Examining The Fragments of G

Steven Perron
University of Toronto
Department of Computer Science,
Toronto, Ontario, Canada
sperron@cs.toronto.edu

Abstract

When restricted to proving Σ_i^q formulas, the quantified propositional proof system G_i^* is closely related to the Σ_i^b theorems of Buss's theory S_2^i. Namely, G_i^* has polynomial-size proofs of the translations of theorems of S_2^i, and S_2^i proves that G_i^* is sound. However, little is known about G_i^* when proving more complex formulas. In this paper, we prove a witnessing theorem for G_1^* similar in style to the KPT witnessing theorem for T_2^i. This witnessing theorem is then used to show that S_2^i proves G_1^* is sound with respect to prenex Σ_{i+1}^q formulas. Note that unless the polynomial hierarchy collapses S_2^i is the weakest theory in the S_2 hierarchy for which this is true. The witnessing theorem is also used to show that G_1^* is p-equivalent to a quantified version of extended-Frege. This is followed by a proof that G_i p-simulates G_{i+1}^*. We finish by proving that S_2 can be axiomatized by S_2^1 plus axioms stating that the cut-free version of G^* is sound. All together this shows that the connection between G_i^* and S_2^i does not extend to more complex formulas.

1 Introduction

In [9], Krajicek and Pudlak introduced the quantified propositional proof system G and its fragments. These fragments have close connections with bounded arithmetic and computational complexity. In particular, the collapse of the polynomial-time hierarchy, the bounded arithmetic hierarchy S_2, and the fragments of G are all related [9, 8, 7, 10]. Even with these close connections to important open problems in logic and computer science, little work has been done investigating the fragments of G. In this paper, we take a closer look at them.

The proof system G_i^* has informally been described as the non-uniform version of S_2^i. This is often expressed by describing the close connection between the Σ_i^b theorems of S_2^i and G_i^* proofs of Σ_i^q formulas [7]. The same type of connection exists between the theory PV and the treelike version of extended-Frege [6]; and the theory T_2^i and G_i. In this paper, we compare these proof systems to each other and the theories to see how accurate these informal descriptions of the proof systems are.

Following Morioka, the proof system G_i^* is defined by restricting G to treelike proofs where all cut formulas are Σ_i^q [3, 11]. Note that originally G_i^* was defined by restricting all formulas, not just cut formulas, to Σ_i^q formulas [9, 7]. Informally, we can think of G_i^* as reasoning with lemmas that can be described as predicates in the ith level of the polynomial-time hierarchy.

First, we examine G_1^* by comparing it to extended-Frege directly. In [7], it was shown that treelike extended-Frege is p-equivalent to G_1^* with respect to quantifier-free formulas. The interesting part of this result is that G_1^* only needs to cut the extension axioms and quantifier-free formulas. This raises the question of whether or not this holds when G_1^* is used to prove more complicated formulas. We define a quantified version of extended-Frege called GPV^*, and we prove that GPV^* and G_1^* are p-equivalent with respect to all prenex formulas. This result is surprising because the class of formulas that GPV^* can cut is much less expressive than the class of formulas that G_1^* can cut. As well, this result does not fit with the view that GPV^* corresponds to PV and G_1^* with S_2^1.

We also take a look at G_i and G_{i+1}^*. If we used the connections with bounded arithmetic as a guide, we would expect G_{i+1}^* to be a strictly stronger proof system than G_i. However, in [12], Nguyen showed that this is probably not the case. This was done by showing that, under an appropriate complexity assumption, G_{i+1}^* does not simulate G_i or even cut-free G for Σ_{i+2}^q formulas. This is in contrast to a result that shows that G_{i+1}^* p-simulates G_i for Σ_{i+1}^q formulas. In this paper, we prove that, in fact, G_i is stronger than G_{i+1}^*, which is surprising. This is done by showing that G_i p-simulates G_{i+1}^* for all formulas, not just Σ_i^q formula as in [7].

Another way of examining G_1^* is to find the weakest fragment of S_2 that can prove that G_1^* is sound. So, we are looking for a theory that proves that, if there is a G_1^* proof of a formula, then that formula is valid. Informally, this gives an upper bound on the reasoning power of G_1^*. This type of question first appeared in [6], where Cook showed that PV proves that extended-Frege is sound. This kind of result played an important role in establishing the connection between the collapse of S_2 and G [9].

In [7], it was shown that S_2^1 proves that G_1^* is sound with respect to Σ_1^q formulas. However, in [11], Morioka proved that, assuming the polynomial hierarchy does not collapse, S_2^1 does not prove that G_1^* is sound with respect to Σ_3^q formulas. This does not fit with the view that G_1^* is the non-uniform version of S_2^1. In fact, it seems that, as the quantifier complexity of the formulas we are proving grows, the reasoning power of G_1^* grows beyond any finite level of the S_2 hierarchy. For the same proof also shows that, assuming the polynomial hierarchy does not collapse, T_2^i does not prove that G_1^* is sound with respect to Σ_{i+2}^q formulas; however, we show that S_2^{i+1} does. Informally this means that the reasoning power of G_1^* relative to Σ_{i+2}^q formulas is not stronger than the reasoning power of S_{i+1}^2.

This leads to the final way of examining G_1^*. In [9], Krajicek and Pudlak were able to prove that S_2 can be axiomatized by S_2^1 plus axioms stating G_i is sound relative to Σ_i^q formulas, for $i \in \mathbb{N}$. We show that the same is true when G_i is replaced by G_1^*. In fact, we can replace G_i by the cut-free version of G^*. This is interesting because it confirms that the reasoning power of G_i^* is not closely related to any finite level of S_2, but, in some sense, it captures the reasoning power of all of S_2.

The main tool used to prove some of these theorems is a witnessing theorem in the style of the KPT witnessing theorem [8]. The KPT witnessing theorem describes how hard it is to witness Σ_{i+3}^b theorems of T_2^i, for $i > 0$. It also holds for $i = 0$ with PV in place of T_2^i. This theorem has been used to prove that the collapse of the S_2 hierarchy implies the collapse of the polynomial-time hierarchy [8], and to show that certain weak theories do not prove the Σ_1^b replacement scheme, relative to some complexity assumptions [5]. In this paper, we adapt the statement of the KPT witnessing theorem to G_1^*, and then prove it. The main difficulty is that proofs of the KPT witnessing theorem rely on the cut-elimination theorem, which, unfortunately, causes the size of the proof to increase exponentially. We need to avoid this increase, so we have to find a way to work around cut formulas.

