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 $| \cdot$ 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 $= =^1$ 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 $\land, \lor, \lnot$ and both number and string quantifiers $\exists x, \forall x, \forall X$. Bounded number quantifiers are defined as usual, and the bounded string quantifier $\exists X \leq t \varphi$ stands for $\exists X (|X| \leq t \land \varphi)$ and the bounded string quantifier $\forall X \leq t \varphi$ stands for $\forall X (|X| \leq t \lor \varphi)$, 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 $\mathbb{N}$, $U_1$ is $\mathbb{N}$ and $U_2$ is the set of finite subsets of $\mathbb{N}$. Each symbol has its intended interpretation.

In general we will consider a vocabulary $\mathcal{L}$ which extends $\mathcal{L}_A^2$. A formula is $\Sigma^B_0(\mathcal{L})$ if it has no string quantifiers and all number quantifiers are bounded. A formula is $\Sigma^B_1(\mathcal{L})$ (resp. $\Pi^B_1(\mathcal{L})$) if it is a $\Sigma^B_0(\mathcal{L})$ 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-$\Sigma^B_1(\mathcal{L})$ (single-$\Pi^B_1(\mathcal{L})$, single-$\Sigma^B_0(\mathcal{L})$, resp.). A formula is $\exists g\Sigma^B_1(\mathcal{L})$ (resp. $g\Pi^B_1(\mathcal{L})$) if it is obtained from $\Sigma^B_0(\mathcal{L})$ formulas using the connectives $\land$ and $\lor$, bounded number quantifiers and bounded string existential (resp. universal) quantifier ("$g$" for "general"). A formula is $\exists g\Sigma^B_1(\mathcal{L})$ if it is of the form $\exists X \varphi$, where $\varphi$ is $g\Sigma^B_1(\mathcal{L})$. We will omit $\mathcal{L}$ if it is $\mathcal{L}_A^2$.
Whereas here we remark that the {\mathcal{L}^2}\textsuperscript{\bar{A}}-terms $\bar{t}$ (technical $\bar{t}$ terms $\bar{t}^2$) and $\Pi^B(\mathcal{L})$ are defined similarly to $\Sigma^B$ and $\Pi^B$, with the additional requirement that all the quantifiers are $\mathcal{L}^2_A$ terms.

\[
\text{axiomatized by the set of axioms 2-BASIC (Figure 1) and the $\Sigma^B_0$-COMP axiom.}
\]

\[
\exists X < a \forall z < a, X(z) \leftrightarrow \varphi(z)
\]  

(1)

not containing $X$.

\[
\begin{array}{ll}
B1. & x + 1 \neq 0 \\
B2. & x + 1 = y + 1 \supset x = y \\
B3. & x + 0 = x \\
B4. & x + (y + 1) = (x + y) + 1 \\
B5. & x \cdot 0 = 0 \\
B6. & x \cdot (y + 1) = (x \cdot y) + x \\
L1. & X(y) \supset y < |X| \\
SE. & |X| = |Y| \land \forall i < |X|(X(i) \leftrightarrow Y(i)) \supset X = Y
\end{array}
\]

Figure 1: 2-BASIC

It has been shown [4, 5] that $V^0$ characterizes $AC^0$ in the sense that the $AC^0$ functions are precisely the $\Sigma^B_1$-definable functions of $V^0$. An important $AC^0$ function is $Row(z, X)$ (also $X[i]$) which is defined as

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

Using $Row$ we can code any finitely many strings into one string. Also, it is $\Sigma^B_0$-definable in $V^0$, and $V^0(\text{Row})$ proves $\Sigma^B(\text{Row})$-COMP. Indeed, every $\Sigma^B_0(\text{Row})$-formula is provably equivalent in $V^0(\text{Row})$ to a $\Sigma^B_0$ formula.

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

\[
\text{NUMONES} \equiv \forall X \exists Y, \forall z < |X| \exists y \leq |X|Y(z, y) \land Y(0, 0) \land \\
\forall z < |X| \forall y \leq |X|, Y(z, y) \supset [(X(z) \supset Y(z + 1, y + 1)) \land (\neg X(z) \supset Y(z + 1, y + 1))].
\]  

(2)

Notice that $Y$ has length bounded by $1 + |X| = |X|$.

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

Proposition 1.2. The function $\text{numones}$ is $\Sigma^B_1$-definable in $VTC^0$.

Proof. It is easy to see that $\text{numones}$ can be defined by the following $\Sigma_1$-formula:

\[
\text{numones}(x, X) = y \leftrightarrow \exists Y, \forall z < x \exists y \leq x Y(z, y) \land Y(0, 0) \land \\
\forall z < x \forall y \leq x, Y(z, y) \supset [(X(z) \supset Y(z + 1, y + 1)) \land (\neg X(z) \supset Y(z + 1, y + 1))].
\]  

(3)

Using $\text{NUMONES}$ (for existence) and $\Sigma^B_0$-IND (for uniqueness), it is straightforward that $VTC^0(\text{numones})$ proves $\forall x \forall X \exists y \text{numones}(x, X) = y$. \qed
Let $\text{VTC}\theta(n\text{umones})$ denote the extension of $\text{VTC}\theta$ obtained by adding $\Sigma_1^B$-defining axiom (3) for $n\text{umones}$. It follows from the above proposition that $\text{VTC}\theta(n\text{umones})$ is a conservative extension of $\text{VTC}\theta$, because any model of $\text{VTC}\theta$ can be expanded to a model of $\text{VTC}\theta(n\text{umones})$.

