathsf{Th}_m(\vec{p}) \longrightarrow \neg\mathsf{Th}_{m+2}(\vec{p}, p_m)$

The last sequent is derived from the axiom $\bot \longrightarrow$ (note that $\mathsf{Th}_{m+2}(\vec{p}, p_m)$ is syntactically $\bot$).

To prove the sequent (10), we reasoning backward as follows:

1. $\longrightarrow \bigvee_{0\le k\le m} [\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p})]$
2. $\longrightarrow \neg\mathsf{Th}_1(\vec{p}), \mathsf{Th}_1(\vec{p}) \wedge \neg\mathsf{Th}_2(\vec{p}), \ldots, \mathsf{Th}_{m-1}(\vec{p}) \wedge \neg\mathsf{Th}_m(\vec{p}), \mathsf{Th}_m(\vec{p})$ — 1. $\vee$-**right**
3. $\mathsf{Th}_1(\vec{p}) \longrightarrow \mathsf{Th}_1(\vec{p}) \wedge \neg\mathsf{Th}_2(\vec{p}), \ldots, \mathsf{Th}_{m-1}(\vec{p}) \wedge \neg\mathsf{Th}_m(\vec{p}), \mathsf{Th}_m(\vec{p})$ — 2. $\neg$-**right**
4. $\mathsf{Th}_1(\vec{p}) \longrightarrow \neg\mathsf{Th}_2(\vec{p}), \ldots, \mathsf{Th}_{m-1}(\vec{p}) \wedge \neg\mathsf{Th}_m(\vec{p}), \mathsf{Th}_m(\vec{p})$ — 3. $\wedge$-**right**

...

$2m$. $\mathsf{Th}_1(\vec{p}), \ldots, \mathsf{Th}_{m-1}(\vec{p}) \longrightarrow \neg\mathsf{Th}_m(\vec{p}), \mathsf{Th}_m(\vec{p})$

The last sequent is obtained from the axiom $\mathsf{Th}_m(\vec{p}) \longrightarrow \mathsf{Th}_m(\vec{p})$. □

Lemma 2.12. *The translations of the axioms (4), (5) and (6) have polynomial-size bounded-depth* **PTK** *proofs.*

Proof. Consider the axiom (5)

$$X(z) \supset numones(X, z+1) = numones(X, z) + 1.$$

First, we translate $numones(X, z+1) = numones(X, z) + 1$. Here, $t_0 = z+1, t_1 = z$. The set S is $S = \{(k_1+1, k_1) \mid k_1 \le m\}$. This atomic formula translates into A, where (let $\vec{p}$ denote $p_0, \ldots, p_{m-1}$):

$$A \equiv \bigvee_{0\le k\le m} [(\mathsf{Th}_{k+1}(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m)) \wedge (\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p}))]$$

Now (5) translates into $\neg p_m \vee A$. We show how to derive $p_m \longrightarrow A$ in **PTK**.

For $k \leq m$ the sequent

$$p_m, \mathsf{Th}_k(\vec{p}) \longrightarrow \mathsf{Th}_{k+1}(\vec{p}, p_m) \tag{a}$$

is derivable in **PTK** (using Th_{k+1}-**right** rule). Therefore we also derive

$$p_m, \neg\mathsf{Th}_{k+2}(\vec{p}, p_m) \longrightarrow \neg\mathsf{Th}_{k+1}(\vec{p}) \tag{b}$$

From (a), (b) we derive (for $k \leq m$)

$$p_m, \mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m) \longrightarrow (\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p})) \wedge (\mathsf{Th}_{k+1}(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m)).$$

By weakening we get

$$p_m, \mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m) \longrightarrow A$$

Combine these sequents for $k \leq m$, using $\vee$-**left**, we get

$$p_m, \bigvee_{k \leq m} [\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+2}(\vec{p}, p_m)] \longrightarrow A$$

We obtain $p_m \longrightarrow A$ from the last sequent and (9).