The paper is organized as follows. In Section 2, we give the basic definitions and notations. Note that we will be using two-sorted theories of bounded arithmetic (V^i) in place of the single-sorted bounded arithmetic (S_2^i). In Section 3, we state and sketch a proof of the witnessing theorem for G_1^*. In Section 4, we use the witnessing theorem to prove that GPV^* p-simulates G_1^*. In Section 5, we show that G_i p-simulates G_{i+1}^*. In Section 6, we prove the prenex-Σ_{i+1}^q reflection principle for G_1^* in S_2^i. In section 7, we give a new axiomatization of S_2.

At this point, I would like to thank the reviewers and, especially, my supervisor Stephen Cook for their useful comments on earlier versions of this paper.

2 Basic Definitions And Notation

2.1 Two-Sorted Bounded Arithmetic

In the introduction, the results were stated for the theories S_2^i. However, we will use two-sorted theories of bounded arithmetic. We follow the presentation in [2, 4]. The two sorts are numbers and binary strings (aka finite sets). The numbers are intended to range over the natural numbers and will be denoted by lower-case letters. For example, i, j, x, y, and z will often be used for number variables; r, s, and t will be used for number terms; and f, g and h will be used for functions that return numbers. The sets are intended to be finite sets of natural numbers. Since the sets are finite, they can be coded by binary strings where the ith bit is 1 if i is in the set. The strings will be denoted by upper- case letters. The letters X,Y, and Z will often be used for string variables; and F, G and H will be used for functions that return strings.

The base language is

$$\mathcal{L}_A^2 = \{0, 1, +, \times, <, =, =_2, \in, ||\} .$$

The constants 0 and 1 are number constants. The functions $+$ and $\times$ take two numbers as input and return a number–the intended meanings are the obvious ones. The language also includes two binary predicates that take two numbers: $<$ and $=$. The predicate $=_2$ is meant to be equality between strings, instead of numbers. In practice, the 2 will not be written because which equality is meant is obvious from the context. The membership predicate $\in$ takes a number i and a string X. It is meant to be true if the ith bit of X is 1 (or i is in the set X). This will also be written as $X(i)$. The final function $|X|$ takes a string as input and returns a number. It is intended to be the number of bits needed to write X when leading zeros are removed (or the least upper bound of the set X). The set of axioms 2BASIC is the set of defining axioms for $\mathcal{L}_A^2$.

We use $\exists X < b\ \phi$ as shorthand for $\exists X[(|X| < b) \wedge \phi]$. The shorthand $\forall X < b\ \phi$ means $\forall X[(|X| < b) \supset \phi]$. The set $\Sigma_0^B = \Pi_0^B$ is the set of formulas whose only quantifiers are bounded number quantifiers. For $i > 0$, the set Σ_i^B is the set of formulas of the form $\exists \vec{X} < \vec{t}\phi$ where ϕ is a Π_{i-1}^B formula. For $i > 0$, the set Π_i^B is the set of formulas of the form $\forall \vec{X} < \vec{t}\phi$ where ϕ is a Σ_{i-1}^B formula.

Now we can define the two main axiom schemes:

Σ_i^B-comp:
$\exists X \leq b \forall i < b[X(i) \leftrightarrow \phi(i)]$,

Σ_i^B-string-ind:
$[\phi(\emptyset) \wedge \forall X[\phi(X) \supset \phi(S(X))]] \supset \phi(Y)$

where $\phi(i)$ is a Σ_i^B formula, and, for Σ_i^B-COMP, ϕ does not contain X, but may contain other free variables. The constant $\emptyset$ is the empty string, and the function $S(X)$ interprets X as a binary number and adds 1 to it. Note that we still view Σ_i^B-string-ind as a formula over $\mathcal{L}_A^2$. We simply replace the instances of $\emptyset$ and $S(X)$ by their Σ_0^B bit-definition.

We can now define two hierarchies of theories.

Definition 2.1. The theory $\mathbf{V^i}$ is axiomatized by the 2BASIC axioms plus Σ_i^B-comp. The theory $\mathbf{TV^i}$ is axiomatized by the 2BASIC axioms, Σ_0^B-comp, and Σ_i^B-string-ind.

For $i > 0$, V^i corresponds to S_2^i, and TV^i corresponds to T_2^i in that they are RSUV-isomorphic [4].

Another theory we often use is VPV, a universal theory with a function symbol for every polynomial time function. The function symbols have defining axioms based on Cobham's Theorem. These function symbols are known as the PV functions. The theory VPV is axiomatized by quantifier-free equivalents of the 2BASIC axioms plus induction on all open $\Sigma_0^B(PV)$ formulas. See [2, 4] for more information on VPV.

Another scheme of formulas we use is the Σ_i^B-MAX scheme:

$$\exists x < b\phi(x) \supset$$
$$\exists x < b[\phi(x) \wedge \forall y < b(x < y \supset \neg\phi(y))]$$

where ϕ is Σ_i^B. This scheme essentially says that, if there exists a value for x less than b that satisfies $\phi(x)$, then there exists a maximum x less than b that satisfies $\phi(x)$. It can be shown that Σ_i^B-MAX is provable in V^i ([4], Corollary 5.8).

From time to time, we will use functions symbols that are not in $\mathcal{L}_A^2$. The first is $X(i,j) \equiv X(< i, j >)$, where $< i, j >= (i+j)(i+j+1) + 2j$ is the pairing function. We index a string by two (or more) numbers instead of one. Even though this is not officially part of the language, it can be thought of as a two dimensional array of bits. The second is the row function. The notation we use is $X^{[i]}$. This functions returns the ith row of the two dimensional array X. Note that, if we add these functions with their Σ_0^B defining axioms to the theory V^i, we get a conservative extension. They can also be used in the induction axioms [2]. This means we can freely use the functions.

2.2 Quantified Propositional Calculus

We are also interested in quantified propositional proof systems. The proof systems we use were originally defined in [9]. They were redefined in [3, 11], which is the presentation we follow.

The set of connectives we use are $\{\wedge, \vee, \neg, \exists, \forall, \top, \bot\}$. Formulas are build using these connectives in the usual way. We will often refer to formulas by the number of quantifier alternations.

Definition 2.2. The set of formulas $\Sigma_0^q = \Pi_0^q$ is the set of quantifier-free propositional formulas. For $i > 0$, the set of Σ_i^q (Π_i^q) formulas is the smallest set of formulas that contains Π_{i-1}^q (Σ_{i-1}^q) and is closed under $\wedge$, $\vee$, existential (universal) quantification, and if $A \in \Pi_i^q$ ($A \in \Sigma_i^q$) then $\neg A \in \Sigma_i^q$ ($\neg A \in \Pi_i^q$).

