Bounded Arithmetic, Part II

Raheleh Jalali

Proof Society 2025

Outline

Hierarchy of theories

Definition (Fragments of bounded arithmetic)

- S_2^i: BASIC + Σ_i^b-PIND
- T_2^i: BASIC + Σ_i^b-IND
- $S_2 = \bigcup_i S_2^i$ and $T_2 = \bigcup_i T_2^i$

Looking ahead:

Theorem (Buss '85, '90)

1. $S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq S_2^3 \subseteq \dots$
2. *Thus,* $S_2 = T_2$.

Theorem

Let $i \geqslant 0$.

- $S_2^1 + \Sigma_i^b$-*IND is equivalent to* $S_2^1 + \Pi_i^b$-*IND.*
- $S_2^1 + \Sigma_i^b$-*LIND is equivalent to* $S_2^1 + \Pi_i^b$-*LIND.*
- $S_2^1 + \Sigma_i^b$-*PIND is equivalent to* $S_2^1 + \Pi_i^b$-*PIND.*

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(|x|)$$

- Define $B(x) := A(|x|)$.

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(|x|)$$

- Define $B(x) := A(|x|)$. It is easy to see $S_2^i \vdash A(0) \to B(0)$.

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(|x|)$$

- Define $B(x) := A(|x|)$. It is easy to see $S_2^i \vdash A(0) \to B(0)$.
- $S_2^i \vdash \forall x(A(x) \to A(x+1)) \to \forall x(A(|\lfloor \frac{1}{2}x \rfloor|) \to A(|\lfloor \frac{1}{2}x \rfloor| + 1))$

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(|x|)$$

- Define $B(x) := A(|x|)$. It is easy to see $S_2^i \vdash A(0) \to B(0)$.
- $S_2^i \vdash \forall x(A(x) \to A(x+1)) \to \forall x(A(|\lfloor \frac{1}{2}x \rfloor|) \to A(|\lfloor \frac{1}{2}x \rfloor| + 1))$
- We have $B(\lfloor \frac{1}{2}x \rfloor) = A(|\lfloor \frac{1}{2}x \rfloor|)$ and $|\lfloor \frac{1}{2}x \rfloor| + 1 = |x|$.

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \rightarrow A(x+1)) \rightarrow \forall x A(|x|)$$

- Define $B(x) := A(|x|)$. It is easy to see $S_2^i \vdash A(0) \rightarrow B(0)$.
- $S_2^i \vdash \forall x(A(x) \rightarrow A(x+1)) \rightarrow \forall x(A(|\lfloor \frac{1}{2}x \rfloor|) \rightarrow A(|\lfloor \frac{1}{2}x \rfloor| + 1))$
- We have $B(\lfloor \frac{1}{2}x \rfloor) = A(|\lfloor \frac{1}{2}x \rfloor|)$ and $|\lfloor \frac{1}{2}x \rfloor| + 1 = |x|$.
- $S_2^i \vdash \forall x(A(x) \rightarrow A(x+1)) \rightarrow \forall x(B(\lfloor \frac{1}{2}x \rfloor) \rightarrow B(x))$

PIND $\Longrightarrow$ LIND

Theorem

The Σ_i^b-LIND axioms are theorems of S_2^i, for $i \geqslant 1$.

Proof.

We want to show

$$S_2^i \vdash A(0) \wedge \forall x(A(x) \to A(x+1)) \to \forall x A(|x|)$$

- Define $B(x) := A(|x|)$. It is easy to see $S_2^i \vdash A(0) \to B(0)$.
- $S_2^i \vdash \forall x(A(x) \to A(x+1)) \to \forall x(A(|\lfloor \frac{1}{2}x \rfloor|) \to A(|\lfloor \frac{1}{2}x \rfloor| + 1))$
- We have $B(\lfloor \frac{1}{2}x \rfloor) = A(|\lfloor \frac{1}{2}x \rfloor|)$ and $|\lfloor \frac{1}{2}x \rfloor| + 1 = |x|$.
- $S_2^i \vdash \forall x(A(x) \to A(x+1)) \to \forall x(B(\lfloor \frac{1}{2}x \rfloor) \to B(x))$
- $B \in \Sigma_i^b$, so by Σ_i^b-PIND, $S_2^i \vdash \forall x(A(x) \to A(x+1)) \to \forall x B(x)$. $\square$

Similarly, we can prove:

Theorem

Let $i \geqslant 1$. T^i_2 proves the Σ^b_i-PIND axioms. Hence, $S^i_2 \subseteq T^i_2$ for $i \geqslant 1$.

Proof.

Let $A(x)$ be a Σ^b_i formula. For c a new free variable, we prove

$$A(0) \wedge (\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor) \rightarrow A(x)) \rightarrow A(c)$$

- We observed Σ^b_i-IND $\Longrightarrow$ Σ^b_i-LIND. Now, show Σ^b_i-LIND $\Longrightarrow$ Σ^b_i-PIND.
- Use LIND on $B(i) := A(t(i))$, where $t(i) = \lfloor \frac{c}{2^{|c|-i}} \rfloor$.
- $B(0)$ and $B(|c|)$ are equivalent to $A(c)$ and $A(0)$.
- $t(i) = \left\lfloor \frac{1}{2} t(i+1) \right\rfloor$
- $(\forall x)(A(\lfloor \frac{1}{2}x \rfloor) \rightarrow A(x))$ implies $(\forall i)(B(i) \rightarrow B(i+1))$.
- By LIND, $B(|c|)$, thus $A(c)$. $\square$

Theorem

$T_2^i \subseteq S_2^{i+1}$ *for* $i \geqslant 1$.

Theorem

$T_2^i \subseteq S_2^{i+1}$ *for* $i \geqslant 1$.

Proof sketch.