**Lemma 1.3.** The theory $\text{VTC}\theta(n\text{umones})$ can be equivalently axiomatized by 2-BASIC, $\Sigma_1^B(n\text{umones})$-COMP and the following axioms:

\begin{align*}
\text{numones}(X, 0) &= 0 \\
X(z) \supset \text{numones}(X, z + 1) &= \text{numones}(X, z) + 1 \\
\neg X(z) \supset \text{numones}(X, z + 1) &= \text{numones}(X, z).
\end{align*}

**Proof.** For one direction, we prove that $\text{VTC}\theta(n\text{umones})$ proves $\Sigma_1^B(n\text{umones})$-COMP and the axioms (4), (5) and (6). It is easy to see that (4), (5) and (6) are provable in $\text{VTC}\theta(n\text{umones})$. For $\Sigma_1^B(n\text{umones})$-COMP, it is shown [12] that $\text{VTC}\theta$ is conservative over $\text{VTC}\theta(\text{Row}, \text{numones})$, and that $\text{VTC}\theta$ proves $\Sigma_1^B(\text{L FTC}^\theta)$-COMP. Since $\text{VTC}\theta(\text{Row}, \text{numones})$ is a conservative extension of $\text{VTC}\theta(n\text{umones})$, it follows that the latter proves $\Sigma_1^B(n\text{umones})$-COMP.

For the other direction, it suffices to show that $\text{VTC}\theta(n\text{umones})$ proves $\text{NUMONES}$. The string $Y$ in (2) is proved to exist in $\text{VTC}\theta(n\text{umones})$ by $\Sigma_0^B(n\text{umones})$-COMP:

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

The correctness of $Y$ is proved in $\text{VTC}\theta(n\text{umones})$ using the properties of $n\text{umones}$ (4), (5) and (6). \qed

## 2 Propositional Translation

We will now discuss the connection between our theory $\text{VTC}\theta(n\text{umones})$ and the propositional threshold logic $\text{PTK}$ [3] (a member of the so-called $\text{TC}\theta$-$\text{Frege}$ family [1]). We will show that each $\Sigma_0^B(n\text{umones})$ theorem of $\text{VTC}\theta$ translates into a family of tautologies which have short $\text{PTK}$ proofs, where the depths of the propositional formulas are bounded by some constant. This is done by translating the $\text{VTC}\theta(n\text{umones})$-proofs into $\text{PTK}$-proofs. First, we recall the definition of $\text{PTK}$.

### 2.1 PTK

The system $\text{PTK}$ is introduced in [3]. It extends $\text{Frege}$ systems by the threshold connectives $\text{Th}_k^n(\phi_1, \ldots, \phi_n)$ true ($\top$) if and only if there are at least $k$ true $\phi_i$'s. In $\text{PTK}$ $\land$ and $\lor$ 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, $\text{Th}_k(\phi_1, \ldots, \phi_n)$ is syntactically $\bot$ if $k > n$, and $\top$ if $k = 0$. Axioms of $\text{PTK}$ include

$$\dashv \top$$

$$\bot \rightarrow$$

$$\varphi \rightarrow \varphi$$

for any $\text{PTK}$ formula $\varphi$. The rules of $\text{PTK}$ 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.\footnote{We slightly modify the rules $\text{Th}_k$-right and $\text{Th}_k$-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.}

Buss and Clote [3] also introduce the system $\text{PTK}'$, which is $p$-equivalent to $\text{PTK}$. The rules of $\text{PTK}'$ are the same as that of $\text{PTK}$, except for the $\text{Th}_k$ introduction rules, which are given in Figure 3.
Both PTK and PTK' are sound and complete, but PTK enjoys cut elimination while this is unknown for PTK'. Also PTK' seems at first sight more powerful than PTK. Nevertheless, they are p-equivalent.\footnote{And both are p-equivalent to FC\textsuperscript{c} \cite{89}, where FC is the system that extends Frege proof systems by new connectives C\textsubscript{h,k} which count exactly the number of true arguments. A drawback of these connectives is that they are not distributive over either \& or \lor.} We show in Theorem 2.2 that PTK p-simulates PTK'. This also verifies that PTK is p-equivalent to its original form defined in \cite{3}. The proof is interesting, but it is independent of the rest of the paper.

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

\textbf{Definition 2.1.} If \( \varphi \) is \( \bot \), \( \top \) or a propositional variable, then \( \text{size}(\varphi) = 1 \), \( \text{depth}(\varphi) = 0 \). If \( \varphi \equiv \neg \psi \) then \( \text{size}(\varphi) = 1 + \text{size}(\psi) \) and \( \text{depth}(\varphi) = 1 + \text{depth}(\psi) \). If \( \varphi \equiv Th_k(\varphi_1, \ldots, \varphi_n) \) \((1 \leq k \leq n)\), then \( \text{size}(\varphi) = n + k + 1 + \sum \text{size}(\varphi_i) \), and \( \text{depth}(\varphi) = 1 + \max \{\text{depth}(\varphi_i)\} \). The size (resp. depth) of a sequent is the total size (resp. maximal depth) of the formulas in the sequent.

\textbf{Theorem 2.2.} PTK p-simulates PTK'.

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

\[
Th_k(p_1, \ldots, p_m) \rightarrow Th'_k(p_1, \ldots, p_{m'}),
\]

Figure 2: Introduction rules of PTK

Figure 3: Th\textsubscript{k}-introduction rules of PTK'
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'.

\[\square\]

**Lemma 2.3.** There is a polynomial \(p(n)\) so that for \(n \in \mathbb{N}, n \geq 1\), there is a PTK proof \(\pi_n\) of size \(\leq p(n)\) 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 \(\varphi_i\)’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:

\(a)\) \(\text{Th}_{k+1}(p_1, \ldots, p_{m'}) \rightarrow \text{Th}_1(p_1, \ldots, p_m), \text{ for } m \geq 1, m' \leq m + k\).

