Kripke Platek set theory over polynomial time computable arithmetic I

Dieter Probst and Thomas Strahm

Institut für Informatik und angewandte Mathematik, Universität Bern

Swansea, April 13, 2007

1. Introduction
2. The case of KPu^r
3. Polynomial time computable arithmetic and extensions
4. Two admissible closures of PTCA
5. Main results

Kripke Platek set theories in proof theory

- the theory of admissibles sets, i.e. Kripke Platek set theory, is one of the most familiar subsystems of Zermelo Fraenkel set theory
- great significane for definability theory and generalized recursion theory
- theories for (iterated) admissibles have long been central for a unifying approach to proof theory

The theory of urelements

- especially in the context of weak set theories it is natural to consider Kripke Platek set theory with urelements
- usually, the theory of urelements is taken to be Peano arithmetic
- in this talk we consider considerably weaker theories of urelements from bounded arithmetic

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic
- we will define two natural closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ for T a theory in the language of PTCA

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic
- we will define two natural closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ for T a theory in the language of PTCA
- the main result will be that

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic
- we will define two natural closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ for T a theory in the language of PTCA
- the main result will be that
  - $\mathbb{A}_0(\mathsf{PTCA})$ is conservative over polynomial time computable arithmetic

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic
- we will define two natural closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ for T a theory in the language of PTCA
- the main result will be that
  - $\mathbb{A}_0(\mathsf{PTCA})$ is conservative over polynomial time computable arithmetic
  - $\mathbb{A}_1(\mathsf{PTCA})$ is conservative over full bounded arithmetic

In this talk

- we will consider weak admissible closures of systems of bounded arithmetic, namely Ferreira's system PTCA of polynomial time computable arithmetic
- we will define two natural closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ for T a theory in the language of PTCA
- the main result will be that
  - $\mathbb{A}_0(\mathsf{PTCA})$ is conservative over polynomial time computable arithmetic
  - $\mathbb{A}_1(\mathsf{PTCA})$ is conservative over full bounded arithmetic
- I will present the systems and main results; Dieter Probst will show you some details of the (quite involved) proofs

1 Introduction

2 The case of KPu^r

3 Polynomial time computable arithmetic and extensions

4 Two admissible closures of PTCA

5 Main results

The theory KPu^r

The theory KPu^r is an extension of Peano arithmetic PA by

- the usual axioms of Kripke Platek set theory, namely pairing, union, Δ_0 separation and Δ_0 collection

The theory KPu^r

The theory KPu^r is an extension of Peano arithmetic PA by

- the usual axioms of Kripke Platek set theory, namely pairing, union, Δ_0 separation and Δ_0 collection
- foundation in the form of the regularity axiom and induction along the natural numbers N for sets

The theory KPu^r

The theory KPu^r is an extension of Peano arithmetic PA by

- the usual axioms of Kripke Platek set theory, namely pairing, union, Δ_0 separation and Δ_0 collection
- foundation in the form of the regularity axiom and induction along the natural numbers N for sets
- an axiom claiming that N forms a set

The theory KPu^r

The theory KPu^r is an extension of Peano arithmetic PA by

- the usual axioms of Kripke Platek set theory, namely pairing, union, Δ_0 separation and Δ_0 collection
- foundation in the form of the regularity axiom and induction along the natural numbers N for sets
- an axiom claiming that N forms a set

The theory KPu^r

The theory KPu^r is an extension of Peano arithmetic PA by

- the usual axioms of Kripke Platek set theory, namely pairing, union, Δ_0 separation and Δ_0 collection
- foundation in the form of the regularity axiom and induction along the natural numbers N for sets
- an axiom claiming that N forms a set

Theorem (Jäger)

KPu^r *is a conservative extension of Peano arithmetic* PA.

Related systems

- the subsystem of second order arithmetic $\Sigma^1_1\text{-AC}\!\restriction$
- the system of explicit mathematics $\mathsf{EM}_0\!\restriction + \mathsf{J}$
- Jäger's fixed point theory with ordinals PA^r_Ω
- similar subsystems of CZF and Martin-Löf type theory

1 Introduction

2 The case of KPu^r

3 Polynomial time computable arithmetic and extensions

4 Two admissible closures of PTCA

5 Main results

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \ldots$ (ranging over binary words)

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \ldots$ (ranging over binary words)
- constants $\varepsilon, 0, 1$ (empty word, zero, one)