Let ϕ be a Σ_i^b-formula satisfying

$$\phi(0) \wedge \forall x(\phi(x) \to \phi(x+1)) \wedge \neg\phi(a)$$

Define $\phi'(y)$ as $y \leqslant a \to \phi(y)$, and formula ψ by

$$\psi(x) := \forall y \leqslant a\ (\phi'(y) \to \phi'(x+y))$$

Then ψ satisfies the assumptions of the PIND scheme and hence $\psi(a)$ follows from S_2^{i+1}. But $\psi(a)$ implies $\phi(a)$. $\square$

Theorem

$T_2^i \subseteq S_2^{i+1}$ for $i \geqslant 1$.

Proof sketch.

Let ϕ be a Σ_i^b-formula satisfying

$$\phi(0) \wedge \forall x(\phi(x) \to \phi(x+1)) \wedge \neg\phi(a)$$

Define $\phi'(y)$ as $y \leqslant a \to \phi(y)$, and formula ψ by

$$\psi(x) := \forall y \leqslant a\ (\phi'(y) \to \phi'(x+y))$$

Then ψ satisfies the assumptions of the PIND scheme and hence $\psi(a)$ follows from S_2^{i+1}. But $\psi(a)$ implies $\phi(a)$. □

Whether the theories are different is unknown; they may all coincide. We don't know if the theories prove P=NP. If so, the hierarchy collapses.

Theorem

Let $i \geqslant 0$.

- $S_2^1 + \Sigma_i^b$*-LIND is equivalent to* $S_2^1 + \Sigma_i^b$*-PIND.*

Outline

1. Hierarchy of theories
2. Σ_i^b-definable functions
3. Main Theorem for S_2^1
   - Sequent calculus
   - Witnessing Lemma
4. Hierarchy of theories, revisited

- "Bootstrap" in computer science means writing a small initial program (a bootstrap) that loads or builds more complex software, "pulling yourself up by your own bootstraps."

- "Bootstrap" in computer science means writing a small initial program (a bootstrap) that loads or builds more complex software, "pulling yourself up by your own bootstraps."
- Similarly we need to bootstrap S^1_2. That is, do a lot of work to define some simple functions and predicates in S^1_2 (for example, subtraction).

- "Bootstrap" in computer science means writing a small initial program (a bootstrap) that loads or builds more complex software, "pulling yourself up by your own bootstraps."
- Similarly we need to bootstrap S^1_2. That is, do a lot of work to define some simple functions and predicates in S^1_2 (for example, subtraction).
- After the bootstrapping it will be easy to show that S^1_2 is a fairly strong system which can define a variety of functions and predicates.

- "Bootstrap" in computer science means writing a small initial program (a bootstrap) that loads or builds more complex software, "pulling yourself up by your own bootstraps."
- Similarly we need to bootstrap S_2^1. That is, do a lot of work to define some simple functions and predicates in S_2^1 (for example, subtraction).
- After the bootstrapping it will be easy to show that S_2^1 is a fairly strong system which can define a variety of functions and predicates.
- We are particularly interested in Σ_i^b-definable functions of S_2^i and T_2^i. We study the strength of the theories of bounded arithmetic by asking which functions are Σ_i^b-definable in the theory.

Σ_i^b-definable functions

Let R be one of our theories, such as S_2^i or T_2^i.

Definition

Let $f : \mathbb{N}^k \to \mathbb{N}$. The function f is Σ_i^b-definable by a theory R iff there is a formula $A \in \Sigma_i^b$ such that

- For all $\bar{n} \in \mathbb{N}^k$, $A(\bar{n}, f(\bar{n}))$ is true (soundness)
- $R \vdash (\forall \bar{x})(\exists y \leqslant t)A(\bar{x}, y)$ for some term t (totality)
- $R \vdash (\forall \bar{x}, y, z)(A(\bar{x}, y) \wedge A(\bar{x}, z) \to y = z)$ (uniqueness)

The second bullet says there's at least one y, third says at most one y.

Why Σ_i^b?

Let f be a Σ_i^b-definable function in S_2^i with the definition A_f and the bound t_f.

- Add f as a new symbol to the language.
- Start with S_2^i over the new language. Note that f can be used freely in the induction axioms.
- Add the defining axiom:

$$(f(x) = y) \leftrightarrow (A_f(x, y) \wedge y \leqslant t_f)$$

- Call the new theory $S_2^i(f)$.

Theorem

$S^i_2(f)$ is conservative over S^i_2, i.e., if $S^i_2(f) \vdash \phi$ (for ϕ in the original language), then already $S^i_2 \vdash \phi$. The same also holds for T^i_2.

So, adding f does not let us prove anything new in the original language.

Theorem

$S_2^i(f)$ is conservative over S_2^i, i.e., if $S_2^i(f) \vdash \phi$ (for ϕ in the original language), then already $S_2^i \vdash \phi$. The same also holds for T_2^i.

So, adding f does not let us prove anything new in the original language.

Proof.

- Key idea: for any r, there's a *unique* y such that $A(r, y)$ holds.

Theorem

$S_2^i(f)$ is conservative over S_2^i, i.e., if $S_2^i(f) \vdash \phi$ (for ϕ in the original language), then already $S_2^i \vdash \phi$. The same also holds for T_2^i.

So, adding f does not let us prove anything new in the original language.

Proof.

- Key idea: for any r, there's a *unique* y such that $A(r, y)$ holds.
- Any $\phi(\ldots f(r) \ldots)$ in the extended language can be re-expressed in the original language:

$$(\exists y \leqslant t)(A(r, y) \wedge \phi(\ldots y \ldots)) \quad \text{or} \quad (\forall y \leqslant t)(A(r, y) \rightarrow \phi(\ldots y \ldots))$$

Theorem

$S^i_2(f)$ is conservative over S^i_2, i.e., if $S^i_2(f) \vdash \phi$ (for ϕ in the original language), then already $S^i_2 \vdash \phi$. The same also holds for T^i_2.

So, adding f does not let us prove anything new in the original language.

Proof.

