Swansea University
Department of Computer Science

Transformation of Equational Theories and the Separation Problem of Bounded Arithmetic

MRes Thesis

by David R. Sherratt

supervised by Prof. Arnold Beckmann

Swansea University
Prifysgol Abertawe

Submitted to Swansea University in fulfilment of the requirements for the Degree of Masters of Research

March 2016

ABSTRACT

This project is about finding a translation between two different types of equational theories which we call expanded equational theories and pure equational theories. We create a translation which is infinitely axiomatizable and we show that S_2^1 cannot prove the consistency of the translated result of Buss and Ignjatovič.

DECLARATIONS

This work has not previously been accepted in substance for any degree and is not being currently submitted for any degree.

Date :

Signed :

Statement 1

This dissertation is being submitted in partial fulfillment of the requirements for the degree of a BSc in Computer Science.

Date :

Signed :

Statement 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are specifically acknowledged by clear cross referencing to author, work, and pages using the references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this dissertation and the degree examination as a whole.

Date :

Signed :

Statement 3

I hereby give consent for my dissertation to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Date :

Signed :

CONTENTS

1. Introduction 7

2. Background 10

2.1 Complexity Classes 10

2.1.1 Polynomial Hierarchy 11

2.1.2 Polynomial Hierarchy of Function 14

2.2 Bounded Arithmetic 17

2.3 Sequent Calculus 23

2.4 Gödel's Completeness Theorem 25

2.5 Gödel's Incompleteness Theorems 25

2.6 Proof System PV 28

2.7 Equational Theories 30

2.8 Formalization in S_2^1 33

3. First Translation 37

3.1 Translation of Boolean Formulas 39

3.2 Translation of Sequents 40

3.3 Consistency Property 41

3.4 Finding Axioms 43

3.5 Criticism of Translation 47

3.6 Evaluation of Translation 49

4. Improved Translation 51
4.1 Translation of Boolean Formulas 51
4.2 Translation of Sequents 52
4.3 Consistency Property 53
4.4 The Provability Property 55
4.4.1 Axioms 56
4.4.2 General Axioms 58
4.4.3 Inference Rules 60
4.5 Evaluation of Translation 75

5. Conclusion 78
5.1 Future Work 78

Appendix 80

A. BASIC 81

B. PK 84

1. INTRODUCTION

The most famous open problem in theoretical computer science formulated by Cook in [12] asks whether the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ are equal. In this thesis we define the polynomial hierarchy as described by Stockmeyer [17] and Buss [3], including the complexity classes $\mathcal{P}$ and $\mathcal{NP}$.

Bounded Arithmetic is a weak fragment of Peano arithmetic [3] which consists of relating first order theories. Since Bounded Arithmetic was introduced, there has always been the open problem of Bounded Arithmetic, is Bounded Arithmetic finitely axiomatizable or not, which can also be expressed as, is the hierarchy of Bounded Arithmetic theories proper or does it collapse [1]. Bounded arithmetic is of great interest to us because of its relation to the polynomial hierarchy. Since the connection to the polynomial hierarchy, this is comparable to the open problem whether the polynomial hierarchy is proper or collapses [3].

One attempt is to use consistency statements to solve the open Bounded Arithmetic problem. Gödel's Incompleteness Theorems [16] apply to the different theories of Bounded Arithmetic such as S_2^1 and T_2^1. Because of this, we can use consistency statement to try and prove that S_2^i is not equivalent to S_2^{i+1} [3], because if S_2^{i+1} can prove the consistency of S_2^i then the two theories must be separate, thus solving the open Bounded Arithmetic problem. We say $Con(S)$ to mean *the consistency of* S. We know that $S_2^{i+1} \nvdash Con(S_2^i)$ as [18] shows that $S_2 \nvdash Con(Q)$ where Q is Robinson's open, induction free sub-theory of Peano arithmetic. Thus the usual notion of consistency is too strong. Buss shows in [3] that $S_2^{i+1} \vdash BDCon(S_2^i)$ holds for at most one i, so that a theory S_2^i can have its consistency proven by S_2^{i+1} where we only refer to proofs only with bounded formulas for at most one i. Pudlák shows in [15] that $S_2 \nvdash BDCon(S_2^1)$ hence only $S_2 \vdash BDCon(S_2^0)$ remains possible. Other notions of consistency have also been considered [3].

This leads to the conjecture by *Gaisi Takeuti* which states *"Let R be an equational theory involving equations $s = t$ where s and t are closed terms in the language of S_2, with natural rules based on recursive definitions of the function symbols. Show that $S_2 \nvdash Con(R)$"* [9]. If this conjecture were to be true, then consistency statement would likely be impractical to use to solve the finitely axiomatizable problem of Bounded Arithmetic. Beckmann disproves this conjecture in [1], leaving hope that consistency statements can be used to obtain an answer, that Bounded Arithmetic is not finitely axiomatizable.

An equational theory is in a *pure equational setting* if every line in a proof is an equation. Beckmann proves in [1] that an equational theory in this setting, whose axioms are restricted to recursive definitions of the function symbols in the language chosen, can have its consistency proven in S_2^1. An equational theory is in an *expanded equational setting* if every line in a proof is a boolean formula. Buss and Ignjatovič show in [5] that an equational theory in this setting, with a proof system of the propositional variant of PK, and a set of axioms defining the non-logical symbols in the language, cannot have its consistency proven in S_2^1. Our result transforms Buss and Ignjatovič equational theory to a pure equation setting, requiring new axioms used for the translation. To do this we define a translation from Buss and Ignjatovič equational theory to a pure equation setting which we call PET, which is formalizable in S_2^1. The translation requires an addition of further axioms to PET. Beckmann's result just uses recursive definitions of the function symbols for the axioms used, however our result will have additional axioms needed for the translation. Our result states S_2^1 cannot prove the consistency of PET, by proving that if S_2^1 could prove the consistency of PET then the consistency of Buss and Ignjatovič's result is also proven by implication.

The aim of this thesis is to develop a translation from expanded equational theories to pure equational theories and to make precise what additional axioms are needed for the translation. We use this translation to transform Buss and Ignjatovič result written in expanded equation theory which we call EET, to a pure equational theory which we call PET. We want this transformation to be formalizable in S_2^1 so that we can show that $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$, and ultimately that S_2^1 cannot prove the consistency of our the equational theory PET, this is explained more in Section 2.8.

This thesis begins by discussing the different research areas looked into for this project. Chapter 2

introduces the background research needed to understand the thesis. Section 2.1 introduces the polynomial hierarchy as described by Stockmeyer in [17]. Section 2.1.2 expands the polynomial hierarchy to the polynomial hierarchy of functions as described by Buss in [3]. Section 2.2 introduces Bounded Arithmetic and its importance to us. Section 2.3 introduces Sequent Calculus and describes where it is used. Section 2.4 describes Gödel's Completeness Theorem and Section 2.5 describes Gödel's Incompleteness Theorems, and why they are important. Section 2.6 introduces the proof system PV, and tells us about different attempts to show that S_2^1 can or cannot prove the consistency of its induction-free version. Section 2.7 explains in more detail EET and some of the foundation for PET. Chapter 3 introduces our first attempt of a translation that was not successful as the translation was not formalizable in S_2^1. Section 3.5 details why this translation is not formalizable in S_2^1. Chapter 4 describes the improved translation where we take into consideration the faults in the first attempt. Section 4.4 describes the additional axioms needed for this translation to work as well as to translate EET into PET. Section 4.5 evaluates the translation and shows that $S_2^1 \nvdash Con(PET)$. Chapter 5 is the conclusion of the thesis and Section 5.1 describes the future research that can be done.

2. BACKGROUND

The background section introduces in detail the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ as well as the famous open problem $\mathcal{P}$ vs $\mathcal{NP}$. We define the polynomial hierarchy as described by Stockmeyer [17] and Buss [3] and describe the problem about whether this hierarchy collapses or not. We introduce Bounded Arithmetic and the Bounded Arithmetic hierarchy and describe the connection between the Bounded Arithmetic hierarchy and the polynomial hierarchy, motivating why we are interested in the separation problem of Bounded Arithmetic. We then describe Gödel's Incompleteness Theorems and explain that the use of consistency statements are a natural candidate for separating the different theories of Bounded Arithmetic, since if $S_2^{i+1} \vdash Con(S_2^i)$ then we can conclude that S_2^{i+1} and S_2^i are separate theories as a theory cannot demonstrate its own consistency. We then describe the different settings of equational theories, *pure* and *expanded*, and describe the results of Beckmann [1] and Buss and Ignjatovič [5].

2.1 Complexity Classes

A complexity class is a set of problems that have similar resource complexity, where the resource can be space or time. Chapter 2.1.1 explains more about complexity classes, and gives in detail examples such as $\mathcal{P}$ and $\mathcal{NP}$, and describes the polynomial hierarchy as described by Stockmeyer in [17]. Chapter 2.1.2 describes the polynomial hierarchy of functions as described by Buss in [3], and the open problems surrounding it such as the $\mathcal{P}$ vs $\mathcal{NP}$ problem.

2.1.1 Polynomial Hierarchy

A complexity class is a set of problems that have similar resource complexity, where the resource can be space or time for example. The problems can be solved by a Turing machine $\mathcal{M}$ within $O(f(n))$ resources, where n is the size of the input of the problem. Examples of complexity classes are $\mathcal{P}$ and $\mathcal{NP}$. Informally, a problem in $\mathcal{P}$ means that the problem can be solved within polynomial resources, that is, a problem can be solved within a number of steps polynomially bounded by the length of the input. A problem in the class $\mathcal{NP}$ would mean that the problem, given a solution, is checkable within polynomial resources.

To be precise, we can define $\mathcal{P}$ as follows. Let Σ be an alphabet with at least 2 elements. Σ^* is then the set of all possible finite strings over Σ. A language, L, over the set Σ is a subset of Σ^*. Each $\mathcal{M}$ has an associated input alphabet Σ. For each $w \in \Sigma^*$, there is a computation associated with $\mathcal{M}$ and w. $\mathcal{M}$ accepts w if $\mathcal{M}$ terminates in the accepting state. $\mathcal{M}$ doesn't accept w if $\mathcal{M}$ terminates in the rejecting state, or if $\mathcal{M}$ fails to terminate. $L(\mathcal{M})$ is the language accepted by $\mathcal{M}$, that is it is the set of all elements of Σ^* that $\mathcal{M}$ accepts. Let $t_{\mathcal{M}}(w)$ be the number of computational steps $\mathcal{M}$ takes on input w, then for natural numbers n, which represent the length of the input, we can find the worst case run time of $\mathcal{M}$, $T_{\mathcal{M}}(n)$. $T_{\mathcal{M}}(n) = max\{t_{\mathcal{M}}(w)|w \in \Sigma^n\}$ where Σ^n is the set of all strings over Σ with length n. We can say $\mathcal{M}$ runs in polynomial time if $\exists_k \forall_n (T_{\mathcal{M}}(w) \leq n^k + k)$. So that the number of computational steps in the worst case run time is less then or equal to some polynomial equation in the form $n^k + k$.

Definition 1. *The complexity class $\mathcal{P} = \{L|L = L(\mathcal{M})$ for some Turing machine $\mathcal{M}$ that runs in polynomial time$\}$[12].*

We can also define $\mathcal{NP}$ as follows. Let R be a binary relation $\Sigma_1^* \times \Sigma_2^*$ for finite non-empty sets Σ_1 and Σ_2. We then associate with each R a language L_R over $\Sigma_1 \cup \Sigma_2 \cup \{\#\}$ where L_R can be defined as $\{w\#y|R(w,y)\}$ and the symbol $\#$ is not in the alphabets Σ_1 and Σ_2. R is polynomial time iff $L_R \in \mathcal{P}$.

Definition 2. *A language L over Σ is in $\mathcal{NP}$ iff there exists some natural number k and binary relation R such that $L_R \in \mathcal{P}$, where $\forall_{w \in \Sigma^*}(w \in L \Leftrightarrow \exists_y(|y| \leq |w|^k$ and $R(w,y)))$ [12].*

The open problem that many computer scientists and mathematicians try to solve is whether the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ are equal or not. This is known as the $\mathcal{P}$ vs $\mathcal{NP}$ problem. The open problem is about whether every language that is accepted by a nondeterministic algorithm in polynomial time (that is in $\mathcal{NP}$) is accepted by a deterministic algorithm in polynomial time as well (is also in $\mathcal{P}$) [12].

It is known that the polynomial class $\mathcal{P}$ is a subset of the class $\mathcal{NP}$, but not known whether or not the classes are equal. The class $\mathcal{P}$ can be expressed as Δ_0^P and Σ_0^P and Π_0^P and Δ_1^P. The class $\mathcal{NP}$ is denoted by Σ_1^P. We can define the polynomial hierarchy in terms of polynomial-time bounded *oracle machines* [17].

Let $\mathcal{O}$ be an oracle machine. $\mathcal{O}$ is a deterministic or nondeterministic Turing machine with a distinguished work tape that we call the *query tape* and three special states called the *query state*, *yes state* and the *no state*. Computations of $\mathcal{O}$ not only depend on the input but on a given set, B, of words which we call the *oracle*. The actions of an oracle machine with oracle B are identical to those of a Turing machine, with one exception. If the machine enters the query state, it next enters the yes state if the nonblank portion of the query tape contains a word from B, otherwise it enters the no state [17].

An oracle machine operates within time $T(n)$ iff for every input x, every computation halts within $T(|x|)$ steps, where $|x|$ denotes the length of the word x. We can now define the polynomial hierarchy according to [17]:

Definition 3. *Let $\mathcal{O}(B)$ denote the language accepted by an oracle machine $\mathcal{O}$ with oracle B.*

Case 1: B is a set of words then

$\mathcal{P}(B)$ = {$\mathcal{O}(B)$ *where $\mathcal{O}$ is a deterministic oracle machine which operates within time $p(n)$ for some*

polynomial $p(n)\}$

$\mathcal{NP}(B) = \{\mathcal{O}(B)$ *where* $\mathcal{O}$ *is a nondeterministic oracle machine which operates within time* $p(n)$ *for some polynomial* $p(n)\}$

Case 2: $\mathcal{B}$ *is a class of sets then*

$$\mathcal{P}(\mathcal{B}) = \bigcup_{B\in\mathcal{B}} \mathcal{P}(B)$$

$$\mathcal{NP}(\mathcal{B}) = \bigcup_{B\in\mathcal{B}} \mathcal{NP}(B)$$

$co\mathcal{NP}(\mathcal{B})$ is the set of languages which are complements of languages in $\mathcal{NP}(\mathcal{B})$

Definition 4. *The polynomial hierarchy is* $\{\Sigma_k^P, \Pi_k^P, \Delta_k^P : k \geq 0\}$ *where*

$$\Sigma_0^P = \Pi_0^P = \Delta_0^P = \mathcal{P}$$

and for $k > 0$

$$\Sigma_{k+1}^P = \mathcal{NP}(\Sigma_k^P)$$

$$\Pi_{k+1}^P = co\mathcal{NP}(\Sigma_k^P)$$

$$\Delta_{k+1}^P = \mathcal{P}(\Sigma_k^P)$$

So the complexity class $\mathcal{NP}$ is $\mathcal{NP}(\mathcal{P}) = \Sigma_1^P$. However, what we do not know and remains an open problem is [17];
Does $\Sigma_k^P \neq \Sigma_{k+1}^P$ for all $k \geq 0$
Does $\Sigma_k^P \neq \Pi_k^P$ for all $k \geq 1$

Does $\Delta_k^P \neq \Sigma_k^P \cap \Pi_k^P$ for all $k \geq 1$

2.1.2 Polynomial Hierarchy of Function

We have already defined the Stockmeyer polynomial hierarchy [17] in the previous Section. In addition, we also have $\square_k^P$ which is a class of functions, e.g. $\square_1^P$ is the class of polynomial-time computable functions. These classes extend our current definition of the hierarchy. The definition found in [3] is as shown;

Definition 5. *N is the set of the natural numbers*

Definition 6. *B is the following set of basic functions from N^k to N [3] :*

0, the constant zero function

$x \to Sx$, the successor function

$x \to \lfloor \frac{1}{2}x \rfloor$, the shift right function

$x \to 2 \cdot x$, the shift left function

$$(x, y) \to x \leq y = \begin{cases} 0 & x > 0, \\ 1 & x \leq y. \end{cases}$$

$$(x, y, z) \to Choice(x, y, z) = \begin{cases} y & x > 0, \\ z & x = 0. \end{cases}$$

Definition 7. *A polynomial is a* suitable polynomial *if it has nonnegative integer coefficients.*

Definition 8. *$\overrightarrow{x}$ is a vector of numbers $x_1, ..., x_n$*

Definition 9. *$|x|$ is $\lceil log_2(x+1) \rceil$ which is the length in binary (except $|0| = 0$).*

We use $|\overrightarrow{x}|$ to denote the vector $|x_1|, ..., |x_n|$

Definition 10. Limited iteration *[3] is defined as follows. Let $k \geq 0$ and let $g : N^k \to N$ and $h : N^{k+2} \to N$ be arbitrary functions and let p and q be suitable polynomials. $f : N^k \to N$ is defined from g and h with time bound p and space bound q iff the following conditions hold for $\tau : N^{k+1} \to N$;*

$$\tau(x_1, ..., x_k, 0) = g(x_1, ..., x_k)$$

$$\tau(x_1, ..., x_k, n+1) = h(x_1, ..., x_k, n, \tau(x_1, ..., x_k, n))$$

where we have n-fold applications of h such that

$$(\forall_{n \leq p(|\vec{x}|)}(|\tau(\vec{x}, n)| \leq q(|\vec{x}|)))$$

where $f(\vec{x})$ *is defined by*

$$f(\vec{x}) = \tau(\vec{x}, p(|\vec{x}|))$$

Using B, *limited iteration*, and composition [3], we can define the *Polynomial-Time Closure* which we use to extend the polynomial hierarchy to a hierarchy of functions [3].

Definition 11. *A function* $f : N^k \to N$ *has* polynomial growth rate *iff there is a suitable polynomial* p *such that for all* $\vec{x}$*, we have* $|f(\vec{x})| \leq p(|\vec{x}|)$.

Definition 12. *Let* C *be a set of functions with polynomial growth rate. The* Polynomial-Time Closure *of* C*,* $PTC(C)$*, is the smallest class of functions which contains* C *and* B *and is closed under composition and limited iteration with functions* g *and* h *with time bound* p *and space bound* q.

A function $f \in PTC(C)$ iff there is a finite set of functions $\{h_1, ..., h_k\} \subseteq C$ and an oracle machine $\mathcal{O}$ with oracles $h_1, ..., h_k$ such that $\mathcal{O}$ computes f in polynomial time [3]. Buss shows in [3] that $PTC(\varnothing)$ = the set of functions computable in polynomial time Turing machines.

Definition 13. *Predicates are functions with the range* $\{0, 1\}$.

Definition 14. *Let* C *be a set of functions. Then* $PRED(C)$ *is the set of predicates in* C.

Definition 15. *Let* C *be a set of functions closed under composition [3]. Then* $LB\exists(C)$ *is the set of predicates* Q *such that:*

$Q : N^i \to N$ *for some* $i \geq 0$.