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \ldots$ (ranging over binary words)
- constants $\varepsilon, 0, 1$ (empty word, zero, one)
- the binary function symbols $*$ and $\times$ (word concatenation and multiplication)

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \dots$ (ranging over binary words)
- constants $\varepsilon, 0, 1$ (empty word, zero, one)
- the binary function symbols $*$ and $\times$ (word concatenation and multiplication)
- the binary relation symbol $\sqsubseteq$ (initial subword relation)

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \ldots$ (ranging over binary words)
- constants $\varepsilon, 0, 1$ (empty word, zero, one)
- the binary function symbols $*$ and $\times$ (word concatenation and multiplication)
- the binary relation symbol $\sqsubseteq$ (initial subword relation)

The language $\mathcal{L}$

$\mathcal{L}$ is a first order language including

- variables $a, b, c, x, y, z, \ldots$ (ranging over binary words)
- constants $\varepsilon, 0, 1$ (empty word, zero, one)
- the binary function symbols $*$ and $\times$ (word concatenation and multiplication)
- the binary relation symbol $\sqsubseteq$ (initial subword relation)

Defining the relations $\sqsubseteq^*$ and $\leq$

$$s \sqsubseteq^* t \quad := \quad (\exists x)(x \sqsubseteq t \land xs \sqsubseteq t) \qquad (s \text{ is a subword of } t)$$

$$s \leq t \quad := \quad 1 \times s \sqsubseteq 1 \times t \qquad (\text{the length of } s \text{ is lte the length of } t)$$

The language $\mathcal{L}_p$

$\mathcal{L}_p$ is a first order language extending $\mathcal{L}$

The language $\mathcal{L}_p$

$\mathcal{L}_p$ is a first order language extending $\mathcal{L}$

- by adding a function symbol for each description of a polynomial time computable function

The language $\mathcal{L}_p$

$\mathcal{L}_p$ is a first order language extending $\mathcal{L}$

- by adding a function symbol for each description of a polynomial time computable function
- the terms of $\mathcal{L}$ act as bounding terms, similar to Cobham's characterization of the polynomial time computable functions

The language $\mathcal{L}_p$

$\mathcal{L}_p$ is a first order language extending $\mathcal{L}$

- by adding a function symbol for each description of a polynomial time computable function
- the terms of $\mathcal{L}$ act as bounding terms, similar to Cobham's characterization of the polynomial time computable functions
- i.e. the polytime functions are generated inductively with the schemata of composition and bounded iteration from a set of initial functions

Some formula classes

Some formula classes

Definition (Δ_0^b, Σ_1^b, Σ_∞^b formulas)

Some formula classes

Definition (Δ_0^b, Σ_1^b, Σ_∞^b formulas)

1. the class of Δ_0^b formulas of $\mathcal{L}_p$ is generated from literals by means of conjunction, disjunction and sharply bounded (i.e. subword) quantification.

Some formula classes

Definition (Δ^b_0, Σ^b_1, Σ^b_∞ formulas)

1. the class of Δ^b_0 formulas of $\mathcal{L}_p$ is generated from literals by means of conjunction, disjunction and sharply bounded (i.e. subword) quantification.
2. an $\mathcal{L}_p$ formula is called Σ^b_1 if it is of the form $(\exists x \leq t)A(x)$ with A Δ^b_0.

Some formula classes

Definition (Δ_0^b, Σ_1^b, Σ_∞^b formulas)

1. the class of Δ_0^b formulas of $\mathcal{L}_p$ is generated from literals by means of conjunction, disjunction and sharply bounded (i.e. subword) quantification.
2. an $\mathcal{L}_p$ formula is called Σ_1^b if it is of the form $(\exists x \leq t)A(x)$ with A Δ_0^b.
3. a formula is called *bounded* or Σ_∞^b if all its quantifiers are bounded in the sense of $\leq$.

The theories PTCA, PTCA$^+$ and PHCA

The theories PTCA, PTCA$^+$ and PHCA

- Ferreira's system PTCA of polynomial time computable arithmetic is based on classical logic and comprises defining axioms for the function and relation symbols of the language $\mathcal{L}_p$. PTCA includes the schema of notation induction on binary words for *quantifier free* formulas, i.e.

$$A(\varepsilon) \wedge (\forall x)(A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow (\forall x)A(x)$$

for each quantifier-free formula $A(x)$ of $\mathcal{L}_p$.