- Key idea: for any r, there's a *unique* y such that $A(r, y)$ holds.
- Any $\phi(\ldots f(r) \ldots)$ in the extended language can be re-expressed in the original language:

  $$(\exists y \leqslant t)(A(r, y) \wedge \phi(\ldots y \ldots)) \quad \text{or} \quad (\forall y \leqslant t)(A(r, y) \rightarrow \phi(\ldots y \ldots))$$

  and if ϕ is Σ^b_i in the new language, then its re-expressed form is Σ^b_i in the original language (choose whichever preserves the quantifier alternation).

$\square$

Similar definitions and results hold for predicates:

Definition

A predicate P is Δ_i^b-definable in R provided there are a Σ_i^b-formula A and a Π_i^b-formula B which are R-provably equivalent and define P.

We can conservatively add a Δ_i^b-definable predicate P to S_2^i or T_2^i and use it freely in the induction schemes.

Bootstrapping S_2^1

An exhausting amount of work.

Example

Some Σ_1^b-definable functions and Δ_1^b-definable predicates in S_2^1

- The predecessor function
- Subtraction
- Numones: number of ones in the binary representation of a number
- $Seq(w)$ true iff w is a valid sequence
- $Len(w)$ the number of elements in w
- $\beta(i, w)$ the value of the ith element of a sequence w

  $\vdots$

Σ_1^b-definability of predecessor in S_2^1

Example

The predecessor function (an inverse to the successor function):

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

Σ_1^b-definability of predecessor in S_2^1

Example

The predecessor function (an inverse to the successor function):

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

The uniqueness condition only uses BASIC and no induction.

Σ_1^b-definability of predecessor in S_2^1

Example

The predecessor function (an inverse to the successor function):

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

The uniqueness condition only uses BASIC and no induction. For totality, let $M(a, b)$ be the defining equation for $P(a) = b$. We want to prove $S_2^1 \vdash \forall x \exists y \leqslant x \; M(x, y)$.

Σ_1^b-definability of predecessor in S_2^1

Example

The predecessor function (an inverse to the successor function):

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

The uniqueness condition only uses BASIC and no induction. For totality, let $M(a, b)$ be the defining equation for $P(a) = b$. We want to prove $S_2^1 \vdash \forall x \exists y \leqslant x\ M(x, y)$. Using BASIC we can prove:

$$\exists y \leqslant 0\ M(0, y)$$

$$(\exists y \leqslant \lfloor \frac{1}{2} x \rfloor)\ M(\lfloor \frac{1}{2} x \rfloor, y) \to (\exists y \leqslant x)\ M(x, y)$$

Σ_1^b-definability of predecessor in S_2^1

Example

The predecessor function (an inverse to the successor function):

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

The uniqueness condition only uses BASIC and no induction. For totality, let $M(a, b)$ be the defining equation for $P(a) = b$. We want to prove $S_2^1 \vdash \forall x \exists y \leqslant x\ M(x, y)$. Using BASIC we can prove:

$$\exists y \leqslant 0\ M(0, y)$$

$$(\exists y \leqslant \lfloor \frac{1}{2} x \rfloor)\ M(\lfloor \frac{1}{2} x \rfloor, y) \rightarrow (\exists y \leqslant x)\ M(x, y)$$

By Σ_1^b-PIND:

$$S_2^1 \vdash \forall x \exists y \leqslant x\ M(x, y)$$

Theorem (Buss'85)

Every polynomial time function is Σ_1^b-definable in S_2^1.

Theorem (Buss'85)

Every polynomial time function is Σ_1^b-definable in S_2^1.

Proof sketch.

By induction on the complexity of the definition of f (using Cobham's characterization of polytime functions).

Theorem (Buss'85)

Every polynomial time function is Σ_1^b-definable in S_2^1.

Proof sketch.

By induction on the complexity of the definition of f (using Cobham's characterization of polytime functions). Important case: limited recursion on notation. Very simple case:

$$f(0) = a$$
$$f(x) = h(f(\lfloor \frac{1}{2}x \rfloor))$$

where $|f(x)| \leqslant p(|x|)$.

Theorem (Buss'85)

Every polynomial time function is Σ_1^b-definable in S_2^1.

Proof sketch.

By induction on the complexity of the definition of f (using Cobham's characterization of polytime functions). Important case: limited recursion on notation. Very simple case:

$$f(0) = a$$
$$f(x) = h(f(\lfloor \frac{1}{2}x \rfloor))$$

where $|f(x)| \leqslant p(|x|)$. By the induction hypothesis, let $H \in \Sigma_1^b$ define h, with the intended meaning $H(x, y) := y = h(x)$.

Theorem (Buss'85)

Every polynomial time function is Σ_1^b-definable in S_2^1.

Proof sketch.

By induction on the complexity of the definition of f (using Cobham's characterization of polytime functions). Important case: limited recursion on notation. Very simple case:

$$f(0) = a$$
$$f(x) = h(f(\lfloor \frac{1}{2}x \rfloor))$$

where $|f(x)| \leqslant p(|x|)$. By the induction hypothesis, let $H \in \Sigma_1^b$ define h, with the intended meaning $H(x, y) := y = h(x)$. We define

$$F(x, y) := \exists w \leqslant t_F(Seq(w) \wedge w_0 = a \wedge \forall i < |x| \underbrace{(w_{i+1} = h(w_i))}_{H(w_i, w_{i+1})} \wedge w_{|x|} = y)$$

Proof sketch.

w codes the steps of the computation. w has $|x|$ many elements w_i; each w_i is f of something; $|w_i|$ is bounded by $p(|x|)$. So, $|w|$ is approximately bounded by $|x| \cdot p(|x|)$ and hence w is bounded by a term t_F.
We have to prove $S_2^1 \vdash \forall x \exists y \leqslant t_F F(x, y)$ and the uniqueness. $\square$

Proof sketch.

w codes the steps of the computation. w has $|x|$ many elements w_i; each w_i is f of something; $|w_i|$ is bounded by $p(|x|)$. So, $|w|$ is approximately bounded by $|x| \cdot p(|x|)$ and hence w is bounded by a term t_F.
We have to prove $S_2^1 \vdash \forall x \exists y \leqslant t_F F(x, y)$ and the uniqueness. $\square$

The above theorem shows that S_2^1, and the Σ_1^b-definable functions are sufficiently strong to introduce polynomial time properties. (The converse holds too: every Σ_1^b-definable function is polynomial time.)