The proof for the axiom (6) is slightly different. It translates into

$$\longrightarrow p_m \vee \bigvee_{k \leq m} [(\mathsf{Th}_k(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{k+1}(\vec{p}, p_m)) \wedge (\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p}))]$$

For each $k \leq m$, it is straightforward to derive the following sequents:

$$\mathsf{Th}_k(\vec{p}) \longrightarrow p_m, \mathsf{Th}_k(\vec{p}, p_m) \qquad \neg\mathsf{Th}_{k+1}(\vec{p}) \longrightarrow p_m, \neg\mathsf{Th}_{k+1}(\vec{p}, p_m)$$

Hence, we can derive

$$\mathsf{Th}_k(\vec{p}) \wedge \mathsf{Th}_{k+1}(\vec{p}) \longrightarrow p_m, (\mathsf{Th}_k(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{k+1}(\vec{p}, p_m)) \wedge (\mathsf{Th}_k(\vec{p}) \wedge \neg\mathsf{Th}_{k+1}(\vec{p}))$$

and thus

$$\mathsf{Th}_k(\vec{p}) \wedge \mathsf{Th}_{k+1}(\vec{p}) \longrightarrow p_m, \bigvee_{i \leq m} [(\mathsf{Th}_i(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{i+1}(\vec{p}, p_m)) \wedge (\mathsf{Th}_i(\vec{p}) \wedge \neg\mathsf{Th}_{i+1}(\vec{p}))]$$

for all $k \leq m$.

As a result, we can derive

$$\bigvee_{k \leq m} [\mathsf{Th}_k(\vec{p}) \wedge \mathsf{Th}_{k+1}(\vec{p})] \longrightarrow p_m, \bigvee_{i \leq m} [(\mathsf{Th}_i(\vec{p}, p_m) \wedge \neg\mathsf{Th}_{i+1}(\vec{p}, p_m)) \wedge (\mathsf{Th}_i(\vec{p}) \wedge \neg\mathsf{Th}_{i+1}(\vec{p}))].$$

From the last sequent and (10) we get the desired sequent. □

It is useful to recall the notion of the Free Variable Normal Form for **LK** proofs.

Definition 2.13 (Free Variable Normal Form). *Let π be an* $\mathbf{LK}^2$ *proof of a formula φ. A free variable in φ is called a parameter variable of π. We say π is in* free variable normal form *if (i) no parameter variable is eliminated from a sequent by any rule; (ii) each nonparameter free variable in π is used exactly once as an eigenvariable; and (iii) each nonparameter free variable does not occur below the sequent where it is used as the eigenvariable.*

Proof of the Translation Theorem for $\mathbf{VTC}^0$. Let $\varphi(\vec{X})$ be a $\mathbf{\Sigma}_0^B(numones)$ theorem of $\mathbf{VTC}^0(numones)$. Then there is an anchored (aka free cut-free) $\mathbf{LK}^2$ proof π of φ which is in Free Variable Normal Form, and which has (by Lemma 1.3) nonlogical axioms from 2-**BASIC**, $\mathbf{\Sigma}_0^B(numones)$-**COMP** and (4), (5) and (6). Because π is in Free Variable Normal Form, if a nonparameter string variable γ is used as the eigenvariable in

$$\frac{\forall x < t(|\vec{\alpha}|),\ \gamma(x) \leftrightarrow \psi(x, \vec{\alpha}), \Gamma \longrightarrow \Delta}{\exists Z \forall x < t(|\vec{\alpha}|),\ Z(x) \leftrightarrow \psi(x, \vec{\alpha}), \Gamma \longrightarrow \Delta}$$

then we can associate γ with the pair $\langle t_\gamma, \psi_\gamma \rangle = \langle t(|\vec{\alpha}|), \psi(x, \vec{\alpha}) \rangle$, which may contain other nonparameter free variables.

If α is a parameter variable, we translate any atomic formula $\alpha(t)$ as before, using the variables $p_0^\alpha, p_1^\alpha, \ldots$. The translation for nonparameter free variables is more complicated, and is explained below.

For each nonparameter free variable γ, we will not use the propositional variable $p_0^\gamma, \ldots$ as for the parameter variables. Instead, we translate each bit $\gamma(z)$ by translating the associated formula ψ_γ. Since ψ_γ may contain other nonparameter variable, there is the danger of going into circularity. This is not a problem, because the dependence relation between nonparameter free string variables, defined below, is an accyclic relation.

Notation We say that γ *depends on* β if β occurs in t_γ or ψ_γ, or if there is another nonparameter free variable γ' such that γ depends on γ', and γ' depends on β. A sequent $\mathcal{S}$ is said to *depend on* γ if $\mathcal{S}$ contains γ, or $\mathcal{S}$ contains some variable β that depends on γ.