The theories PTCA, PTCA$^+$ and PHCA

- Ferreira's system PTCA of polynomial time computable arithmetic is based on classical logic and comprises defining axioms for the function and relation symbols of the language $\mathcal{L}_p$. PTCA includes the schema of notation induction on binary words for *quantifier free* formulas, i.e.

$$A(\varepsilon) \wedge (\forall x)(A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow (\forall x)A(x)$$

  for each quantifier-free formula $A(x)$ of $\mathcal{L}_p$.
- PTCA$^+$ extends PTCA by the schema of notation induction for Σ_1^b formulas of $\mathcal{L}_p$

The theories PTCA, PTCA$^+$ and PHCA

- Ferreira's system PTCA of polynomial time computable arithmetic is based on classical logic and comprises defining axioms for the function and relation symbols of the language $\mathcal{L}_p$. PTCA includes the schema of notation induction on binary words for *quantifier free* formulas, i.e.

$$A(\varepsilon) \wedge (\forall x)(A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow (\forall x)A(x)$$

  for each quantifier-free formula $A(x)$ of $\mathcal{L}_p$.
- PTCA$^+$ extends PTCA by the schema of notation induction for Σ_1^b formulas of $\mathcal{L}_p$
- Σ_∞^b-NIA is the extension of PTCA$^+$ where notation induction is permitted for all bounded or Σ_∞^b formulas of $\mathcal{L}_p$. We will use the name PHCA (polynomial hierarchy computable arithmetic) instead of Σ_∞^b-NIA.

Reflection and collection principles

Reflection and collection principles

(Σ-sRef) and (Σ-bColl)

Reflection and collection principles

(Σ-sRef) and (Σ-bColl)

- Sharp Σ reflection [$A(x, y)$ is a Δ_0^b formula of $\mathcal{L}_p$]

$$(\forall x \sqsubseteq^* b)(\exists y)A(x, y) \rightarrow (\exists z)(\forall x \sqsubseteq^* b)(\exists y \sqsubseteq^* z)A(x, y) \quad (\Sigma\text{-sRef})$$

Reflection and collection principles

(Σ-sRef) and (Σ-bColl)

- Sharp Σ reflection [$A(x, y)$ is a Δ_0^b formula of $\mathcal{L}_p$]

$$(\forall x \sqsubseteq^* b)(\exists y)A(x, y) \rightarrow (\exists z)(\forall x \sqsubseteq^* b)(\exists y \sqsubseteq^* z)A(x, y) \quad (\Sigma\text{-sRef})$$

- Bounded collection [$A(x, y)$ is a Σ_∞^b formula of $\mathcal{L}_p$]

$$(\forall x \leq b)(\exists y)A(x, y) \rightarrow (\exists z)(\forall x \leq b)(\exists y \leq z)A(x, y) \quad (\Sigma\text{-bColl})$$

Conservation results

Conservation results

Theorem (Cantini)

We have that PTCA + (Σ-sRef) *is a conservative extenions of* PTCA *for* $\forall\exists\Sigma_1^b$ *statements of* $\mathcal{L}_p$.

Conservation results

Theorem (Cantini)

We have that PTCA + (Σ-sRef) *is a conservative extenions of* PTCA *for* $\forall\exists\Sigma_1^b$ *statements of* $\mathcal{L}_p$.

Theorem (Buss, Ferreira)

We have that PHCA + (Σ-bColl) *is a conservative extenions of* PHCA *for* $\forall\exists\Sigma_\infty^b$ *statements of* $\mathcal{L}_p$.

1 Introduction

2 The case of KPu^r

3 Polynomial time computable arithmetic and extensions

4 Two admissible closures of PTCA

5 Main results

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) informally

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) informally

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) are admissible closures of PTCA, i.e. the urelements are the binary words $\mathbb{W} = \{0, 1\}^*$. However, we do not claim that $\mathbb{W}$ forms a set; it is merely a class in our setting. We have two basic set existence principles for collections of words, namely

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) informally

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) are admissible closures of PTCA, i.e. the urelements are the binary words $\mathbb{W} = \{0,1\}^*$. However, we do not claim that $\mathbb{W}$ forms a set; it is merely a class in our setting. We have two basic set existence principles for collections of words, namely

Two basic set existence principles

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) informally

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) are admissible closures of PTCA, i.e. the urelements are the binary words $\mathbb{W} = \{0, 1\}^*$. However, we do not claim that $\mathbb{W}$ forms a set; it is merely a class in our setting. We have two basic set existence principles for collections of words, namely

Two basic set existence principles

(W.0) *The collection of all subwords of a given binary word forms a set;*

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) informally

$\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA) are admissible closures of PTCA, i.e. the urelements are the binary words $\mathbb{W} = \{0, 1\}^*$. However, we do not claim that $\mathbb{W}$ forms a set; it is merely a class in our setting. We have two basic set existence principles for collections of words, namely

Two basic set existence principles

(W.0) *The collection of all subwords of a given binary word forms a set;*

(W.1) *The collection of all words whose length is less than or equal to length of a given binary word forms a set.*

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$

- We fix any theory T in the language $\mathcal{L}_p$ of binary strings. Our aim is to define two admissible closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ of T.