Proof sketch.

w codes the steps of the computation. w has $|x|$ many elements w_i; each w_i is f of something; $|w_i|$ is bounded by $p(|x|)$. So, $|w|$ is approximately bounded by $|x| \cdot p(|x|)$ and hence w is bounded by a term t_F.
We have to prove $S^1_2 \vdash \forall x \exists y \leqslant t_F F(x, y)$ and the uniqueness. □

The above theorem shows that S^1_2, and the Σ^b_1-definable functions are sufficiently strong to introduce polynomial time properties. (The converse holds too: every Σ^b_1-definable function is polynomial time.)

Hence, we can w.l.o.g. assume that all polynomial time functions are present in the languages of our theories of bounded arithmetic.

Theorem (Buss '85)

Every polynomial time predicate is Δ_1^b-definable by S_2^1.

(Again, a converse holds.)

Thus, every polynomial time predicate can be conservatively introduced to S_2^i or T_2^i with its defining axioms, and used freely in induction axioms.

Outline

Theorem (Parikh '71)

Let $i \geqslant 1$. Suppose that A is a bounded formula and that S_2^i or T_2^i proves $\forall \bar{x} \exists y A(\bar{x}, y)$. Then there is a term $t(\bar{x})$ such that the same theory proves $\forall \bar{x} \exists y \leqslant t(\bar{x}) A(\bar{x}, y)$.

Theorem (Parikh '71)

Let $i \geqslant 1$. Suppose that A is a bounded formula and that S_2^i or T_2^i proves $\forall \bar{x} \exists y A(\bar{x}, y)$. Then there is a term $t(\bar{x})$ such that the same theory proves $\forall \bar{x} \exists y \leqslant t(\bar{x}) A(\bar{x}, y)$.

Therefore, we cannot define the exponentiation function in bounded arithmetic (and prove its unbounded totality). More generally, we cannot prove unbounded existential statements.

Theorem (Main Theorem for S_2^1, Buss'85)

Suppose f is Σ_1^b-defined by S_2^1. Then f is computable in polynomial time.

Theorem (Main Theorem for S_2^1, Buss'85)

Suppose f is Σ_1^b-defined by S_2^1. Then f is computable in polynomial time.

Corollary

The Σ_1^b-definable functions of S_2^1 are precisely the polytime functions.

This shows S_2^1 has proof-theoretic strength corresponding to polynomial time computation.

Theorem (Main Theorem for S_2^1, Buss'85)

Suppose f is Σ_1^b-defined by S_2^1. Then f is computable in polynomial time.

Corollary

The Σ_1^b-definable functions of S_2^1 are precisely the polytime functions.

This shows S_2^1 has proof-theoretic strength corresponding to polynomial time computation.

The so-called "main theorems" of S_2^i give an exact characterization of the Σ_i^b-definable functions of S_2^i in terms of the computational complexity.

Theorem

$(i \geqslant 1)$ *The Σ_i^b-definable functions of S_2^i are precisely the polynomial time computable functions with an oracle from Σ_{i-1}^p.*

Theorem (corresponding theorem for predicates)

The Δ_i^b-definable predicates of S_2^i are precisely the Δ_i^p- predicates.

An interesting case for predicates:

Corollary

If A is a formula which is S_2^1-provably in NP $\cap$ coNP, then A defines a polynomial time predicate (provably in S_2^1). Being provably in NP $\cap$ coNP means provably equivalent to a Σ_1^b- and to a Π_1^b-formula

- To prove Main Theorem, we formalize S_2^1 in sequent calculus.
- In Buss' thesis: sequent calculus is called natural deduction (e.g. LK: Gentzen's natural deduction calculus)
- The advantage of sequent calculus is that it provides an elegant framework for proof by induction on the complexity of proofs.

Sequent calculus LK

For Γ, Δ finite multisets of formulas, $\Gamma \Rightarrow \Delta$ means $\bigwedge \Gamma \rightarrow \bigvee \Delta$.

$$A \Rightarrow A \quad (\mathrm{Ax})$$

$$\frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\,(Lw) \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}\,(Rw)$$

$$\frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\,(Lc) \qquad \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}\,(Rc)$$

$$\frac{A, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\,(L\wedge_1) \qquad \frac{B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\,(L\wedge_2)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\,(R\wedge) \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A \vee B}\,(R\vee_1)$$

$$\frac{\Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \vee B}\,(R\vee_2) \qquad \frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\,(L\vee)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta}\,(L\rightarrow) \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}\,(R\rightarrow)$$

$$\frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}\,(L\neg) \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}\,(R\neg)$$

LK cont'

$$\frac{A(b), \Gamma \Rightarrow \Delta}{\exists x A(x), \Gamma \Rightarrow \Delta}\,(L\exists) \qquad \frac{\Gamma \Rightarrow \Delta, A(t)}{\Gamma \Rightarrow \Delta, \exists x A(x)}\,(R\exists)$$

$$\frac{A(t), \Gamma \Rightarrow \Delta}{\forall x A(x), \Gamma \Rightarrow \Delta}\,(L\forall) \qquad \frac{\Gamma \Rightarrow \Delta, A(b)}{\Gamma \Rightarrow \Delta, \forall x A(x)}\,(R\forall)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\,(Cut)$$

In the quantifier inferences, the free variable b is called the eigenvariable and must not appear in the lower sequent.

LK cont'

$$\frac{A(b), \Gamma \Rightarrow \Delta}{\exists x A(x), \Gamma \Rightarrow \Delta}\ (L\exists) \qquad \frac{\Gamma \Rightarrow \Delta, A(t)}{\Gamma \Rightarrow \Delta, \exists x A(x)}\ (R\exists)$$

$$\frac{A(t), \Gamma \Rightarrow \Delta}{\forall x A(x), \Gamma \Rightarrow \Delta}\ (L\forall) \qquad \frac{\Gamma \Rightarrow \Delta, A(b)}{\Gamma \Rightarrow \Delta, \forall x A(x)}\ (R\forall)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (Cut)$$

In the quantifier inferences, the free variable b is called the eigenvariable and must not appear in the lower sequent.

