CTFM, Feb 19, 2013, Tokyo Institute of Technology

Inductive Definitions in Bounded Arithmetic: A New Way to Approach P vs. PSPACE

Naohi Eguchi

Mathematical Institute, Tohoku University, Japan

Introduction 1/2

- Purpose in computational complexity:
  Find limits of realistic computations.
- Theoretically: Comparing different notions about computational complexity, e.g. P $\neq$? NP

Introduction 1/2

- Purpose in computational complexity:
  Find limits of realistic computations.
- Theoretically: Comparing different notions about computational complexity, e.g. P $\neq$? NP
- Difficult: to compare complexity classes directly.
  $\Longrightarrow$ Machine-independent logical approaches.
- This talk: new Bounded Arithmetic characterisations of P and PSPACE.
  (P $\subseteq$ NP $\subseteq$ PSPACE, P $\neq$? PSPACE)

Introduction 2/2

In finite model theory (N. Immermann et al.)

1. P is captured by monotone inductive definitions.
2. PSPACE is captured by non-monotone inductive definitions.

Introduction 2/2

In finite model theory (N. Immermann et al.)

1. P is captured by monotone inductive definitions.
2. PSPACE is captured by non-monotone inductive definitions.

Can 1 or 2 be formalised in bounded arithmetic?

- to understand what is the most essential principle in P- or PSPACE-computations.
- to find new aspects of the relationship between P and PSPACE.

Inductive definition (monotone case)

Example of inductive definition: $\mathbb{N}$ is the smallest set containing $\mathbf{0}$ closed under $x \mapsto x + \mathbf{1}$.

Inductive definition (monotone case)

Example of inductive definition: $\mathbb{N}$ is the smallest set containing $\mathbf{0}$ closed under $x \mapsto x + 1$.

More precisely: Define an operator $F : V \to V$ by

$x \in F(X) :\Leftrightarrow x = 0 \vee \exists y \in X(x = y + 1)$.

Inductive definition (monotone case)

Example of inductive definition: $\mathbb{N}$ is the smallest set containing $\mathbf{0}$ closed under $\boldsymbol{x} \mapsto \boldsymbol{x} + \mathbf{1}$.

More precisely: Define an operator $\boldsymbol{F} : \boldsymbol{V} \to \boldsymbol{V}$ by

$\boldsymbol{x} \in \boldsymbol{F}(\boldsymbol{X}) :\Leftrightarrow \boldsymbol{x} = \mathbf{0} \vee \exists \boldsymbol{y} \in \boldsymbol{X}(\boldsymbol{x} = \boldsymbol{y} + \mathbf{1})$.

See:

- $\mathbb{N}$ is the least fixed point of $\boldsymbol{F}$:
  $\boldsymbol{F}(\mathbb{N}) \subseteq \mathbb{N}$, $\forall \boldsymbol{X} \subseteq \boldsymbol{V}[\boldsymbol{F}(\boldsymbol{X}) \subseteq \boldsymbol{X} \to \mathbb{N} \subseteq \boldsymbol{X}]$
- The least fixed point exists since $\boldsymbol{F}$ is monotone:
  $\boldsymbol{X} \subseteq \boldsymbol{Y} \Rightarrow \boldsymbol{F}(\boldsymbol{X}) \subseteq \boldsymbol{F}(\boldsymbol{Y})$.

Inductive definition (general case)

$F : V \to V$;

$x \in F(X) :\Leftrightarrow x = 0 \vee \exists y \in X(x = y + 1)$.

$$\begin{cases} F^0 & := & \emptyset \\ F^{\alpha+1} & := & F(F^\alpha) \\ F^\gamma & := & \bigcup_{\alpha<\gamma} F^\alpha \quad (\gamma : \text{limit}) \end{cases}$$

Inductive definition (general case)

$F : V \to V$;

$x \in F(X) :\Leftrightarrow x = 0 \vee \exists y \in X(x = y + 1)$.

$$\begin{cases} F^0 & := & \emptyset \\ F^{\alpha+1} & := & F(F^\alpha) \\ F^\gamma & := & \bigcup_{\alpha<\gamma} F^\alpha \quad (\gamma : \text{limit}) \end{cases}$$

See:

- $\exists \alpha_0 < \#\mathcal{P}(V)$ such that $F^{\alpha_0+1} = F(F^{\alpha_0}) = F^{\alpha_0}$.
- $\mathbb{N} = F_{\alpha_0}$.