\(b)\) \(\text{Th}_n(p_1, \ldots, p_n) \rightarrow \text{Th}_n(p_1, \ldots, p_m), \text{ for } 1 \leq k \leq m \leq n\).

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

\[\text{Th}_{k+1}(p_1, \ldots, p_{m+k}) \rightarrow \text{Th}_1(p_1, \ldots, p_m)\]

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

1. \(\text{Th}_{k+1}(p_1, \ldots, p_{m+k}) \rightarrow \text{Th}_1(p_1, \ldots, p_m)\)
2. \(\text{Th}_{k+1}(p_1, \ldots, p_{m+k}) \rightarrow p_1, \ldots, p_m\) from 1, using \(\lor\)-right
3. (i). \(\text{Th}_{k+1}(p_2, \ldots, p_{m+k}) \rightarrow p_1, \ldots, p_m\) from 2, using \(\text{Th}_{k+1}\)-left
3. (ii). \(p_1, \text{Th}_k(p_2, \ldots, p_{m+k}) \rightarrow p_1, \ldots, p_m\) from 2, using \(\text{Th}_{k+1}\)-left

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

\[\text{Th}_{k+1}(p_{m}, \ldots, p_{m+k}) \rightarrow p_1, \ldots, p_m\]

This is derivable using \(\land\)-left rule and the structural rules.

Part b) is proved similarly. \(\square\)

**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 \(\pi_n\) from \(\pi_{n-1}\). It will be evident from the construction that in general, the size of \(\pi_n\) is bounded by some polynomial \(p(n)\).

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 \(\pi_{n-1}\). 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 \(\pi_{n-1}\). Suppose that either \(m = n\) or \(m' = n\). Reasoning backward. Since both \(k' < n\) and \(k < n\), we can apply the appropriate rule (either \(\text{Th}_n\)-left or \(\text{Th}_n\)-right) to obtain the sequent \((8)\) from the existing sequents in \(\pi_{n-1}\). \(\square\)

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

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

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

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

We will prove that for a PTK proof $\pi$,

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

there is a treelike PTK proof of

$$\rightarrow \text{Th}_n(\mathcal{S}_1, \ldots, \mathcal{S}_n).$$

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

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

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

a) $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n \rightarrow \hat{\mathcal{S}}$,

b) $\text{Th}_{n+m}(\Lambda, \hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n) \rightarrow \text{Th}_{n+m+1} (\Lambda, \hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_n, \hat{\mathcal{S}})$.

**Proof.** Since $\Lambda$ contains exactly $m$ formulas, it is straightforward to derive the sequent in part b) from the sequent in part a) using the $\land$-right and $\land$-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 $\text{Th}_k$-right rule. Suppose that

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

By definition,

$$\hat{\mathcal{S}}_1 \equiv \text{Th}_1(\lnot \text{Th}_k(\varphi_2, \ldots, \varphi_n), \Lambda)$$

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

$$\hat{\mathcal{S}} \equiv \text{Th}_1(\lnot \text{Th}_k(\varphi_1, \ldots, \varphi_n), \Lambda)$$
Reasoning backward (ignore sequents which can be obtained from axioms just by structural rules).

1. \( \text{Th}_1 (\neg \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda), \text{Th}_1 (\neg \varphi_1, \neg \text{Th}_{k-1} (\varphi_2, \ldots, \varphi_n), \Lambda), \longrightarrow \text{Th}_1 (\neg \text{Th}_k (\varphi_1, \ldots, \varphi_n), \Lambda) \)

2. \( \text{Th}_1 (\neg \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda), \text{Th}_1 (\neg \varphi_1, \neg \text{Th}_{k-1} (\varphi_2, \ldots, \varphi_n), \Lambda), \longrightarrow \neg \text{Th}_k (\varphi_1, \ldots, \varphi_n), \Lambda \)  \( (1. \ \vee\text{-right}) \)

3. \( \text{Th}_k (\varphi_2, \ldots, \varphi_n), \text{Th}_1 (\neg \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda), \text{Th}_1 (\neg \varphi_1, \neg \text{Th}_{k-1} (\varphi_2, \ldots, \varphi_n), \Lambda), \longrightarrow \Lambda \)  \( (2. \ \neg\text{-right}) \)

4. \( \text{Th}_k (\varphi_2, \ldots, \varphi_n), \neg \text{Th}_k (\varphi_2, \ldots, \varphi_n), \text{Th}_1 (\neg \varphi_1, \neg \text{Th}_{k-1} (\varphi_2, \ldots, \varphi_n), \Lambda), \longrightarrow \Lambda \)  \( (3. \ \vee\text{-left}) \)

5. \( \text{Th}_k (\varphi_2, \ldots, \varphi_n), \neg \text{Th}_k (\varphi_2, \ldots, \varphi_n), \neg \varphi_1, \longrightarrow \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda \)  \( (4. \ \neg\text{-left}) \)

6. \( \text{Th}_k (\varphi_2, \ldots, \varphi_n), \neg \varphi_1, \longrightarrow \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda \)  \( (5. \ \vee\text{-left}) \)

7. \( \text{Th}_k (\varphi_2, \ldots, \varphi_n), \neg \text{Th}_{k-1} (\varphi_2, \ldots, \varphi_n), \longrightarrow \text{Th}_k (\varphi_2, \ldots, \varphi_n), \Lambda \)  \( (5. \ \vee\text{-left}) \)

Now, 6. and 7. are easily obtained from the \( \neg\text{-left} \) and \( \text{Th}_k\text{-left} \) rules.