The first proof system, from which all others will be defined, is the proof system G. This proof system is a sequent calculus based on Gentzen's system LK. The system G is essentially the DAG-like, propositional version of LK. We will not give all of the rules, but will mention a few of special interest.

The cut rule is

$$\text{cut}\ \frac{A, \Gamma \rightarrow \Delta \qquad \Gamma \rightarrow \Delta, A}{\Gamma \rightarrow \Delta}$$

In this rule, we call A the cut formula. There are also four rules that introduce quantifiers:

$$\exists\text{-left}\ \frac{A(x), \Gamma \rightarrow \Delta}{\exists z A(z), \Gamma \rightarrow \Delta} \qquad \exists\text{-right}\ \frac{\Gamma \rightarrow \Delta, A(B)}{\Gamma \rightarrow \Delta, \exists z A(z)}$$

$$\forall\text{-left}\ \frac{\Gamma \rightarrow \Delta, A(x)}{\Gamma \rightarrow \Delta, \forall z A(z)} \qquad \forall\text{-right}\ \frac{A(B), \Gamma \rightarrow \Delta}{\forall z A(z), \Gamma \rightarrow \Delta}$$

These rules have conditions on them. In $\exists$-left and $\forall$-right, the variable x must not appear in the bottom sequent. In these rules, x is called the eigenvariable. In the other two rules, the formula B must be a Σ_0^q formula, and no variable that appears free in B can be bound in $A(x)$.

The initial sequents of G are sequents of the form $\rightarrow \top$, $\bot \rightarrow$, or $x \rightarrow x$, where x is any propositional variable. A G proof is a series of sequents such that each sequent is either an initial sequent or can be derived from previous sequents using one of the rules of inference. The proof system G_i is G with cut formulas restricted to Σ_i^q formulas.

We define G^* as the treelike version of G. So, a G^* proof is a G proof where each sequent in used as an upper sequent in an inference at most once. A G_i^* proof is a G^* proof in which cut formulas are prenex Σ_i^q. In [11], it was shown that, for treelike proofs, it did not matter if the cut formulas in G_i^* were prenex or not. So when we construct G_i^* proofs the cut formulas will not always be prenex, but that does not matter.

To make proofs simpler, we assume that all treelike proofs are in *free-variable normal form*.

Definition 2.3. A parameter variable for a G_i^* proof π is a variable that appears free in the final sequent of π. A proof π is in *free-variable normal form* if (1) every non-parameter variable is used as an eigenvariable exactly once in π, and (2) parameter variables are not used as eigenvariables.

Note that, if a proof is treelike, we can always put it in free-variable normal form by simply renaming variables.

2.3 Truth Definitions

In order to reason about the proof systems in the theories, we must be able to reason about quantified propositional formulas. Due to space considerations, we will introduce the notation and informal definitions, but not the formal definitions. We follow the presentation in [7, 9]. This is an abuse of notation, but we will not distinguish between a formula and its string encoding. If we let F be a Σ_i^q formula and let A be an assignment to the free variables of F, then

$$(A \models_i F) \equiv \text{“}A \text{ is a satisfying assignment for } F\text{”}.$$

It is important that, for $i > 0$, $A \models_i F$ has a Σ_i^B definition, and, for $i = 0$, it has a $\Sigma_0^B(PV)$ definition. If F is a Π_i^q formula, we will use the same notation for satisfaction, but in this case the definition is Π_i^B.

Given a formula $F \equiv \bigwedge_{i=0}^n F_i$, there is a PV function $Parse_\wedge(F, j)$ that outputs $F_{min(j,n)}$. The same goes for $\vee$ in place of $\wedge$. The theory VPV proves the Tarski conditions for the truth definition:

- $(A \models_i F) \leftrightarrow (\forall j \le |F|\ A \models_i Parse_\wedge(F, j))$ (where $F \equiv \bigwedge\limits_{j=0}^{n} F_j$)
- $(A \models_i F) \leftrightarrow (\exists j \le |F|\ A \models_i Parse_\vee(F, j))$ (where $F \equiv \bigvee\limits_{j=0}^{n} F_j$)
- $(A \models_i \neg F) \leftrightarrow (A \not\models_i F)$
- $(A \models_i \exists \vec{x} F(\vec{x})) \leftrightarrow \exists X(A \cup X \models_i F(\vec{x}))$ (for $F \in \Sigma_i^q$)
- $(A \models_i \forall \vec{x} F(\vec{x})) \leftrightarrow \forall X(A \cup X \models_i F(\vec{x}))$ (for $F \in \Pi_i^q$)
- $(A \models_i F) \leftrightarrow (A \models_{i-1} F)$ (for $F \in \Sigma_{i-1}^q \cup \Pi_{i-1}^q$).

Valid formulas (or tautologies) are defined as

$$TAUT_i(F) \equiv \forall A,$$
$$(\text{“A is an assignment to the variables of } F\text{”} \supset A \models_i F)$$

This truth definition can be extended to define the truth of a sequent. So, if $\Gamma \to \Delta$ is a sequent of $\Sigma_i^q \cup \Pi_i^q$ formulas, then

$$\begin{aligned}&(A \models_i\ \Gamma \to \Delta) \equiv \\ &\quad \text{“there exists a formula in } \Gamma \text{ that } A \text{ does not satisfy”} \\ &\quad \vee \text{ “there exists a formula in } \Delta \text{ that } A \text{ satisfies”}\end{aligned}$$

Another important formula we will use is the reflection principle for a proof system. We define the Σ_i^q reflection principle for a proof system P as

$$\begin{aligned}&\Sigma_i^q\text{-RFN}(P) \equiv \forall F \forall \pi, \\ &\quad (\text{“}\pi \text{ is a } P \text{ proof of } F\text{”} \wedge F \in \Sigma_i^q) \supset TAUT_i(F)\end{aligned}$$

This formula essentially says that, if there exists a P proof of a Σ_i^q formula F, then F is valid. Another way of putting it is to say that P is sound when proving Σ_i^q formulas. In this paper, we will sometimes replace Σ_i^q with prenex Σ_i^q formulas.

3 KPT Witnessing for G_1^*

In bounded arithmetic, a useful tool has been the KPT witnessing theorem [8]. In the simplest case, the KPT witnessing theorem describes how to witness the Σ_2^B theorems of VPV. The original theorem was more general, but we state it here for the simplest case.