Theorem (Gentzen '36)

- *LK is sound and complete for first-order logic.*
- *LK without the Cut inference is complete.*

Sequent Calculus Formulation of Bounded Arithmetic

We enlarge LK as follows:

1. Allow equality axioms and BASIC axioms as initial sequents (call them BASIC$^{=}$). An initial sequent will contain only *atomic* formulas (i.e., contain no quantifiers or propositional connectives).

Sequent Calculus Formulation of Bounded Arithmetic

We enlarge LK as follows:

1. Allow equality axioms and BASIC axioms as initial sequents (call them BASIC$^=$). An initial sequent will contain only *atomic* formulas (i.e., contain no quantifiers or propositional connectives).
2. Add inferences for bounded quantifiers (call them *BQ*):

$$\frac{b \leqslant s, A(b), \Gamma \Rightarrow \Delta}{(\exists x \leqslant s)A(x), \Gamma \Rightarrow \Delta} \qquad \frac{\Gamma \Rightarrow \Delta, A(t)}{t \leqslant s, \Gamma \Rightarrow \Delta, (\exists x \leqslant s)A(x)}$$

$$\frac{A(t), \Gamma \Rightarrow \Delta}{t \leqslant s, (\forall x \leqslant s)A(x), \Gamma \Rightarrow \Delta} \qquad \frac{b \leqslant s, \Gamma \Rightarrow \Delta, A(b)}{\Gamma \Rightarrow \Delta, (\forall x \leqslant s)A(x)}$$

Sequent Calculus Formulation of Bounded Arithmetic

We enlarge LK as follows:

1. Allow equality axioms and BASIC axioms as initial sequents (call them BASIC$^=$). An initial sequent will contain only *atomic* formulas (i.e., contain no quantifiers or propositional connectives).
2. Add inferences for bounded quantifiers (call them BQ):

$$\frac{b \leqslant s, A(b), \Gamma \Rightarrow \Delta}{(\exists x \leqslant s)A(x), \Gamma \Rightarrow \Delta} \qquad \frac{\Gamma \Rightarrow \Delta, A(t)}{t \leqslant s, \Gamma \Rightarrow \Delta, (\exists x \leqslant s)A(x)}$$

$$\frac{A(t), \Gamma \Rightarrow \Delta}{t \leqslant s, (\forall x \leqslant s)A(x), \Gamma \Rightarrow \Delta} \qquad \frac{b \leqslant s, \Gamma \Rightarrow \Delta, A(b)}{\Gamma \Rightarrow \Delta, (\forall x \leqslant s)A(x)}$$

3. Induction inferences:

$$\frac{A(b), \Gamma \Rightarrow \Delta, A(b+1)}{A(0), \Gamma \Rightarrow \Delta, A(t)}\ (\Sigma_i^b\text{-IND}) \qquad \frac{A(\lfloor \frac{1}{2} b \rfloor), \Gamma \Rightarrow \Delta, A(b)}{A(0), \Gamma \Rightarrow \Delta, A(t)}\ (\Sigma_i^b\text{-PIND})$$

for $A \in \Sigma_i^b$, b eigenvariable, t term.

Definition

$GS_2^i := LK + \mathrm{BASIC}^{=} + BQ + \Sigma_i^b\text{-PIND}$
$GT_2^i := LK + \mathrm{BASIC}^{=} + BQ + \Sigma_i^b\text{-IND}$

Definition

$GS_2^i := LK + \mathrm{BASIC}^{=} + BQ + \Sigma_i^b\text{-PIND}$
$GT_2^i := LK + \mathrm{BASIC}^{=} + BQ + \Sigma_i^b\text{-IND}$

By abuse of notation, we also write S_2^i (resp. T_2^i) for GS_2^i (resp. GT_2^i).

Theorem

The Σ_i^b-IND (respectively, Σ_i^b-PIND) rule is equivalent to the Σ_i^b-IND (respectively, Σ_i^b-PIND) axioms. Hence the new definitions of S_2^i and T_2^i agree with the earlier definitions.

Example (Σ_i^b-IND rule can derive the Σ_i^b-IND axiom)

Let A be any Σ_i^b–formula, and let a and b be any free variables not appearing in A. Then we can derive the *IND* axiom for A by:

$$\dfrac{A(a) \longrightarrow A(a) \qquad A(Sa) \longrightarrow A(Sa)}{A(a) \supset A(Sa), A(a) \longrightarrow A(Sa)}$$

$$\dfrac{A(a) \supset A(Sa), A(a) \longrightarrow A(Sa)}{(\forall x)(A(x) \supset A(Sx)), A(a) \longrightarrow A(Sa)}$$

$$\dfrac{(\forall x)(A(x) \supset A(Sx)), A(a) \longrightarrow A(Sa)}{(\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow A(b)}$$

$$\dfrac{(\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow A(b)}{(\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow (\forall x)A(x)}$$

$$\dfrac{(\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow (\forall x)A(x)}{A(0) \wedge (\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow (\forall x)A(x)}$$

$$\dfrac{A(0) \wedge (\forall x)(A(x) \supset A(Sx)), A(0) \longrightarrow (\forall x)A(x)}{A(0), A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}$$

$$\dfrac{A(0), A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}{A(0) \wedge (\forall x)(A(x) \supset A(Sx)), A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}$$

$$\dfrac{A(0) \wedge (\forall x)(A(x) \supset A(Sx)), A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}{A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}$$

$$\dfrac{A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \longrightarrow (\forall x)A(x)}{\longrightarrow A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)A(x)}$$

Definition

A cut inference is *free* unless the cut formula is the direct descendant either of a formula in an initial sequent or of a principal formula of an induction inference.