For a nonparameter variable γ translating $\gamma(s)$ requires the lengths of γ, other nonparameter free variables that it depends on, and the parameter variables. We define the translation of $\gamma(s)$ inductively. For the base case, γ does not depend on any other nonparameter variable, then we simply translate $\gamma(s)$ by translating $\psi_\gamma(s)$. For the induction step, let $\vec{\alpha}$ be all parameter variables, and $\beta_1, \ldots, \beta_\ell$ be all the nonparameter free variables that γ depends on, then for $n_\gamma \leq t_\gamma(n_{\vec{\beta}})$, define

$$\gamma(s)[n_\gamma, n_{\vec{\alpha}}, n_{\vec{\beta}}] =_{\text{def}} \begin{cases} \psi_\gamma(s)[n_{\vec{\alpha}}, n_{\vec{\beta}}] & \text{if } i < n_\gamma - 1 \\ \top & \text{if } i = n_\gamma - 1 \\ \bot & \text{if } i \geq n_\gamma \end{cases}$$

where $i = val(s)$. Here $\psi_\gamma(s)[n_{\vec{\alpha}}, n_{\vec{\beta}}]$ is the translation of $\psi_\gamma(s)$, which is already defined by our induction hypothesis.

A $\mathbf{\Sigma}_0^B(numones)$ formula in π is translated by translating its atomic subformulas. Consider now an instance of $\mathbf{\Sigma}_0^B(numones)$-**COMP** in π:

$$\exists X \leq t \forall z < t,\ X(z) \leftrightarrow \psi(z)$$

Let $\vec{\beta}$ be all nonparameter variables that this instance depends on, and $\vec{\alpha}$ be all parameter variables in π. It is translated into

$$\bigvee_{n=0}^{v} A[n, n_{\vec{\alpha}}, n_{\vec{\beta}}] \tag{14}$$

where $v = val(t)$, and

$$A[0, n_{\vec{\alpha}}, n_{\vec{\beta}}] \equiv \bigwedge_{i=0}^{v-1} \neg\psi(z)[i; n_{\vec{\alpha}}, n_{\vec{\beta}}]$$

$$A[n, n_{\vec{\alpha}}, n_{\vec{\beta}}] \equiv \psi(z)[n-1; n_{\vec{\alpha}}, n_{\vec{\beta}}] \wedge \bigwedge_{i=n}^{v-1} \neg\psi(z)[i; n_{\vec{\alpha}}, n_{\vec{\beta}}] \qquad \text{for } n \geq 1$$

Let $\mathcal{S}(\vec{b}, \vec{\alpha})$ be a sequent in π with all free variables indicated. Let $\vec{\beta}$ be all nonparameter free variables that some nonparameter free variable in $\mathcal{S}$ depends on. We prove by induction on the depth of $\mathcal{S}$ in π that the translation sequent $\mathcal{S}[\vec{m}; n_{\vec{\alpha}}, n_{\vec{\beta}}]$ has a bounded depth **PTK** proof of size bounded by a polynomial in $\vec{m}, n_{\vec{\alpha}}, n_{\vec{\beta}}$.

For the base case, $\mathcal{S}$ is an axiom of $\mathbf{LK}^2$-$\mathbf{VTC}^0(numones)$. It is easy to check that the axioms of 2-**BASIC** translates into tautologies having short, **cut**-free proofs in **PTK**. (Although for **L1** and **L2** the situation is more complicated than in the translation of $\mathbf{V}^0$ proofs into bounded depth **Frege** proofs.) It is also easy to show that the translation (14) of the $\mathbf{\Sigma}_0^B(numones)$-**COMP** axioms has short **PTK** proofs. For the induction step, we consider the interesting case of the rule **string** $\exists$ **left**. Suppose that

$$\frac{\mathcal{S}_1}{\mathcal{S}} = \frac{\forall x < t(|\vec{\alpha}|),\ \gamma(x) \leftrightarrow \psi(x, \vec{\alpha}), \Gamma \longrightarrow \Delta}{\exists Z \forall x < t(|\vec{\alpha}|),\ Z(x) \leftrightarrow \psi(x, \vec{\alpha}), \Gamma \longrightarrow \Delta}$$

Notice that γ depends only on variables in $\vec{\alpha}$ and $\vec{\beta}$. For each $n_\gamma \leq v$, where $v = val(t(n_{\vec{\alpha}}))$, $\mathcal{S}_1$ translates into

$$\mathcal{S}_1[n_\gamma, n_{\vec{\alpha}}, n_{\vec{\beta}}] =_{\text{def}} A[n_\gamma, n_{\vec{\alpha}}, n_{\vec{\beta}}], ||\Gamma|| \longrightarrow ||\Delta||$$

where $||\Gamma||$ and $||\Delta||$ denote the translation of Γ and Δ, respectively. Now $\mathcal{S}$ translates into

$$\bigvee_{n_\gamma=0}^{v} A[n, n_{\vec{\alpha}}, n_{\vec{\beta}}], ||\Gamma|| \longrightarrow ||\Delta||$$