Theorem 3.1 (KPT Witnessing [8])**.** *Suppose $VPV \vdash \forall X \exists Y \forall Z \phi(X, Y, Z)$, where ϕ is a Σ_0^B formula. Then there exists a finite sequence of PV function symbols $F_1, F_2, \ldots, F_k$ such that*

$$\begin{aligned}VPV \vdash \forall X \forall W\ \ &\phi(X, F_1(X), W^{[1]}) \\ &\vee \phi(X, F_2(X, W^{[1]}), W^{[2]}) \\ &\quad \vdots \\ &\vee \phi(X, F_k(X, W^{[1]}, W^{[2]}, \ldots, W^{[k-1]}), W^{[k]})\end{aligned}$$

Informally, this can be viewed as an interactive computation between a student, who runs in polynomial time, and an all-knowing teacher. Given a value for X, the student's goal is the find a witness for $\exists Y \forall Z \phi(X, Y, Z)$. The student starts by computing $F_1(X)$. If that is not a witness, the teacher responds with a counter example $W^{[1]}$. Using that the students makes a second guess by computing F_2. The teacher responds with $W^{[2]}$, and this process continues.

Our goal is to get a similar theorem for G_1^*. The first obstacle comes in the statement of the theorem. The theory VPV has access to function symbols that correspond to the polynomial-time functions, but, in G_1^*, there are no function symbols. To fix this, we use the idea of an extension cedent from [4].

Definition 3.2. An *extension cedent* is a series of formulas of the form

$$e_1 \leftrightarrow E_1, e_2 \leftrightarrow E_2, \ldots, e_n \leftrightarrow E_n$$

such that E_i is a Σ_0^q formula that does not mention the variables $e_i, e_{i+1}, \ldots, e_n$. We say that e_i depends on a variable q if E_i mentions q or E_i mentions a variable that depends on q.

Observe that an extension cedent is really a description of a circuit, and that polynomial-size circuits are the nonuniform version of polynomial-time functions. So extension cedents replace the functions.

Theorem 3.3 (KPT Witnessing for G_1^*). *There exists a PV (polynomial-time) function F such that VPV proves the following. Let π be a G_1^* proof of a prenex Σ_2^q formula $A(\vec{p}) \equiv \exists\vec{x}\forall\vec{y}B(\vec{x},\vec{y},\vec{p})$, where $B(\vec{x},\vec{y},\vec{p})$ is a Σ_0^q formula with all free variables shown. Then, given π, F outputs a G_0^* proof of a sequent $\Lambda \rightarrow \Theta$ where*

1. *Θ is a series of formulas of the form $B(\vec{C}^i, \vec{q}^i, \vec{p})$,*
2. *Λ is an extension cedent defining a new set of variables E in terms of $\vec{q}^1, \ldots, \vec{q}^n$ and $\vec{p}$,*
3. *$\vec{C}^i$ are Σ_0^q formulas that do not mention $\vec{q}^j$ for $j \geq i$,*
4. *$\vec{C}^i$ does not mention any variable in E that depends on a variable in $\vec{q}^j$ for $j \geq i$.*

Before we prove this theorem, notice that this is similar to the KPT Witnessing theorem for VPV. The row $W^{[i]}$ corresponds to $\vec{q}^i$, and F_i corresponds to $\vec{C}^i$. The major difference is that the number of functions is not constant; it can grow polynomially in the size of the proof.

One way of proving the KPT Witnessing Theorem is to observe that it is a corollary to the Herbrand Theorem. So the idea behind our proof is to adjust the proof-theoretic proof of the Herbrand Theorem. See [1] for an outline of this proof. The main difference between our proof and that proof is that cut elimination cannot be used since it causes an exponential increase in the size of the proof. To get around this problem, we use the idea used in [4] to prove that extended-Frege p-simulates G_1^*. This idea is to turn the Σ_1^q cut formulas into Σ_0^q cut formulas by witnessing the existential quantifiers with extension variables.

We prove the Herbrand Theorem for all Σ_i^q formulas, but before we can state the general theorem, we need a few definitions. The first one has more to do with notation. The q variables come from the eigenvariables in the G_1^* proof. To make it easier to refer to these variables, we use the following notation:

Notation 3.4. Let π be a G^* proof. Then the set Q_π will be the set of variables that are used as eigenvariables in π. If S is a sequent in π, then $Q_{\pi,S}$ will be the set of variables that are used as eigenvariables in the subproof of π ending with S. We will refer to $Q_{\pi,S}$ as Q_S when π is understood.

Note that π is treelike, and, if it is in free-variable normal form and S is derived from S_1 and S_2, then $Q_S = Q_{S_1} \cup Q_{S_2}$, and $Q_{S_1} \cap Q_{S_2} = \emptyset$.

The general witnessing theorem will be for G_1^* proofs of any prenex Σ_i^q formula A. In the end, we want a G_0^* proof of a sequent $\Lambda \rightarrow \Theta$, where Θ is a series of instances as defined as follows.

Definition 3.5. Let A be the formula

$$\forall\vec{y}^0\exists\vec{x}^1\forall\vec{y}^1 \ldots \exists\vec{x}^n\forall\vec{y}^n B(\vec{y}^0, \vec{x}^1, \vec{y}^1, \ldots, \vec{x}^n, \vec{y}^n),$$

where B is a Σ_0^q formula with all free variables shown. An *instance* of A is a Σ_0^q formula obtained by

1. replacing each universal variable y_j^i by a variable $q_j^i \in Q$,
2. replacing each existential variable x_j^i by a Σ_0^q formula C_j^i, and
3. removing the quantifiers.

A *partial instance* of A is the same as an instance of A except only an initial segment of the quantified variables are replaced.

Observe that in Theorem 3.3, there is an ordering on the variables. Namely the variables $\vec{q}^i$ come before the variables $\vec{q}^{i+1}$. We could also extend this ordering to include the extension variables. An extension variable would have to be larger than every variable it depends on. Then the formulas $\vec{C}^i$ can only mention variables smaller than $\vec{q}^i$. For the general case, we want something similar. To make the proof simpler, we will use $\prec$ to refer to this ordering. The ordering $\prec$ orders the eigenvariables Q and the extension variables E. Then Θ will be more than a series of instances; it will be a series of instances relative to $\prec$.

Definition 3.6. Let $\prec$ be a partial ordering of the variables $Q \cup E$, and A be a formula as in Definition 3.5. Let

$$B' \equiv B(\vec{q}^0, \vec{C}^1, \vec{q}^1, \ldots, \vec{C}^n, \vec{q}^n)$$

be an instance of A. Then B' is an instance of A relative to $\prec$ if

1. for $i < i'$, $q^{i'}_{j'} \not\prec q^i_j$

2. for $j < j'$, $q^i_{j'} \not\prec q^i_j$

3. $\vec{C}^i$ does not mention q^i_1 or any variable $v \in Q \cup E$ such that $q^i_1 \prec v$

The first two points in this definition essentially say that $\prec$ preserves the quantifier order, where the outermost quantifiers are smaller. The third point sets $\vec{q}^i$ as the upper bound on the variables that $\vec{C}^i$ can mention. The idea of defining an instance comes from [1] and a conversation with Stephen Cook.