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$

- We fix any theory T in the language $\mathcal{L}_p$ of binary strings. Our aim is to define two admissible closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ of T.
- $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ are formulated in the extension $\mathcal{L}_p^* = \mathcal{L}_p(\in, \mathsf{W}, \mathsf{S})$ of $\mathcal{L}_p$ by the membership relation symbol $\in$ and the unary relation symbols W and S for the *class* of binary words and sets, respectively.

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$

- We fix any theory T in the language $\mathcal{L}_p$ of binary strings. Our aim is to define two admissible closures $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ of T.
- $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ are formulated in the extension $\mathcal{L}_p^* = \mathcal{L}_p(\in, \mathsf{W}, \mathsf{S})$ of $\mathcal{L}_p$ by the membership relation symbol $\in$ and the unary relation symbols W and S for the *class* of binary words and sets, respectively.
- Formulas, Δ_0 formulas and Σ formulas of $\mathcal{L}_p^*$ are defined as usual.

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

The logical axioms of $\mathbb{A}_0(\mathsf{T})$ comprise the usual axioms of classical first order logic with equality. The non-logical include:

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

The logical axioms of $\mathbb{A}_0(\mathsf{T})$ comprise the usual axioms of classical first order logic with equality. The non-logical include:

I. Ontological axioms, part A. We have for all function symbols h and relation symbols R of the language $\mathcal{L}_p$:

$$\mathsf{W}(a) \leftrightarrow \neg\mathsf{S}(a), \quad \mathsf{W}(\vec{b}) \rightarrow \mathsf{W}(h(\vec{b})),$$
$$R(\vec{b}) \rightarrow \mathsf{W}(\vec{b}), \quad a \in b \rightarrow \mathsf{S}(b).$$

UNIVERSITÄT BERN

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

The logical axioms of $\mathbb{A}_0(\mathsf{T})$ comprise the usual axioms of classical first order logic with equality. The non-logical include:

I. Ontological axioms, part A. We have for all function symbols h and relation symbols R of the language $\mathcal{L}_p$:

$$\mathsf{W}(a) \leftrightarrow \neg\mathsf{S}(a), \quad \mathsf{W}(\vec{b}) \rightarrow \mathsf{W}(h(\vec{b})),$$
$$R(\vec{b}) \rightarrow \mathsf{W}(\vec{b}), \quad a \in b \rightarrow \mathsf{S}(b).$$

II. Ontological axioms, part B.

(W.0) $\mathsf{W}(a) \rightarrow \exists x(\mathsf{S}(x) \wedge x = \{y : \mathsf{W}(y) \wedge y \sqsubseteq^* a\})$

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

The logical axioms of $\mathbb{A}_0(\mathsf{T})$ comprise the usual axioms of classical first order logic with equality. The non-logical include:

I. Ontological axioms, part A. We have for all function symbols h and relation symbols R of the language $\mathcal{L}_p$:

$$\mathsf{W}(a) \leftrightarrow \neg\mathsf{S}(a), \quad \mathsf{W}(\vec{b}) \rightarrow \mathsf{W}(h(\vec{b})),$$
$$R(\vec{b}) \rightarrow \mathsf{W}(\vec{b}), \quad a \in b \rightarrow \mathsf{S}(b).$$

II. Ontological axioms, part B.

(W.0) $\mathsf{W}(a) \rightarrow \exists x(\mathsf{S}(x) \wedge x = \{y : \mathsf{W}(y) \wedge y \sqsubseteq^* a\})$

III. Axioms about T. We have for all axioms $A(\vec{x})$ of T whose free variables belong to the list $\vec{x}$:

(T axioms) $\mathsf{W}(\vec{a}) \rightarrow A^{\mathsf{W}}(\vec{a})$.

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

IV. Kripke Platek axioms. We have for all Δ_0 formulas $A(x)$ and $B(x, y)$ of the language $\mathcal{L}_p^*$:

(Pair) $\exists x(a \in x \wedge b \in x)$.