There is a $R \in PRED(C)$ *and a suitable polynomial* p *such that for all* $\vec{x}$*:*

$$Q(\vec{x}) = \exists_{y \leq p(|\vec{x}|)} R(\vec{x}, y)$$

Definition 16. *Let* C *be a set of functions closed under composition [3]. Then* $LB\forall(C)$ *is the set of predicates* Q *such that:*

$Q : N^i \to N$ *for some* $i \geq 0$.

There is a $R \in PRED(C)$ *and a suitable polynomial* p *such that for all* $\vec{x}$*:*

$$Q(\vec{x}) = \forall_{y \leq p(|\vec{x}|)} R(\vec{x}, y)$$

Using PTC, $LB\exists$, $LB\forall$, and composition [3] we can now extend our current definition of the polynomial hierarchy to include the polynomial-time computable functions $\square_k^P$. $\square_0^P$ is smallest set of functions that contain B and is closed under composition, $LB\exists$ and $LB\forall$.

Definition 17. *The polynomial hierarchy is* $\{\Sigma_k^P, \Pi_k^P, \Delta_k^P, \square_k^P : k \geq 0\}$ *where*

$\square_0^P$ *is the smallest set of functions containing* B *and closed under composition,* $LB\exists$*, and* $LB\forall$
$\Sigma_0^P = \Pi_0^P = \Delta_0^P = \mathcal{P}$

and for $k > 0$

$\square_k^P = PTC(\Sigma_k^P)$

$\Sigma_{k+1}^P = \mathcal{NP}(\Sigma_k^P)$

$\Pi_{k+1}^P = co\mathcal{NP}(\Sigma_k^P)$

$\Delta_{k+1}^P = \mathcal{P}(\Sigma_k^P)$

$PH = \bigcup_k \Sigma_k^P$

We refer to the complexity class $\square_1^P = PTC(\mathcal{P})$ as $\mathcal{FP}$ and $\square_k^P$ as $\mathcal{FP}^{\Sigma_{k-1}^P}$ from now on. This leads us to our next Chapter, Bounded Arithmetic. We are interested in knowing whether the polynomial hierarchy collapses or not, and since [3] we know that the polynomial hierarchy is closely linked to the Bounded Arithmetic hierarchy so we are therefore interested in whether the Bounded Arithmetic hierarchy collapses or not. This is explained in more detail in Section 2.2.

2.2 Bounded Arithmetic

Bounded Arithmetic is a family of formal theories which are weak fragments of Peano arithmetic [3]. It takes interest because of the connections it has to the polynomial hierarchy [3].

Definition 18. *A variable occurrence in a first-order formula is called a* free variable *if it is not occurring in the scope of the quantifier*

Definition 19. *A* sentence *is a formula without the occurrence of free variables. A* theory *is a set of sentences.*

Bounded arithmetic theories are first-order theories for the natural numbers. The first-order language of Bounded Arithmetic, L_b, contains the predicates $=$ and $\leq$. The language L_b also consists of the symbols;

S	successor function
0	zero constant
$+$	addition
$\cdot$	multiplication
$\lvert x\rvert$	binary length of x, can also be expressed as $\lceil log_2(x+1)\rceil$
$\lfloor \frac{1}{2}x \rfloor$	greatest integer less than or equal to $\frac{x}{2}$
$\#$	*smash* function with 2 inputs, x and y, defined to be $2^{\lvert x\rvert\cdot\lvert y\rvert}$

The growth rate of $\#$ is necessary to define functions in the polynomial hierarchy. The smash function can express the term $2^{q(|x|)}$ where q is any polynomial with non-negative coefficients, since $1\#x = 2^{|x|}$ and $|\lfloor \frac{1}{2}(x\#y)\rfloor| = |x|\cdot|y|$.

Using the symbols $0, S, +$, and $\cdot$ one can construct terms that represent the natural numbers. For example, one can express the number 3 as the term $SSS0$ or even as $(SS0)+(S0)$. S^k0 is also a term that can be used to represent k applications of the successor function on 0. Formulae use natural numbers frequently, a number in an equation is intended to be replaced by any closed term representing the value of the number [3].

Definition 20. *Quantifiers of the form* $(\forall x)$ *and* $(\exists x)$ *are known as* unbounded *quantifiers. A bounded quantifier is of the form* $(\forall x \leq t)$ *and* $(\exists x \leq t)$ *where* t *is a term not including* x*. A sharply bounded quantifier is similar to a bounded quantifier, except they are of the form* $(\forall x \leq |t|)$ *and* $(\exists x \leq |t|)$*.*

A bounded formula is a formula with no unbounded quantifiers. Bounded Arithmetic is concerned with bounded quantifiers and eliminates unbounded quantifiers from the proofs of bounded formulae.

A hierarchy of bounded formulae can be defined by counting alterations of bounded quantifiers [7]. The following defines the Bounded Arithmetic hierarchy [3].

$\Pi_0^b = \Sigma_0^b$ is the set of formulae where all quantifiers are sharply bounded.

Σ_{k+1}^b is the smallest set that satisfies;

- $\Pi_k^b \subseteq \Sigma_{k+1}^b$
- If $A \in \Sigma_{k+1}^b$ then $(\exists x \leq t)A \in \Sigma_{k+1}^b$ and $(\forall x \leq |t|)A \in \Sigma_{k+1}^b$
- If $A, B \in \Sigma_{k+1}^b$ then $A \wedge B \in \Sigma_{k+1}^b$ and $A \vee B \in \Sigma_{k+1}^b$
- If $A \in \Sigma_{k+1}^b$ and $B \in \Pi_{k+1}^b$ then $\neg B \in \Sigma_{k+1}^b$ and $B \rightarrow A \in \Sigma_{k+1}^b$

Also, Π_{k+1}^b is the smallest possible set that satisfies;

- $\Sigma_k^b \subseteq \Pi_{k+1}^b$
- If $A \in \Pi_{k+1}^b$ then $(\exists x \leq |t|)A \in \Pi_{k+1}^b$ and $(\forall x \leq t)A \in \Pi_{k+1}^b$
- If $A, B \in \Pi_{k+1}^b$ then $A \wedge B \in \Pi_{k+1}^b$ and $A \vee B \in \Pi_{k+1}^b$
- If $A \in \Pi_{k+1}^b$ and $B \in \Sigma_{k+1}^b$ then $\neg B \in \Pi_{k+1}^b$ and $B \rightarrow A \in \Pi_{k+1}^b$

Peano Arithmetic is axiomatized by a number of axioms and an induction schema. Since Bounded Arithmetic is a weak fragment of Peano Arithmetic [3], one can form the axioms of Bounded Arithmetic by extending the open axioms and restricting the induction axioms of Peano Arithmetic.

BASIC is a finite set of true open formulae of the natural numbers. They are suitable to define the properties relating the function and predicate symbols of Bounded Arithmetic [3]. The formulas that make *BASIC* can be found in Appendix A.

Bounded Arithmetic has various types of induction axioms. These are the IND axioms which are the usual induction axioms, the PIND axioms which are *polynomial* induction axioms, and the LIND axioms which are *length* induction axioms.

Definition 21. *Let Ψ be a set of formulae. Then axiom schemes can be defined as*

Ψ-*IND* $\quad A(0) \wedge (\forall x(A(x) \rightarrow A(Sx)) \rightarrow (\forall x)A(x) \quad$ *for* $A \in \Psi$

Ψ-*PIND* $\quad A(0) \wedge (\forall x(A(\lfloor \frac{1}{2}x \rfloor) \rightarrow A(x)) \rightarrow (\forall x)A(x) \quad$ *for* $A \in \Psi$

Ψ-*LIND* $\quad A(0) \wedge (\forall x(A(x) \rightarrow A(Sx)) \rightarrow (\forall x)A(|x|) \quad$ *for* $A \in \Psi$

The axiom scheme Ψ-IND is stronger than Ψ-PIND and Ψ-LIND. Ψ-IND can be shown to be potentially stronger than Ψ-PIND after inspection. For example, if we know that $A(0)$ is true and we wish to deduce that $A(100)$ is also true then Ψ-IND would deduce $A(1)$ from $A(0)$, and $A(2)$ from $A(1)$, and so on until we reach the 100th deduction $A(100)$. On the other hand, Ψ-PIND would make seven deductions to reach $A(100)$. Therefore, because Ψ-IND took 100 inferences and Ψ-PIND only took 7 to reach the same conclusion we can argue that the hypothesis of Ψ-PIND is stronger than the Ψ-IND hypothesis and hence the Ψ-PIND axioms are weaker than the Ψ-IND axioms [3].

There are different theories in Bounded Arithmetic that can be obtained by using different induction schemes [4]. The theory T_2^i is a theory of Bounded Arithmetic axiomatized by the Σ_i^b-IND axioms and the additional finite set of open axioms *BASIC*, and S_2^i is a theory of Bounded Arithmetic axiomatized by the Σ_i^b-PIND axioms and *BASIC* [4]. Buss shows in [3] that for $i \geq 1$, the Σ_i^b-IND axioms imply the Σ_i^b-PIND axioms, and the Σ_{i+1}^b-PIND axioms imply the Σ_i^b-IND axioms. It is because of that the theory S_2^{i+1} contains the theory T_2^i which in turn contains S_2^i, that is, $S_2^i \subseteq T_2^i \subseteq S_2^{i+1} \subseteq ...$ [4].

S_2^1 is a first-order theory in the language L_b that consists of the axioms;

- *BASIC* axioms
- Σ_i^b-PIND axioms

T_2^1 is a first-order theory in the language L_b that consists of the axioms;

- *BASIC* axioms
- Σ_i^b-IND axioms

S_2^1 is the theory this paper concentrates on. An example of something that is *true* that S_2^1 cannot

prove is the consistency of S_2^1 itself. That is, S_2^1 is unable to show the consistency statement of S_2^1. This is explained in more detail in Section 2.5.

However, statements that S_2^1 can prove using the *BASIC* axioms and the Σ_i^b-PIND induction scheme include the existence and uniqueness of the solutions to the predecessor function [3]. The predecessor function can be defined as;

$$P(a) = b \Leftrightarrow (a = 0 \wedge b = 0) \vee (Sb = a)$$

Let $M(a, b)$ be the defining formula for the predecessor function (the right hand side). To show that a formula is Σ_1^b-definable in S_2^1 we must show that S_2^1 can prove the *uniqueness* condition and the *existence* condition.

The uniqueness condition can be expressed as

$$M(a, b) \wedge M(a, c) \rightarrow M(a, c)$$

which can be proven with the *BASIC* axioms with no induction.

For the existence condition,

$$\exists_{z \leq 0} M(0, z)$$

can be proven from the *BASIC* axioms without any use of induction axioms.

$$\exists_{z \leq \lfloor \frac{x}{2} \rfloor} M(\lfloor \frac{x}{2} \rfloor, z) \rightarrow \exists_{z \leq x} M(x, z)$$

can also be proven also from the *BASIC* axioms without any use of induction axioms. Below is a proof of the second equation of the existence condition.

Proof. We argue in *BASIC*. Let x be arbitrarily. Let us assume that the premise, $\exists_{z \leq \lfloor \frac{x}{2} \rfloor} M(\lfloor \frac{x}{2} \rfloor, z)$ is true. We have to show that $\exists_{z \leq x} M(x, z)$ is also true. Let $a = \lfloor \frac{x}{2} \rfloor$, then we have $\exists_{z \leq a} M(a, z)$ is true. Let us choose a $z \leq a$ such that $M(a, z)$ is true. We need to find some $z' \leq x$ such that $M(x, z')$ is true.

If we substitute x as a, and y as x, in *BASIC* axiom 32. which can be found in Appendix A we get the formula

$$a = \lfloor \frac{x}{2} \rfloor \leftrightarrow (2 \cdot a = x) \vee (S(2 \cdot a) = x)$$

Here we have two cases, $(2 \cdot a = x)$ and $(S(2 \cdot a) = x)$.

Let us look at the first case, $2 \cdot a = x$. We need to find $z' \leq z$ such that $M(2 \cdot a, z')$ is true. Here we have another two cases. In case 1.1, if $a = 0$ then we choose $z' = 0$. In case 1.2, if $a \neq 0$ then we choose $z' = S(2 \cdot z)$. It follows from the *BASIC* axioms that $S(z') = x$, as $S(z') = S(S(2 \cdot z)) = 2 \cdot S(z)$ *(this can be proven using the BASIC axioms)* $= 2 \cdot a = x$.

The second case is easier to see, where $S(2 \cdot a) = x$. We need to find some z' such that $z' \leq S(2 \cdot a)$ and such that $M(S(2 \cdot a), z')$ is true. If we choose $z' = 2 \cdot a$, then it is obvious that $M(S(2 \cdot a), 2 \cdot a)$ holds.

Therefore, $\exists_{z \leq \lfloor \frac{x}{2} \rfloor} M(\lfloor \frac{x}{2} \rfloor, z) \rightarrow \exists_{z \leq x} M(x, z)$ can be proven by the *BASIC* axioms and there is no need for any induction axioms. □

Finally, since $\exists_{z \leq x} M(x, z)$ is a Σ_1^b formula and S_2^1 can prove $\exists_{z \leq 0} M(0, z)$ is true and S_2^1 can prove $\exists_{z \leq \lfloor \frac{x}{2} \rfloor} M(\lfloor \frac{x}{2} \rfloor, z) \rightarrow \exists_{z \leq x} M(x, z)$ is true; Σ_1^b-PIND yields

$$S_2^1 \vdash \forall_x \exists_{z \leq x} M(x, z)$$

The theories S_2^{-1} and T_2^{-1} are the theories with only the *BASIC* axioms and no induction axioms. S_2 is the union $\bigcup_i S_2^i$ and T_2 is the union $\bigcup_i T_2^i$ [3].

Theorem 1. *Let $i \geq 1$. $T_2^i \vdash S_2^i$ and $S_2^i \vdash T_2^{i-1}$. Therefore $S_2 \equiv T_2$ [3].*

Theorem 1 describes how the different theories are related and describes the hierarchy of Bounded Arithmetic theories. What we do not know is if the hierarchy collapses or not. That is, we have the inclusions $S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq ...$ but we do not know whether these are proper inclusions. Do stronger axioms in Bounded Arithmetic prove more true statements?

Definition 22. *A theory T is* finitely axiomatizable *iff there is a finite set of axioms Δ where T is the set of all consequences of the formulas in Δ.*

Theorem 2. *If $T_2^i = S_2^{i+1}$ then $T_2^i = S_2$.*

Theorem 2 was proven in [6] and states that if the condition $T_2^i = S_2^{i+1}$ holds then this implies that S_2 collapses to T_2^i. This is because if $T_2^i = S_2^{i+1}$, then $T_2^i \vdash \Sigma_{i+1}^B$-IND and $T_2^i = T_2^{i+1}$. The full proof can be found in [6]. It would also be that $T_2^i = S_2$ is *finitely axiomatizable.*

Definition 23. *Let f be a function. Then f is Σ_i^b-definable in S_2^1 iff there is a Σ_i^b formula $A(x, y)$ such that;*

- $\forall_x A(x, f(x))$ *is true*
- $S_2^1 \vdash \forall_x \exists_y A(x, y)$
- $S_2^1 \vdash \forall_{x,y,z}(A(x, y) \wedge A(x, z) \rightarrow y = z)$

Buss shows in [3] that any function in $\mathcal{FP}^{\Sigma_{i-1}^P}$ can be Σ_i^b-defined in S_2^i, and vice versa.

Theorem 3 ([3])**.** *The Σ_i^b-definable functions in S_2^i are exactly $\mathcal{FP}^{\Sigma_{i-1}^P}$, for $i > 0$.*

The connection between the separation problem for Bounded Arithmetic and the complexity classes is that if the Bounded Arithmetic hierarchy collapses, then we can prove that the polynomial time hierarchy of complexity classes also collapses and we could prove this in S_2^1. If the Bounded Arithmetic theories are separate, then the polynomial time hierarchy could still collapse.

2.3 Sequent Calculus

Formal proofs in theories of Bounded Arithmetic such as S_2^1 use Gentzen-style sequent calculus proof systems [3]. First order theories with bounded quantifiers use the sequent calculus system described by Buss in [3], which is the usual Gentzen sequent calculus proof system augmented with inference rules for the bounded quantifiers [5].

Terms are built from constants, functions and free variables. From these terms we can build formulae. A finite sequence of formulae is called a *cedent.*

Definition 24. *A* sequent *is composed of the antecedent, succedent, and the sequent arrow* $\Rightarrow$*. Let* Δ *and* Γ *be cedents. Then the sequent* $\Delta \Rightarrow \Gamma$ *has* Δ *as a antecedent and* Γ *as a succedent. The sequent is true iff all of the formulae in* Δ *being true implies at least one of the formulae in* Γ *is true [3].*

The inference rules for the system PK we are using and Buss and Ignjatovič used for their result can be located in Appendix B. Buss used a sequent calculus system to define Bounded Arithmetic called LK [3]. LK consists of the *structural inferences*, *propositional inferences* and the *Cut rule* in PK as well as the following additional rules found in [3]. Buss and Ignjatovič did not use these rules as they worked with quantifier-free formulas only [5].

Definition 25. *An* eigenvariable *is a free variable that may not appear in the lower sequent of an inference [3].*

$$\forall\text{:left} \quad \frac{A(t), \Gamma \Rightarrow \Delta}{\forall_x A(x), \Gamma \Rightarrow \Delta}$$

$$\forall\text{:right} \quad \frac{\Gamma \Rightarrow \Delta, A(a)}{\Gamma \Rightarrow \Delta, \forall_x A(x)}$$

where a is an eigenvariable.

$$\exists\text{:left} \quad \frac{A(a), \Gamma \Rightarrow \Delta}{\exists_x A(x), \Gamma \Rightarrow \Delta}$$

where a is an eigenvariable.

$$\exists\text{:right} \quad \frac{\Gamma \Rightarrow \Delta, A(t)}{\Gamma \Rightarrow \Delta, \exists_x A(x)}$$

LK also includes the logical axiom $A \Rightarrow A$ where A is any cedent.

Theorem 4. *LK is sound and complete.*

Theorem 4 is proven in [3].

2.4 Gödel's Completeness Theorem

Gödel's completeness theorem is a result about first-order logic that we can extend to the theories of Bounded Arithmetic. The theorem applied to a theory of Bounded Arithmetic states that a sentence is *valid* in the theory iff the sentence is derivable (we can find a proof of it) from the axioms of that theory. We can define the completeness theorem as follows

Definition 26. *Let L be a first-order language and T be a formal system for L . For a sentence ϕ in L, we use the notion $\vdash \phi$ to mean ϕ* is provable in *T [2].*

Given that T is a formal system and ϕ is a sentence in the language of T, we use $T \vDash \phi$ to say that ϕ is true under every interpretation of T [2].