Definition

A cut inference is *free* unless the cut formula is the direct descendant either of a formula in an initial sequent or of a principal formula of an induction inference.

Theorem (Free-cut elimination, Gentzen)

If a sequent is provable in S^i_2 (or T^i_2), it has a proof in the same theory with no free cuts.

Definition

A cut inference is *free* unless the cut formula is the direct descendant either of a formula in an initial sequent or of a principal formula of an induction inference.

Theorem (Free-cut elimination, Gentzen)

If a sequent is provable in S_2^i (or T_2^i), it has a proof in the same theory with no free cuts.

- In a free-cut free proof, every formula is a subformula of an induction formula, an axiom, or the conclusion.
- In S_2^i and T_2^i, cuts can be restricted to Σ_i^b-formulas.
- Hence any sequent of Σ_i^b-formulas provable in S_2^i (resp. T_2^i) has a proof in which all formulas are Σ_i^b.

Outline of the proof of Main Theorem

Theorem (Main Theorem for S_2^1)

If f is Σ_1^b defined by S_2^1, then f is polynomial time computable.

Outline of the proof of Main Theorem

Theorem (Main Theorem for S^1_2)

If f is Σ^b_1 defined by S^1_2, then f is polynomial time computable.

Step 1:

- Suppose $S^1_2 \vdash \forall x \exists y \leqslant t(x) A(x, y)$

Outline of the proof of Main Theorem

Theorem (Main Theorem for S^1_2)

If f is Σ^b_1 defined by S^1_2, then f is polynomial time computable.

Step 1:

- Suppose $S^1_2 \vdash \forall x \exists y \leqslant t(x) A(x,y)$
- Then, $S^1_2 \vdash \exists y \leqslant t(c)\ A(c,y)$

Outline of the proof of Main Theorem

Theorem (Main Theorem for S^1_2)

If f is Σ^b_1 defined by S^1_2, then f is polynomial time computable.

Step 1:

- Suppose $S^1_2 \vdash \forall x \exists y \leqslant t(x) A(x, y)$
- Then, $S^1_2 \vdash \exists y \leqslant t(c)\ A(c, y)$
- Free-cut elim: $S^1_2 \vdash \exists y \leqslant t(c)\ A(c, y)$. (The proof π only has Σ^b_1-formulas.)

Outline of the proof of Main Theorem

Theorem (Main Theorem for S_2^1)

If f is Σ_1^b defined by S_2^1, then f is polynomial time computable.

Step 1:

- Suppose $S_2^1 \vdash \forall x \exists y \leqslant t(x) A(x, y)$
- Then, $S_2^1 \vdash \exists y \leqslant t(c)\ A(c, y)$
- Free-cut elim: $S_2^1 \vdash \exists y \leqslant t(c)\ A(c, y)$. (The proof π only has Σ_1^b-formulas.)

Step 2: Use π to extract an algorithm to compute f.

Outline of the proof of Main Theorem

Theorem (Main Theorem for S^1_2)

If f is Σ^b_1 defined by S^1_2, then f is polynomial time computable.

Step 1:

- Suppose $S^1_2 \vdash \forall x \exists y \leqslant t(x) A(x, y)$
- Then, $S^1_2 \vdash \exists y \leqslant t(c)\ A(c, y)$
- Free-cut elim: $S^1_2 \vdash \exists y \leqslant t(c)\ A(c, y)$. (The proof π only has Σ^b_1-formulas.)

Step 2: Use π to extract an algorithm to compute f.

Step 1: immediate (from free-cut elim.). Step 2: Witnessing lemma.

Witness

For Step 2, we need a notion of a witness. Witnesses are just values for the x that the formula $(\exists x \leqslant t)\phi(\bar{y}, x)$ is true, where ϕ is sharply bounded.

Witness

For Step 2, we need a notion of a witness. Witnesses are just values for the x that the formula $(\exists x \leqslant t)\phi(\bar{y}, x)$ is true, where ϕ is sharply bounded.

Definition

Let $A(\bar{c})$ be a formula of the above form. The predicate $\mathrm{Wit}_A(\bar{c}, u)$ is defined so that

- If A is $(\exists x \leqslant t)B(\bar{c}, x)$, $B \in \Delta_0^b$, then $\mathrm{Wit}_A(\bar{c}, u)$ is the formula

$$u \leqslant t \wedge B(\bar{c}, u).$$

Witness

For Step 2, we need a notion of a witness. Witnesses are just values for the x that the formula $(\exists x \leqslant t)\phi(\bar{y}, x)$ is true, where ϕ is sharply bounded.

Definition

Let $A(\bar{c})$ be a formula of the above form. The predicate $\mathrm{Wit}_A(\bar{c}, u)$ is defined so that

- If A is $(\exists x \leqslant t)B(\bar{c}, x)$, $B \in \Delta_0^b$, then $\mathrm{Wit}_A(\bar{c}, u)$ is the formula

$$u \leqslant t \wedge B(\bar{c}, u).$$

- If A is in Δ_0^b, then $\mathrm{Wit}_A(\bar{c}, u)$ is just $A(\bar{c})$.

Witness

For Step 2, we need a notion of a witness. Witnesses are just values for the x that the formula $(\exists x \leqslant t)\phi(\bar{y}, x)$ is true, where ϕ is sharply bounded.

Definition

Let $A(\bar{c})$ be a formula of the above form. The predicate $\mathrm{Wit}_A(\bar{c}, u)$ is defined so that

- If A is $(\exists x \leqslant t)B(\bar{c}, x)$, $B \in \Delta_0^b$, then $\mathrm{Wit}_A(\bar{c}, u)$ is the formula

$$u \leqslant t \wedge B(\bar{c}, u).$$

- If A is in Δ_0^b, then $\mathrm{Wit}_A(\bar{c}, u)$ is just $A(\bar{c})$.