This is derived from the translations of $\mathcal{S}_1[n_\gamma, n_{\vec{\alpha}}, n_{\vec{\beta}}]$ by the $\vee$-**left** rule. □

3 Propositional Translation For $\mathbf{V}^0(m)$ And VACC

For $m \geq 2$, $\varphi_{MOD_m}(X, Y)$ is the formula stating that Y is the "counting modulo m" array for X:

$$\begin{aligned}\varphi_{MOD_m}(X, Y) \equiv &[\forall z \leq |X| \exists! y < m Y(z, y)] \ \wedge \ Y(0,0) \ \wedge \ \forall z < |X| \forall y < m, \\ &Y(z,y) \supset [(X(z) \supset Y(z+1, y+1 \mod m)) \wedge (\neg X(z) \supset Y(z+1, y))].\end{aligned} \tag{15}$$

Here, we identify the natural number m with the corresponding numeral $\underline{m}$. Note that $\mod m$ is not really a new function. In deed, $\phi(y \mod m)$ can be seen as an abbreviation of

$$\exists r < m, \exists q < y,\ y = qm + r \wedge \phi(r). \tag{16}$$

Thus if $\phi(y)$ is $\mathbf{\Sigma}_0^B$, then $\phi(y \mod m)$ is also $\mathbf{\Sigma}_0^B$.

Definition 3.1. *For each $m \geq 2$, let*

$$MOD_m \equiv \forall X \exists Y \varphi_{MOD_m}(X, Y) \qquad \textit{and} \qquad \mathbf{V}^0(m) = \mathbf{V}^0 \cup \{MOD_m\} \tag{17}$$

Note that the string Y in MOD_m can be bounded by $\langle |X|, m \rangle$. Also, let

$$\mathbf{VACC} = \mathbf{V}^0 \cup \{MOD_m \mid m \geq 2\}. \tag{18}$$

For each $m \geq 2, m \in \mathbb{N}$, the function $mod_m(X, z)$ can be defined as follows.

$$mod_m(X, 0) = 0 \tag{19}$$

$$X(z) \wedge mod_m(X, z) = m - 1 \supset mod_m(X, z+1) = 0 \tag{20}$$

$$X(z) \wedge mod_m(X, z) < m - 1 \supset mod_m(X, z+1) = mod_m(X, z) + 1 \tag{21}$$

$$\neg X(z) \supset mod_m(X, z+1) = mod_m(X, z) \tag{22}$$

We present the systems $\mathbf{PK}(m)$, for each $m \in \mathbb{N}, m \geq 2$. For $m = 2$, the system $\mathbf{PK}(2)$ is the same as $\mathbf{PK}\oplus$ in [9]. In general, $\mathbf{PK}(m)$ extends **Frege** proof systems by the connective M_m^k connectives, where $0 \leq k < m$. The meaning of $\mathsf{M}_m^k(\varphi_1, \dots, \varphi_n)$ is that the number of true φ_i's modulos m is exactly k. The logical rules of $\mathbf{PK}(m)$ are:

$$\frac{\Lambda \longrightarrow \varphi, \Gamma}{\neg\varphi, \Lambda \longrightarrow \Gamma}\ \neg\textbf{-left} \qquad \frac{\varphi, \Lambda \longrightarrow \Gamma}{\Lambda \longrightarrow \neg\varphi, \Gamma}\ \neg\textbf{-right}$$

$$\frac{\varphi_1, \dots, \varphi_n, \Lambda \longrightarrow \Gamma}{\wedge(\varphi_1, \dots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \wedge\textbf{-left} \qquad \frac{\Lambda \longrightarrow \varphi_1, \Gamma \quad \dots \quad \Lambda \longrightarrow \varphi_n, \Gamma}{\Lambda \longrightarrow \wedge(\varphi_1, \dots, \varphi_n), \Gamma}\ \wedge\textbf{-right}$$

$$\frac{\varphi_1, \Lambda \longrightarrow \Gamma \quad \dots \quad \varphi_n, \Lambda \longrightarrow \Gamma}{\vee(\varphi_1, \dots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \vee\textbf{-left} \qquad \frac{\Lambda \longrightarrow \varphi_1, \dots, \varphi_n, \Gamma}{\Lambda \longrightarrow \vee(\varphi_1, \dots, \varphi_n), \Gamma}\ \vee\textbf{-right}$$