Definition 27. *Let L be a first-order language and T be a formal system for L. For a sentence ϕ in L, we use the notion $\vDash \phi$ to mean ϕ* is valid *[2].*

Theorem 5 (Completeness Theorem)**.** *Let L be a first-order language and T be a formal system for L. If $\vDash \phi$ then $\vdash \phi$.*

We can also extend this theorem for first-order theories such as the theories of Bounded Arithmetic. For example, given a sentence ϕ in the language of Bounded Arithmetic, $S_2^1 \vdash \phi$ iff $S_2^1 \vDash \phi$.

If $S_2^1 \vdash \phi$, then ϕ can be proven using the inference rules and logical axioms from LK as well as the axioms of S_2^1. There is a proof of ϕ. If $S_2^1 \vDash \phi$, then ϕ is valid for all the interpretations that are a model of S_2^1.

2.5 Gödel's Incompleteness Theorems

Gödel's incompleteness theorems describe certain limitations on sets of axioms. We are interested in Gödel's Incompleteness Theorems for separating the theories of Bounded Arithmetic. Statements of a formal theory are usually written in a symbolic form, it is common to use sets of axioms to describe the statement of a formal theory. A set of axioms is said to be *effectively generated* if the axioms can

be listed by an effective procedure (a computer program for example) without listing any statements that are not axioms.

David Hilbert proposed in 1920 a research project that became known as *Hilbert's Program* [13, 20]. The program had two parts (*i*) the descriptive and (*ii*) the justificatory. The descriptive part aimed to *"produce the sort of description of our natural ways of mathematical reasoning that makes precise evaluation of that reasoning possible"* [13]. However, we are more focused on the aim of the second part which was to produce a finitary proof of the reliability of our natural modes of mathematical reasoning. He wanted mathematics to be formulated on a solid and complete logical foundation. He believed it could be done by showing that all of mathematics follows from a correctly chosen finite system of axioms and that such a system is provable consistent. However, these theorems become a technical fault of Hilbert's Program and prove that such a program is impossible [13].

A *theory* is the consequences of a set of axioms. A set of axioms is *complete* if all statements in the language of the theory of the axioms are either provable or its negation is provable from the given set of axioms [16]. The set of axioms is *incomplete* if there exists at least one statement where that statement or its negation is not provable from the given set of axioms. A set of axioms is said to be *consistent* if there does not exist a sentence in the axioms langauge is both provable and refutable (so that there doesn't exist a statement such that the statement and its negation is provable from the axioms) [16].

Gödel's theorems' in plain English are

1. No consistent system of axioms whose theorems can be listed by an effective procedure is capable of proving all truths about the relations of the natural numbers. That is, there will always exist a statement about the natural numbers that is true but not provable within the set of axioms.

2. No consistent system of axioms whose theorems can be listed by an effective procedure can demonstrate it's own consistency.

Gödel's first incompleteness theorem states there is no complete, axiomatizable, and consistent

theory for the natural numbers. Let T be an axiomatizable theory and Con_T be the consistency statement expressing that T is consistent [8].

Theorem 6 (Gödel's First Incompleteness Theorem)**.** *Let T be a consistent theory and $S_2^1 \subseteq T$. Then there is a true sentence ϕ such that $T \nvdash \phi$.*

Intuitively, ϕ would be the sentence formalizing *"I am not provable in T"*. If ϕ were false, then ϕ would be provable in T, making it true and resulting in a contradiction. Therefore ϕ has to be true.

Gödel's first incompleteness theorem kills the possibility of Hilbert's program succeeding, by giving a true formula which is not provable by a consistent theory T.

Theorem 7 (Gödel's Second Incompleteness Theorem)**.** *Let T be a consistent theory and $S_2^1 \subseteq T$. Then $T \nvdash Con_T$.*

Buss proves in [3] that S_2^1 cannot prove its own consistency for the usual notation. If there did exist a proof in some theory of Bounded Arithmetic of the consistency of S_2^1, then by Gödel's incompleteness theorems, one would know that that theory and S_2^1 were separate theories. If that theory could demonstrate S_2^1's consistency and it was proven that those theories were equivalent, then we would have a contradiction to Gödel's second incompleteness theorem.

The usual notion of consistency of S_2^1 is considered to be too strong for any theory of Bounded Arithmetic to show. The usual notion of consistency for S_2^1 works by formalizing a notion of a proof, like sequent calculus, in S_2^1 and expressing that there does not exist a proof of contradiction, and proving this for all possible proofs of S_2^1. The usual approach does not work because it becomes impossible to feasibly evaluate the closed terms from the language of Bounded Arithmetic, as their values grow exponentially in their Gödel numbers in general.

However, the consistency of S_2^1 may still be demonstrated using a weaker notion of consistency. Pudlák shows in [15] that there does not exist a theory of Bounded Arithmetic that can prove the bounded consistency of S_2^1. Bounded consistency statements are weaker in the sense that they only refer to proofs that use bounded formulas, and are still too strong. Equational theories notion of a

proof is considered to be the weakest because we only reason in equational reasoning and not arbitrary formulas with unbounded quantifiers.

This leads to the conjecture by *Gaisi Takeuti* which states *"Let R be an equational theory involving equations $s = t$ where s and t are closed terms in the language of S_2, with natural rules based on recursive definitions of the function symbols. Show that $S_2 \nvdash Con(R)$"* [9]. If this conjecture were to be true, then consistency statements would likely be impractical to use to solve the finite axiomatizability problem of Bounded Arithmetic. Beckmann disproves this in [1], leaving hope that consistency statements can be used to obtain a negative answer, that Bounded Arithmetic is not finitely axiomatizable.

2.6 Proof System PV

PV was originally introduced in 1975 in [10]. PV relates propositional calculus proofs lengths to feasibly constructive proofs. The motivation for Cook developing PV came from the open problem $\mathcal{P}$ vs $\mathcal{NP}$ described in Section 2.1. He aimed to prove the inequality of the two classes. Also, a constructive proof of the statement $\forall_x A$ provides finding methods for an efficient proof of A for all possible values of x, but it does not say anything about the growth of the proofs length in terms of x. This issue lead Cook to the notion of feasibly constructive proofs. Cook showed that provable equations in PV are polynomial-verifiable, and that an equation $t = u$ of PV is provable in PV if and only if it is polynomially verifiable [10]. Cook uses equational theories for the definition of PV. He only reasoned about equations.

As we can recall in Section 2.2 Buss introduced S_2^1 in 1985. Buss proved in his thesis [3] that S_2^1 can Σ_1^b-define all the functions of PV and that S_2^1 is a conservative extension of PV and that S_2^1 and PV have the same Σ_1^b-formulae as theorems. It can also be viewed that PV is the equational version of S_2^1 [3].

L_p can be defined as the language containing L_e, defined later in Section 2.7, plus the symbols for all polynomial-time computable functions. $BASIC_p$ are the axioms $BASIC_e$ plus the axioms defining the polynomial time functions in terms of their definition by limited recursion on notation [5]. We can

also define a set of axioms known as the *equality axioms* known as;

1. $t = t$ (*Reflexivity*)
2. $s = t \to t = s$ (*Symmetry*)
3. $(s = t \wedge t = u) \to s = u$ (*Transitivity*)
4. $f \in L_p, ar(f) = k$
   $(\wedge_{i=1}^{k} s_i = t_i) \to f(s_1, ..., s_k) = f(t_1, ..., t_k)$ (*Function compatibility*)
5. *in particular for the predicate symbol* $\leq$ *we have*
   $s_1 = t_1, s_2 = t_2, t_1 \leq t_2 \to s_1 \leq s_2$

If we define 2 rules of inference, substitution and induction as;

$$\text{substitution}: \quad \frac{\Gamma(a) \Rightarrow \Delta(a)}{\Gamma(t) \Rightarrow \Delta(t)}$$

$$\text{induction}: \quad \frac{\Gamma, A(b) \Rightarrow A(b+1), \Delta}{\Gamma, A(0) \Rightarrow A(t(\overrightarrow{a})), \Delta}$$

we can then define PV precisely as

Definition 28. *PV is an equational theory which is determined by the language* L_p*, set of axioms* $BASIC_p$*, equality axioms, the rules of inference in the sequent calculus PK defined in Appendix B plus the rules of inference substitution and induction.*

Definition 29. *PV⁻ is an equational theory which is determined by the language* L_p*, set of axioms* $BASIC_p$*, equality axioms, the rules of inference in the sequent calculus PK defined in Appendix B.*

PV⁻ is the proof system PV without induction and substitution. That is, they do not use the induction rule that the original PV has. Buss and Ignjatovič claimed to prove that S_2^1 couldn't prove the consistency of PV⁻ but what they actually proved was that S_2^1 couldn't prove the consistency of

PV^- extended by propositional logic and $BASIC_e$. In Buss and Ignjatovič version of PV^-, each line in a proof is a sequent of boolean formulas. The language used were the function symbols for all in PV, and the axioms used were the recursive definitions of the functions in PV as well as the additional axioms $BASIC_e$. S_2^1 could not prove the consistency of this version of PV^-.

Beckmann however proves that S_2^1 can prove the consistency of PV^- [1]. The version of PV^- used in [1] is in pure equational theory, that is, every line in a proof is an equation. Beckmann proves that the consistency of any equational theory can be proven for any language if the only axioms used are recursive definitions for the symbols in that language. So for this case, the language would be the function symbols for all the functions in PV and the axioms used would be the recursive definitions of the functions in PV.

Yamagata claims to improve on Beckmann's result in [19]. He points out that Beckmann does not use substitution and Yamagata attempts to prove that S_2^1 can prove the consistency of PV^- including the substitution rule. He uses a slightly different version of PV however described in [11] where this version of PV uses binary notation of the natural numbers instead of the usual dyadic notation of the natural numbers.

2.7 Equational Theories

This project is interested in sets of axioms for equations. Each set of such axioms forms an equational theory in the following sense. An equational theory of a class of structures is a set of universal atomic formulas that hold in all members of the class. Given a language, a set of axioms and some proof system, an equational theory is the set of all equations provable from these.

If we take a notion of proof (such as sequent calculus for example), and a notion of arithmetization of proof and syntax (such as Gödel numbering), then a consistency statement expresses that for all potential proofs of a given theory, for each notion of proof there is not a proof of contradiction. For example, in the theory of numbers a contradiction would be $0 = 1$.

A *consistency* of a mathematical theory is a formal sentence expressing that the theory is error-

free. When we talk about *"consistency of an equational theory"*, we say there exists a formal sentence describing that the theory of equations is free of contradictions. For example, if a true consistency statement of a mathematical theory exists one cannot use that theory to prove the statement $1 = 0$ i.e. that $1 = 0$ is not derivable in that theory.

Theorem 8. *S_2^1 can prove consistency of a pure equational theory where the set of axioms are recursively defining the functions in a language [1].*

Beckmann shows in [1] that the weak fragment of Bounded Arithmetic S_2^1 can prove consistency of equational theories with functions defined via recursion. That is, given a set of axioms defining the functions in the language, the consistency of the equational theory given by closed instances of these axioms can be proved in S_2^1. Examples of these axioms include recursive axioms to define +; (1) $x + 0 = x$ (2) $x + Sy = S(x + y)$.

Pure equational theories are equational theories where every line written in the proof is an equation. Beckmann used pure equational theories to obtain his result in [1]. Beckmann uses a version of pure equational theory where substitution was omitted and axioms were only recursive definitions of the function symbols in the language. These functions will all have an arity, 0-arity functions will be constants such as 0 and 1, 1-arity functions will include functions such as the successor function, S. Functions with arity 2 will include functions such as addition and multiplication. From this language we can build terms and from these terms we can build formulas. For example, $s = t$ is a formula where s and t are both terms.

We develop a system used for our result which we call PI (Pure Inferences). PI consists of the *identity axioms* such as $s = s$. There are also the rules

Note : $u[x/s]$ denotes the term u where all occurrences of variable x in u are replaced by s;

$$\text{symmetry} : \frac{s = t}{t = s}$$

$$\text{transitivity}: \quad \frac{s = t \qquad t = u}{s = u}$$

$$\text{function extensionality}: \quad \frac{s = t}{u[x/s] = u[x/t]}$$

Definition 30. *PI consists of the* identity axioms*, the symmetry rule, transitivity rule and the function extensionality rule.*

PI will be used to define our result, which we call PET, in Chapter 4.

Expanded equational theory can be seen as an extension of pure equational theory. Expanded equational theory also allows inequalities and propositional connectives. Expanded equational theory must use a proof system that allows propositional rules of inference. The chosen proof system to write formal proofs in the Buss and Ignjatovič version of expanded equational theory is Gentzen-style sequent calculus. Each line in a expanded equational theory proof is not an equation, but a sequent of boolean formulas.

Buss and Ignjatovič show in [5] that no fragment of Bounded Arithmetic proves the bounded consistency of S_2^{-1} for proofs which contain only formulas as the least closure of Σ_i^b formulas under Boolean connectives and sharply bounded quantifiers. They worked with quantifier free theories rather than purely equational ones. The symbols of the language used were those in Bounded Arithmetic plus the additional set of symbols $\{2^a_{|b|}, \dot{-}, sq(a), < a, b >, (a)_1, (a)_2\}$. $2^a_{|b|}$ is the function $2^{\min(a,|b|)}$, is limited subtraction where $a \dot{-} b = a - b$ if $a \geq b$ or 0 otherwise. $sq(a) = a \cdot a$. $< a, b >$ is the pairing function. $(a)_1, (a)_2$ are projection functions. We call this new language L_e.

The additional axioms used in [5] were those in $BASIC_e$ which can be found in Appendix A.

Definition 31. *EET is the theory over the language L_e, which considers boolean formulas in L_e, and is given by the axioms $BASIC_e$ and the proof system PK as defined in Appendix B*

An example of an EET formula is $(\neg(a \leq b) \vee (a \dot{-} b = 0)) \wedge (\neg(a \dot{-} b = 0) \vee (a \leq b))$. Some more examples of EET formulas include all those in $BASIC$ which can be found in Appendix A.

Buss and Ignjatovič show in [5] that they could get rid of the induction axioms by translating into a boolean setting. We will not use induction, and EET does not consist of any induction axioms.

The Buss and Ignjatovič version of expanded equational theory (EET) uses this new language and set of axioms that extends what Beckmann used in [1], and the Gentzen-style sequent calculus.

B_i^b denotes the class of formulas obtained as the least closure of Σ_i^b formulas under boolean connectives and sharply bounded quantifiers [5]. The hierarchy of B_i^b-formulas is explained in [5]. A B_i^b-proof is a sequent calculus proof in which every formula is in B_i^b. Buss and Ignjatovič prove in [5] that S_2^i cannot prove the consistency of S_2^{-1} proofs in which only B_i^b-formulas appear.

Definition 32. *When we say Con we are referring to B_i^b-consistency as explained in [5]*

As a result of this, we also have the theorem that was proven in [5];

Theorem 9. $S_2^1 \nvdash Con(EET)$.

This thesis and project aims to develop a translation from EET into a pure equational theory (which we call PET). A well formed equation in PET is in the form $s = t$ where s and t are terms in the underlying language of PET. An example of a typical PET formula would be $\max(\min(b,1)\dot{-}a, \min(a,1)\dot{-}b) = 0$ *this is actually the translation of the formula* $(a \leftrightarrow b)$*, explained in Section 4.4.1.* In our version of PET we want to use the same language and rules used by Beckmann [1] with an additional set of axioms that simulate the propositional reasoning that was used in Buss and Ignjatovič EET. We want to make precise what this additional set of axioms are. A well formed equation in PET is in the form $s = t$ where s and t are terms in the underlying language of PET.

We also wish to prove that $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$, so that we may use Theorem 9 to prove that $S_2^1 \nvdash Con(PET)$.

2.8 Formalization in S_2^1

To formalize the translation of formulas and the translation of proofs in bounded arithmetic, we will need to use Gödel numbers. Gödelization is the process of encoding of syntax of some formal language

into the natural numbers. It uses numbers to represent reasoning in mathematics. We follow the same Gödel numbering system used by Buss in [3] to encode both EET and PET formulas and proofs.

Before we show the translation of formulas and proofs can be formalized in bounded arithmetic, we must show that the syntax can be encoded in S_2^1. We use binary notation to represent Gödel numbers. Each symbol of EET and PET will have its own Gödel number. Here we extend the way metamathematics is encoded in Buss uses in [3]. For EET we will use the following:

Logical Symbols

$\forall$	- 0,	$\vee$	- 5
$\exists$	- 1,	$($	- 6
$\neg$	- 2,	$)$	- 7
$\subset$	- 3,	,	- 8
$\wedge$	- 4,	$\rightarrow$	- 9

Non-logical Symbols

Constants:	0	- 16					
Unary Functions:	S	- 20,	$\|x\|$	- 24,	$\lfloor \frac{1}{2}x \rfloor$	- 28	
	$(a)_1$	- 26,	$(a)_2$	- 30,	$sq(a)$	- 34	
Binary Functions:	$+$	- 32,	$\cdot$	- 36,	$\#$	- 40	
	$2^a_{\|b\|}$	- 38,	$\dot{-}$	- 42,	$< a, b >$	- 44	
Binary Relation:	$=$	- 18,	$\leq$	- 22,			

Free Variables		*Free Variables*	
a_1	- 19	x_1	- 17
a_2	- 23	x_2	- 21
a_3	- 27	x_3	- 25
...	...	...	...

and similarly for PET where we have all the symbols for polynomial-time computable functions. We can use the Σ_1^b-defined function symbols and Δ_1^b-defined predicate symbols defined in [3] for handling Gödel numberings for the metamathematical concepts such as "formula" and "term". $Fmla(w)$ is a unary predicate which is Δ_1^b-defined in S_2^1 to mean w codes a formula. $Term(w)$ means that w encodes a term. The full inductive definition for these predicates can be found in [3].

Given a Σ_1^b-defined translation, tr, this translation should have the property: given p is the Gödel number of a EET formula, then $tr(p)$ is the gödel number of the equivalent PET equation, $p^* = 0$.

The translation of proofs works in a similar way. For a translation of proofs to be formalizable in S_2^1 we need a method of coding tree-like proofs in S_2^1. We can encode trees the same way in [3]. They can be coded by sequences, a tree is coded by a sequence with two special symbols "[" and "]" for denoting the structure of the tree.

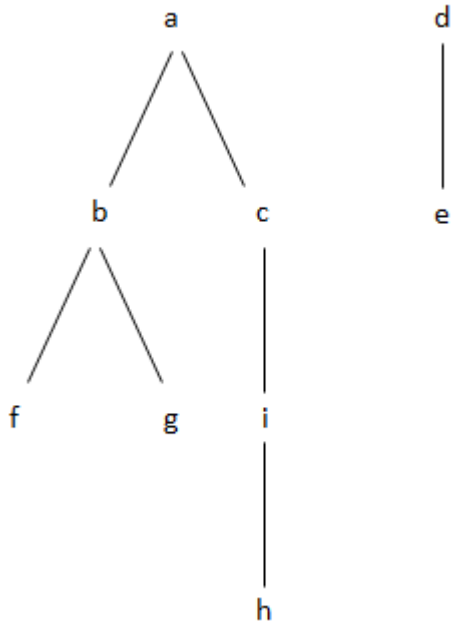

Fig 1: A tree that can be encoded as $a[b[fg]c[i[h]]]d[e]$

The sequence with special characters "[" and "]" can easily be encoded in S_2^1, with the Σ_1^b-defined

predciates and Δ_1^b-defined functions described in [3], including the function $SubTree(i, w)$ for extracting subtrees of w with root node i, the function $Father(i, w)$ that returns the father node of i (the node directly above it) in w, and the function $Son(k, j, w)$ that returns the k^{th} son of node j in tree w.