The last definition we need before we state the general theorem was something we were able to avoid in the simple case. It is possible that the same variable $q \in Q$ is substituted for a universal variable in two different instances of Θ. When this happens, we must be sure it happens consistently. The variable q must replace the same universal variable y^i_j in both instances, and the instances must be i, j-contractable.

Definition 3.7. We say two instances of A are i, j-contractable if y^i_j and all quantified variables to the left of y^i_j are replaced by the same formula or variable in both instances.

The idea behind the name is that these instances would have been contracted in π before the $\forall y^i_j$ quantifier was added.

Now we are prepared to state the general theorem.

Theorem 3.8 (Main Theorem)**.** *There exists a PV function F such that VPV proves the following. Let π be a G^*_1 proof of A in free-variable normal form, where*

$$A \equiv \forall \vec{y}^0 \exists \vec{x}^1 \forall \vec{y}^1 \ldots \exists \vec{x}^n \forall \vec{y}^n B(\vec{y}^0, \vec{x}^1, \vec{y}^1, \ldots, \vec{x}^n, \vec{y}^n),$$

and B is a Σ^q_0 formula with all free variables shown. Then, given π, F outputs a G^*_0 proof of a sequent $\Lambda \rightarrow \Theta$ and a total ordering $\prec$ of the variables $Q_\pi \cup E$, where E is a set of variables that do not appear in π, with the following properties:

- *Λ is an extension cedent defining the variables in E;*
- *for $e \in E$, if e depends on a variable $p \in Q_\pi \cup E$, then $p \prec e$;*
- *Θ is a series of instances of A, relative to $\prec$, and*
- *if two of the instances in Θ use $q \in Q_\pi$ to replace a universal variable, then those two instances are i, j-contractable, where q replaced y^i_j in the instances.*

Proof. The G^*_0 proof that we are looking for will be constructed by changing π one sequent at a time starting with the initial sequents and working our way down. To simplify the construction, we will ignore the order of the formulas in the sequents. So a sequent is a pair of multisets. One set for the left side of the sequent, and one set for the right side.

Let S be any sequent in π. By the subformula property of G^*_1, S is of the form

$$\Gamma \rightarrow \Delta, \Omega,$$

where Γ and Δ are possibly empty sets of Σ^q_1 formulas and Ω is a possibly empty set of formulas that are partial instances of A. We want to define a PV function that outputs a G^*_0 proof of a sequent

$$S' \equiv \Lambda, \Gamma' \rightarrow \Delta', \Theta,$$

and a total ordering $\prec$ on $Q_S \cup E$ where

1. Γ' is obtained from Γ by replacing each formula $\exists \vec{z} D(\vec{z})$ by $D(\vec{q})$, where D is Σ^q_0 and $\vec{q} \in Q_S$. (We use different $\vec{q}$ for different formulas.)

2. Δ' is obtained from Δ be replacing each formula $\exists \vec{z} D(\vec{z})$ by $D(\vec{e})$, where D is Σ^q_0 and $\vec{e} \in E$. (We use different $\vec{e}$ for different formulas.)

3. Λ is an extension cedent defining E;

4. for $e \in E$, if e depends on a variable $p \in Q_S \cup E$, then $p \prec e$;

5. Θ is a set of instances of A, relative to $\prec$, and

6. if two of the instances in Θ use $q \in Q_S$ to replace a universal variable, then those two instances are i, j-contractable, where q replaced y^i_j in the instances.

Note that $\prec$ is only defined on the extension variables and eigenvariables used so far. Initially, $\prec$ is an ordering where nothing is comparable. As we move down the proof, we order the variables.

The proof is done by induction on the depth of S in the proof π. If we let S be the final sequent, we get a proof of the theorem since $Q_\pi = Q_S$, and conditions 3-6 are the conditions we need for the theorem. Also, note that the induction hypothesis can be stated as a $\Sigma^B_0(PV)$ formula (is a polynomial predicate) by saying that the output of the function F on the first i sequents of π meets all of the conditions. This means the induction can be carried out in VPV.

The function F is described and proved correct by induction. There is a separate construction for each rule of inference. For the sake of space, we will not give many details. The construction in most cases is done the same way it is in the proof that extended-Frege p-simulates G^*_1 (Theorem 7.48 of [4]). The difference is that the variables need to be ordered. As new extension variables come along, they are made the largest variables so far. As new eigenvariables come along, they become the smallest variables so far. When cutting, the extension variables and eigenvariables from the subproof with the cut formula on the right become larger than the variables from the other subproof. This gives an idea of how the main cases are handled. $\square$

4 GPV^* and G_1^*

We now move on to applications of the main theorem. The first application deals with a seemingly weaker proof system.

Definition 4.1. The proof system GPV^* is G^* where cut formulas are restricted to Σ_0^q formulas or formulas of the form $\exists x[x \leftrightarrow A]$, where A is a Σ_0^q formula that does not mention x.

At first glance, it seems like GPV^* would be a weaker proof system than G_1^* because the cut formulas are less expressive. The cut formulas in GPV^* can be trivially witnessed, but the cut formulas in G_1^* are NP-hard. Nevertheless, it can be shown that GPV^* and G_1^* are p-equivalent for prenex formulas. One direction is easy since every GPV^* proof is a G_1^* proof, so all that is left is to prove the other direction.

Theorem 4.2. *VPV proves that GPV^* p-simulates G_1^* for prenex formulas.*

Proof. Let π be a G_1^* proof of a formula A of the form

$$\forall \vec{y}^0 \exists \vec{x}^1 \forall \vec{y}^1 \ldots \exists \vec{x}^n \forall \vec{y}^n B(\vec{y}^0, \vec{x}^1, \vec{y}^1, \ldots, \vec{x}^n, \vec{y}^n).$$

By the main theorem (Theorem 3.8), VPV proves that there exists a G_0^* proof π' of a sequent $\Lambda \rightarrow \Theta$ and a total ordering $\prec$ of the variables $Q_\pi \cup E$ meeting the conditions of the theorem.