\( \square \)

**Corollary 2.8.** Tredike PTK \( p \)-simulates daglike PTK.

### 2.2 Translating \( \Sigma_0^0\) (numones) Formulas

First we recall the translation of a \( \Sigma_0^0 \) formula \( \varphi (\overline{X}) \) [4]: \( \varphi (\overline{X}) \) translates into a family of propositional formulas \( \{ \varphi (\overline{X}) [\overline{n}] : \overline{n} \geq 0 \} \), so that for all \( \overline{n} \in \mathbb{N} \), \( \varphi (\overline{X}) [\overline{n}] \) is valid iff \( \mathbb{N} \models \exists \overline{x} (|\overline{x}| = \overline{n} \land \varphi (\overline{x})) \). 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 \( \text{val} (t) \) be the numerical value of a closed term \( t \). The translation is defined inductively. For the base case, for example,

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

and

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

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 \( \varphi (\overline{X}) \equiv s = t \) that contains numones. Translation of this function requires the threshold connectives in PTK. Let numones \( \text{numones}(X_i, t_i) \) (for \( i = 0, \ldots, \ell \)) be all occurrences of numones in \( \varphi \) (possibly with repetitions). Fix \( \overline{n} \). Let \( S \) be the set of all tuples \( (k_0, \ldots, k_\ell) \) (where \( k_i \leq \min \{|X_i|, t_i|\} \)) that make \( \varphi (\overline{X}) \) true when \( |\overline{X}| = \overline{n} \) and \( \text{numones}(X_{j_i}, t_{j_i}) = k_j \). Then,

\[
(s = t)[\overline{n}] = \text{def} \bigvee_{k \in S} \bigwedge_{j=0}^{\ell} [\text{Th}_{k_j} (p_j^X) \land \neg \text{Th}_{k_{j+1}} (p_j^X)],
\]

where \( \text{Th}_0 (\ldots) = \text{syn} \top \) and \( \text{Th}_k (p_1, \ldots, p_n) = \text{syn}\bot \) 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 \( \exists z < |X| (z = t \land X(z)) \). Also, to translate any atomic formula of the form \( \alpha = \beta \), we translate the LHS of SE (Figure 1).

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

**Lemma 2.9.** For each simple \( \Sigma_0^0 \) (numones) formula \( \varphi (\overline{x}, \overline{Y}) \), there is a constant \( d \) and a polynomial \( p \) so that for all sequences \( \overline{m}, \overline{n} \), the propositional formula \( \varphi (\overline{x}, \overline{Y}) [\overline{m}; \overline{n}] \) has depth \( d \) and size bounded by \( p(|\overline{n}|) \).

The remaining of this section is devoted to the proof of the following theorem.
Theorem 2.10 (Propositional Translation Theorem for VTC\(^0\)). For each \(\Sigma^P_0\) (numones) theorem \(\varphi(X)\) of VTC\(^0\) (numones), there is a constant \(d\) and a polynomial \(p\) such that the family of tautologies \(\{\varphi(X)[\vec{\alpha}] : \vec{\alpha} \in \mathbb{N}\}\) has \(d\)-PTK proofs of sizes bounded by \(p(\vec{\alpha})\).

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:

\[
p_m \to \bigvee_{0 \leq k \leq m} [\text{Th}_k(\vec{p}) \land \neg \text{Th}_{k+2}(\vec{p}, p_m)] \tag{9}
\]

\[
p_m \to \bigvee_{0 \leq k \leq m} [\text{Th}_k(\vec{p}) \land \neg \text{Th}_{k+1}(\vec{p})] \tag{10}
\]

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

\[
p_m \to \text{Th}_0(\vec{p}) \land \neg \text{Th}_2(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m).
\]

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

1. \(p_m \to \text{Th}_0(\vec{p}) \land \neg \text{Th}_2(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m)\)
2. \(p_m \to \neg \text{Th}_2(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m)\) \quad \text{1. \ -right}
3. \(p_m, \text{Th}_2(\vec{p}, p_m) \to \text{Th}_1(\vec{p}) \land \neg \text{Th}_3(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m)\) \quad \text{2. \ -right}
4. \(p_m, \text{Th}_1(\vec{p}) \to \text{Th}_0(\vec{p}) \land \neg \text{Th}_3(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m)\) \quad \text{3. \ Th}_2\text{-left}
5. \(p_m, \text{Th}_1(\vec{p}) \to \neg \text{Th}_0(\vec{p}, p_m), \ldots, \text{Th}_m(\vec{p}) \land \neg \text{Th}_{m+2}(\vec{p}, p_m)\) \quad \text{4. \ -right}

\[\vdots\]

\[3m + 2. \ p_m, \text{Th}_1(\vec{p}), \ldots, \text{Th}_m(\vec{p}) \to \neg \text{Th}_{m+2}(\vec{p}, p_m)\]