$$\frac{\mathsf{M}_m^k(\varphi_2, \dots, \varphi_n), \Lambda \longrightarrow \Gamma \qquad \varphi_1, \mathsf{M}_m^{k-1}(\varphi_2, \dots, \varphi_n), \Lambda \longrightarrow \Gamma}{\mathsf{M}_m^k(\varphi_1, \dots, \varphi_n), \Lambda \longrightarrow \Gamma}\ \mathsf{M}_m^k\textbf{-left}$$

$$\frac{\Lambda \longrightarrow \varphi_1, \mathsf{M}_m^k(\varphi_2, \dots, \varphi_n), \Gamma \qquad \Lambda \longrightarrow \mathsf{M}_m^{k-1}(\varphi_2, \dots, \varphi_n), \Gamma}{\Lambda \longrightarrow \mathsf{M}_m^k(\varphi_1, \dots, \varphi_n), \Gamma}\ \mathsf{M}_m^k\textbf{-right}$$

For each $d \in \mathbb{N}$, a d-$\mathbf{PK}(m)$ proof is a $\mathbf{PK}(m)$ proof where cut formulas have depth at most d. We obtain the analog of Theorem 2.10.

Theorem 3.2. *For each $\mathbf{\Sigma}_0^B$ theorem $\varphi(\vec{X})$ of $\mathbf{V}^0(m)$, there is a constant d and a polynomial $\mathbf{p}$ such that the family of tautologies $\{\varphi(\vec{X})[\vec{n}] : \vec{n} \in \mathbb{N}\}$ has d-$\mathbf{PK}(m)$ proofs of sizes bounded by $\mathbf{p}(\vec{n})$.*

This theorem can be proved in the same way as Theorem 2.10.

Let $\varphi(\vec{x}, \vec{Y})$ be an atomic formula of the form $s = t$ or $s \leq t$, $\varphi(\vec{x}, \vec{Y})$ contains mod_m. Let $mod_m(Y_0, t_0), \dots, mod_m(Y_\ell, t_\ell)$ be all occurrences of mod_m in φ. Let S be the set of all tuples $(k_0, \dots, k_\ell)$ where $0 \leq k_i < m$ such that $\varphi(\vec{x}, \vec{Y})$ is true when $\vec{x} = \vec{m}, |\vec{Y}| = \vec{n}$, and $mod_m(Y_i, t_i) = k_i$. Let $val(t_i)[\vec{m}; \vec{n}; \vec{k}]$ denote the value of t_i when $\vec{x} = \vec{m}, |\vec{Y}| = \vec{n}$, and $mod_m(Y_i, t_i) = k_i$. Then define

$$\varphi(\vec{x}, \vec{Y})[\vec{m}; \vec{n}] \equiv \bigvee_{\vec{k} \in S} \bigwedge_{j=0}^{\ell} \mathsf{M}_m^{k_j}(\vec{p}^j), \tag{23}$$

where $\vec{p}^j$ denotes $p_0^{Y_j}, \dots, p_{val(t_j)[\vec{m};\vec{n};\vec{k}]-1}^{Y_j}$.

Proof of Theorem 3.2. Similarly to the Proof of Theorem 2.10, it suffices to show that the defining axioms of mod_m translate into tautologies with short proof in $\mathbf{PK}(m)$. We will only consider (21); the other axioms are dealt with similarly. This axiom is translated into

$$(p_m \wedge \bigvee_{k=0}^{a-1} \mathsf{M}_a^k(\vec{p})) \supset \bigvee_{k=0}^{a-1} [\mathsf{M}_a^{k+1}(\vec{p}, p_m) \wedge \mathsf{M}_a^k(\vec{p})],$$

where $\vec{p}$ is the list of $p_0, \dots, p_{m-1}$. It is easy to derive in $\mathbf{PK}(a)$ the sequent

$$p_m, \mathsf{M}_a^k(\vec{p}) \longrightarrow \mathsf{M}_a^{k+1}(\vec{p}, p_m) \wedge \mathsf{M}_a^k(\vec{p}),$$

for $0 \leq k < m$. Hence we can derive

$$p_m, \mathsf{M}_a^k(\vec{p}) \longrightarrow \bigvee_{k=0}^{a-1} [\mathsf{M}_a^{k+1}(\vec{p}, p_m) \wedge \mathsf{M}_a^k(\vec{p})],$$

for $0 \leq k < m$. From this, we obtain the desired sequent using $\vee$-**right**. $\square$

Note that the depths of the propositional formulas do not depend on a. Define **PKM** to be **PK** together with the connectives M_a^k, for $a, k \in \mathbb{N}, 0 \leq k < a, a \geq 2$, which have introduction rules as in $\mathbf{PK}(m)$.