We describe an algorithm that takes as input π' and $\prec$. The algorithm extends π' into a GPV^* proof of A. At any stage, π' will be a proof of a sequent $\Lambda' \rightarrow \Theta'$, where Λ' is a subsequence of Λ and Θ' is a series of partial instances of A relative to $\prec$. The algorithm has four steps:

Step 1: Add as many existential quantifiers to the formulas in Θ' as possible using $\exists$-right rules.
Step 2: Use contraction to combine as many formulas in Θ' as possible.
Step 3: Find the largest variable that is mentioned in Λ' or Θ'.
Step 3a: If it is an extension variable e, apply $\exists$-left to the formula $e \leftrightarrow E$ with e as the eigenvariable. Then cut the formula $\exists e[e \leftrightarrow E]$ after deriving $\rightarrow \exists e[e \leftrightarrow E]$.
Step 3b: If it was an eigenvariable q in π, then apply $\forall$-right with q as the eigenvariable.
Step 4: Repeat steps 1 to 3 until there is no change.

At first, it may not be obvious that this algorithm works. For example, it is not obvious that the eigenvariable restriction for $\exists$-left or $\forall$-right rules in Step 3 is met. To show that the eigenvariable restriction is met, we make two observations. First, if p is the largest variable in Λ' and Θ', then no extension variable can depend on p. Otherwise, that variable would be larger than p. Second, if we are in Step 3 and p is the largest variable in Λ' and Θ', then p cannot be mentioned in any formula C that was used to replace an existential variable in a partial instance of A. This is because C would have been used as the target formula in an $\exists$-right rule in Step 1. Otherwise, an eigenvariable that appears to the right of C is still present, and this variable must be larger than p. For the same reason, we know that there cannot be two partial instances with p replacing a universal variable that have not been contracted yet. This means the eigenvariable restriction is met in Step 3.

When the algorithm is done, we will have a proof of the formula we want. Notice that Λ' would be empty because every extension variable has been removed. Also, Θ' would be the single formula A since every instance in Θ would have every quantifier added by now, and every instance would have been contracted to a single formula. We know the algorithm eventually stops because we continually reduce the number of variables in π'. $\square$

5 G_i and G_{i+1}^*

As has already been mentioned, G_i is commonly connected with the theory TV^i and G_{i+1}^* is commonly connected with V^{i+1}. Since the two theories have the same Σ_{i+1}^B theorems, it was natural that the two proof systems are p-equivalent when proving Σ_{i+1}^q formulas. However, we want to extend this to more general formulas. In [12], it was shown that one direction is probably not possible. Namely that, under an appropriate complexity assumption, there is a family of Σ_2^q formulas for which G_{i+1}^* does not p-simulate G_i. Here we prove that G_i p-simulates G_{i+1}^* for all formulas.

The proof is based on the proof of Krajicek that depth d, DAG-like PK can p-simulate depth $d+1$, tree-like PK. The observation of the similarity between the two theorems is due to Toni Pitassi.

Definition 5.1 (The i-Substitution Rule)**.** The i-substitution rule is

$$\frac{A_1(p), \ldots, A_m(p), \Gamma \to \Delta, B_1(p), \ldots, B_n(p)}{A_1(C), \ldots, A_m(C), \Gamma \to \Delta, B_1(C), \ldots, B_n(C)}$$

where C is a quantifier-free formula, $A_1, \ldots, A_m, B_1, \ldots, B_n$ are $\Sigma_i^q \cup \Pi_i^q$ formulas, and p does not appear in the bottom sequent.

Lemma 5.2. *G_i^* p-simulates the i-substitution rule.*

Proof. We will describe how to do the simulation for the case where there is one A and B. The general case is done the same way.

Suppose we have a derivation of

$$A(p), \Gamma \to \Delta, B(p). \tag{5.1}$$

We want to derive

$$A(C), \Gamma \to \Delta, B(C).$$

First we derive

$$p \leftrightarrow C, A(C) \to A(p),$$

and cut this with (5.1), where $A(p)$ is the cut formula. This gives

$$p \leftrightarrow C, A(C), \Gamma \to \Delta, B(p). \tag{5.2}$$

Then we derive

$$p \leftrightarrow C, B(p) \to B(C),$$

and cut this with (5.2), where $B(p)$ is the cut formula. This gives

$$p \leftrightarrow C, A(C), \Gamma \to \Delta, B(C). \tag{5.3}$$

We then apply $\exists$-left to this sequent with p as the eigenvariable, and then cut $\exists p[p \leftrightarrow C]$ after deriving $\to \exists p[p \leftrightarrow C]$. $\square$

Theorem 5.3. *G_i p-simulates G_{i+1}^*.*

Proof. Let π be a G_{i+1}^* proof. The reason π is not a G_i proof is that it would contain cut formulas that are not Σ_i^q or Π_i^q. We can assume these formulas are Σ_{i+1}^q and are of the form

$$\exists x_1 \ldots \exists x_n C(x_1, \ldots, x_n).$$

We need to turn these cut formulas into Π_i^q cut formulas. To do this, we change all of the non-($\Sigma_i^q \cup \Pi_i^q$) formulas that are ancestors of these cut formulas. They are of the form

$$\exists x_l \ldots \exists x_n C(D_1, \ldots, D_{l-1}, x_l, \ldots, x_n), \tag{5.4}$$

where D_j is a Σ_0^q formula for $j < l$, and $C(\vec{x})$ is a Π_i^q formula. Note that, if this formula is on the left side of a sequent, then the formula D_i will actually be variables that eventually get used as eigenvariables. From now on, we will assume all formulas of the form (5.4) are ancestors of cut formulas. Those that are not are simply ignored.

For each sequent $S \equiv \Gamma \to \Delta$ in π, we give a G_i proof π' of a sequent $S' \equiv \Gamma' \to \Delta'$ where

1. Γ' is obtain from Γ by replacing every formula of the form (5.4) by $C(D_1, \ldots, D_{l-1}, x_l^C, \ldots, x_n^C)$,
2. Δ' is obtained from Δ by removing every formula of the form (5.4),
3. the sequent

   $$C(D_1, \ldots, D_{l-1}, x_l^C, \ldots, x_n^C) \to$$

   can be used as an axiom if and only if Δ contains a formula of the form (5.4).

For example, if S is the sequent

$$\exists x_2, x_3 C_1(q_1, x_2, x_3), \Gamma \to \Delta, \exists x_3, x_4 C_2(D_1, D_2, x_3, x_4),$$

S' would be

$$C_1(q_1, x_2^{C_1}, x_3^{C_1}), \Gamma \to \Delta,$$

and when we prove S', we are allowed to use

$$C_2(D_1, D_2, x_3^{C_2}, x_4^{C_2}) \to$$

as an axiom. In essence, we are saying, if we can derive

$$C_2(D_1, D_2, x_3^{C_2}, x_4^{C_2}) \to,$$

we can prove S'. Note that, when we get to the final sequent, no formula is an ancestor of a cut formula. Therefore, if S is the final formula in π, $S' = S$ and the only initial sequents are of the form $x \to x$. So this will give us a proof of the theorem.