Inductive definition (finite case)

$F : S \to S\ (\#S < \omega)$

- There does not always exist $m < \omega$ such that $F^{m+1} = F(F^m) = F^m$.
- However $\exists k \leq 2^{\#S}$, $\exists l > 0$ such that $\forall n \geq l,\ F^{k+n} = F^n$.

Inductive definition (finite case)

$F : S \to S$ ($\#S < \omega$)

- There does not always exist $m < \omega$ such that $F^{m+1} = F(F^m) = F^m$.
- However $\exists k \leq 2^{\#S}$, $\exists l > 0$ such that $\forall n \geq l,\ F^{k+n} = F^n$.

Inductive definition (finite case)

$F : S \to S$ ($\#S < \omega$)

- There does not always exist $m < \omega$ such that $F^{m+1} = F(F^m) = F^m$.
- However $\exists k \leq 2^{\#S}$, $\exists l > 0$ such that $\forall n \geq l,\ F^{k+n} = F^n$.

Note:

- Choice of k and l is not unique.
- But F^n plays a role similar to the least fixed point like in infinite case.

Connection to time-complexity

Suppose:

1. A function $f(x)$ is computable in $T(x)$ steps.
2. TAPE^l denotes the tape description at the lth step in computing $f(x)$;

   $\text{TAPE}^0 =$

B	i_1	$\cdots$	$i_{\lvert x\rvert}$	B	$\cdots$	B

   ($x = i_1 \cdots i_{|x|}$ (input), $i_1, \ldots, i_{|x|} \in \{0, 1\}$)

Connection to time-complexity

Suppose:

1. A function $f(x)$ is computable in $T(x)$ steps.
2. TAPE^l denotes the tape description at the lth step in computing $f(x)$;

   TAPE^0 =

B	i_1	$\cdots$	$i_{\lvert x\rvert}$	B	$\cdots$	B

   ($x = i_1 \cdots i_{|x|}$ (input), $i_1, \ldots, i_{|x|} \in \{0, 1\}$)

Connection to time-complexity

Suppose:

1. A function $f(x)$ is computable in $T(x)$ steps.
2. TAPE^{l} denotes the tape description at the lth step in computing $f(x)$;

   TAPE^{0} = $\boxed{B}\boxed{i_1}\boxed{\cdots}\boxed{i_{|x|}}\boxed{B}\boxed{\cdots}\boxed{B}$

   ($x = i_1 \cdots i_{|x|}$ (input), $i_1, \ldots, i_{|x|} \in \{0, 1\}$)

Then

- TAPE$^{T(x)+1}$ = TAPE$^{T(x)}$.
- This gives rise to (finite) inductive definition!

Formalising computations 1/2

$$f \text{ is computable} \quad \Leftrightarrow \quad \underbrace{\exists \text{ program to compute } f}_{\boldsymbol{\Sigma}^0_1\text{-formula}}$$

This gives rise to:

Formalising computations 1/2

$$\boldsymbol{f} \text{ is computable} \quad \Leftrightarrow \quad \underbrace{\exists \text{ program to compute } \boldsymbol{f}}_{\boldsymbol{\Sigma^0_1}\text{-formula}}$$

This gives rise to:

Def Let $\boldsymbol{\Phi}$: a set of formulas $\subseteq \boldsymbol{\Sigma^0_1}$ & $\boldsymbol{f}$: a function.
$\boldsymbol{f}$ is $\boldsymbol{\Phi}$-definable in $\boldsymbol{T}$ if ${}^{\exists}\boldsymbol{A}(\vec{\boldsymbol{x}}, \boldsymbol{y}) \in \boldsymbol{\Phi}$ such that