Now that EET formulas and trees can be encoded, we can encode EET tree-like proofs. A Gödel number of a proof codes a tree of sequents labeled precisely as to how the rules of inference in EET are applied. Each node of the tree is a ordered pair $< x, w >$ where w is the coded formula and x codes the inference rule used to deduce w from the sons of w. Here, the sons of w are the sequents directly above w in the proof tree in EET. Axioms in a proof will be encoded as a node with no son nodes. The root of the tree is the encoding of the sequent that is proved in the EET proof. Coding tree-like proofs in PET will be in a similar way except we use the encoding of equations and not sequents. The inductive definition of proofs as metamathematical objects in S_2^1 can be found in [3].

Now EET proofs and PET proofs can be encoded and defined metamathematically through the use of Σ_1^b-defined function symbols and Δ_1^b-defined predicate symbols, showing that these trees can be built in S_2^1.

Given a Σ_1^b-defined translation, tr, this translation should have the property: given p is the Gödel number of a EET proof with end sequent s, then $tr(p)$ is the gödel number of the equivalent PET proof of the equation, $s^* = 0$.

To make a Σ_1^b-defined translation, we want that the translation is defined by recursion. If this is achieved, then we achieve a function that follow the *p-inductive definitions* described in [3]. This means it would be a Σ_1^b-defined function, so that the translation would be formalizable in S_2^1.

Given an EET proof tree, we can recursively define a translation by first translating the sequent that is proven. This will be the root of the tree. The sons of the root of the tree will be the assumptions of the last rule used to obtain the proven sequent. This process is recursively repeated until we reach the axioms used in the proof. If a translation can be defined in this way, then it will be a *p-inductive* definition as defined in [3], making it a Σ_1^b-defined function meaning that the translation would be formalizable in S_2^1.

3. FIRST TRANSLATION

We aim for a translation of Boolean formulas over L_e into equations, which can be used between two equational theories, the result of Buss and Ignjatovič's paper [5] proving that $S_2^1 \nvdash PV^-$ extended by propositional logic and $BASIC_e$ axioms, which we call EET, and translating it into a pure equational setting. PET is the result of this paper, and when we refer to PET we are actually referring to what PET should be. A translation helps us achieve this because a translation means we have the Theorem $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$, and in combination with Theorem 9 to prove that $S_2^1 \nvdash Con(PET)$ by the extension of our new axioms required for the translation. In this Chapter we describe our first attempt of a translation, where we attempt to simulate propositional reasoning in a pure equational setting. We reach problems described later in this Chapter about how the axioms are inefficient and how we reach exponential growth because of how we define the translation and how many times we use the assumption in a proof, this decreases the likelihood that the proof is formalizable in S_2^1.

We begin by fixing a standard model for the natural numbers; where we have a structure with domain of the natural numbers and a standard interpretation. All non-logical symbols will have standard interpretation. Then;

Definition 33. *A formula is valid if and only if the formula is true in the standard model under any interpretation of its free variables.*

Let $\mathcal{T}$ be the set of terms such that; terms with no variables that are interpreted in our standard model to either 0 or 1 are in $\mathcal{T}$, and terms with variables are seen as functions with input, such that for any input the terms are evaluated and interpreted in the standard model of the natural numbers in the range $\{0, 1\}$.

We have some arbitrary first order language, $\mathcal{L}$, for the natural numbers, which will be used for both PET and EET. When we have a language, we can have a translation of formulas in EET into terms in PET. A translation is a map, $*$, from boolean formulas and sequents of boolean formulas, to terms.

Definition 34. *A* translation *is a map, $*$, from boolean formulas and sequents of boolean formulas to terms in $\mathcal{T}$*

The simplest of translations would be the translation of any boolean formula and sequent to the term 0. This is an example of a translation that lacks the properties of a *good translation.*

A good translation has two essential properties, which our translation need to have in order to be usable. First there is the *consistency property.* A translation satisfies the consistency property if and only if a boolean formula A is valid then its translation A^* equals 0, is valid. The property also applies to sequents, as in the sequent $\Gamma \Rightarrow \Delta$ is valid if and only if $(\Gamma \Rightarrow \Delta)^* = 0$, is valid.

We chose to fix 0 to represent *"truth"*, so any term that equals 0 is considered true.

Definition 35. *A translation, *, has the* consistency property *iff for a valid formula or sequent, A, $A^* = 0$ is a valid formula, and vice versa.*

From this we get the equivalence, A is valid iff $A^* = 0$ is valid. We also have a *provability property.* A statement, A, is provable in EET if and only if PET can prove that the translation, $A^* = 0$. Formally, EET $\vdash A$ implies PET $\vdash A^* = 0$. This also applies to sequents, EET $\vdash \Gamma \Rightarrow \Delta$ implies PET$\vdash (\Gamma \Rightarrow \Delta)^* = 0$.

Definition 36. *A translation, *, has the* provability property *iff EET $\vdash A$ implies PET $\vdash A^* = 0$, where A is a formula or sequent.*

Definition 37. *A* good translation *is one which has the consistency property and the provability property.*

So our goal for this project is to find a good translation that works that can also prove the rules of Gentzen style sequent calculus by reaching the conclusion of the rule from the translation of the

premise. As an additional condition we'll need this good translation also has to be formalizable in S_2^1 so that we can prove that $S_2^1 \nvdash Con(PET)$.

3.1 Translation of Boolean Formulas

To translate the extended setting into a pure equational setting one has to come up with a way to arithmetize all boolean connectives. For example, we know that for the formula $A \wedge B$ to be true both terms A and B have to be true. When translating this particular formula we have $(A \wedge B)^* = A^* + B^*$. If $A \wedge B$ then $A^* + B^* = 0$ and vice versa.

To ensure that the translation stays within the range $\{0, 1\}$, we use the sg-function. That is;

$sg(0) = 0$

$sg(x) = 1$ where x is any natural number greater than 0.

This way, we can always stay within the range $\{0, 1\}$ so that we stick to the definition of a translation.

An encoding for atomic formulas which are listed below, where s and t are terms.

$$\begin{aligned} (s = t)^* &= sg((s \dot{-} t) + (t \dot{-} s)) \\ (s \leq t)^* &= sg((s \dot{-} t)) \end{aligned}$$

We can then define a translation for the boolean connectives $\neg$, $\vee$, and $\wedge$ by recursion.

So if A is an atomic formula and we know the translation works for atomic formulas, we can define a translation for negation by recursion on the build-up of the formula A. $(\neg A)^* = 1 \dot{-} A^*$, where the next step is to look at the translation of the formula A.

We can also define conjunction and disjunction. $(A \vee B)^* = A^* \cdot B^*$ where the next recursive steps are to look at the translations of the formulas A and B. Also $(A \wedge B)^* = sg(A^* + B^*)$ where the next recursive steps are to look at the translations of the formulas A and B.

A summary of the way to arithmetize all the boolean connectives is as shown

$$\begin{aligned} (\neg A)^* &= 1 \dot{-} A^* \\ (A \vee B)^* &= A^* \cdot B^* \\ (A \wedge B)^* &= sg(A^* + B^*) \end{aligned}$$

3.2 Translation of Sequents

A sequent in the form $\Gamma \Rightarrow \Delta$ is true if some $A \in \Gamma$ is false or some $B \in \Delta$ is true. To arithmetize this first we need a way to encode to see for all formulas in Γ are true (or translations of formulas are equal to 0) and at least one of the formulas in Δ is true (or at least one of the translations of the formulas is equal to 0).

Γ and Δ are lists of sequents. $\Gamma = A_1, A_2...A_n$ and $\Delta = B_1, B_2...B_n$. Then by Γ^* we mean $A_1^*, A_2^*...A_n^*$ and by Δ^* we mean $B_1^*, B_2^*...B_n^*$. $(\Gamma \Rightarrow \Delta)^*$ is the translation from the sequent, $\Gamma \Rightarrow \Delta$, to a term, which we will define below.

We also need a definition for the minimum and maximum of an arbitrary length of a list of numbers for the translation of a sequent. We can recursively define $\min_{|x|}^{L/R}(x)$. If we state that

$\min_0() = 1$

$\min_1(x) = x$

$\min_{n+1}^{L}(x_0, x_1, ...x_n) = \min(x_0, \min_n(x_1, ...x_n))$.

$\min_{n+1}^{R}(x_0, x_1, ...x_n) = \min(\min_n(x_0, x_1, ...), x_n))$.

The maximum function can be defined in a similar way, such that

$\max_0() = 0$

$\max_1(x) = x$

$\max_{n+1}^{L}(x_0, x_1, ...x_n) = \max(x_0, \max_n(x_1, ...x_n))$.

$\max_{n+1}^{R}(x_0, x_1, ...x_n) = \max(\max_n(x_0, x_1, ...), x_n))$.

For simplicity, we will write $\min(\bar{x})$ instead of $\min_{|\bar{x}|}^{L/R}(\bar{x})$ and $\max(\bar{x})$ instead of $\max_{|\bar{x}|}^{L/R}(\bar{x})$.

If all formulas were true then the translation of these would all be 0. Therefore to check that all the formulas are true, one can find the maximum of the translated formulas and if it equals 0 then all the formulas are true. If we find that 1 is the maximum then it means at least one of the formulas is false. If we look for the minimum of the translated formulas, and we find a 0, then we know at least one of the formulas is true. If we find 1 for the minimum then none of the formulas are true. Thus, we define

Definition 38. $(\Gamma \Rightarrow \Delta)^*$ *is the translation from the sequent* $\Gamma \Rightarrow \Delta$ *to the term* $\min(\Delta^*) \dot{-} \max(\Gamma^*)$.

3.3 Consistency Property

The consistency property as defined in Definition 35 states that if the formula A is true, then it's translation $A^* = 0$ is true. Here we wish to prove that the translation suggested has the consistency property. For boolean connectives we can prove this by induction on the build-up of boolean formulas.

To prove our translation of sequents has the consistency property, we can analyse the different cases. Our translation states that $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*) \dot{-} \max(\Gamma^*)$. As a reminder, a sequent is only true if some $A \in \Gamma$ is false or some $B \in \Delta$ is true and the axiom would be false if all formulas in Γ are true and all formulas in Δ are false.

Lemma 1. *The translation,* $*$*, has the consistency property.*

Proof. The base case would be proving that the atomic formulas translate. For example, if A is $s = t$ then we want to prove that the translation is equal to 0, that if A is valid iff $A^* = 0$ is valid. This can be proven through explanation. Suppose s and t were equal, then we know that both $s \dot{-} t$ and $t \dot{-} s$ would be 0 and the equation would evaluate to 0. If they were not equal, then either $s \dot{-} t$ would be 0 and $t \dot{-} s$ would be greater than 0, or $s \dot{-} t$ would be greater than 0 and $t \dot{-} s$ would be 0. Either way, we know that $(s \dot{-} t) + (t \dot{-} s)$ could not be 0 if s and t are not equal.

A translation of the atomic formula $s \leq t$ we can prove is that $s \dot{-} t = 0$ when $s \leq t$ is true. Similar to above, is s was less than or equal to t then $s \dot{-} t$ would have to equal 0. If s was not less than or equal to t, or t was greater than s, then $s \dot{-} t$ would be greater than 0. Therefore we have proven that the consistency property holds for translation of the atomic formulas.

To prove that the translation of the boolean connectives holds the consistency property we would use an inductive step. One can use the assumption that A is true iff $A^* = 0$ is true, to prove that the translations of negation, conjunction and disjunction hold the consistency property.

For negation, we have the translation $(\neg A)^* = 1 \dot{-} A^*$. If A is true we expect $\neg A$ to be false, that is, $(\neg A)^* = 1$. If we use the induction hypothesis that $A^* = 0$, then $(\neg A)^* = 1 \dot{-} 0 = 1$. If A was false, and

that $A^* = 1$, then $\neg A$ would be true so $(\neg A)^*$ should equal 0. If we use the hypothesis that $(\neg A)^* = 0$, then $(\neg A)^* = 1 \dot{-} 1 = 0$. Therefore, the translation of the negation holds the consistency property.

For disjunction, we have the translation $(A \vee B)^* = A^* \cdot B^*$. If we use the assumption that $A^* = 0$, then $(A \vee B)^* = 0 \cdot B^*$. We are under the assumption that A is true, since A is true iff $A^* = 0$ is true, so we know that $A \vee B$ is also true, so we would expect $(A \vee B)^* = 0$. $(A \vee B)^* = 0 \cdot B^*$ and $0 \cdot B^*$ will always equal 0. We know that $A \vee B$ is false if A is false and B is false, therefore, $(A \vee B)^* = 1$ if $A^* = 1$ and $B^* = 1$. Since we know $A^* = 1$ and $B^* = 1$ and $(A \vee B)^* = 1 \cdot 1$, it's easy to see that $A^* \cdot B^* = 1$. Therefore the consistency property holds for translation of disjunction.

For conjunction, we have the translation $(A \wedge B)^* = A^* + B^*$. If we use the assumption, that $A^* = 0$, then $(A \wedge B)^* = 0 + B^*$. Since we are using the hypothesis that A is true, we know that $(A \wedge B)$ will only be true if B is also true. That is, $(A \wedge B)^* = 0$ if and only if $B^* = 0$ and $(A \wedge B)^* = 1$ if and only if $B^* = 1$. If we substitute $B^* = 0$ into $(A \wedge B)^* = 0 + B^*$ we would get $(A \wedge B)^* = 0 + 0 = 0$ and if we substitute $B^* = 1$ we would get $(A \wedge B)^* = 0 + 1 = 1$. If $A^* = 1$, then $(A \wedge B)^* = A^* + B^* = 1 + B^*$. It is easy to see that $1 + B^*$ can never equal 0, therefore, $(A \wedge B)^* = 1$. Therefore the consistency property holds for the translation of conjunction.

Now we are left to prove that the translation for the sequents holds the consistency property. The first case is where at least one of the formulas in Δ is true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*) \dot{-} \max(\Gamma^*)$ should equal 0. If at least one of the formulas in Δ is true, then Δ^* will have at least one 0 by induction hypothesis. Therefore, $\min(\Delta^*) = 0$. Therefore we are left with the term $0 \dot{-} \max(\Gamma^*)$. Regardless of value of $\max(\Gamma^*)$, we know that $0 \dot{-} \max(\Gamma^*) = 0$. Therefore this translation works for this case.

The next case is where not all the formulas in Γ are true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*) \dot{-} \max(\Gamma^*)$ should equal 0. If there is a formula in Γ that is false, the translated equation of that formula would equate to 1. Therefore $\max(\Gamma^*) = 1$. If none of the formulas in Δ are true, then all formulas translated would have equations that are equal to 1. Therefore, $\min(\Delta^*) = 1$. We are left with the equation in this case, $1 \dot{-} 1 = 0$. Therefore this translation works for this case.

The next case is where all the formulas in Γ are true and none of the formulas in Δ are true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*) \dot{-} \max(\Gamma^*)$ should equal 1. If all the formulas in Γ are true, then the translation

of these formulas will all have equations that equate to 0. Therefore, $\max(\Gamma^*) = 0$. If none of the formulas in Δ are true, then all formulas translated would have equations that equate to 1. Therefore, $\min(\Delta^*) = 1$. Therefore in this case, $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*) \dot{-} \max(\Gamma^*) = 1 \dot{-} 0 = 1$. Therefore this translation works for this case.

Therefore the translation, *, has the consistency property.

□

3.4 Finding Axioms

The translation must be able to prove each rule of EET in order to obtain the provability property. The rules of EET can be found in Appendix B. Therefore we will create axioms in PET that aid with these proofs. These axioms must be universally mathematically true, and will be a part of the system PET.

$$\neg\text{:left} \quad \frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}$$

Using our translation, our premise would be $\min(\Delta^*, A^*) \dot{-} \max(\Gamma^*) = 0$ and our conclusion would be $\min(\Delta^*) \dot{-} \max(1 \dot{-} A^*, \Gamma^*) = 0$. If we use the axiom

Axiom 1. $\min(\min(a, 1), b) \dot{-} c = \min(a, 1) \dot{-} \max(1 \dot{-} b, c)$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$ and $c = \Gamma^*$.

$$\neg\text{:right} \quad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}$$

Using our translation, out premise would be $\min(\Delta^*) \dot{-} \max(\Gamma^*, A^*) = 0$ and our conclusion would be $\min(\Delta^*, 1 \dot{-} A^*) \dot{-} \max(\Gamma^*) = 0$. If we use the axiom

Axiom 2. $\min(a, 1) \dot{-} \max(b, c) = \min(\min(a, 1), 1 \dot{-} b) \dot{-} c$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$ and $c = \Gamma^*$.

$$\wedge\text{:left} \quad \frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}$$

Using our translation, our premise would be $\min(\Delta^*) \dot{-} \max(A^*, B^*, \Gamma^*) = 0$ and our conclusion would be $\min(\Delta^*) \dot{-} \max(A^* + B^*, \Gamma^*) = 0$. Using the axiom

Axiom 3. $\min(a, 1) \dot{-} \max(b, \max(c, d)) = \min(a, 1) \dot{-} \max(b + c, d)$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$, $c = B^*$ and $d = \Gamma^*$.

$$\wedge\text{:right} \quad \frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}$$

Using our translation, the premises of this rule would be $\min(\Delta^*, A^*) \dot{-} \max(\Gamma^*) = 0$ and $\min(\Delta^*, B^*) \dot{-} \max(\Gamma^*) = 0$. The conclusion would be $\min(\Delta^*, A^* + B^*) \dot{-} \max(\Gamma^*) = 0$. Using the axiom

Axiom 4. $\max((\min(\min(a, 1), b) \dot{-} c), (\min(\min(a, 1), d) \dot{-} c)) = \min(\min(a, 1), b + d) \dot{-} c$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$, $d = B^*$ and $c = \Gamma^*$.

$$\vee\text{:left} \quad \frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}$$

Using our translation, the premises of this rule would be $\min(\Delta^*) \dot{-} \max(A^*, \Gamma^*) = 0$ and $\min(\Delta^*) \dot{-} \max(B^*, \Gamma^*) = 0$. The conclusion would be $\min(\Delta^*) \dot{-} \max(A^* \cdot B^*, \Gamma^*) = 0$. Using the axiom

Axiom 5. $\max((\min(a, 1) \dot{-} \max(b, c)), (\min(a, 1) \dot{-} \max(d, c))) = \min(a, 1) \dot{-} \max(b \cdot d, c)$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$, $d = B^*$ and $c = \Gamma^*$.

$$\vee\text{:right}\ \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}$$

Using our translation, we would have the premise $\min(\Delta^*, A^*, B^*) \dot{-} \max(\Gamma^*) = 0$ and we would have the conclusion $\min(\Delta^*, A^* \cdot B^*) \dot{-} \max(\Gamma^*) = 0$. Using the axiom

Axiom 6. $\min(\min(\min(a, 1), b), d) \dot{-} c = \min(\min(a, 1), b \cdot d) \dot{-} c$

then we can reach the conclusion by using this axiom and substituting $a = \Delta^*$, $b = A^*$, $d = B^*$ and $c = \Gamma^*$.

$$\text{exchange:left}\ \frac{\Gamma, A, B, \Gamma' \Rightarrow \Delta}{\Gamma, B, A, \Gamma' \Rightarrow \Delta}$$

The premise of this rule, under our translation, would be $\min(\Delta^*) \dot{-} \max(\Gamma^*, A^*, B^*, \Gamma'^*) = 0$ and the conclusion would be $\min(\Delta^*) \dot{-} \max(\Gamma^*, B^*, A^*, \Gamma'^*) = 0$.