(Union) $\exists x(\forall y \in a)(\forall z \in y)(z \in x)$.

(Δ_0-Sep) $\exists x(\mathsf{S}(x) \wedge x = \{y \in a : A(y)\})$.

(Δ_0-Coll) $(\forall x \in a)\exists y B(x, y) \rightarrow \exists z(\forall x \in a)(\exists y \in z)B(x, y)$.

Defining $\mathbb{A}_0(\mathrm{T})$ and $\mathbb{A}_1(\mathrm{T})$ (ctd.)

IV. Kripke Platek axioms. We have for all Δ_0 formulas $A(x)$ and $B(x, y)$ of the language $\mathcal{L}_p^*$:

(Pair) $\exists x(a \in x \wedge b \in x)$.

(Union) $\exists x(\forall y \in a)(\forall z \in y)(z \in x)$.

(Δ_0-Sep) $\exists x(\mathsf{S}(x) \wedge x = \{y \in a : A(y)\})$.

(Δ_0-Coll) $(\forall x \in a)\exists y B(x, y) \rightarrow \exists z(\forall x \in a)(\exists y \in z)B(x, y)$.

V. Foundation. Here we include the usual regularity axiom:

(Fund) $\mathsf{S}(a) \wedge a \neq \emptyset \rightarrow (\exists x \in a)(\forall y \in x)(y \notin a)$.

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

IV. Kripke Platek axioms. We have for all Δ_0 formulas $A(x)$ and $B(x,y)$ of the language $\mathcal{L}_p^*$:

$$\begin{array}{rl} \text{(Pair)} & \exists x(a \in x \ \wedge \ b \in x). \\ \text{(Union)} & \exists x(\forall y \in a)(\forall z \in y)(z \in x). \\ (\Delta_0\text{-Sep}) & \exists x(\mathsf{S}(x) \ \wedge \ x = \{y \in a : A(y)\}). \\ (\Delta_0\text{-Coll}) & (\forall x \in a)\exists y B(x,y) \ \rightarrow \ \exists z(\forall x \in a)(\exists y \in z)B(x,y). \end{array}$$

V. Foundation. Here we include the usual regularity axiom:

$$\text{(Fund)} \quad \mathsf{S}(a) \wedge a \neq \emptyset \ \rightarrow \ (\exists x \in a)(\forall y \in x)(y \notin a).$$

VI. Set induction on W

$$\begin{array}{rl} (\text{Set-I}_{\mathsf{W}}) & \mathsf{W}(b) \ \wedge \ \varepsilon \in a \ \wedge \ (\forall x \sqsubset b) \bigwedge_{i=0,1} [x \in a \wedge xi \sqsubseteq b \ \rightarrow \ xi \in a] \\ & \qquad \rightarrow \ b \in a. \end{array}$$

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

In the stronger closure $\mathbb{A}_1(\mathsf{T})$ *it is claimed that for each word a we have the set of all words b whose length is less than or equal to the length of a.*

Defining $\mathbb{A}_0(\mathsf{T})$ and $\mathbb{A}_1(\mathsf{T})$ (ctd.)

In the stronger closure $\mathbb{A}_1(\mathsf{T})$ it is claimed that for each word a we have the set of all words b whose length is less than or equal to the length of a.

More precisely, $\mathbb{A}_1(\mathsf{T})$ is obtained from $\mathbb{A}_0(\mathsf{T})$ by replacing (W.0) by the stronger axiom (W.1):

$$\text{(W.1)}\quad \mathsf{W}(a) \rightarrow \exists x(\mathsf{S}(x) \wedge x = \{y : \mathsf{W}(y) \wedge y \leq a\}).$$

Clearly, $\mathbb{A}_1(\mathsf{PTCA})$ proves the weaker axiom (W.0).