Corollary 3.3. *For each $\mathbf{\Sigma}_0^B$ theorem $\varphi(\vec{x}, \vec{Y})$ of* **VACC***, there is a constant d and a polynomial* $\mathbf{p}$ *such that the family of tautologies $\{\varphi(\vec{x}, \vec{Y})[\vec{m}; \vec{n}]\}_{\vec{m}, \vec{n} \in \mathbb{N}}$ has d-*$\mathbf{PKM}$ *proofs of sizes bounded by $\mathbf{p}(\vec{n})$.*

4 Translation for $\mathbf{TV}^0$ and other Subsystems

It is known that each $\mathbf{\Sigma}_0^B$ theorem of $\mathbf{V}^1$ translates into a family of tautologies having polynomial size $\mathbf{G}_1^\star$ (or equivalently **eFrege**) proofs. Since $\mathbf{V}^1$ is $\mathbf{\Sigma}_1^B$ conservative over $\mathbf{TV}^0$, the result also holds for $\mathbf{TV}^0$. Here we will briefly explain a direct proof of this statement, which helps to define subsystems of $\mathbf{G}_1^\star$ that correspond naturally to subsystems of $\mathbf{TV}^0$, such as those discussed in [12].

4.1 Propostional Translation for $\mathbf{TV}^0$

Consider the $\mathbf{AC}^0$ function $Chop(x, X)$ (also $X^{<x}$) which is the initial segment of X defined by

$$|X^{<x}| \leq x \wedge \forall z < x\, X^{<x}(z) \leftrightarrow X(z)$$

In general, any $\mathbf{\Sigma}_0^B(Chop)$ formula can be transformed to a $\mathbf{\Sigma}_0^B$ formula by successivly eliminating the occurrence of *Chop* (see also [6]). Defining $\mathbf{TV}^0$ requires only $\mathbf{\Sigma}_0^B(Chop)$ formulas that are obtained from a $\mathbf{\Sigma}_0^B$ formula $\varphi(Z)$ by replacing Z everywhere by $X^{<t}$, where X does not occur in $\varphi(Z)$, and t does not contain X, Z. Therefore to transform $\varphi(X^{<t})$ into its $\mathbf{\Sigma}_0^B$ equivalence, we first replace each atomic formula $\eta(|Z|)$ of $\varphi(Z)$ by

$$\eta' \equiv (\forall x < t \neg Z(x) \wedge \eta(0)) \vee \exists z < t,\ Z(z) \wedge \forall x < t(z < x \supset \neg Z(x)) \wedge \eta(z+1)$$

Now the $\mathbf{\Sigma}_0^B$ equivalence of $\varphi(X^{<t})$ is obtained by replacing each atomic formula $Z(s)$ by $s \leq t \wedge X(s)$.[3]

We will work with the following definition of $\mathbf{TV}^0$:

Definition 4.1 ([5]). *The theory $\mathbf{TV}^0$ has vocabulary $\mathcal{L}_A^2$ and is axiomatized by $\mathbf{V}^0$ and the $\mathbf{\Sigma}_0^B$-Bit-Recursion scheme:*

$$\exists X \leq y \forall x < y\, (X(x) \leftrightarrow \varphi(x, X^{<x})) \tag{24}$$

where φ is any $\mathbf{\Sigma}_0^B$ formula that only contains X in the context $X^{<x}$, and $\varphi(x, X^{<x})$ is understood to be its $\mathbf{\Sigma}_0^B$ equivalence which is obtained as descibed above.

Recall that the $\|\varphi\|$ denotes the family of propositional formulas translated from φ.

Theorem 4.2 (Proposition Translation Theorem for $\mathbf{TV}^0$). *For each theorem φ of $\mathbf{TV}^0$, the family $\|\varphi\|$ has polynomial size $\mathbf{G}_1^\star$ proofs.*

Proof. We will prove the Theorem for a $\mathbf{\Sigma}_0^B$ theorem of $\mathbf{TV}^0$. It is easy to extend the proof to consider other theorems of $\mathbf{TV}^0$. Consider an anchored $\mathbf{LK}^2$-$\mathbf{TV}^0$ proof in free variable normal form π of φ. We may also assume that the contraction rule is not used for $\mathbf{\Sigma}_1^B$ formulas. As before, a nonparameter free string variable γ may be associated with a pair $\langle t_\gamma, \psi_\gamma \rangle$, but here it is used as the eigenvariable to introduce either an instance of $\mathbf{\Sigma}_0^B$-**COMP** or an instance of the $\mathbf{\Sigma}_0^B$-Bit-Recursion axiom scheme (24).