The construction of π' is given inductively. There is a separate case for each rule of inference. Most cases are simple and are left to the reader. The only cases we will give are cut, $\exists$-left, and $\exists$-right.

Cut: Suppose $S \equiv \Gamma \to \Delta$ is derived from S_1 and S_2 using cut. Let the cut formula be $\exists \vec{x} C(\vec{x})$. By induction with S_1, we have a G_i proof π_1' of

$$S_1' \equiv C(\vec{x}^C), \Gamma' \to \Delta'.$$

By induction with S_2, we have a G_i proof π_2' of $\Gamma' \to \Delta'$ using the axiom $C(\vec{x}^C) \to$. Notice that π_2' is a proof of the sequent we want, but it uses an axiom we are no longer able to use. However, π_1' gives us a derivation of this axiom, with a few extra formulas.

The first step in the construction of π' is to add Γ' to the left and Δ' to the right of every sequent in π_2'. This makes the axiom we want to remove $C_i(\vec{x}^i), \Gamma' \to \Delta'$, which is the final sequent π_1'. So, π' is π_1' followed by the new π_2'. Note that the axiom could have been used multiple times; however, since we are constructing a DAG-like proof, we do not need to repeat π_1' multiple times. This gives a proof of $\Gamma', \Gamma' \to \Delta', \Delta'$, from which we can derive $\Gamma' \to \Delta'$ using contraction.

$\exists$-left: Suppose S is

$$\exists x_j \ldots \exists x_n C(q_1, \ldots, q_{j-1}, \mathbf{x_j}, x_{j+1}, \ldots, x_n), \Gamma \to \Delta,$$

and it was derived from S_1

$$\exists x_{j+1} \ldots \exists x_n C(q_1, \ldots, q_{j-1}, \mathbf{q_j}, x_{j+1}, \ldots, x_n), \Gamma \to \Delta.$$

By induction with S_1, we get a G_i proof of

$$C(q_1, \ldots, q_{j-1}, \mathbf{q_j}, x^C_{j+1}, \ldots, x^C_n), \Gamma' \to \Delta'.$$

Since q_j was used as an eigenvariable, it only appears in that one formula. Therefore we can substitute q_j by x^C_j using the i-substitution rule. This gives us π'.

$\exists$-right: Suppose S is

$$\Gamma \to \Delta, \exists x_j \ldots \exists x_n C(D_1, \ldots, D_{j-1}, \mathbf{x_j}, x_{j+1}, \ldots, x_n),$$

and it was derived from S_1

$$\Gamma \to \Delta, \exists x_{j+1} \ldots \exists x_n C(D_1, \ldots, D_{j-1}, \mathbf{D_j}, x_{j+1}, \ldots, x_n).$$

First assume $j < n$. That is we had at least one quantifier already. By induction with S_1, we get a G_i proof of $\Gamma' \to \Delta'$ using the axiom

$$C(\ldots, \mathbf{D_j}, \ldots) \to . \tag{5.5}$$

We cannot use this axiom anymore. Instead, we use the axiom

$$C(\ldots, \mathbf{x^C_j}, \ldots) \to$$

to derive (5.5) using the i-substitution rule.

If $j = n$, the construction is a little different. By induction with S_1, we get a G_i proof of

$$\Gamma' \to \Delta', C(\ldots, D_{n-1}, D_n). \tag{5.6}$$

To construct π', we take the axiom we can now use,

$$C(\ldots, D_{n-1}, x^C_n) \to,$$

and derive

$$C(\ldots, D_{n-1}, D_n) \to$$

using the i-substitution rule. Then we cut with (5.6). $\square$

6 G_1^* Reflection Principles

We can also use the main theorem to prove reflection principles. Proving reflection principles is the standard method of assessing the strength of a proof system relative to a theory. For example, the Σ^q_1 reasoning of G_1^* is not stronger than the Σ^B_1 reasoning of V^1 because V^1 proves Σ^q_1-RFN(G_1^*) [7]. Our goal is to find the weakest fragment of V that proves Σ^q_i-RFN(G_1^*). In [11], it was shown that TV^0 does not prove Σ^q_2-RFN(G_1^*) unless the polynomial-time hierarchy collapses. Using the same ideas, it is possible to show that TV^i does not prove Σ^q_{i+2}-RFN(G_1^*), for $i \geq 0$, unless the polynomial-time hierarchy collapses. This still leaves open whether or not V^i proves Σ^q_{i+1}-RFN(G_1^*) for $i \geq 1$. We do not resolve this problem completely, but we do take a big step. We prove that V^i proves (prenex Σ^q_{i+1})-RFN(G_1^*).

We first prove the base case. Namely, that V^1 proves (prenex Σ^q_2)-RFN(G_1^*). The proof serves as a template for the general case, which we prove right after.

Theorem 6.1. *V^1 proves (prenex Σ^q_2)-RNF(G_1^*).*

Proof. Let π be a G_1^* proof of a prenex Σ^q_2 formula A. So A is of the form

$$\exists \vec{x} \forall \vec{y} B(\vec{x}, \vec{y}, \vec{p}),$$

where B is a Σ^q_0 formula. In this formula, $\vec{p}$ is all of the free variables in A, and should be understood as being implicitly universally quantified. We want to prove in V^1 that, given values for $\vec{p}$, there exists values for $\vec{x}$ that witness the formula.

By the KTP witnessing theorem for G_1^* (Theorem 3.3), V^1 proves that there is a G_0^* proof of a sequent

$$S \equiv \Lambda \to \Theta,$$

meeting the conditions of the theorem.

Let

$$\begin{aligned}
&\psi(m, \Lambda, \Theta, P) \equiv \\
&\exists E\ \exists Q \text{ “E is a truth assignment to the extension variables”} \\
&\qquad \wedge \text{ “Q is a truth assignment to the eigenvariables”} \\
&\qquad \wedge \forall i < m\ (P \cup E \cup Q) \models_0 \neg B(\vec{C}^i, \vec{q}^i, \vec{p}) \\
&\qquad \wedge (P \cup E \cup Q) \models_0 \Lambda
\end{aligned}$$

This formula says that there exists assignments E and Q that satisfy Λ and make to first m formulas in Θ false. It is easy to bound the size of E and Q. This means that ψ is equivalent to a Σ^B_1 formula.

Using Σ^B_1-MAX, we find the maximum value m_0 for m that satisfies ψ given values for Λ, Θ, and P. Then $\vec{C}^{m_0+1}$ are the witnesses we are looking for.