1. All free variables in $\boldsymbol{A}(\vec{\boldsymbol{x}}, \boldsymbol{y})$ are indicated.
2. $\boldsymbol{n} = \boldsymbol{f}(\vec{\boldsymbol{m}}) \Leftrightarrow \mathbb{N} \models \boldsymbol{A}(\underline{\vec{\boldsymbol{m}}}, \underline{\boldsymbol{n}})$ for ${}^{\forall}\vec{\boldsymbol{m}}, \boldsymbol{n} \in \mathbb{N}$.
3. $\boldsymbol{T} \vdash \forall \vec{\boldsymbol{x}} \exists! \boldsymbol{y} \boldsymbol{A}(\vec{\boldsymbol{x}}, \boldsymbol{y})$.

Formalising computations 2/2

Classical facts:

1. $\boldsymbol{f}$: primitive recursive $\Leftrightarrow$ $\boldsymbol{f}$: $\boldsymbol{\Sigma^0_1}$-definable in $\mathbf{I\Sigma_1}$.

(Parsons '70, Mints '73, Buss '86 and Takeuti '87)

Formalising computations 2/2

Classical facts:

1. f: primitive recursive $\Leftrightarrow$ f: $\mathbf{\Sigma^0_1}$-definable in $\mathbf{I\Sigma_1}$.

   (Parsons '70, Mints '73, Buss '86 and Takeuti '87)

2. $f \in$ FP $\Leftrightarrow$ f: $\mathbf{\Sigma^b_1}$-definable in $\mathbf{S^1_2}$. (Buss '86)
   - The start of bounded-arithmetic characterisations of complexity classes.

Note: By Gödel's incompleteness theorem, not all the computable functions are definable in any reasonable system.

Inductive definitions in 2nd order arithmetic

- Inductive definition can be axiomatised in 2nd order arithmetic in the most natural way.

  Fact

  1. $\mathbf{\Pi^1_0}$-MID$_\mathbf{0}$ $= \mathbf{\Pi^1_1}$-CA$_\mathbf{0}$.

     (MID: Monotone Inductive definition)
  2. $\mathbf{\Pi^1_0}$-MID$_\mathbf{0}$ $= \mathbf{\Pi^0_1}$-MID$_\mathbf{0}$ $\subsetneq \mathbf{\Pi^0_2}$-ID$_\mathbf{0}$ $\subsetneq$ $\mathbf{\Pi^0_3}$-ID$_\mathbf{0}$ $\subsetneq \cdots$.

Inductive definitions in 2nd order arithmetic

- Inductive definition can be axiomatised in 2nd order arithmetic in the most natural way.

  Fact

  1. $\mathbf{\Pi^1_0}$-MID$_\mathbf{0}$ $= \mathbf{\Pi^1_1}$-CA$_\mathbf{0}$.

     (MID: Monotone Inductive definition)

  2. $\mathbf{\Pi^1_0}$-MID$_\mathbf{0}$ $= \mathbf{\Pi^0_1}$-MID$_\mathbf{0}$ $\subsetneq \mathbf{\Pi^0_2}$-ID$_\mathbf{0}$ $\subsetneq$ $\mathbf{\Pi^0_3}$-ID$_\mathbf{0}$ $\subsetneq \cdots$.

- Finitary inductive definition can be axiomatised in 2nd order bounded arithmetic.

Foundations of 2nd order bounded arithmetic 1/3

Languages of 2nd order bounded arithmetic:

1. $\mathbf{0}$, $\boldsymbol{S}$, $+$ and $\cdot$.
2. $\lfloor \frac{\boldsymbol{x}}{\mathbf{2}} \rfloor$, $|\boldsymbol{x}| = \lceil \mathbf{log_2}(\boldsymbol{x} + \mathbf{1}) \rceil$ and $|\boldsymbol{X}|$.