Notation For the discussion below, we say that γ is a *comprehension variable* (resp. *recursion variable*) if it used as the eigenvariable to introduce either an instance of $\mathbf{\Sigma}_0^B$-**COMP** (resp. $\mathbf{\Sigma}_0^B$-Bit-Recursion).

[3]For many theories in [12], the formula $\varphi(x, X^{<x})$ in Definition 4.1 contains X only in the form $X(z)$ for some $z < x$. Hence even the simple transformation described here is not necessary.

Although we can simply translate the $\mathbf{\Sigma}_0^B$-**COMP** axioms into $\mathbf{\Sigma}_1^q$ formulas (and thus cutting such an axiom translates into cutting a $\mathbf{\Sigma}_1^q$ formula), here we translate them into $\mathbf{\Sigma}_0^q$ formulas, by translating the comprehension variables as in the proof of Theorem 2.10.

The "depend" relation between nonparameter free string variables is defined as in the proof of Theorem 2.10. We will translate each nonparameter variables γ inductively (based on the "depend" relation) just as before, but for each recursion variable γ we introduce the corresponding propositional variables $p_0^\gamma, p_1^\gamma, \ldots$, just as for the parameter variables.

There is a complication in getting the right length for each string that is known to exist from the $\mathbf{\Sigma}_0^B$-Bit-Recursion axioms. (Recall that for the strings that exist by the $\mathbf{\Sigma}_0^B$-**COMP** axioms, we try all possible lengths, see the formula (14).) Here we simplify the matter by using the following form of the $\mathbf{\Sigma}_0^B$-Bit-Recursion scheme:

$$\exists X(|X| = y + 1 \wedge \forall x < y,\, X(x) \leftrightarrow \psi(x, X^{<x})) \tag{25}$$

It is straightforward that $\mathbf{TV}^0$ can be equivalently axiomatized by $\mathbf{V}^0$ and the above version of the $\mathbf{\Sigma}_0^B$-Bit-Recursion axiom scheme.

Thus the length of each recursion variable is uniquely determined by the lengths of the parameter variables and comprehension variables in the proof. Therefore translating each formula in a sequent $\mathcal{S}$ in π requires the length of all variables that $\mathcal{S}$ depends on. We discuss the interesting atomic formulas. If γ is a recursion variables that depends on the comprehension variables $\beta_1, \ldots, \beta_\ell$, then given the values $n_{\vec\beta}$ for $|\beta_j|$ and $n_{\vec\alpha}$ for $\vec\alpha$ (the parameter variables), we translate $\gamma(s)$ into

$$\gamma(s)[n_{\vec\alpha}, n_{\vec\beta}] =_{\text{def}} \begin{cases} p_i^\gamma & \text{if } s < v - 1 \\ \top & \text{if } s = v - 1 \\ \bot & \text{otherwise} \end{cases}$$

where $i = val(s)$, and $v = val(t_\gamma)$.

Also, if γ is a comprehension variable that depends other comprehension variables $\beta_1, \ldots, \beta_\ell$, then

$$\gamma(s)[n_\gamma, n_{\vec\alpha}, n_{\vec\beta}] =_{\text{def}} \begin{cases} \psi_\gamma(s)[n_{\vec\alpha}, n_{\vec\beta}] & \text{if } i < n_\gamma - 1 \\ \top & \text{if } i = n_\gamma - 1 \\ \bot & \text{if } i \geq n_\gamma \end{cases}$$

Next we translate $\mathbf{\Sigma}_1^B$ formulas in π. An instance of $\mathbf{\Sigma}_0^B$-**COMP** is handled as in the proof of the Translation Theorem for $\mathbf{VTC}^0$. Consider an instance of (25) that occurs in a sequent $\mathcal{S}$ of π. Given the lengths $n_{\vec\alpha}, n_{\vec\beta}$ of the parameter and comprehension string variables that $\mathcal{S}$ depends on, the instance of (25) in $\mathcal{S}$ is translated into

$$\exists p_0^\gamma \ldots \exists p_{v-1}^\gamma (\bigwedge_{i=0}^{v-1} p_i^\gamma \leftrightarrow \overline{\psi}_i)$$

where $v = val(t_\gamma(n_{\vec\alpha}, n_{\vec\beta}))$, and $\overline{\psi}_i$ is the translation of $\psi(i, \gamma^{<i})$. Note that $\overline{\psi}_0$ does not contain p_j^γ, and $\overline{\psi}_{i+1}$ contains $p_0^\gamma, \ldots, p_i^\gamma$. The next Claim is straightforward:

Claim Let $\mathcal{S}$ be any sequent in π. There is a polynomial $\mathbf{p}$ that satisfies the following condition. Given the lengths $n_{\vec\alpha}$ of the parameter variables in π, and $n_{\vec\beta}$ of the comprehension variables that $\mathcal{S}$ depends on. Then the translation $\mathcal{S}[n_{\vec\alpha}, n_{\vec\beta}]$ has a $\mathbf{G}_1^\star$ proof of size bounded by $\mathbf{p}(n_{\vec\alpha}, n_{\vec\beta})$.

Applying the Claim for the endsequent of π yields the desired result. □

4.2 Subsystems of $\mathbf{G}_1^\star$

From the above proof, a way to define subsystems of $\mathbf{G}_1^\star$ that correspond naturally to subtheories of $\mathbf{TV}^0$ is as follows. For a two-sorted formula ψ, we consider closure of ψ under substitution of $\mathbf{\Sigma}_0^B$ formulas for free string variables. Call this the $\mathbf{\Sigma}_0^B$-closure of ψ. Given a $\mathbf{\Sigma}_0^B$ formula ψ, consider the subtheory $\mathcal{T}$ of $\mathbf{TV}^0$ which is axiomatized by $\mathbf{V}^0$ and the instance (25) of $\mathbf{\Sigma}_0^B$-Bit-Recursion. (Examples of $\mathcal{T}$ include $\mathbf{VTC}^0$, **VNL** and others in [12].) We define the subsytem $\mathbf{G}$-$\mathcal{T}$ of $\mathbf{G}_1^\star$:

Definition 4.3. *A* $\mathbf{G}$-$\mathcal{T}$ *proof is a* $\mathbf{G}_1^\star$ *proof where the cut formulas contain only parameter variables, and are restricted to the* $\mathbf{\Sigma}_1^q$ *translation of the Bit-Recursion axioms for the* $\mathbf{\Sigma}_0^B$*-closure of* ψ*.*

The following Lemma is obvious from the proof of Theorem 4.2.

Lemma 4.4. *Every theorem of* $\mathcal{T}$ *translates into a family of tautologies having polynomial size proofs in* $\mathbf{G}$-$\mathcal{T}$*.*

References

[1] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On Interpolation and Automatization for Frege Systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[2] Samuel Buss. *Bounded Arithmetic.* Bibliopolis, Naples, 1986.

[3] Samuel Buss and Peter Clote. Cutting Planes, Connectivity and Threshold Logic. *Archive for Mathematical Logic*, 35:33–62, 1996.

[4] Stephen Cook. Proof Complexity and Bounded Arithmetic. Course Notes for CSC 2429S. `http://www.cs.toronto.edu/~sacook/`.

[5] Stephen Cook. Theories for Complexity Classes and Their Propositional Translations. In Jan Krajíček, editor, *Complexity of computations and proofs*, pages 175–227. Quaderni di Matematica, 2005.

[6] Stephen Cook and Phuong Nguyen. Introduction to Proof Complexity: Bounded Arithmetic and Propositional Translations. (In progress), 2005.

[7] Jan Krajíček. On the number of steps in proofs. *Annals of Pure and Applied Logic*, 41:153–178, 1989.

[8] Jan Krajíček. On Frege and Extended Frege Proof Systems. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II.* Birkhäuser, 1995.

[9] Jan Johannsen. Equational calculi and constant-depth propositional proofs. In Paul Beame and Samuel Buss, editors, *Proof Complexity and Feasible Arithmetics*, volume 39. AMS DIMACS Series, 1998.

[10] Phuong Nguyen. VTC^0: A Second-Order Theory for TC^0. Master's thesis, University of Toronto, 2004. `http://www.cs.toronto.edu/~ntp/`.

[11] Phuong Nguyen and Stephen Cook. VTC^0: A Second-Order Theory for TC^0. In *Proc. 19th IEEE Symposium on Logic in Computer Science*, 2004.

[12] Phuong Nguyen and Stephen Cook. Theory for $\mathbf{TC}^0$ and Other Small Complexity Classes. *Logical Methods in Computer Science*, 2005.

[13] Domenico Zambella. Notes on Polynomially Bounded Arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.

[14] Domenico Zambella. End Extensions of Models of Linearly Bounded Arithmetic. *Annals of Pure and Applied Logic*, 88:263–277, 1997.