If that were not the case, we could find values for $\vec{q}^{m_0+1}$ that would falsify $B(\vec{C}^{m_0+1}, \vec{q}^{m_0+1}, \vec{p})$. Since $\vec{C}^i$ does not mention $\vec{q}^{m_0+1}$, it follows, with a little work, that we could falsify the first $m_0 + 1$ instances. However, that violates the choice of m_0.

Also note that $\vec{C}^{m_0+1}$ exists since it is not possible to falsify all of the instances. $\square$

Theorem 6.2. *V^i proves (prenex Σ^q_{i+1})-RFN(G_1^*).*

Proof. Given a G_1^* proof of a Σ_{i+1}^q formula, we use the main theorem (Theorem 3.8) to get a G_0^* proof of $\Lambda \to \Theta$ and an ordering $\prec$ meeting the conditions of that theorem. Let

$$B(\vec{C}^1, \vec{q}^1, \ldots, \vec{C}^n, \vec{q}^n)$$

be one of the instances in Θ. Then $\vec{q}^i$ is a block of eigenvariables, and for each block $\vec{q}^i$ we associate the formula

$$\exists \vec{x}^{i+1} \forall \vec{y}^{i+1} \ldots \exists \vec{x}^n \forall y^n B(\vec{C}^1, \vec{q}^1, \ldots, \vec{q}^i, \vec{x}^{i+1}, \ldots, \vec{x}^n, \vec{y}^n).$$

That is we add all of the quantifiers to the right of $\forall \vec{y}^i$ to the instance. Note this formula will always be Σ_{i-1}^q. Using $\prec$, we are able to order each block of eigenvariables in Θ.

Now, as in the previous theorem, we define a formula $\psi(m, \Lambda, \Theta, P)$, which says there are values falsifying the formulas corresponding to the first m blocks of eigenvariables. This formula is Σ_i^B, so we find the the maximum value m_0 satisfying the formula using Σ_i^B-MAX. From this, we can find our witness. □

7 New Axiomatization of V

In this section, we will strengthen a result from [9]. In that paper, Krajicek and Pudlak showed that V can be axiomatized by $V^1 + \{\Sigma_i^q\text{-RFN}(G_i) \mid i \in \mathbb{N}\}$. A similar proof can be used to prove that V can be axiomatized by $V^1 + \{\Sigma_i^q\text{-RFN}(G_i^*) \mid i \in \mathbb{N}\}$. In this section, we show that V can also be axiomatized by $V^1 + \{\Sigma_i^q\text{-RFN}(CFG^*) \mid i \in \mathbb{N}\}$, where CFG^* is the cut-free version of G^*. Note that CFG^* is a weaker proof system than any of the other fragments of G.

Just a bit of notation. If A is a formula with free variables $\vec{p}$, then $\exists A$, called the existential closure of A, is the formula $\exists \vec{p} A$.

Lemma 7.1. *V^1 proves*

$$\Sigma_{i+1}^q\text{-}\mathit{RFN}(CFG^*) \leftrightarrow \Sigma_{i+1}^q\text{-}\mathit{RFN}(G_i^*).$$

Proof. The if direction is easy since a CFG^* proof is also a G_i^* proof. The only if direction is not as easy. Assume Σ_{i+1}^q-RFN(CFG^*), and argue in V^1. Given a G_i^* proof π of a Σ_{i+1}^q formula A, we change it into a CFG^* proof of a formula

$$B \equiv A \vee \bigvee_{j=1}^{n} \exists (C_j \wedge \neg C_j),$$

where $C_1, \ldots, C_n$ are all of the cut formulas in π.

This is done by, first, replacing each cut by

$$\dfrac{\dfrac{\Gamma \to \Delta, C \qquad \dfrac{C, \Gamma \to \Delta}{\Gamma \to \Delta, \neg C}}{\Gamma \to \Delta, C \wedge \neg C}}{\Gamma \to \Delta, \exists (C \wedge \neg C)}$$

The sequents in the rest of the proof are changed to include $\exists (C_i \wedge \neg C_i)$. Note that none of the inferences are affected by adding this formula. The only problem could be the eigenvariable restriction in $\exists$-left and $\forall$-right inferences; however, since the new formula does not have any free variables, there is no problem. At the end of the proof, the A is combined with the new formulas using $\vee$-right inferences.

Since the cut formulas are Σ_i^q, B is a Σ_{i+1}^q. By Σ_{i+1}^q-RFN(CFG^*), B is true, and, since $\exists (C_i \wedge \neg C_i)$ cannot be true, A must be true. This can be done in V^1 since it proves the Tarski conditions for the true definition. □

Corollary 7.2. $V = V^1 + \{\Sigma_i^q\text{-}\mathit{RFN}(CFG^*) \mid i \in \mathbb{N}\}$.

Proof. Follows from the lemma above, Krajicek's and Pudlak's axiomatization of V, and the fact that Σ_{i+1}^q-RFN(G_i^*) implies Σ_i^q-RFN(G_i^*) □

References

[1] S. R. Buss. On Herbrand's theorem. *Lecture Notes in Computer Science*, 960:195–209, 1995.

[2] S. Cook. *Theories for Complexity Classes and their Propositional Translations*, pages 175–227. Quaderni di Matematica. 2003.

[3] S. Cook and T. Morioka. Quantified propositional calculus and a second-order theory for NC^1. *Archive for Math. Logic*, 44(6):711–749, August 2005.

[4] S. Cook and P. Nguyen. Foundations of proof complexity: Bounded arithmetic and propositional translations. Available from http://www.cs.toronto.edu/˜sacook/csc2429h/book, 2006.

[5] S. Cook and N. Thapen. The strength of replacement in weak arithmetic. *ACM Trans. Comput. Logic*, 7(4):749–764, 2006.

[6] S. A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the 7-th ACM Symposium on the Theory of computation*, pages 83–97, 1975.

[7] J. Krajicek. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Cambridge University Press, 1995.

[8] J. Krajícek, P. Pudlák, and G. Takeuti. Bounded arithmetic and the polynomial hierarchy. *Ann. Pure Appl. Logic*, 52(1-2):143–153, 1991.

[9] J. Krajicek and P. Pulak. Quantified propostitional calculi and fragments of bounded arithmetic. *Zeitschr. f. math. Logik und Grendlagen d. Math.*, 36:29–46, 1990.

[10] A. Maciel and T. Pitassi. Conditional lower bound for a system of constant-depth proofs with modular connectives. In *LICS*, pages 189–200. IEEE Computer Society, 2006.

[11] T. Morioka. *Logical Approaches to the Complexity of Search Problems: Proof Complexity, Quantified Propositional Calculus, and Bounded Arithmetic*. PhD thesis, University Of Toronto, 2005.

[12] P. Nguyen. Separating dag-like and tree-like proof systems. Accepted in LICS, 2007.