We immediately have:

Lemma

- *$A(\bar{c}) \leftrightarrow (\exists u) Wit_A(\bar{c}, u)$ in S_2^1.*
- *Wit_A is a Δ_0^b-formula.*

Theorem (Witnessing Lemma)

Let $\Gamma \Rightarrow \Delta$ be an S_2^1-provable sequent of Σ_1^b formulas with free variables $\bar{c}$. Then there is a polynomial time function $f(\bar{c}, \bar{u})$ such that

$$\bigwedge_{\gamma_i \in \Gamma} Wit_{\gamma_i}(\bar{c}, u_i) \rightarrow \bigvee_{\delta_j \in \Delta} Wit_{\delta_j}(\bar{c}, f(\bar{c}, \bar{u})).$$

Proof sketch (of Witnessing Lemma (for S^1_2))

By induction on the number of lines in a free-cut free S^1_2-proof of $\Gamma \Rightarrow \Delta$, where all the formulas in Γ, Δ are Σ^b_1.

Proof sketch (of Witnessing Lemma (for S^1_2))

By induction on the number of lines in a free-cut free S^1_2-proof of $\Gamma \Rightarrow \Delta$, where all the formulas in Γ, Δ are Σ^b_1. Thus (simple case):

$$S^1_2 \vdash (\exists x_1 \leqslant t_1)\gamma_1(x_1, c), \ldots, (\exists x_k \leqslant t_k)\gamma_k(x_k, c) \Rightarrow$$

$$(\exists y_1 \leqslant s_1)\delta_1(y_1, c), \ldots, (\exists y_\ell \leqslant s_\ell)\delta_\ell(y_\ell, c)$$

Proof sketch (of Witnessing Lemma (for S^1_2))

By induction on the number of lines in a free-cut free S^1_2-proof of $\Gamma \Rightarrow \Delta$, where all the formulas in Γ, Δ are Σ^b_1. Thus (simple case):

$$S^1_2 \vdash (\exists x_1 \leqslant t_1)\gamma_1(x_1, c), \ldots, (\exists x_k \leqslant t_k)\gamma_k(x_k, c) \Rightarrow$$

$$(\exists y_1 \leqslant s_1)\delta_1(y_1, c), \ldots, (\exists y_\ell \leqslant s_\ell)\delta_\ell(y_\ell, c)$$

- Inner formulas, γ_i's, δ_j's, are sharply bounded; take polytime to check.

Proof sketch (of Witnessing Lemma (for S^1_2))

By induction on the number of lines in a free-cut free S^1_2-proof of $\Gamma \Rightarrow \Delta$, where all the formulas in Γ, Δ are Σ^b_1. Thus (simple case):

$$S^1_2 \vdash (\exists x_1 \leqslant t_1)\gamma_1(x_1, c), \ldots, (\exists x_k \leqslant t_k)\gamma_k(x_k, c) \Rightarrow$$

$$(\exists y_1 \leqslant s_1)\delta_1(y_1, c), \ldots, (\exists y_\ell \leqslant s_\ell)\delta_\ell(y_\ell, c)$$

- Inner formulas, γ_i's, δ_j's, are sharply bounded; take polytime to check.
- Witnessing lemma: there is a polynomial time procedure f such that
  - f takes $c, x_1, \ldots, x_k$, as input. These have the property that they make all γ_i's true, i.e., satisfy $\bigwedge_{i=1}^k \gamma_i(x_i, c)$.
  - f outputs y, which is a witness to one of δ_j's.

Proof sketch (of Witnessing Lemma (for S^1_2))

By induction on the number of lines in a free-cut free S^1_2-proof of $\Gamma \Rightarrow \Delta$, where all the formulas in Γ, Δ are Σ^b_1. Thus (simple case):

$$S^1_2 \vdash (\exists x_1 \leqslant t_1)\gamma_1(x_1, c), \ldots, (\exists x_k \leqslant t_k)\gamma_k(x_k, c) \Rightarrow$$

$$(\exists y_1 \leqslant s_1)\delta_1(y_1, c), \ldots, (\exists y_\ell \leqslant s_\ell)\delta_\ell(y_\ell, c)$$

- Inner formulas, γ_i's, δ_j's, are sharply bounded; take polytime to check.
- Witnessing lemma: there is a polynomial time procedure f such that
  - f takes $c, x_1, \ldots, x_k$, as input. These have the property that they make all γ_i's true, i.e., satisfy $\bigwedge_{i=1}^{k} \gamma_i(x_i, c)$.
  - f outputs y, which is a witness to one of δ_j's.

As an example, suppose that the proof ends with

$$\frac{B(\lfloor \frac{1}{2} a \rfloor) \Rightarrow B(a)}{B(0) \Rightarrow B(t)}$$

Proof.

Use IH, and limited recursion on notation to define f. By the induction hypothesis, there is a witness function g for the premise.

Proof.

Use IH, and limited recursion on notation to define f. By the induction hypothesis, there is a witness function g for the premise.
Now define f by recursion on notation as

$$f(w, 0, \bar{c}) = w$$
$$f(w, a, \ c) = g(f(w, \lfloor \frac{1}{2} a \rfloor, \bar{c}), a, \bar{c})$$

Proof.

Use IH, and limited recursion on notation to define f. By the induction hypothesis, there is a witness function g for the premise.
Now define f by recursion on notation as

$$f(w, 0, \bar{c}) = w$$

$$f(w, a, \ c) = g(f(w, \lfloor \frac{1}{2} a \rfloor, \bar{c}), a, \bar{c})$$

We show that f is polynomial time computable by showing that $|f|$ is bounded.

$\square$

Proving Main Theorem using Witnessing Lemma

Theorem

Let f be Σ_1^b-defined by S_2^1. Then f is polytime computable.

Proving Main Theorem using Witnessing Lemma

Theorem

Let f be Σ_1^b-defined by S_2^1. Then f is polytime computable.

Importing Witnessing Lemma.

Recall by Step 1:

$$S_2^1 \vdash \Rightarrow (\exists y \leqslant t)\, A(c, y)$$

by a free-cut free π, where $A \in \Sigma_1^b$.