Lower bounds

It is immediately seen that PTCA and PHCA are contained in $\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA), respectively:

Lower bounds

It is immediately seen that PTCA and PHCA are contained in $\mathbb{A}_0(\mathrm{PTCA})$ and $\mathbb{A}_1(\mathrm{PTCA})$, respectively:

- quantifier notation induction in PTCA translates into set induction on W in $\mathbb{A}_0(\mathrm{PTCA})$, using Δ_0 separation, (W.0) and the fact that quantifier free formulas of $\mathcal{L}_p$ are Δ_0 formulas of $\mathcal{L}_p^*$;

Lower bounds

It is immediately seen that PTCA and PHCA are contained in $\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA), respectively:

- quantifier notation induction in PTCA translates into set induction on W in $\mathbb{A}_0$(PTCA), using Δ_0 separation, (W.0) and the fact that quantifier free formulas of $\mathcal{L}_p$ are Δ_0 formulas of $\mathcal{L}_p^*$;
- in order to treat induction for Σ^b_∞ formulas in the case of PHCA, observe that

Lower bounds

It is immediately seen that PTCA and PHCA are contained in $\mathbb{A}_0$(PTCA) and $\mathbb{A}_1$(PTCA), respectively:

- quantifier notation induction in PTCA translates into set induction on W in $\mathbb{A}_0$(PTCA), using Δ_0 separation, (W.0) and the fact that quantifier free formulas of $\mathcal{L}_p$ are Δ_0 formulas of $\mathcal{L}_p^*$;
- in order to treat induction for Σ^b_∞ formulas in the case of PHCA, observe that
  - each term s of $\mathcal{L}_p$ is majorized by a *monotone* term t of $\mathcal{L}$ and thus

Lower bounds

It is immediately seen that PTCA and PHCA are contained in $\mathbb{A}_0(\mathrm{PTCA})$ and $\mathbb{A}_1(\mathrm{PTCA})$, respectively:

- quantifier notation induction in PTCA translates into set induction on W in $\mathbb{A}_0(\mathrm{PTCA})$, using Δ_0 separation, (W.0) and the fact that quantifier free formulas of $\mathcal{L}_p$ are Δ_0 formulas of $\mathcal{L}_p^*$;
- in order to treat induction for Σ^b_∞ formulas in the case of PHCA, observe that
  - each term s of $\mathcal{L}_p$ is majorized by a *monotone* term t of $\mathcal{L}$ and thus
  - each Σ^b_∞ formula $A[\vec{x}]$ can be written in the form

$$(\mathcal{Q}_1 y_1 \leq t_1[\vec{x}])(\mathcal{Q}_2 y_2 \leq t_2[\vec{x}]) \dots (\mathcal{Q}_n y_n \leq t_n[\vec{x}]) B[\vec{x}, y_1, y_2, \dots, y_n]$$

  where $\mathcal{Q}_i \in \{\exists, \forall\}$ and B quantifier-free. Hence, we can define A by a Δ_0 formula in $\mathcal{L}_p^*$ by using (W.1) in order to define the sets

$$a_i \ := \ \{z \in \mathrm{W} : z \leq t_i[\vec{x}]\} \quad (1 \leq i \leq n)$$

1. Introduction
2. The case of KPu^r
3. Polynomial time computable arithmetic and extensions
4. Two admissible closures of PTCA
5. Main results

The strength of $\mathbb{A}_0(\mathrm{PTCA})$

Theorem (Strength of $\mathbb{A}_0(\mathrm{PTCA})$)

$\mathbb{A}_0(\mathrm{PTCA})$ *is a conservative extension of* PTCA *for* $\forall\exists\Sigma_1^b$ *sentences of* $\mathcal{L}_p$*.*

The strength of $\mathbb{A}_0(\mathsf{PTCA})$

Theorem (Strength of $\mathbb{A}_0(\mathsf{PTCA})$)

$\mathbb{A}_0(\mathsf{PTCA})$ *is a conservative extension of* PTCA *for* $\forall\exists\Sigma_1^b$ *sentences of* $\mathcal{L}_p$*.*

Corollary

The Σ_1^b *definable functions of* $\mathbb{A}_0(\mathsf{PTCA})$ *are exactly the polytime functions.*

The strength of $\mathbb{A}_1(\mathsf{PTCA})$

Theorem (Strength of $\mathbb{A}_1(\mathsf{PTCA})$)

$\mathbb{A}_1(\mathsf{PTCA})$ *is a conservative extension of* PHCA *for* $\forall\exists\Sigma^b_\infty$ *sentences of* $\mathcal{L}_p$.

The strength of $\mathbb{A}_1(\mathsf{PTCA})$

Theorem (Strength of $\mathbb{A}_1(\mathsf{PTCA})$)

$\mathbb{A}_1(\mathsf{PTCA})$ *is a conservative extension of* PHCA *for* $\forall\exists\Sigma^b_\infty$ *sentences of* $\mathcal{L}_p$.

Corollary

The Σ^b_∞ *definable functions of* $\mathbb{A}_1(\mathsf{PTCA})$ *are exactly the functions in the polynomial time hierarchy.*