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

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

1. \(\to \bigvee_{0 \leq k \leq m} [\text{Th}_k(\vec{p}) \land \neg \text{Th}_{k+1}(\vec{p})]\)
2. \(\to \neg \text{Th}_1(\vec{p}), \text{Th}_1(\vec{p}) \land \neg \text{Th}_2(\vec{p}), \ldots, \text{Th}_{m-1}(\vec{p}) \land \neg \text{Th}_m(\vec{p}), \text{Th}_m(\vec{p})\) \quad \text{1. \ V-right}
3. \(\text{Th}_1(\vec{p}) \to \text{Th}_0(\vec{p}) \land \neg \text{Th}_2(\vec{p}), \ldots, \text{Th}_{m-1}(\vec{p}) \land \neg \text{Th}_m(\vec{p}), \text{Th}_m(\vec{p})\) \quad \text{2. \ -right}
4. \(\text{Th}_1(\vec{p}) \to \neg \text{Th}_0(\vec{p}), \ldots, \text{Th}_{m-1}(\vec{p}) \land \neg \text{Th}_m(\vec{p}), \text{Th}_m(\vec{p})\) \quad \text{3. \ -right}

\[\vdots\]

\[2m. \ \text{Th}_1(\vec{p}), \ldots, \text{Th}_{m-1}(\vec{p}) \to \neg \text{Th}_m(\vec{p}), \text{Th}_m(\vec{p})\]

The last sequent is obtained from the axiom \(\text{Th}_m(\vec{p}) \to \text{Th}_m(\vec{p})\). \(\square\)

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

**Proof.** Consider the axiom (5)

\[X(z) \supset \text{numones}(X, z + 1) = \text{numones}(X, z) + 1.\]

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

\[A \equiv \bigvee_{0 \leq k \leq m} [(\text{Th}_{k+1}(\vec{p}, p_m) \land \neg \text{Th}_{k+2}(\vec{p}, p_m)) \land (\text{Th}_k(\vec{p}) \land \neg \text{Th}_{k+1}(\vec{p}))]\]
Now (5) translates into $\neg p_m \lor A$. We show how to derive $p_m \rightarrow A$ in PTK.

For $k \leq m$ the sequent
\[ p_m, \text{Th}_k(\bar{p}) \rightarrow \text{Th}_{k+1}(\bar{p}, p_m) \] (a)
is derivable in PTK (using Th$_{k+1}$-right rule). Therefore we also derive
\[ p_m, \neg \text{Th}_{k+2}(\bar{p}, p_m) \rightarrow \neg \text{Th}_{k+1}(\bar{p}) \] (b)

From (a), (b) we derive (for $k \leq m$)
\[ p_m, \text{Th}_k(\bar{p}) \land \neg \text{Th}_{k+2}(\bar{p}, p_m) \rightarrow (\text{Th}_k(\bar{p}) \land \neg \text{Th}_{k+1}(\bar{p})) \land (\text{Th}_{k+1}(\bar{p}, p_m) \land \neg \text{Th}_{k+2}(\bar{p}, p_m)). \]

By weakening we get
\[ p_m, \text{Th}_k(\bar{p}) \land \neg \text{Th}_{k+2}(\bar{p}, p_m) \rightarrow A \]

Combine these sequents for $k \leq m$, using $\lor$-left, we get
\[ p_m, \bigvee_{k \leq m} [\text{Th}_k(\bar{p}) \land \neg \text{Th}_{k+2}(\bar{p}, p_m)] \rightarrow A \]

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

The proof for the axiom (6) is slightly different. It translates into
\[ \rightarrow p_m \lor \bigvee_{k \leq m} [(\text{Th}_k(\bar{p}, p_m) \lor \neg \text{Th}_{k+1}(\bar{p}, p_m)) \lor (\text{Th}_k(\bar{p}) \lor \neg \text{Th}_{k+1}(\bar{p}))] \]

For each $k \leq m$, it is straightforward to derive the following sequents:
\[ \text{Th}_k(\bar{p}) \rightarrow p_m, \text{Th}_k(\bar{p}, p_m) \quad \neg \text{Th}_{k+1}(\bar{p}) \rightarrow p_m, \neg \text{Th}_{k+1}(\bar{p}, p_m) \]

Hence, we can derive
\[ \text{Th}_k(\bar{p}) \land \text{Th}_{k+1}(\bar{p}) \rightarrow p_m, (\text{Th}_k(\bar{p}, p_m) \land \neg \text{Th}_{k+1}(\bar{p}, p_m)) \land (\text{Th}_k(\bar{p}) \land \neg \text{Th}_{k+1}(\bar{p})) \]

and thus
\[ \text{Th}_k(\bar{p}) \land \text{Th}_{k+1}(\bar{p}) \rightarrow p_m, \bigvee_{i \leq m} [(\text{Th}_i(\bar{p}, p_m) \land \neg \text{Th}_{i+1}(\bar{p}, p_m)) \land (\text{Th}_i(\bar{p}) \land \neg \text{Th}_{i+1}(\bar{p}))] \]

for all $k \leq m$.

As a result, we can derive
\[ \bigvee_{k \leq m} [\text{Th}_k(\bar{p}) \land \text{Th}_{k+1}(\bar{p})] \rightarrow p_m, \bigvee_{i \leq m} [(\text{Th}_i(\bar{p}, p_m) \land \neg \text{Th}_{i+1}(\bar{p}, p_m)) \land (\text{Th}_i(\bar{p}) \land \neg \text{Th}_{i+1}(\bar{p}))]. \]

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

\[ \square \]