If we look at the definitions we gave for *max*, we recursively call smaller versions of the list by removing the first or last element and comparing that the maximum of the rest of the list. Looking at out conclusion, can recursively call the *max* function until we are left with $\max(B^*, A^*)$. We can then use the axiom

Axiom 7. $\max(x, y) = \max(y, x)$

to make the conclusion look like the premise, $\min(\Delta^*) \dot{-} \max(\Gamma^*, A^*, B^*, \Gamma'^*) = 0$.

$$\text{exchange:right}\ \frac{\Gamma \Rightarrow \Delta, A, B, \Delta'}{\Gamma \Rightarrow \Delta, B, A, \Delta'}$$

The premise of this rule, under our translation, would be $\min(\Delta^*, A^*, B^*, \Gamma'^*) \dot{-} \max(\Gamma^*) = 0$ and the conclusion would be $\min(\Delta^*, B^*, A^*, \Gamma'^*) \dot{-} \max(\Gamma^*) = 0$.

Using similar logic to Axiom 7, we can recursively call smaller lists until we reach $\min(B^*, A^*)$. Then we can use an axiom;

Axiom 8. $\min(x, y) = \min(y, x)$

to prove that the conclusion matches the premise.

$$\text{contraction:left} \quad \frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}$$

The premise for this rule, under our translation, would be $\min(\Delta^*) \dot{-} \max(A^*, A^*, \Gamma^*) = 0$. Our conclusion would be $\min(\Delta^*) \dot{-} \max(A^*, \Gamma^*) = 0$.

Using similar logic as Axioms 7, we can recursively call smaller lists of max in the premise until we reach $\max(A^*, A^*)$. Then using a new axiom

Axiom 9. $\max(x, x) = x$

we can make the premise match the conclusion.

$$\text{contraction:right} \quad \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}$$

The premise for this rule, under our translation, would be $\min(\Delta^*, A^*, A^*) \dot{-} \max(\Gamma^*) = 0$. Our conclusion would be $\min(\Delta^*, A^*) \dot{-} \max(\Gamma^*) = 0$.

Using similar logic as Axioms 8, we can recursively call smaller lists of min in the premise until we reach $\min(A^*, A^*)$. Then using a new axiom

Axiom 10. $\min(x, x) = x$

we can make the premise match the conclusion.

Some of these axioms created aren't very elegant and are quite complicated. This makes the probability that the translation is formalizable in S_2^1 unlikely, which is bad because this is what we're aiming for. One may consider revising the translation used and try an alternative translation that has simple, more elegant axioms.

3.5 Criticism of Translation

Here we point out some flaws with the current suggested translation *. We explain why some of these flaws may make * an unsuitable translation for what we are trying to achieve, as it is likely that * is not formalizable in S_2^1. This starts off by proving that $\max_n^L(\bar{x}) = \max_n^R(\bar{x})$ in a pure equational setting using PI defined in Section 2.7. Because we want to use both $\max_n^L(\bar{x})$ and $\max_n^R(\bar{x})$ interchangeably, a proof of this would be used frequently and here we show why this proof creates a problem that would make it unlikely that $*$ is formalizable in S_2^1.

Lemma 2. $\max_n^L(\bar{x}) = \max_n^R(\bar{x})$ *is provable in PET for all natural numbers* n.

Proof. By inudction on n, we can show that $\max_n^L(\bar{x}) = \max_n^R(\bar{x})$ has a proof in PET. First we can try the two base cases, when $n = 0$ and $n = 1$.

If $n = 0$, then $\max_0^L() \equiv 0 \equiv \max_0^R()$. This can be proven in PET using the identity axioms.

If $n = 1$, then $\max_1^L(x) \equiv x \equiv \max_1^R(x)$. This can be proven in PET using the identity axioms.

Now we can prove the inductive step. That is, assuming the induction hypothesis $\max_n^L(x_0, \bar{x}) = \max_n^R(x_0, \bar{x})$ is provable in PET we can prove $\max_{n+1}^L(x_0, \bar{x}, x_n) = \max_{n+1}^R(x_0, \bar{x}, x_n)$ is provable in PET. Note that $\bar{x}$ are the terms $x_1, x_2, ..., x_{n-1}$. Let $\pi_n(\bar{x})$ denote the proof of the induction hypothesis $\max_n^L(\bar{x}) = \max_n^R(\bar{x})$.

$$\text{ext}\ \frac{\pi_n(x_0, \bar{x})}{\max(\max_n^L(x_0, \bar{x}), x_n) = \max(\max_n^R(x_0, \bar{x}), x_n)}$$

$$\text{ext}\ \frac{\pi_{n-1}(\bar{x})}{\max(\max(x_0, \max^L_{n-1}(\bar{x})), x_n) = \max(\max(x_0, \max^R_{n-1}(\bar{x})), x_n)}$$

Since $\max(\max^L_n(x_0, \bar{x}), x_n)$ and $\max(\max(x_0, \max^L_{n-1}(\bar{x})), x_n)$ are equivalent terms and are equal in the language level, we can use the transitivity rule

$$\text{symmetry}\ \frac{\max(\max^L_n(x_0, \bar{x}), x_n) = \max(\max^R_n(x_0, \bar{x}), x_n)}{\max(\max^R_n(x_0, \bar{x}), x_n) = \max(\max^L_n(x_0, \bar{x}), x_n)}$$

Now let α_1 be the equation $\max(\max^R_n(x_0, \bar{x}), x_n) = \max(\max^L_n(x_0, \bar{x}), x_n)$ and α_2 be the equation $\max(\max^L_n(x_0, \bar{x}), x_n) = \max(\max(x_0, \max^R_{n-1}(\bar{x})), x_n)$.

$$\text{transitivity}\ \frac{\alpha_1 \qquad \alpha_2}{\max(\max^R_n(x_0, \bar{x}), x_n) = \max(\max(x_0, \max^R_{n-1}(\bar{x})), x_n)}$$

Now if we use the induction hypothesis

$$\text{ext}\ \frac{\pi_n(\bar{x}, x_n)}{\max(x_0, \max^L_n(\bar{x}, x_n)) = \max(x_0, \max^R_n(\bar{x}, x_n))}$$

Since $\max(x_0, \max^R_n(\bar{x}, x_n))$ and $\max(\max(x_0, \max^R_{n-1}(\bar{x})), x_n)$ are equivalent terms in the language level, we can use the transitivity rule.

$$\text{symmetry}\ \frac{\max(x_0, \max^L_n(\bar{x}, x_n)) = \max(x_0, \max^R_n(\bar{x}, x_n))}{\max(x_0, \max^R_n(\bar{x}, x_n)) = \max(x_0, \max^L_n(\bar{x}, x_n))}$$

Now let α_3 be the equation $\max(\max^R_n(x_0, \bar{x}), x_n) = \max(\max(x_0, \max^R_{n-1}(\bar{x})), x_n)$ and α_4 be the equation $\max(x_0, \max^R_n(\bar{x}, x_n)) = \max(x_0, \max^L_n(\bar{x}, x_n))$.

$$\text{transitivity}\ \frac{\alpha_3 \qquad \alpha_4}{\max(\max^R_n(x_0, \bar{x}), x_n) = \max(x_0, \max^L_n(\bar{x}, x_n))}$$

Since $\max^R_{n+1}(x_0, \bar{x}, x_n) = \max(\max^R_n(x_0, \bar{x}), x_n)$ and $\max^L_{n+1}(x_0, \bar{x}, x_n) = \max(x_0, \max^L_n(\bar{x}, x_n))$, we have proven that $\max^R_{n+1}(x_0, \bar{x}, x_n) = \max^L_{n+1}(x_0, \bar{x}, x_n)$ has a proof in PET and therefore we have proven our Lemma by induction.

□

Lemma 3. $\min^L_n(\bar{x}) = \min^R_n(\bar{x})$ *is provable in PET for all natural numbers* n.

The proof for this lemma can be obtained by using a similar proof to Lemma 2.

If we look at the complexity of this proof we can see that this chosen method becomes too complicated for the PET proof system and is even *exponential*. If we say that induction hypothesis, π_n, has a proof of length S_n and π_{n-1} has a proof of length S_{n-1}, then we can look at the length of the proof S_{n+1} as a recursive formula $2S_n + S_{n-1} + C$ where C is some constant that counts the number of transitivity, extensionality and symmetry rules used. We can argue that $S_n = \Omega(2^n)$, if we use induction hypothesis $S_n \geq C \cdot 2^n$, we know that $S_{n+1} \geq 2 \cdot 2_n$, we can use the induction hypothesis here to say that $S_{n+1} \geq 2 \cdot C \cdot 2^n \geq C \cdot 2^{n+1}$, hence $S_n = \Omega(2^n)$. This creates an exponential growth as n increases. This is bad because we are aiming for a polynomial translation, as an exponential translation would mean that the equational theories created are not formalizable in S^1_2 as exponential procedures can not be formalizable in S^1_2.

This leaves us with the task of redefining the terms *max* and *min* such that we can still prove the consistency property and prove the provability property.

3.6 Evaluation of Translation

The aim of a translation was to translate Buss and Ignjatovič's result in [5] which we call EET into some pure equational setting which we call PET. We still want our result to be formalizable in S^1_2 like EET. This translation uses a mixture of $\max^L_n$, $\max^R_n$, $\min^L_n$ and $\min^R_n$ which causes an exponential growth when we want a polynomial translation to keep the result formalizable in S^1_2. One way of tackling this problem would be to choose only one style for *min* and one style for *max* throughout the whole of the translation.

We create some axioms for the translation that are complicated, which can lead to probability that this translations is not formalizable in S_2^1 because these axioms are so specific. If we look at the translations of conjunction and disjunction, we can replace these with a new translation. A conjunction statement $A \wedge B$ can be translated as $\max(A^*, B^*)$ and a disjunction statement $A \vee B$ can be translated as $\min(A^*, B^*)$. This can help towards more elegant and simpler axioms.

4. IMPROVED TRANSLATION

This is the second attempt of a translation we do. We address the problems and issues raised in Chapter 3. We redefine the translation hoping to create more elegant axioms and we focus on using assumptions only exactly once in proofs. We redefine the translation $*$ in the following sections to be a translation from boolean formulas and sequents of boolean formulas into terms such that the range of the terms is $\{0, 1\}$.

4.1 Translation of Boolean Formulas

We keep the same translation for the atomic formulas as defined in Section 3.1. We now make changes to how we translate the boolean connectives, $\neg, \wedge$ and $\vee$.

We define a translation for the boolean connectives $\neg$, $\vee$, and $\wedge$ by recursion. We know the translations for the atomic formulas hold.

So if A is an atomic formula and we know the translation works for atomic formulas, we can define a translation for negation by recursion on the build-up of the formula A. $(\neg A)^* = 1 \dot{-} A^*$, as before, where the next step is to look at the translation of the formula A.

We can also define conjunction and disjunction. $(A \vee B)^* = \min(A^*, B^*)$ where the next recursive steps are to look at the translations of the formulas A and B. Also $(A \wedge B)^* = \max(A^*, B^*)$ where the next recursive steps are to look at the translations of the formulas A and B.

A summary of the way to arithmetize all the boolean connectives is as shown

$$\begin{aligned}(\neg A)^* &= 1 \dot{-} A^* \\ (A \vee B)^* &= \min(A^*, B^*) \\ (A \wedge B)^* &= \max(A^*, B^*)\end{aligned}$$

4.2 Translation of Sequents

We make slight changes to this translation, especially from those described in Section 3.2. We change how we define *min* and *max* and how we translate a sequent. We change the definition for the minimum and maximum of an arbitrarily long length of a list of numbers for the translation of a sequent. We use only the definitions $\min_n^R$ and $\max_n^L$ as before. We can recursively define $\min(x, y)$. If we state that

$\min_0() = 1$

$\min_1(x) = x$

$\min_{n+1}(x_0, x_1, ...x_n) = \min(\min_n(x_0, x_1, ...), x_n))$.

The maximum function can be defined in a similar way, such that

$\max_0() = 0$

$\max_1(x) = x$

$\max_{n+1}(x_0, x_1, ...x_n) = \max(x_0, \max_n(x_1, ...x_n))$.

For simplicity, we will write $\min(\bar{x})$ instead of $\min_{|\bar{x}|}(\bar{x})$ and $\max(\bar{x})$ instead of $\max_{|\bar{x}|}(\bar{x})$.

If all formulas were true then the translation of these would be a list of 0's. Therefore to check that all the formulas are true, one can find the maximum of the translated formulas and if it equals 0 then all the formulas are true. If we find that 1 is the maximum then it means at least one of the formulas is false. If we look for the minimum of the translated formulas, and it is 0, then we know at least one of the formulas is true. If we find 1 for the minimum then none of the formulas are true. Thus, we define

Definition 39. $(\Gamma \Rightarrow \Delta)^*$ *is the translation from the sequent* $\Gamma \Rightarrow \Delta$ *to the term* $\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$.

4.3 Consistency Property

The consistency property, as defined in Definition 35, states that if the formula A is true, then it's translation $A^* = 0$ is true. Here we wish to prove that the translation suggested has the consistency property by proving that our translation of the boolean connective and sequents have the consistency property. For boolean connectives we can prove this by induction on the build-up of boolean formulas.

To prove our translation of sequents has the consistency property, we can analyse the different cases. Our translation states that $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$. As a reminder, a sequent is only true if some $A \in \Gamma$ is false or some $B \in \Delta$ is true.

Lemma 4. *The translation, $*$, has the consistency property.*

Proof. The base case would be proving that the atomic formulas translate. For example, if A is $s = t$ then we want to prove that the translation is equal to 0, that if A is valid iff $A^* = 0$ is valid. This can be proven through explanation. Suppose s and t were equal, then we know that both $s \dot{-} t$ and $t \dot{-} s$ would be 0 and the equation would evaluate to 0. If they were not equal, then either $s \dot{-} t$ would be 0 and $t \dot{-} s$ would be greater than 0, or $s \dot{-} t$ would be greater than 0 and $t \dot{-} s$ would be 0. Either way, we know that $\max(s \dot{-} t, t \dot{-} s)$ could not be 0 if s and t are not equal.

A translation of the atomic formula $s \leq t$ we can prove is that $s \dot{-} t = 0$ when $s \leq t$ is true. Similar to above, is s was less than or equal to t then $s \dot{-} t$ would have to equal 0. If s was not less than or equal to t, s t was greater than s, then $s \dot{-} t$ would be greater than 0. Therefore we have proven that the consistency property holds for translation of the atomic formulas.

To prove that the translation of the boolean connectives holds the consistency property we would use an inductive step. One can use the induction hypothesis that $A^* = 0$ to prove that the translations of negation, conjunction and disjunction hold the consistency property.

For negation, we have the translation $(\neg A)^* = 1 \dot{-} A^*$. If A is true we expect $\neg A$ to be false, that is, $(\neg A)^* = 1$. If we use the assumption that $A^* = 0$, then $(\neg A)^* = 1 \dot{-} 0 = 1$. If A was false, and that $A^* = 1$, then $\neg A$ would be true so $(\neg A)^*$ should equal 0. If we use the assumption that $(\neg A)^* = 0$, then $(\neg A)^* = 1 \dot{-} 1 = 0$. Therefore, the translation of the negation holds the consistency property.

For disjunction, we have the translation $(A \vee B)^* = \min(A^*, B^*)$. If we use the assumption that $A^* = 0$, then $(A \vee B)^* = \min(0, B^*)$. Since we are using the assumption that A is true, we know that $A \vee B$ is also true, so we would expect $(A \vee B)^* = 0$. $(A \vee B)^* = \min(0, B^*)$ and $\min(0, B^*)$ will always equal 0. We know that $A \vee B$ is false if A is false and B is false, therefore, $(A \vee B)^* = 1$ if $A^* = 1$ and $B^* = 1$. Since we know $A^* = 1$ and $B^* = 1$ and $(A \vee B)^* = \min(A^*, B^*)$, it's easy to see that $\min(A^*, B^*) = 1$. Therefore the consistency property holds for translation of disjunction.

For conjunction, we have the translation $(A \wedge B)^* = \max(A^*, B^*)$. If we use the assumption, that $A^* = 0$, then $(A \wedge B)^* = \max(0, B^*)$. Since we are using the assumption that A is true (since we are assuming that A is true iff $A^* = 0$ is true), we know that $(A \wedge B)$ will only be true if B is also true. That is, $(A \wedge B)^* = 0$ if and only if $B^* = 0$ and $(A \wedge B)^* = 1$ if and only if $B^* = 1$. If we substitute $B^* = 0$ into $(A \wedge B)^* = \max(0, B^*)$ we would get $(A \wedge B)^* = \max(0, 0) = 0$ and if we substitute $B^* = 1$ we would get $(A \wedge B)^* = \max(0, 1) = 1$. If $A^* = 1$, then $(A \wedge B)^* = \max(A^*, B^*) = \max(1, B^*)$. It is easy to see that $\max(1, B^*)$ can never equal 0, therefore, $(A \wedge B)^* = 1$. Therefore the consistency property holds for the translation of conjunction.

Now we are left to prove that the translation for the sequents holds the consistency property. The first case is where at least one of the formulas in Δ is true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$ should equal 0. If at least one of the formulas in Δ is true, then Δ^* will have at least one 0 by induction hypothesis. Therefore, $\min(\Delta^*, 1) = 0$. Therefore we are left with the term $0 \dot{-} \max(\Gamma^*)$. Regardless of value of $\max(\Gamma^*)$, we know that $0 \dot{-} \max(\Gamma^*) = 0$. Therefore this translation works for this case.

The next case is where not all the formulas in Γ are true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$ should equal 0. If there is a formula in Γ that is false, the translated equation of that formula would equate to 1. If none of the formulas in Δ are true, then all formulas translated would have equations that are equal to 1. Therefore, $\min(\Delta^*, 1) = 1$. We are left with the equation in this case, $1 \dot{-} 1 = 0$. Therefore this translation works for this case.

The next case is where all the formulas in Γ are true and none of the formulas in Δ are true. Then $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$ should equal 1. If all the formulas in Γ are true, then the translation of these formulas will all have values 0. Therefore, $\max(\Gamma^*) = 0$. If none of the formulas in Δ are true,

then all formulas translated would have equations that equate to a value greater than 0. Therefore, $\min(\Delta^*, 1) = 1$. Therefore in this case, $(\Gamma \Rightarrow \Delta)^* = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 1 \dot{-} 0 = 1$. Therefore this translation works for this case.