Proving Main Theorem using Witnessing Lemma

Theorem

Let f be Σ_1^b-defined by S_2^1. Then f is polytime computable.

Importing Witnessing Lemma.

Recall by Step 1:

$$S_2^1 \vdash \Rightarrow (\exists y \leqslant t)\ A(c, y)$$

by a free-cut free π, where $A \in \Sigma_1^b$.
Apply Witnessing Lemma to the last line of the proof
$(\Rightarrow (\exists y \leqslant t)A(y, c))$:

Proving Main Theorem using Witnessing Lemma

Theorem

Let f be Σ_1^b-defined by S_2^1. Then f is polytime computable.

Importing Witnessing Lemma.

Recall by Step 1:

$$S_2^1 \vdash \Rightarrow (\exists y \leqslant t)\ A(c, y)$$

by a free-cut free π, where $A \in \Sigma_1^b$.
Apply Witnessing Lemma to the last line of the proof
$(\Rightarrow (\exists y \leqslant t)A(y, c))$: Given c and witnesses for all the formulas on the left, we can find in polynomial time a witness for y on the right. So, we get a polytime function of c.

$\square$

Outline

Explanation about $\preceq_{\forall\Sigma^b_{i+1}}$

We have: $S^1_2 \subseteq T^1_2 \preceq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall\Sigma^b_3} S^3_2 \subseteq \dots$

Explanation about $\preceq_{\forall\Sigma^b_{i+1}}$

We have: $S^1_2 \subseteq T^1_2 \preceq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall\Sigma^b_3} S^3_2 \subseteq \dots$

We already discussed that T^i_2 contains S^i_2. Now:

$T^i_2 \preceq_{\forall\Sigma^b_2} S^{i+1}_2$: S^{i+1}_2 is $\forall\Sigma^b_{i+1}$ conservative over T^i_2

Explanation about $\preceq_{\forall\Sigma^b_{i+1}}$

We have: $S^1_2 \subseteq T^1_2 \preceq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall\Sigma^b_3} S^3_2 \subseteq \dots$

We already discussed that T^i_2 contains S^i_2. Now:

$$T^i_2 \preceq_{\forall\Sigma^b_2} S^{i+1}_2 : \quad S^{i+1}_2 \text{ is } \forall\Sigma^b_{i+1} \text{ conservative over } T^i_2$$

Two parts of being $\forall\Sigma^b_{i+1}$ conservative are:

1. $T^i_2 \subseteq S^{i+1}_2$, i.e., S^{i+1}_2 is a possibly stronger, at least no weaker, theory containing T^i_2.
2. if ϕ is of the form $\forall\bar{x}\psi$ with $\psi \in \Sigma^b_{i+1}$ and if $S^{i+1}_2 \vdash \phi$ then $T^i_2 \vdash \phi$.

Explanation about $\preceq_{\forall\Sigma^b_{i+1}}$

We have: $S^1_2 \subseteq T^1_2 \preceq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall\Sigma^b_3} S^3_2 \subseteq \dots$

We already discussed that T^i_2 contains S^i_2. Now:

$$T^i_2 \preceq_{\forall\Sigma^b_2} S^{i+1}_2 : \quad S^{i+1}_2 \text{ is } \forall\Sigma^b_{i+1} \text{ conservative over } T^i_2$$

Two parts of being $\forall\Sigma^b_{i+1}$ conservative are:

1. $T^i_2 \subseteq S^{i+1}_2$, i.e., S^{i+1}_2 is a possibly stronger, at least no weaker, theory containing T^i_2.
2. if ϕ is of the form $\forall\bar{x}\psi$ with $\psi \in \Sigma^b_{i+1}$ and if $S^{i+1}_2 \vdash \phi$ then $T^i_2 \vdash \phi$.

$\preceq_{\forall\Sigma^b_{i+1}}$ means: subset plus nothing new of this complexity class (i.e., theories are equal on the class $\forall\Sigma^b_{i+1}$). The stronger theory adds no new low-complexity facts.

Theorem (Buss '90)

Let $i \geqslant 1$.

1. S_2^{i+1} *is* $\forall\Sigma_{i+1}^b$ *conservative over* T_2^i.
2. *In particular,* T_2^i *can* Σ_{i+1}^b *define precisely the functions in* $FP^{\Sigma_i^p}$.

Theorem (Buss '90)

Let $i \geqslant 1$.

1. S_2^{i+1} *is* $\forall\Sigma_{i+1}^b$ *conservative over* T_2^i.
2. *In particular,* T_2^i *can* Σ_{i+1}^b *define precisely the functions in* $FP^{\Sigma_i^p}$.

Proof idea

With some work, one can show that the witnessing lemma carries over to the base theory when using T_2^i in place of S_2^{i+1}.

This highlights once more the strength of the witnessing lemma: it precisely characterizes what can be done in these theories.

One extra theorem:

Theorem (Krajicek-Pudlak-Takeuti'91, Buss'95, Zambella'96)

If $T_2^i = S_2^{i+1}$, then the polynomial time hierarchy collapses (provably) to $\Sigma_{i+1}^p/poly$ and to $B(\Sigma_{i+2}^b)$.

One extra theorem:

Theorem (Krajicek-Pudlak-Takeuti'91, Buss'95, Zambella'96)

If $T_2^i = S_2^{i+1}$, then the polynomial time hierarchy collapses (provably) to $\Sigma_{i+1}^p/poly$ and to $B(\Sigma_{i+2}^b)$.

- $\Sigma_{i+1}^p/poly$ means the hierarchy collapses to non-uniform Σ_{i+1}^p with polynomial amount of advice or $B(\Sigma_{i+2}^b)$ means to uniform boolean combinations of Σ_{i+2}^b and thus to $\Sigma_{i+3}^b = \Pi_{i+3}^b$.
- In such case, the hierarchy collapses precisely to these levels.
- Conjecture: the hierarchy is strict.
- But, separating these theories means that they do not prove the collapse of the polynomial hierarchy. A major breakthrough.