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

**Definition 2.13 (Free Variable Normal Form).** Let $\pi$ be an LK$^2$ proof of a formula $\varphi$. A free variable in $\varphi$ is called a parameter variable of $\pi$. We say $\pi$ 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 $\pi$ 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 $VTC^0$. Let $\varphi(\vec{X})$ be a $\Sigma_0^B(\text{numones})$ theorem of $VTC^0(\text{numones})$. Then there is an anchored (aka free cut-free) $LK^2$ proof $\pi$ of $\varphi$ which is in Free Variable Normal Form, and which has (by Lemma 1.3) nonlogical axioms from 2-BASIC, $\Sigma_0^B(\text{numones})$-COMP and (4), (5) and (6). Because $\pi$ is in Free Variable Normal Form, if a nonparameter string variable $\gamma$ is used as the eigenvariable in

$$\forall x < t([\vec{a}]), \gamma(x) \leftrightarrow \psi(x, \vec{a}), \Gamma \rightarrow \Delta$$

$$\exists y < t([\vec{a}]), Z(x) \leftrightarrow \psi(x, \vec{a}), \Gamma \rightarrow \Delta$$

then we can associate $\gamma$ with the pair $\langle t, \gamma \rangle$ = $\langle t([\vec{a}]), \psi(x, \vec{a}) \rangle$, which may contain other nonparameter free variables.

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

For each nonparameter free variable $\gamma$, we will not use the propositional variable $p_0, p_1, \ldots$ as for the parameter variables. Instead, we translate each bit $\gamma(z)$ by translating the associated formula $\psi_\gamma$. Since $\psi_\gamma$ 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 acyclic relation.

Notation We say that $\gamma$ depends on $\beta$ if $\beta$ occurs in $t_\gamma$ or $\psi_\gamma$, or if there is another nonparameter free variable $\gamma'$ such that $\gamma$ depends on $\gamma'$, and $\gamma'$ depends on $\beta$. A sequent $S$ is said to depend on $\gamma$ if $S$ contains $\gamma$, or $S$ contains some variable $\beta$ that depends on $\gamma$.

For a nonparameter variable $\gamma$ translating $\gamma(s)$ requires the lengths of $\gamma$, other nonparameter free variables that it depends on, and the parameter variables. We define the translation of $\gamma(s)$ inductively. For the base case, $\gamma$ 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 $\gamma$ depends on, then for $n_\gamma \leq t_\gamma(n_\beta)$, define

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

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

A $\Sigma_0^B(\text{numones})$ formula in $\pi$ is translated by translating its atomic subformulas. Consider now an instance of $\Sigma_0^B(\text{numones})$-COMP in $\pi$:

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

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

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

where $v = val(t)$, and

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

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

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

For the base case, $\mathcal{S}$ is an axiom of LK$^2$-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 $V^0$ proofs into bounded depth Frege proofs.) It is also easy to show that the translation (14) of the $\Sigma^B_0$-COMP axioms has short PTK proofs. For the induction step, we consider the interesting case of the rule string $\exists$ left. Suppose that

$$\frac{\mathcal{S}_1}{\mathcal{S}} \quad \forall x < t(|\bar{a}|), \gamma(x) \leftrightarrow \psi(x, \bar{a}), \Gamma \longrightarrow \Delta$$

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

$$\mathcal{S}_1[n_\gamma, n_\alpha, n_\beta] = \exists \bigvee_{n_\gamma = 0}^v A[n_\gamma, n_\alpha, n_\beta], ||\Gamma|| \longrightarrow ||\Delta||$$

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

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

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

3 Propositional Translation For $V^0(m)$ And VACC

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

$$\varphi_{MOD_m}(X, Y) \equiv \forall z \leq |X| \exists y < m Y(z, y) \land Y(0, 0) \land \forall z \leq |X| \forall y < m, Y(z, y) \supset [(X(z) \supset Y(z + 1, y + 1 \mod m)) \land (\neg X(z) \supset Y(z + 1, y))].$$

(15)

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

$$\exists r < m, \exists q < y, y = qm + r \land \phi(r).$$

(16)

Thus if $\phi(y)$ is $\Sigma^B_0$, then $\phi(y \mod m)$ is also $\Sigma^B_0$.

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

$$MOD_m \equiv \forall X \exists Y \varphi_{MOD_m}(X, Y) \quad \text{and} \quad V^0(m) = V^0 \cup \{MOD_m\}$$

(17)

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

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

(18)

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

$$\mod_m(X, 0) = 0$$

(19)

$$X(z) \land \mod_m(X, z) = m - 1 \supset \mod_m(X, z + 1) = 0$$

(20)

$$X(z) \land \mod_m(X, z) < m - 1 \supset \mod_m(X, z + 1) = \mod_m(X, z) + 1$$

(21)

$$\neg X(z) \supset \mod_m(X, z + 1) = \mod_m(X, z)$$

(22)
We present the systems \( \text{PK}(m) \), for each \( m \in \mathbb{N}, m \geq 2 \). For \( m = 2 \), the system \( \text{PK}(2) \) is the same as \( \text{PK} \oplus \) in [9]. In general, \( \text{PK}(m) \) extends \( \text{Freg} \) proof systems by the connective \( M^k_m \) connectives, where \( 0 \leq k < m \). The meaning of \( M^k_m(\varphi_1, \ldots, \varphi_n) \) is that the number of true \( \varphi_i \)'s modulo \( m \) is exactly \( k \). The logical rules of \( \text{PK}(m) \) are:

\[
\begin{align*}
\frac{\Lambda \rightarrow \varphi, \Gamma}{\neg \varphi, \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{left}\)} \\
\frac{\varphi_1, \ldots, \varphi_m, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, \ldots, \varphi_m, \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{right}\)} \\
\frac{\Lambda \rightarrow \varphi_1, \ldots, \ldots, \varphi_n, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, \ldots, \varphi_n, \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{right}\)} \\
\frac{\Lambda \rightarrow \Lambda \rightarrow \varphi_1, \ldots, \varphi_m, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, \ldots, \varphi_m, \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{right}\)} \\
\frac{\Lambda \rightarrow \varphi_1, \ldots, \varphi_n \lor \varphi_1, \ldots, \varphi_n, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, \ldots, \varphi_n \lor \varphi_1, \ldots, \varphi_n, \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{right}\)} \\
\frac{\Lambda \rightarrow \varphi_1, \ldots, \varphi_n, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, M^k_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{left}\)} \\
\frac{\Lambda \rightarrow \varphi_1, \ldots, \varphi_n, \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, M^{k-1}_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{left}\)} \\
\frac{\Lambda \rightarrow \varphi_1, M^k_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, M^{k-1}_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{left}\)} \\
\frac{\Lambda \rightarrow \varphi_1, M^k_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma}{\Lambda \rightarrow \varphi_1, M^{k-1}_m(\varphi_2, \ldots, \varphi_n), \Lambda \rightarrow \Gamma} & \quad \text{\(-\text{left}\)}
\end{align*}
\]

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

**Theorem 3.2.** For each \( \Sigma^0_d \) theorem \( \varphi(\bar{x}) \) of \( \forall \theta(m) \), there is a constant \( d \) and a polynomial \( p \) such that the family of tautologies \( \{ \varphi(\bar{x})[\bar{n}] : \bar{n} \in \mathbb{N} \} \) has \( d\text{-PK}(m) \) proofs of sizes bounded by \( p(\bar{n}) \).

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

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

\[
\varphi(\bar{x}, \bar{Y})[\bar{m}; \bar{n}] \equiv \bigvee_{\bar{k} \in S} \bigwedge_{j=0}^\ell M^k_m(\bar{p}^j),
\]

where \( \bar{p}^j \) denotes \( p_0^j, \ldots, p_{\text{val}(t_j[\bar{m}; \bar{n}; \bar{k}])-1} \).

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

\[
(p_m \land \bigvee_{k=0}^{a-1} M^k_m(\bar{p})) \supset \bigvee_{k=0}^{a-1} [M^k_{a+1}(\bar{p}, p_m) \land M^k_a(\bar{p})],
\]

where \( \bar{p} \) is the list of \( p_0, \ldots, p_{m-1} \). It is easy to derive in \( \text{PK}(a) \) the sequent

\[
p_m, M^k_a(\bar{p}) \rightarrow M^{k+1}_a(\bar{p}, p_m) \land M^k_a(\bar{p}),
\]

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

\[
p_m, M^k_a(\bar{p}) \rightarrow \bigvee_{k=0}^{a-1} [M^k_{a+1}(\bar{p}, p_m) \land M^k_a(\bar{p})],
\]

for \( 0 \leq k < m \). From this, we obtain the desired sequent using \(-\text{right}. \)

\( \square \)

12
Note that the depths of the propositional formulas do not depend on \( a \). Define \( \text{PKM} \) to be \( \text{PK} \) together with the connectives \( M_k^a \), for \( a, k \in \mathbb{N}, 0 \leq k < a, a \geq 2 \), which have introduction rules as in \( \text{PK}(m) \).

**Corollary 3.3.** For each \( \Sigma^B_0 \) theorem \( \varphi(\vec{x}, \vec{Y}) \) of \( \text{VACC} \), there is a constant \( d \) and a polynomial \( p \) such that the family of tautologies \( \{ \varphi(\vec{x}, \vec{Y})[n\vec{t}; \vec{\bar{t}}]\}_{n, \vec{t} \in \mathbb{N}} \) has \( d \)-\( \text{PKM} \) proofs of sizes bounded by \( p(\vec{t}) \).

## 4 Translation for \( \text{TV}^0 \) and other Subsystems

It is known that each \( \Sigma^B_0 \) theorem of \( \text{V}^1 \) translates into a family of tautologies having polynomial size \( G^*_1 \) (or equivalently \( \text{eFrege} \) proofs). Since \( \text{V}^1 \) is \( \Sigma^B_0 \) conservative over \( \text{TV}^0 \), the result also holds for \( \text{TV}^0 \). Here we will briefly explain a direct proof of this statement, which helps to define subsystems of \( G^*_1 \) that correspond naturally to subsystems of \( \text{TV}^0 \), such as those discussed in [12].

### 4.1 Propositional Translation for \( \text{TV}^0 \)

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

\[
|X^{<x}| \leq x \land \forall z < x X^{<x}(z) \iff X(z)
\]

In general, any \( \Sigma^B_0(\text{Chop}) \) formula can be transformed to a \( \Sigma^B_0 \) formula by successively eliminating the occurrence of \( \text{Chop} \) (see also [6]). Defining \( \text{TV}^0 \) requires only \( \Sigma^B_0(\text{Chop}) \) formulas that are obtained from a \( \Sigma^B_0 \) 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 \( \Sigma^B_0 \) equivalence, we first replace each atomic formula \( \eta(\{Z\}) \) of \( \varphi(Z) \) by

\[
\eta' \equiv (\forall z < t \neg Z(x) \land \eta(0)) \lor \exists z < t, \ Z(z) \land \forall x < t(z < x \lor \neg Z(x)) \land \eta(z + 1)
\]

Now the \( \Sigma^B_0 \) equivalence of \( \varphi(X^{<t}) \) is obtained by replacing each atomic formula \( Z(s) \) by \( s \leq t \land X(s) \).