Therefore the translation, *, has the consistency property.

□

4.4 The Provability Property

The provability property, defined in Definition 36, states that if EET $\vdash A$ then PET $\vdash A^* = 0$, and if PET $\vdash A^* = 0$ then EET $\vdash A$, where A is a formula. Here we wish to prove that the suggested translation has the provability property.

We intend to show this by looking at EET-proofs. Looking at the definition of EET in Definition 31, every proof will include the axioms found in Appendix B so that the proof tree can have a start node. Each proof can use the axioms of $BASIC_e$ as well. Also proofs use the inference rules of PK found in Appendix B so that a node may produce children nodes, and may again use those nodes to produce further nodes by using axioms and inference rules. For some formula or sequent A, if EET $\vdash A$ then A can be proven from the axioms and inference rules described. We want to translate the axioms $BASIC_e$ and add them to the axioms of PET. We want to then prove the translation of the conclusion of every inference rule in PK from the translation of the assumption using the translation of the assumption at most once. This includes creating additional axioms for PET that each proof from the translation of the conclusion for every inference rule to use. This may increase the size of the original proof, but we can estimate the size of the PET proofs and ensure the size of the PET proof is bounded by a polynomial of the size of the original EET proof.

We want that the transformation is formalizable in S_2^1, so if we show that the transformation is done locally and in polynomial time, and since S_2^1 can define all polynomial time procedures [3], we show that S_2^1 is able to formalize this transformation.

If we can prove that for any EET-proof of a formula or sequent, A, there is an equivalent PET-proof

for $A^* = 0$, then we prove that the translation has the provability property.

4.4.1 Axioms

To prove the translation has the provability property we need to translate the axioms in $BASIC_e$ and develop a pure equational theory setting proof for each inference rule of PK, possibly creating additional axioms. We call the set of translated axioms $BASIC_t$ and these will be a part of our result, PET, defined later in Section 4.5.

To translate these axioms, we will use the following guide;

$$\begin{aligned}
(a = b)^* &= \max(a \dot{-} b, b \dot{-} a) = 0 \\
(a \leq b)^* &= a \dot{-} b = 0 \\
(a \to b)^* &= \min(b, 1) \dot{-} a = 0 \\
(a \leftrightarrow b)^* &= \max(\min(b, 1) \dot{-} a, \min(a, 1) \dot{-} b) = 0 \\
(a \neq b)^* &= 1 \dot{-} \max(a \dot{-} b, b \dot{-} a) = 0 \\
(a \wedge b)^* &= \max(a, b) = 0 \\
(a \vee b)^* &= \min(a, b) = 0
\end{aligned}$$

First there is the translation of the axioms in $BASIC$ found in Appendix A. $BASIC_t$ consists of the following translated axioms. Note that 1 and 2 are used to abbreviate S0 and SS0.

Each formula is labeled with a number, which corresponds to formulas in $BASIC$. If we take the first formula in $BASIC$ and translate it for example;

$(y \leq x \to y \leq Sx)^* \equiv \min(y \dot{-} Sx, 1) \dot{-} (y \dot{-} x) = 0$

Definition 40. *$BASIC_t$ consists of the following 46 formulas*

1. $\min(y \dot{-} Sx, 1) \dot{-} (y \dot{-} x) = 0$
2. $1 \dot{-} (\max(x \dot{-} Sx, Sx \dot{-} x)) = 0$
3. $0 \dot{-} x = 0$
4. $\max((\min(\max(x \dot{-} y, 1 \dot{-} (\max(x \dot{-} y, y \dot{-} x))), 1) \dot{-} (Sx \dot{-} y)), (\min(Sx \dot{-} y, 1)$

$\dot{-}\max(x\dot{-}y, 1\dot{-}(\max(x\dot{-}y, y\dot{-}x))))) = 0$

5. $\min(1\dot{-}(2\cdot x\dot{-}0, 0\dot{-}2\cdot x), 1)\dot{-}(1\dot{-}(x\dot{-}0, 0\dot{-}x)) = 0$
6. $\min(y\dot{-}x, x\dot{-}y) = 0$
7. $\max(x\dot{-}y, \min(\max(x\dot{-}y, y\dot{-}x), 1)\dot{-}y\dot{-}x) = 0$
8. $\min(x\dot{-}z, 1)\dot{-}\max(x\dot{-}y, y\dot{-}z) = 0$
9. $\max(|0|\dot{-}0, 0\dot{-}|0|) = 0$
10. $\min(\max(\max(|2\cdot x|\dot{-}S(|x|), S(|x|)\dot{-}|2\cdot x|), \max(|S(2\cdot x)|\dot{-}S(|x|), S(|x|)\dot{-}|S(2\cdot x)|)), 1)$
$\dot{-}(1\dot{-}(\max(x\dot{-}0, 0\dot{-}x))) = 0$
11. $\max(|S0|\dot{-}S0, S0\dot{-}|S0|) = 0$
12. $\min(|x|\dot{-}|y|, 1)\dot{-}(x\dot{-}y) = 0$
13. $\max(|x\#y|\dot{-}S(|x|\cdot|y|), S(|x|\cdot|y|)\dot{-}|x\#y|) = 0$
14. $\max(|0\#y|\dot{-}S0, S0\dot{-}|0\#y|) = 0$
15. $\min(\max(\max(1\#(2.x)\dot{-}2(1\#x), 2(1\#x)\dot{-}1\#(2.x)), \max(1\#(S(2.x))$
$\dot{-}2(1\#x), 2(1\#x)\dot{-}1\#(S(2.x)))), 1)\dot{-}(1\dot{-}(\max(x\dot{-}0, 0\dot{-}x))) = 0$
16. $\max(x\#y\dot{-}y\#x, y\#x\dot{-}x\#y) = 0$
17. $\min(\max(x\#z\dot{-}y\#z, y\#z\dot{-}x\#z), 1)\dot{-}\max(|x|\dot{-}|y|, |y|\dot{-}|x|) = 0$
18. $\min(\max(x\#y\dot{-}(u\#y)\cdot(v\#y), (u\#y)\cdot(v\#y)\dot{-}x\#y), 1)\dot{-}\max(|x|\dot{-}|u|+|v|, |u|+|v|\dot{-}|x|) = 0$
19. $x\dot{-}(x+y) = 0$
20. $\min(\max(S(2\cdot x)\dot{-}2\cdot y, 1\dot{-}\max(S(2\cdot x)\dot{-}2.y, 2\cdot y\dot{-}S(2.x))), 1)\dot{-}\max(x\dot{-}y, 1\dot{-}\max(x\dot{-}y, y\dot{-}x)) = 0$
21. $\max((x+y)\dot{-}(y+x), (y+x)\dot{-}(x+y)) = 0$
22. $\max((x+0)\dot{-}x, x\dot{-}(x+0)) = 0$
23. $\max(x+Sy\dot{-}S(x+y), S(x+y)\dot{-}x+Sy) = 0$
24. $\max(x+(y+z)\dot{-}(x+y)+z, (x+y)+z\dot{-}x+(y+z)) = 0$
25. $\max(\min(y\dot{-}z, 1)\dot{-}((x+y)\dot{-}(x+z)), \min(((x+y)\dot{-}(x+z)), 1)\dot{-}y\dot{-}z) = 0$
26. $\max((x\cdot 0)\dot{-}0, 0\dot{-}(x\cdot 0)) = 0$
27. $\max(x\cdot(Sy)\dot{-}(x\cdot y)+x, (x\cdot y)+x\dot{-}x\cdot(Sy)) = 0$
28. $\max((x\cdot y)\dot{-}(y\cdot x), (y\cdot x)\dot{-}(x\cdot y)) = 0$

29. $\max(x \cdot (y+z) \dot{-} (x \cdot y) + (x \cdot z), (x \cdot y) + (x \cdot z) \dot{-} x \cdot (y+z)) = 0$

30. $\min(\max(\min(y \dot{-} z, 1) \dot{-} x \cdot y \dot{-} x \cdot z, \min(x \cdot y \dot{-} x \cdot z, 1) \dot{-} y \dot{-} z), 1) \dot{-} (S0 \dot{-} x) = 0$

31. $\min(\max(S(\lfloor \frac{x}{2} \rfloor) \dot{-} |x|, |x| \dot{-} S(\lfloor \frac{x}{2} \rfloor)), 1) \dot{-} (1 \dot{-} \max(x \dot{-} 0, 0 \dot{-} x)) = 0$

32. $\max(\min(\max(\lfloor \frac{y}{2} \rfloor \dot{-} x, x \dot{-} \lfloor \frac{y}{2} \rfloor), 1) \dot{-} \min(\max(2.x \dot{-} y, y \dot{-} 2.x), \max(S(2.x) \dot{-} y, y \dot{-} S(2.x))),$
$\min(\min(\max(2.x \dot{-} y, y \dot{-} 2.x), \max(S(2.x) \dot{-} y, y \dot{-} S(2.x))), 1) \dot{-} \max(\lfloor \frac{y}{2} \rfloor \dot{-} x, x \dot{-} \lfloor \frac{y}{2} \rfloor)) = 0$

Also the translation of the additional axioms in $BASIC_e$:

33. $|a| \dot{-} a = 0$

34. $|a \cdot b| \dot{-} (|a| + |b|) = 0$

35. $\max(2^a_{|0|} \dot{-} 1, 1 \dot{-} 2^a_{|0|}) = 0$

36. $\max(2^0_{|c|} \dot{-} 1, 1 \dot{-} 2^0_{|c|}) = 0$

37. $\min(\max(2^{a+b}_{|c|} \dot{-} 2^a_{|c|} \cdot 2^b_{|c|}, 2^a_{|c|} \cdot 2^b_{|c|} \dot{-} 2^{a+b}_{|c|}), 1) \dot{-} (a + b \dot{-} |c|) = 0$

38. $\min(\max(\max(2^1_{|c|} \dot{-} 2, 2 \dot{-} 2^1_{|c|}), 2^a_{|c|} \dot{-} 2 \cdot c), 1) \dot{-} (1 \dot{-} \max(c \dot{-} 0, 0 \dot{-} c)) = 0$

39. $\max(\min(\max(a \dot{-} b \dot{-} 0, 0 \dot{-} a \dot{-} b), 1) \dot{-} a \dot{-} b, \min(a \dot{-} b, 1) \dot{-} \max(a \dot{-} b \dot{-} 0, 0 \dot{-} a \dot{-} b)) = 0$

40. $\max(\min(\max(a \dot{-} b \dot{-} 0, 0 \dot{-} a \dot{-} b), 1) \dot{-} \max((b \dot{-} a) + a \dot{-} b, b \dot{-} (b \dot{-} a) + a), \min(\max((b \dot{-} a) + a \dot{-} b, b \dot{-} (b \dot{-} a) + a), 1) \dot{-} \max(a \dot{-} b \dot{-} 0, 0 \dot{-} a \dot{-} b)) = 0$

41. $\max(sq(a) \dot{-} a \cdot a, a \cdot a \dot{-} sq(a)) = 0$

42. $\max((< a, b >)_1 \dot{-} a, a \dot{-} (< a, b >)_1) = 0$

43. $\max((< a, b >)_2 \dot{-} a, a \dot{-} (< a, b >)_2) = 0$

44. $\max(< (a)_1, (a)_2 > \dot{-} a, a \dot{-} < (a)_1, (a)_2 >) = 0$

45. $| < a, b > | \dot{-} (2 \cdot (1 + |a| + |b|)) = 0$

46. $\max(< a, b > \dot{-} \lfloor \frac{1}{2} ((a^2 + b^2 + 2ab + a + 1) \dot{-} b) \rfloor, \lfloor \frac{1}{2} ((a^2 + b^2 + 2ab + a + 1) \dot{-} b) \rfloor \dot{-} < a, b >) = 0$

4.4.2 General Axioms

To prove our translation, *, has the provability property as defined in Definition 36, we look at the general proof rules in PK defined in Appendix B. These are logical, reflexivity, symmetry, transitivity and compatibility. For each of these axioms in EET, we want to develop a PET-style axioms to replace

these whenever they may be used within an EET-proof.

Let a be a term, then the EET logical axiom is

$a \Rightarrow a$

The translation of this axiom creates a new axiom in PET

Axiom 1. $\min(a, 1) \dot{-} a = 0$

Let a be a term, then the EET reflexivity axiom is

$\Rightarrow a = a$

The translation of this axiom creates a new axiom in PET

Axiom 2. $\max(a \dot{-} a, a \dot{-} a) = 0$

Let a and b be terms, then the EET symmetry axiom is

$a = b \Rightarrow b = a$

The translation of this axiom creates a new axiom in PET

Axiom 3. $\min(\max(b \dot{-} a, a \dot{-} b), 1) \dot{-} \max(a \dot{-} b, b \dot{-} a) = 0$

Let a, b and c be terms, then the EET transitivity axiom is

$a = b \wedge b = c \Rightarrow a = c$

The translation of this axiom creates a new axiom in PET

Axiom 4. $\min(\max(a \dot{-} c, c \dot{-} a), 1) \dot{-} \max(\max(a \dot{-} b, b \dot{-} a), \max(b \dot{-} c, c \dot{-} b)) = 0$

Lastly, there is the EET-compatibility axiom. Let a, b and U be terms, and x be a variable in U, the EET-compatibility rule is

Note : $u[x/s]$ denotes the term u where all occurrences of variable x in u are replaced by s;

$a = b \Rightarrow U[x/a] = U[x/b]$

A term now is a term built over the language L_p. Since U can be any term, we can use the compatibility axiom to reach any possible conclusion. Therefore here we reach a stage where we create an infinite set of axioms in PET.

Let f be a function symbol of the language L_p. Even for a finite language of function symbols the set of terms will be infinite (assuming there is at least one function symbol of arity > 0). So with our EET-compatibility axiom in the form $a = b \Rightarrow U[x/a] = U[x/b]$, we add the translation of these axioms to new axioms of PET, we are adding infinitely many axioms to PET. The translation of the compatibility axiom creates infinitely new axioms in PET of the form;

Axiom 5. $\min(\max(f(a) \dot{-} f(b), f(b) \dot{-} f(a)), 1) \dot{-} \max(a \dot{-} b, b \dot{-} a) = 0$

Axioms 1, 2, 3, 4, and the infinite set created by 5 are known as $BASIC_g$ for reference.

4.4.3 Inference Rules

To prove Lemma 5 we also need to look into EET. Every line in a proof of EET consists of a sequent of boolean formulas which are connected by Gentzen style sequent calculus. For each rule, we want to be able to prove that the translation of the premise of the rule and the translation of the conclusion of the rule are equal in a PET setting.

We will be assuming the translation of the premise of each rule is true and trying to reach the conclusion of the rule using PET rules and axioms that we develop on the way. Also, a lesson we learnt from Chapter 3.5, we will only be using the premise of each rule exactly once as to avoid an exponential growth of the proof which would mean the translation is unlikely to be formalizable in S^1_2.

We will see a full PET style proof and an informal equational reasoning proof for each rule. The full PET proof, which uses the rules of PET such as extensionality and transitivity, gives great detail about how this translation holds the provability property whereas the equational reasoning simply illustrates how each proof works by proving the conclusion of each rule using the assumption, the rules premise, and the axioms provided. Equational reasoning can be used instead for easier reading of the proofs.

Here's a list of the axioms the following proofs will use, for reference, these will be known as $BASIC_a$;

Definition 41. *$BASIC_a$ consists of the following 18 formulas*

Axiom 6. $\min(a,1) \dot{-} \max(1 \dot{-} b, c) = \min(\min(a,1), b) \dot{-} c$

Axiom 7. $\min(\min(a,1), 1 \dot{-} b) \dot{-} c = \min(a,1) \dot{-} \max(b,c)$

Axiom 8. $\min(\min(a,b),c) = \min(\min(a,c),b)$

Axiom 9. $\max(\max(a,b),c) = \max(a,\max(b,c))$

Axiom 10. $\max(0,a) = a$

Axiom 11. $\max(a,b) \dot{-} c = \max(a \dot{-} c, b \dot{-} c)$

Axiom 12. $\min(a,\max(b,c)) = \max(\min(a,b),\min(a,c))$

Axiom 13. $a \dot{-} \min(b,c) = \max(a \dot{-} b, a \dot{-} c)$

Axiom 14. $\max(\min(a,b),c) = \min(\max(a,c),\max(b,c))$

Axiom 15. $\min(a,\min(b,c)) = \min(\min(a,b),c)$

Axiom 16. $\max(\min(a,b),c) = \min(\max(a,c),\max(b,c))$

Axiom 17. $0 = \min(1 \dot{-} a, a)$

Axiom 18. $\max(a,\max(b,c)) = \max(b,\max(a,c))$

Axiom 19. $\max(a,b) = \max(a,\max(a,b))$

Axiom 20. $\min(a,b) = \min(\min(a,b),b)$

Axiom 21. $0 = (b \dot{-} \max(a,c)) \dot{-} (b \dot{-} c)$

Axiom 22. $a \dot{-} 0 = a$

Axiom 23. $0 = (\min(b,a) \dot{-} c) \dot{-} (b \dot{-} c)$

Now we will create PET-style proofs of the conclusion of each rule found in Appendix B, using the assumption of the rule only at most once and any of these axioms. Since the rules of PK can be used in EET-proofs, we need to create proofs that prove the translation of the conclusion of these rules are true (are equal to 0), given the translation of each assumption.

Negation Left

$$\neg\text{:left}\ \frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}$$

Our assumption would be the translation of the premise which would be $\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion $\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)$

$\overset{Axiom\,6}{=} \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,8}{=} \min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 8 to be $\min(\min(\Delta^*, 1)A^*) = \min(\min(\Delta^*, A^*), 1)$

$$\text{ext}\ \frac{\min(\min(\Delta^*, 1)A^*) = \min(\min(\Delta^*, A^*), 1)}{\min(\min(\Delta^*, 1)A^*) \dot{-} \max(\Gamma^*) = \min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)}$$

If we label the conclusion of the extensionality rule as H, then we can use the assumption here and the transitivity rule.

$$\text{transitivity}\ \frac{H \qquad \min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0}{\min(\min(\Delta^*, 1)A^*) \dot{-} \max(\Gamma^*) = 0}$$

If we instantiate Axiom 6 to be $\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*) = \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)$, then we can apply the transitivity rule. Let us label the instantiated Axiom with F.

$$\text{transitivity}\ \frac{F \qquad \min(\min(\Delta^*, 1)A^*) \dot{-} \max(\Gamma^*) = 0}{\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*) = 0}$$

□

Negation Right

$$\neg\text{:right}\ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0$. We wish to reach the conclusion $\min(\min(\Delta^*, 1 \dot{-} A^*), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\Delta^*, 1 \dot{-} A^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,8}{=} \min(\min(\Delta^*, 1), 1 \dot{-} A^*) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,7}{=} \min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 7 to be
$\min(\min(\Delta^*, 1), 1 \dot{-} A^*) \dot{-} \max(\Gamma^*) = \min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)$ and label this as H then

$$\text{transitivity}\ \frac{H \qquad \min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0}{\min(\min(\Delta^*, 1), 1 \dot{-} A^*) \dot{-} \max(\Gamma^*) = 0}$$

If we then instantiate Axiom 8 to be
$\min(\min(\Delta^*, 1 \dot{-} A^*), 1) = \min(\min(\Delta^*, 1), 1 \dot{-} A^*)$

$$\text{ext}\ \frac{\min(\min(\Delta^*, 1 \dot{-} A^*), 1) = \min(\min(\Delta^*, 1), 1 \dot{-} A^*)}{\min(\min(\Delta^*, 1 \dot{-} A^*), 1) \dot{-} \max(\Gamma^*) = \min(\min(\Delta^*, 1), 1 \dot{-} A^*) \dot{-} \max(\Gamma^*)}$$

If we label the conclusion of this extensionality rule as F then we can apply it with the conclusion of the transitivity rule with a new transitivity rule to reach our goal.

$$\text{transitivity}\ \frac{F \qquad \min(\min(\Delta^*, 1), 1 \dot{-} A^*) \dot{-} \max(\Gamma^*) = 0}{\min(\min(\Delta^*, 1 \dot{-} A^*), 1) \dot{-} \max(\Gamma^*) = 0}$$

□

Conjunction Left

$$\wedge\text{:left}\ \frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(A^*, \max(B^*, \Gamma^*)) = 0$. We wish to reach the conclusion $\min(\Delta^*, 1) \dot{-} \max(\max(A^*, B^*), \Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(\max(A^*, B^*), \Gamma^*)$

$\overset{Axiom\ 9}{=} \min(\Delta^*, 1) \dot{-} \max(A^*, \max(B^*, \Gamma^*))$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 9 to be $\max(\max(A^*, B^*), \Gamma^*) = \max(A^*, \max(B^*, \Gamma^*))$

$$\text{ext}\ \frac{\max(\max(A^*, B^*), \Gamma^*) = \max(A^*, \max(B^*, \Gamma^*))}{\min(\Delta^*, 1) \dot{-} \max(\max(A^*, B^*), \Gamma^*) = \min(\Delta^*, 1) \dot{-} \max(A^*, \max(B^*, \Gamma^*)))}$$

If we label the conclusion of this extensionality rule as H, and use the transitivity rule with our assumption we can reach our conclusion.

$$\text{transitivity}\ \frac{H \qquad \min(\Delta^*, 1) \dot{-} \max(A^*, \max(B^*, \Gamma^*)) = 0}{\min(\Delta^*, 1) \dot{-} \max(\max(A^*, B^*), \Gamma^*) = 0}$$

□

Conjunction Right

$$\wedge\text{:right}\ \frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}$$

Our assumptions would be the translation of the premises which would be
$\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0$ which we label $H = 0$, and
$\min(\min(\Delta^*, B^*), 1) \dot{-} \max(\Gamma^*) = 0$ which we label $F = 0$. We wish to reach the conclusion
$\min(\min(\Delta^*, \max(A^*, B^*)), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\Delta^*, \max(A^*, B^*)), 1) \dot{-} \max(\Gamma^*)$
$\overset{Axiom\,8}{=} \min(\min(\Delta^*, 1), \max(A^*, B^*)) \dot{-} \max(\Gamma^*)$
$\overset{Axiom\,12}{=} \max(\min(\min(\Delta^*, 1), A^*), \min(\min(\Delta^*, 1), B^*)) \dot{-} \max(\Gamma^*)$
$\overset{Axiom\,8}{=} \max(\min(\min(\Delta^*, 1), A^*), \min(\min(\Delta^*, B^*), 1)) \dot{-} \max(\Gamma^*)$
$\overset{Axiom\,8}{=} \max(\min(\min(\Delta^*, A^*), 1), \min(\min(\Delta^*, B^*), 1)) \dot{-} \max(\Gamma^*)$
$\overset{Axiom\,11}{=} \max(\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*), \min(\min(\Delta^*, B^*), 1) \dot{-} \max(\Gamma^*))$
$\overset{Assumption}{=} \max(0, \min(\min(\Delta^*, B^*), 1) \dot{-} \max(\Gamma^*))$
$\overset{Axiom\,10}{=} \min(\min(\Delta^*, B^*), 1) \dot{-} \max(\Gamma^*)$
$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 10 to be $\max(0, F) = F$, then we can say

$$\text{transitivity}\ \dfrac{\text{ext}\ \dfrac{H = 0}{\max(H, F) = \max(0, F)} \qquad \max(0, F) = F}{\text{transitivity}\ \dfrac{\max(H, F) = F \qquad \mathrm{F} = 0}{\max(H, F) = 0}}$$

If we instantiate Axiom 11 to be
$\max(\min(\min(\Delta^*, A^*), 1), \min(\min(\Delta^*, B^*), 1)) \dot{-} \max(\Gamma^*) = \max(H, F)$

$$\text{transitivity}\ \dfrac{\max(\min(\min(\Delta^*, A^*), 1), \min(\min(\Delta^*, B^*), 1)) \dot{-} \max(\Gamma^*) = \max(H, F) \qquad \max(H, F) = 0}{\max(\min(\min(\Delta^*, A^*), 1), \min(\min(\Delta^*, B^*), 1)) \dot{-} \max(\Gamma^*) = 0}$$

If we instantiate Axiom 8 to be $\min(\min(\Delta^*, 1), A^*) = \min(\min(\Delta^*, A^*), 1)$, and label the conclusion of the previous transitivity rule as $G = 0$ then

$$\text{ext}\ \frac{\min(\min(\Delta^*,1),A^*)=\min(\min(\Delta^*,A^*),1)}{\text{transitivity}\ \dfrac{\max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,B^*),1))\dot{-}\max(\Gamma^*)=G \qquad G=0}{\max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,B^*),1))\dot{-}\max(\Gamma^*)=0}}$$

Let us label the conclusion of the last transitivity rule as $I = 0$. If we instantiate Axiom 8 as $\min(\min(\Delta^*,1),B^*) = \min(\min(\Delta^*,B^*),1)$.

$$\text{ext}\ \frac{\min(\min(\Delta^*,1),B^*)=\min(\min(\Delta^*,B^*),1)}{\text{transitivity}\ \dfrac{\max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,1),B^*))\dot{-}\max(\Gamma^*)=I \qquad I=0}{\max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,1),B^*))\dot{-}\max(\Gamma^*)=0}}$$

Let us label the conclusion of the last transitivity rule as $J = 0$. If we instantiate Axiom 12 to be $\min(\min(\Delta^*,1),\max(A^*,B^*)) = \max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,1),B^*))$

$$\text{ext}\ \frac{\min(\min(\Delta^*,1),\max(A^*,B^*))=\max(\min(\min(\Delta^*,1),A^*),\min(\min(\Delta^*,1),B^*))}{\text{transitivity}\ \dfrac{\min(\min(\Delta^*,1),\max(A^*,B^*))\dot{-}\max(\Gamma^*)=J \qquad J=0}{\min(\min(\Delta^*,1),\max(A^*,B^*))\dot{-}\max(\Gamma^*)=0}}$$

Let us label the conclusion of the last transitivity rule as $K = 0$. If we instantiate Axiom 8 to be $\min(\min(\Delta^*,\max(A^*,B^*)),1) = \min(\min(\Delta^*,1),\max(A^*,B^*))$

$$\text{ext}\ \frac{\min(\min(\Delta^*,\max(A^*,B^*)),1)=\min(\min(\Delta^*,1),\max(A^*,B^*))}{\text{transitivity}\ \dfrac{\min(\min(\Delta^*,\max(A^*,B^*)),1)\dot{-}\max(\Gamma^*)=K \qquad K=0}{\min(\min(\Delta^*,\max(A^*,B^*)),1)\dot{-}\max(\Gamma^*)=0}}$$

□

Disjunction Left

$$\vee\text{:left}\ \frac{A,\Gamma\Rightarrow\Delta \qquad B,\Gamma\Rightarrow\Delta}{A\vee B,\Gamma\Rightarrow\Delta}$$

Our assumptions would be the translation of the premises which would be $\min(\Delta^*,1)\dot{-}\max(A^*,\Gamma^*) = 0$ which we label $H = 0$, and $\min(\Delta^*,1)\dot{-}\max(B^*,\Gamma^*) = 0$ which we label $F = 0$. We wish to reach the conclusion $\min(\Delta^*,1)\dot{-}\max(\min(A^*,B^*),\Gamma^*) = 0$.

The equational reasoning for this is;

$$
\begin{aligned}
&\min(\Delta^*, 1) \dot{-} \max(\min(A^*, B^*), \Gamma^*) \\
&\overset{Axiom\,14}{=} \min(\Delta^*, 1) \dot{-} \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*)) \\
&\overset{Axiom\,13}{=} \max(\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*), \min(\Delta^*, 1) \dot{-} \max(B^*, \Gamma^*)) \\
&\overset{Assumption}{=} \max(0, \min(\Delta^*, 1) \dot{-} \max(B^*, \Gamma^*)) \\
&\overset{Axiom\,10}{=} \min(\Delta^*, 1) \dot{-} \max(B^*, \Gamma^*) \\
&\overset{Assumption}{=} 0
\end{aligned}
$$

Proof. If we instantiate Axiom 10 to be $\max(0, F) = F$, then we can say

$$
\text{ext}\ \dfrac{\text{transitivity}\ \dfrac{\dfrac{H = 0}{\max(H, F) = \max(0, F)} \qquad \max(0, F) = F}{\max(H, F) = F} \qquad \mathrm{F} = 0}{\max(H, F) = 0}
$$

If we instantiate Axiom 13 to be $\min(\Delta^*, 1) \dot{-} \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*)) = \max(H, F)$

$$
\text{transitivity}\ \dfrac{\min(\Delta^*, 1) \dot{-} \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*)) = \max(H, F) \qquad \max(H, F) = 0}{\min(\Delta^*, 1) \dot{-} \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*)) = 0}
$$

Let us label the conclusion of the last transitivity rule as $G = 0$. If we instantiate Axiom 14 to be $\max(\min(A^*, B^*), \Gamma^*) = \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*))$

$$
\text{ext}\ \dfrac{\text{transitivity}\ \dfrac{\max(\min(A^*, B^*), \Gamma^*) = \min(\max(A^*, \Gamma^*), \max(B^*, \Gamma^*))}{\min(\Delta^*, 1) \dot{-} \max(\min(A^*, B^*), \Gamma^*) = G} \qquad G = 0}{\min(\Delta^*, 1) \dot{-} \max(\min(A^*, B^*), \Gamma^*) = 0}
$$

□

Disjunction Right

$$
\vee\text{:right}\ \dfrac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}
$$

Our assumption would be the translation of the premise which would be $\min(\min(\min(\Delta^*, A^*), B^*), 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion $\min(\min(\Delta^*, \min(A^*, B^*)), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\Delta^*, \min(A^*, B^*)), 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\ 15}{=} \min(\min(\min(\Delta^*, A^*), B^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 15 to be $\min(\Delta^*, \min(A^*, B^*)) = \min(\min(\Delta^*, A^*), B^*)$ and label this assumption as $H = 0$ then

$$\text{transitivity}\ \frac{\text{ext}\ \dfrac{\min(\Delta^*, \min(A^*, B^*)) = \min(\min(\Delta^*, A^*), B^*)}{\min(\min(\Delta^*, \min(A^*, B^*)), 1) \dot{-} \max(\Gamma^*) = H} \qquad H = 0}{\min(\min(\Delta^*, \min(A^*, B^*)), 1) \dot{-} \max(\Gamma^*) = 0}$$

□

Exchange Left

$$\text{exchange:left}\ \frac{\Gamma, A, B, \Gamma' \Rightarrow \Delta}{\Gamma, B, A, \Gamma' \Rightarrow \Delta}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(A^*, \max(B^*, \Gamma'^*))) = 0$. We wish to reach the conclusion $\min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(B^*, \max(A^*, \Gamma'^*))) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(B^*, \max(A^*, \Gamma'^*)))$

$\overset{Axiom\ 18}{=} \min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(A^*, \max(B^*, \Gamma'^*)))$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 18 to be $\max(B^*, \max(A^*, \Gamma'^*)) = \max(A^*, \max(B^*, \Gamma'^*))$ and label this assumption as $H = 0$ then

$$\text{transitivity}\ \frac{\text{ext}\ \dfrac{\max(B^*, \max(A^*, \Gamma'^*)) = \max(A^*, \max(B^*, \Gamma'^*))}{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(B^*, \max(A^*, \Gamma'^*))) = H} \qquad H = 0}{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*, \max(B^*, \max(A^*, \Gamma'^*))) = 0}$$

□

Exchange Right

$$\text{exchange:right}\ \frac{\Gamma \Rightarrow \Delta, A, B, \Delta'}{\Gamma \Rightarrow \Delta, B, A, \Delta'}$$

Our assumption would be the translation of the premise which would be $\min(\min(\min(\min(\Delta^*, A^*), B^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion $\min(\min(\min(\min(\Delta^*, B^*), A^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\min(\min(\Delta^*, B^*), A^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\ 8}{=} \min(\min(\min(\min(\Delta^*, A^*), B^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 8 to be $\min(\min(\Delta^*, B^*), A^*) = \min(\min(\Delta^*, A^*), B^*)$ and label this assumption as $H = 0$ then

$$\text{transitivity}\ \frac{\text{ext}\ \dfrac{\min(\min(\Delta^*, B^*), A^*) = \min(\min(\Delta^*, A^*), B^*)}{\min(\min(\min(\min(\Delta^*, B^*), A^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*) = H} \qquad H = 0}{\min(\min(\min(\min(\Delta^*, B^*), A^*), \Delta'^*), 1) \dot{-} \max(\Gamma^*) = 0}$$

□

Contraction Left

$$\text{contraction:left}\ \frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(A^*, \max(A^*, \Gamma^*)) = 0$. We wish to reach the conclusion $\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)$

$\overset{Axiom\ 19}{=} \min(\Delta^*, 1) \dot{-} \max(A^*, \max(A^*, \Gamma^*))$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 19 to be $\max(A^*, \Gamma^*) = \max(A^*, \max(A^*, \Gamma^*))$ and label this assumption as $H = 0$ then

$$\text{ext}\ \frac{\text{transitivity}\ \dfrac{\max(A^*, \Gamma^*) = \max(A^*, \max(A^*, \Gamma^*)) \qquad \min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = H}{\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0} \qquad H = 0}{}$$

□

Contraction Right

$$\text{contraction:right}\ \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}$$

Our assumption would be the translation of the premise which would be $\min(\min(\min(\Delta^*, A^*), A^*), 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion $\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\ 20}{=} \min(\min(\min(\Delta^*, A^*), A^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Assumption}{=} 0$

Proof. If we instantiate Axiom 20 to be $\min(\Delta^*, A^*) = \min(\min(\Delta^*, A^*), A^*)$ and label this assumption as $H = 0$ then

$$\text{ext}\ \dfrac{\min(\Delta^*, A^*) = \min(\min(\Delta^*, A^*), A^*)}{\text{transitivity}\ \dfrac{\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = H \qquad H = 0}{\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0}}$$

□

Weakening Left

$$\text{weakening:left}\ \ \dfrac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion $\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)$

$\overset{Axiom\,22}{=} (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} 0$

$\overset{Assumption}{=} (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*))$

$\overset{Axiom\,21}{=} 0$

Proof. If we use the assumption;

$$\text{ext}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0}{(\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)) = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} 0}$$

If we instantiate Axiom 21 to be $0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*))$ and label conclusion of the last extensionality rule as H then

$$\text{transitivity}\ \ \dfrac{0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)) \qquad H}{0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} 0}$$

If we instatiate Axiom 22 to be $(\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} 0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*))$ and label this as G then

$$\text{transitivity}\ \dfrac{0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) \dot{-} 0 \qquad G}{\text{symmetry}\ \dfrac{0 = (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*))}{(\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)) = 0}}$$

□

Weakening Right

$$\text{weakening:right}\ \ \dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}$$

Our assumption would be the translation of the premise which would be $\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0$. We wish to reach the conclusion

$\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,8}{=} \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,22}{=} \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*) \dot{-} 0$

$\overset{Assumption}{=} (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*))$

$\overset{Axiom\,23}{=} 0$

Proof. If we use the assumption;

$$\text{com}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0}{(\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)) = (\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)) \dot{-} 0}$$

If we instantiate Axiom 23 to be $0 = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*))$ and label the conclusion of the last extensionality rule as H then

$$\text{transitivity}\ \ \dfrac{0 = (\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)) \dot{-} (\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)) \qquad H}{0 = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)) \dot{-} 0}$$

If we instatiate Axiom 22 to be

$(\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)) \dot{-} 0 = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*))$ and label this as G then

$$\text{transitivity}\ \dfrac{\text{symmetry}\ \dfrac{0 = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)) \dot{-} 0 \qquad G}{0 = \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)}}{\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*) = 0}$$

If we instantiate Axiom 8 to be

$\min(\min(\Delta^*, A^*), 1) = \min(\min(\Delta^*, 1), A^*)$ and label $\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*) = 0$ as F then

$$\text{transitivity}\ \dfrac{\text{com}\ \dfrac{\min(\min(\Delta^*, A^*), 1) = \min(\min(\Delta^*, 1), A^*)}{\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = \min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)} \qquad F}{\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0}$$

□

Cut Rule

$$\text{cut}\ \dfrac{\Gamma \Rightarrow \Delta, A \qquad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Our assumptions would be the translation of the premises which would be

$\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*) = 0$ which we label $H = 0$, and

$\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*) = 0$ which we label $F = 0$. We wish to reach the conclusion

$\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0$.