We will work with the following definition of \( \text{TV}^0 \):

**Definition 4.1** ([5]). The theory \( \text{TV}^0 \) has vocabulary \( L^2_\lambda \) and is axiomatized by \( \text{V}^0 \) and the \( \Sigma^B_0 \)-Bit-Recursion scheme:

\[
\exists X \leq y \forall x < y (X(x) \iff \varphi(x, X^{<x}))
\]  \hspace{1cm} (24)

where \( \varphi \) is any \( \Sigma^B_0 \) formula that only contains \( X \) in the context \( X^{<x} \), and \( \varphi(x, X^{<x}) \) is understood to be its \( \Sigma^B_0 \) equivalence which is obtained as described above.

Recall that the \( ||\varphi|| \) denotes the family of propositional formulas translated from \( \varphi \).

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

**Proof.** We will prove the Theorem for a \( \Sigma^B_0 \) theorem of \( \text{TV}^0 \). It is easy to extend the proof to consider other theorems of \( \text{TV}^0 \). Consider an anchored \( \text{LK}^2\cdot\text{TV}^0 \) proof in free variable normal form \( \pi \) of \( \varphi \). We may also assume that the contraction rule is not used for \( \Sigma^B_1 \) formulas. As before, a nonparameter free string variable \( \gamma \) may be associated with a pair \( (t, \psi) \), but here it is used as the eigenvariable to introduce either an instance of \( \Sigma^B_0 \)-\text{COMP} or an instance of the \( \Sigma^B_0 \)-Bit-Recursion axiom scheme (24).

**Notation** For the discussion below, we say that \( \gamma \) is a comprehension variable (resp. recursion variable) if it is used as the eigenvariable to introduce either an instance of \( \Sigma^B_0 \)-\text{COMP} (resp. \( \Sigma^B_0 \)-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 $\Sigma_0^B$-COMP axioms into $\Sigma_1^B$ formulas (and thus cutting such an axiom translates into cutting a $\Sigma_1^B$ formula), here we translate them into $\Sigma_0^B$ 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 $\gamma$ inductively (based on the “depend” relation) just as before, but for each recursion variable $\gamma$ we introduce the corresponding propositional variables $p^n_0, p^n_1, \ldots$, 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 $\Sigma_0^B$-Bit-Recursion axioms. (Recall that for the strings that exist by the $\Sigma_0^B$-COMP axioms, we try all possible lengths, see the formula (14).) Here we simplify the matter by using the following form of the $\Sigma_0^B$-Bit-Recursion scheme:

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

(25)

It is straightforward that $T^0V$ can be equivalently axiomatized by $V^0$ and the above version of the $\Sigma_0^B$-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 $\pi$ requires the length of all variables that $\mathcal{S}$ depends on. We discuss the interesting atomic formulas. If $\gamma$ is a recursion variables that depends on the comprehension variables $\beta_1, \ldots, \beta_\ell$, then the values $n_\beta$ for $|\beta|$ and $n_\alpha$ for $\alpha$ (the parameter variables), we translate $\gamma(s)$ into

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

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

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

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

Next we translate $\Sigma_1^B$ formulas in $\pi$. An instance of $\Sigma_0^B$-COMP is handled as in the proof of the Translation Theorem for $\text{VTC}^C$. Consider an instance of (25) that occurs in a sequent $\mathcal{S}$ of $\pi$. Given the lengths $n_\alpha, n_\beta$ of the parameter and comprehension string variables that $\mathcal{S}$ depends on, the instance of (25) in $\mathcal{S}$ is translated into

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

where $v = \text{val}(t_\gamma(n_\alpha, n_\beta))$, and $\overline{\psi_i}$ is the translation of $\psi(i, \gamma^{<i})$. Note that $\overline{\psi_0}$ does not contain $p_0^n$, and $\overline{\psi_{i+1}}$ contains $p_0^n, \ldots, p_i^n$. The next Claim is straightforward:

**Claim** Let $\mathcal{S}$ be any sequent in $\pi$. There is a polynomial $p$ that satisfies the following condition. Given the lengths $n_\alpha$ of the parameter variables in $\pi$, and $n_\beta$ of the comprehension variables that $\mathcal{S}$ depends on. Then the translation $\overline{\mathcal{S}[n_\alpha, n_\beta]}$ has a $G_1^1$ proof of size bounded by $p(n_\alpha, n_\beta)$.

Applying the Claim for the endsequent of $\pi$ yields the desired result. \qed
4.2 Subsystems of $G^i_1$

From the above proof, a way to define subsystems of $G^i_1$ that correspond naturally to subtheories of $TV^0$ is as follows. For a two-sorted formula $\psi$, we consider closure of $\psi$ under substitution of $\Sigma^B_0$ formulas for free string variables. Call this the $\Sigma^B_0$-closure of $\psi$. Given a $\Sigma^B_0$ formula $\psi$, consider the subtheory $T$ of $TV^0$ which is axiomatized by $V^0$ and the instance (25) of $\Sigma^B_0$-Bit-Recursion. (Examples of $T$ include $VTC^1$, $VNL$ and others in [12].) We define the subsystem $G\cdot T$ of $G^i_1$:

**Definition 4.3.** A $G\cdot T$ proof is a $G^i_1$ proof where the cut formulas contain only parameter variables, and are restricted to the $\Sigma^i_1$ translation of the Bit-Recursion axioms for the $\Sigma^B_0$-closure of $\psi$.

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

**Lemma 4.4.** Every theorem of $T$ translates into a family of tautologies having polynomial size proofs in $G\cdot T$.

**References**