The equational reasoning for this is;

$\min(\Delta^*, 1) \dot{-} \max(\Gamma^*)$

$\overset{Axiom\,10}{=} \min(\Delta^*, 1) \dot{-} \max(0, \Gamma^*)$

$\overset{Axiom\,17}{=} \min(\Delta^*, 1) \dot{-} \max(\min(1 \dot{-} A^*, A^*), \Gamma^*)$

$\overset{Axiom\,16}{=} \min(\Delta^*, 1) \dot{-} \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*))$

$\overset{Axiom\,13}{=} \max((\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)), (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)))$

$\overset{Axiom\,6}{=} \max((\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)), (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)))$

$$\overset{Axiom\,8}{=} \max((\min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)), (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)))$$
$$\overset{Assumption}{=} \max(0, (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)))$$
$$\overset{Axiom\,10}{=} \min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*)$$
$$\overset{Assumption}{=} 0$$

Proof. If we instantiate Axiom 10 to be $\max(0, F) = F$, then we can say

$$\text{transitivity}\ \dfrac{\text{ext}\ \dfrac{H = 0}{\max(H, F) = \max(0, F)} \qquad \max(0, F) = F}{\text{transitivity}\ \dfrac{\max(H, F) = F \qquad \mathrm{F} = 0}{\max(H, F) = 0}}$$

If we instantiate Axiom 8 to be $\min(\min(\Delta^*, 1), A^*) = \min(\min(\Delta^*, A^*), 1)$ then

$$\text{ext}\ \dfrac{\min(\min(\Delta^*, 1), A^*) = \min(\min(\Delta^*, A^*), 1)}{\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*) = \min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)}$$

If we label the conclusion of the last extensionality rule as G and instantiate Axiom 6 to be $(\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)) = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*)$ then

$$\text{tran}\ \dfrac{(\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)) = (\min(\min(\Delta^*, 1), A^*) \dot{-} \max(\Gamma^*) \qquad G}{\text{com}\ \dfrac{(\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)) = \min(\min(\Delta^*, A^*), 1) \dot{-} \max(\Gamma^*)}{\text{tran}\ \dfrac{\max((\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)), F) = \max(H, F) \qquad \max(H, F) = 0}{\max((\min(\Delta^*, 1) \dot{-} \max(1 \dot{-} A^*, \Gamma^*)), (\min(\Delta^*, 1) \dot{-} \max(A^*, \Gamma^*))) = 0}}}$$

If we then label the conclusion of out last transitivity rule as $I = 0$ and instantiate Axiom 13 to be $\min(\Delta^*, 1) \dot{-} \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*)) = I$ then

$$\text{tran}\ \dfrac{\min(\Delta^*, 1) \dot{-} \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*)) = I \qquad I = 0}{\min(\Delta^*, 1) \dot{-} \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*)) = 0}$$

If we label the conclusion of our last transitivity rule as $J = 0$ and instantiate Axiom 16 to be $\max(\min(1 \dot{-} A^*, A^*), \Gamma^*) = \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*))$

$$\text{com}\ \dfrac{\max(\min(1 \dot{-} A^*, A^*), \Gamma^*) = \min(\max(1 \dot{-} A^*, \Gamma^*), \max(A^*, \Gamma^*))}{\text{tran}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(\min(1 \dot{-} A^*, A^*), \Gamma^*) = J \qquad J = 0}{\min(\Delta^*, 1) \dot{-} \max(\min(1 \dot{-} A^*, A^*), \Gamma^*) = 0}}$$

If we label the conclusion of the last transitivity rule as $K = 0$ and instantiate Axiom 17 to be $0 = \min(1 \dot{-} A^*, A^*)$ then

$$\text{sub}\ \frac{0 = \min(1 \dot{-} A^*, A^*)}{\text{tran}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(0, \Gamma^*) = K \qquad K = 0}{\min(\Delta^*, 1) \dot{-} \max(0, \Gamma^*) = 0}}$$

Lastly, if we label the conclusion of the last transitivity rule as $L = 0$ and instantiate Axiom 10 as $\max(0, \Gamma^*) = \max(\Gamma^*)$. *Recall that* $\max(0, \Gamma^*) \equiv \max_{n+1}(0, \Gamma^*) \equiv \max(0, \max(\Gamma_1^*, \max(..., \Gamma_n^*)))$

$$\begin{array}{l}\text{com}\ \dfrac{\max(0, \Gamma^*) = \max(\Gamma^*)}{\text{sym}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(0, \Gamma^*) = \min(\Delta^*, 1) \dot{-} \max(\Gamma^*)}{\text{tran}\ \dfrac{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = \min(\Delta^*, 1) \dot{-} \max(0, \Gamma^*) \qquad L = 0}{\min(\Delta^*, 1) \dot{-} \max(\Gamma^*) = 0}}}\end{array}$$

□

Since we can prove that for any EET-proof of a formula or sequent, A, there is an equivalent PET-proof for $A^* = 0$, then we have proven that the translation has the provability property.

Lemma 5. *The translation, $*$, has the provability property.*

4.5 Evaluation of Translation

Since we have proven $*$ has the consistency property and the provability property, we have proven also that * is a good translation.

Lemma 6. *$*$ is a good translation.*

This translation concentrated on using the assumption at most once for every rule. This was to avoid exponential growth in the proofs when translating Buss and Ignjatovič's result in [5]. That is, S_2^1 could not prove the consistency of PV^- extended by propositional logic and the $BASIC_e$ axioms.

Definition 42. *PET is the theory over the language L_p which only considers equations and is given by the axioms $BASIC_g$, $BASIC_t$ and $BASIC_a$ and the proof system PI as defined in Section 2.7*

Lemma 7. ** is formalizable in* S_2^1

Proof. To show * is formalizable in S_2^1, since Buss proved in [3] that S_2^1 can define all $\square_1^P$-procedures, we want to show that the transformation is a $\square_1^P$-procedures. What we actually show is that the transformation can be done within polynomial resources and that it only grows polynomially. The size of a proof is polynomially bounded by the size of the original one. This means it is a $\square_1^P$-procedures and since S_2^1 is able to formalize such constructive procedures [3], * is formalizable in S_2^1.

Let A be a formula with a proof P in EET. Then we can apply the transformation and let P^* be the proof of the formula $A^* = 0$ in PET. We can prove that the size of P^* is polynomially bounded by the size of P.

Proof P^* may use additional axioms that were not used in the original proof P, this creates some build up of the size of the proof. However we can estimate the size of the blow up of P^* given the size of P. In proof P^*, the number of lines is linearly bounded in P by the number of lines before. After inspection of our translation, the number of lines in P^* is at most 15 times more than the number of lines in P, as when we use the *cut rule* in EET we create 15 more steps in PET and this is the most number of steps created for any rule. For any symbol on a line in P, the most number of times it can occur on a line in P^* is 6. After inspection, the number of times a symbol was repeated on the same line was found in the *cut rule* to be at most 6 times. The size of the proof grows polynomially with fixed constants. The number of lines in $P^* = O(\text{size of } P)$ and the number of symbols in each line in $P^* = O(\text{size of } P)$. So the size of $P^* \leq$ the number of lines in $P \cdot$ the number of symbols per line in P. Therefore the size of $P^* \leq O((\text{size of } P)^2)$. Therefore, * grows polynomially, concluding that * is formalizable in S_2^1 □

With this translation, we can prove the theorem;

Theorem 10. $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$

Proof. $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$ is equivalent to saying $S_2^1 \vdash \neg Con(EET) \rightarrow \neg Con(PET)$. If we prove one, then we prove the other.

Let P be a proof of $0 = 1$ in an EET setting. Using our transformation, $*$, we can transform this proof such that P^* is the proof of $\max(0\dot{-}1, 1\dot{-}0) = 0$ in a PET setting. From here we can prove that $0 = 1$ with use of further axioms. Using the axiom from $BASIC_t$ that states $0\dot{-}x = 0$, and axiom 10. $x\dot{-}0 = x$, and axiom 22. $\max(0, x) = x$ from $BASIC_a$ we can show that if $\max(0\dot{-}1, 1\dot{-}0) = 0$ has a PET-proof, then $0 = 1$ has a PET-proof.

By instantiating these axioms to be $0\dot{-}1 = 0$, $1\dot{-}0 = 1$, and $\max(0, 1) = 1$ we can create the proof of $0 = 1$ using the assumption that we have a proof of $\max(0\dot{-}1, 1\dot{-}0) = 0$.

$$\dfrac{\text{tran}\dfrac{\text{sym}\dfrac{\max(0\dot{-}1, 1\dot{-}0) = 0}{0 = \max(0\dot{-}1, 1\dot{-}0)} \quad \text{com}\dfrac{0\dot{-}1 = 0}{\max(0\dot{-}1, 1\dot{-}0) = \max(0, 1\dot{-}0)}}{0 = \max(0, 1\dot{-}0)} \quad \text{com}\dfrac{1\dot{-}0 = 1}{\max(0, 1\dot{-}0) = \max(0, 1)}}{0 = \max(0, 1)}$$

And so;

$$\dfrac{0 = \max(0, 1) \qquad \max(0, 1) = 1}{0 = 1}$$

Since the transformation is a polynomial time procedure, and that the size of the proof doesn't grow exponentially, then we are still arguing in S_2^1.

Here we prove that $S_2^1 \vdash \neg Con(EET) \rightarrow \neg Con(PET)$. This is equivalent to $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$. Hence $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$.

□

Corollary. $S_2^1 \nvdash Con(PET)$

Proof. We prove this by contradiction. If $S_2^1 \vdash Con(PET)$, then by Theorem 10, $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$, and modus ponens we would have $S_2^1 \vdash Con(EET)$ which contradicts Theorem 9, $S_2^1 \nvdash Con(EET)$. Therefore $S_2^1 \nvdash Con(PET)$ has to be true. □

5. CONCLUSION

Buss proves in [3] that the polynomial hierarchy and the Bounded Arithmetic hierarchy are strongly linked. Gödel's Incompleteness Theorems state that no consistent set of axioms can demonstrate its own consistency. If we can prove that some theory in Bounded Arithmetic can prove the consistency of S_2^1 then because of Gödel's Incompleteness Theorems we would have proven that the theories are separate, and since the theories are closely linked to the complexity classes we would have proven that the complexity classes are separate.

Beckmann shows that S_2^1 can prove the consistency of the proof system PV^- excluding substitution [1]. Buss and Ignjatovič prove that S_2^1 can not prove the consistency of PV^- extended by propositional logic and the $BASIC^e$ axioms [5]. Buss and Ignjatovič's version of PV we call expanded equational theory, and created a translation into a pure equational theory so that we work with equations rather than formulas. Buss and Ignjatovič show that $S_2^1 \nvdash Con(EET)$. Our result shows that $S_2^1 \vdash Con(PET) \rightarrow Con(EET)$ by our *good translation*, *, concluding that $S_2^1 \nvdash Con(PET)$ where our version of PET is the Buss and Ignjatovič axiom system with additional axioms needed for the translation viewed in a pure equational theory setting.

5.1 Future Work

In this thesis we prove that $S_2^1 \nvdash Con(PET)$, that is the translated equivalent of Buss and Ignjatovič's result in [5]. However, we only prove this for the usual notion of consistency. Other notions of consistency, such as $BDCon$, can be looked at to prove the consistency of PET with the extension of axioms needed for the translation. This leading to a new problem to be looked at; if the consistency of

this pure equational theory with axioms that are properties of the symbols used rather than recursive definitions can be proven, then the consistency of EET, Buss and Ignjatovič's result can also be proven.

A future research direction is to close the gap between provability and unprovability for pure equational theories. We know Beckmann in [1] proved that the consistency of any theory of equations which is based just on recursion axioms for the underlying language can be proven in a weak theory of Bounded Arithmetic related to polynomial time reasoning. Future work could be looking into the axioms used in PET and the translated axioms $BASIC_t$ to see if they can be simplified. Also one would be interested in making the set of axioms in $BASIC_g$ finite. We would be making this set of additional axioms precise, and to close the gap between provability and unprovability by nailing down precisely which axioms can be added to PET and still have its consistency provable in S_2^1 and which settings will lead to a more complex notion of consistency unprovable in a weak theory of Bounded Arithmetic related to polynomial time reasoning like S_2^1.

APPENDIX

A. BASIC

BASIC consists of 32 mathematically true formulas (Note: 1 and 2 are used to abbreviate S0 and SS0);

1. $y \leq x \rightarrow y \leq Sx$
2. $x \neq Sx$
3. $0 \leq x$
4. $x \leq y \wedge x \neq y \leftrightarrow Sx \leq y$
5. $x \neq 0 \rightarrow 2 \cdot x \neq 0$
6. $y \leq x \vee x \leq y$
7. $x \leq y \wedge y \leq x \rightarrow x = y$
8. $x \leq y \wedge y \leq z \rightarrow x \leq z$
9. $|0| = 0$
10. $x \neq 0 \rightarrow |2 \cdot x| = S(|x|) \wedge |S(2 \cdot x)| = S(|x|)$
11. $|S0| = S0$
12. $x \leq y \rightarrow |x| \leq |y|$
13. $|x\#y| = S(|x| \cdot |y|)$
14. $|0\#y| = S0$
15. $x \neq 0 \rightarrow 1\#(2 \cdot x) = 2(1\#x) \wedge 1\#(S(2 \cdot x)) = 2(1\#x)$
16. $x\#y = y\#x$
17. $|x| = |y| \rightarrow x\#z = y\#z$
18. $|x| = |u| + |v| \rightarrow x\#y = (u\#y) \cdot (v\#y)$
19. $x \leq x + y$

20. $x \leq y \wedge x \neq y \rightarrow S(2 \cdot x) \leq 2.y \wedge S(2 \cdot x) \neq 2 \cdot y$
21. $x + y = y + x$
22. $x + 0 = x$
23. $x + Sy = S(x + y)$
24. $(x + y) + z = x + (y + z)$
25. $x + y \leq x + z \leftrightarrow y \leq z$
26. $x \cdot 0 = 0$
27. $x \cdot (Sy) = (x \cdot y) + x$
28. $x \cdot y = y \cdot x$
29. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
30. $S0 \leq x \rightarrow (x \cdot y \leq x \cdot z \leftrightarrow y \leq z)$
31. $x \neq 0 \rightarrow |x| = S(|\lfloor \frac{x}{2} \rfloor|)$
32. $x = \lfloor \frac{y}{2} \rfloor \leftrightarrow (2 \cdot x = y \vee S(2 \cdot x) = y)$

$BASIC_e$ is an extension to $BASIC$ with the following axioms

1. $|a| \leq a$
2. $|a \cdot b| \leq |a| + |b|$
3. $2^a_{|0|} = 1$
4. $2^0_{|c|} = 1$
5. $a + b \leq |c| \rightarrow 2^{a+b}_{|c|} = 2^a_{|c|} \cdot 2^b_{|c|}$
6. $c \neq 0 \rightarrow (2^1_{|c|} = 2 \wedge 2^a_{|c|} < 2 \cdot c)$
7. $a \leq b \leftrightarrow a \dot{-} b = 0$
8. $a \dot{-} b = 0 \leftrightarrow (b \dot{-} a) + a = b$
9. $sq(a) = a \cdot a$
10. $(< a, b >)_1 = a$
11. $(< a, b >)_2 = b$
12. $< (a)_1, (a)_2 >= a$

13. $|<a,b>| \leq 2 \cdot (1 + |a| + |b|)$
14. $<a,b> = \lfloor \frac{1}{2}((a^2 + b^2 + 2ab + a + 1) \dot{-} b) \rfloor$

B. PK

PK is over a first order language with equality involving only propositional connectives. It consists of a rooted tree where the nodes are sequents. The root of the tree is what the proof proves and the leaves at the top of the tree are the *initial sequents* or *axioms*. Given that A and B are arbitrary formulas and Γ and Δ are arbitrary cedents, the axioms and rules for the proof system consist of the equality axioms, structural rules and the logical rules. The equality axioms are;

logical axiom

$$A \Rightarrow A$$

reflexivity axiom

$$\Rightarrow A = A$$

symmetry axiom

$$A = B \Rightarrow B = A$$

transitivity axiom

$$A = B \wedge B = C \Rightarrow A = C$$

compatibility axiom

$$A = B \Rightarrow U[X/A] = U[X/B]$$

The structural rules are defined as follows.

$$\text{exchange:left} \quad \frac{\Gamma, A, B, \Gamma' \Rightarrow \Delta}{\Gamma, B, A, \Gamma' \Rightarrow \Delta}$$

$$\text{exchange:right}\ \frac{\Gamma \Rightarrow \Delta, A, B, \Delta'}{\Gamma \Rightarrow \Delta, B, A, \Delta'}$$

$$\text{contraction:left}\ \frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}$$

$$\text{contraction:right}\ \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}$$

$$\text{weakening:left}\ \frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}$$

$$\text{weakening:right}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}$$

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

And the logic rules are defined as follows

$$\neg\text{:left}\ \frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}$$

$$\neg\text{:right}\ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}$$

$$\wedge\text{:left}\ \frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}$$

$$\wedge\text{:right}\ \frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}$$

$$\vee\text{:left}\ \frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}$$

$$\vee\text{:right}\ \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}$$

$$\rightarrow\text{:left}\ \frac{\Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta}$$

$$\rightarrow\text{:right}\ \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}$$

BIBLIOGRAPHY

[1] Arnold Beckmann, *Proving consistency of equational theories in Bounded Arithmetic.* J. Symbolic Logic, 67:279-296. 2002.

[2] Guram Bezhanishvili, *Henkin's method and the completeness theorem.* Available from the webpage http://www. cs. nmsu. edu/historical-projects. Last Visited 08/2015.

[3] Samuel R. Buss *Bounded Arithmetic*, Bibliopolis, Naples, Italy, 1986.

[4] Samuel R. Buss *Axiomatizations and conversation results for fragments of Bounded Arithmetic.* Logic and Computation, Contemporary Mathematics, 106: 57-84. Providence, AMS. 1990.

[5] Samuel R. Buss and Aleksandar Ignjatovič, *Unprovability of consistency statements in fragments of Bounded Arithmetic.* Ann. Pure Appl. Logic , 74:3, 221-244. 1995.

[6] Samuel R. Buss *Relating the Bounded Arithmetic and Polynomial-Time Hierarchies.* Annals of Pure and Applied Logic, 75, 67-77, 1995.

[7] Samuel R. Buss *Bounded arithmetic and propositional proof complexity*, Logic of Computation. Springer Berlin Heidelberg, 1997. 67-121.

[8] Samuel R. Buss *First-Order Proof Theory of Arithmetic* in *Handbook of Proof Theory*, Elsevier, Amsterdam, 1998. 112-122.

[9] Peter Clote and Jan Krajíček, Open problems, in *Arithmetic, proof theory, and computational complexity.* Papers from the conference held in Prague, July 2-5 1991. Edited by Clote and Krajíček,

pp 1 - 9. Oxford Logic Guides, 23. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1993.

[10] Stephen Cook *Feasibly constructive proofs and the propositional calculus (preliminary version)*. In Seventh Annual ACM Symposium on Theory of Computing (Albuquerque, N.M., 1975), pages 83-97. Assoc. Comput. Mach., New York, 1975.

[11] Stephen Cook and Alasdair Urquhart, *Functional interpretations of feasibly constructive arithmetic*. Annals of pure and applied logic, 63(2):103-200, September 1993.

[12] Stephen Cook *The P verses NP problem*. Clay Mathematical Institute, The Millennium Prize Problem, 2000.

[13] Michael Detlefsen *Hilbert's program: an essay on mathematical instrumentalism*. Vol 182. Springer Science & Business Media. 1986.

[14] Leon Henkin *The completeness of the first-order functional calculus*. The journal of symbolic logic 14(3): 159-166, 1949

[15] Pavel Pudlák *A note on Bounded Arithmetic*. Fundamenta Mathematicae 136:2, 85-89. 1990.

[16] Raymond M. Smullyan, *Gödel's Incompleteness Theorems*. Oxford Logic Guides, 1992.

[17] Larry J. Stockmeyer, *The polynomial-time hierarchy*. Theoret. Comput. Sci., 3(1):1-22 (1977), 1976.

[18] Alex J. Wilkie and Jeff B. Paris, *On the scheme of induction for Bounded Arithmetic formulas*. Ann. Pure Appl. Logic, pages 261-302. 1987.

[19] Yoriyuki Yamagata *Consistency proof of a feasible arithmetic inside a Bounded Arithmetic*. arXiv preprint arXiv:1411.7087, 26/11/2014.

[20] Richard Zach *Hilbert's program then and now*. Philosophy of Logic, Handbook of the Philosophy of Science, 5: 411-447, 2006.