Importantly $\boldsymbol{x} \# \boldsymbol{y} = \mathbf{2}^{|\boldsymbol{x}| \cdot |\boldsymbol{y}|}$ is not included.

Foundations of 2nd order bounded arithmetic 1/3

Languages of 2nd order bounded arithmetic:

1. $\mathbf{0}$, $\boldsymbol{S}$, $+$ and $\cdot$.
2. $\lfloor \frac{\boldsymbol{x}}{\mathbf{2}} \rfloor$, $|\boldsymbol{x}| = \lceil \log_2(\boldsymbol{x} + \mathbf{1}) \rceil$ and $|\boldsymbol{X}|$.

Importantly $\boldsymbol{x} \# \boldsymbol{y} = \mathbf{2}^{|\boldsymbol{x}| \cdot |\boldsymbol{y}|}$ is not included.

Intuition:

1. $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \cdots \in {}^{<\mathbb{N}}\{\mathbf{0}, \mathbf{1}\}$.
2. $|\boldsymbol{X}| = \boldsymbol{l}$ if $\boldsymbol{X} \equiv \boldsymbol{i}_0 \boldsymbol{i}_1 \cdots \boldsymbol{i}_{\boldsymbol{l}-1}$ & $\boldsymbol{i}_{\boldsymbol{j}} \in \{\mathbf{0}, \mathbf{1}\}$.
3. $\boldsymbol{j} \in \boldsymbol{X} \Leftrightarrow \boldsymbol{i}_{\boldsymbol{j}} = \mathbf{1}$ if $\boldsymbol{X} \equiv \boldsymbol{i}_0 \boldsymbol{i}_1 \cdots \boldsymbol{i}_{\boldsymbol{l}-1}$.

Foundations of 2nd order bounded arithmetic 2/3

Def ($\mathbf{\Sigma}_1^{\mathrm{B}}$-formulas)

1. $\mathbf{\Sigma}_0^{\mathrm{B}} = \mathbf{\Pi}_0^{\mathrm{B}}$: the set of formulas containing only bounded number quantifiers $\exists x \leq t$.
2. $\exists \vec{X}(|\vec{X}| \leq \vec{t} \wedge \varphi(\vec{X})) \in \mathbf{\Sigma}_{n+1}^{\mathrm{B}}$ if $\varphi \in \mathbf{\Pi}_n^{\mathrm{B}}$.

Foundations of 2nd order bounded arithmetic 2/3

Def ($\boldsymbol{\Sigma}_{\mathbf{1}}^{\mathrm{B}}$-formulas)

1. $\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathrm{B}} = \boldsymbol{\Pi}_{\mathbf{0}}^{\mathrm{B}}$: the set of formulas containing only bounded number quantifiers $\exists \boldsymbol{x} \leq \boldsymbol{t}$.
2. $\exists \vec{\boldsymbol{X}}(|\vec{\boldsymbol{X}}| \leq \vec{\boldsymbol{t}} \wedge \boldsymbol{\varphi}(\vec{\boldsymbol{X}})) \in \boldsymbol{\Sigma}_{\boldsymbol{n}+\mathbf{1}}^{\mathrm{B}}$ if $\boldsymbol{\varphi} \in \boldsymbol{\Pi}_{\boldsymbol{n}}^{\mathrm{B}}$.

Def (Bit-comprehension axiom)

$\forall \boldsymbol{x} \exists \boldsymbol{X}^{\leq \boldsymbol{x}}$ s.t. $\forall \boldsymbol{j} < \boldsymbol{x}(\boldsymbol{j} \in \boldsymbol{X} \leftrightarrow \boldsymbol{\varphi}(\boldsymbol{j}))$

($\exists \boldsymbol{X}^{\leq \boldsymbol{x}} \cdots$ denotes $\exists \boldsymbol{X}(|\boldsymbol{X}| \leq \boldsymbol{x} \wedge \cdots))$

Note: $\bigcup_{\boldsymbol{n} \in \mathbb{N}} \boldsymbol{\Sigma}_{\boldsymbol{n}}^{\mathrm{B}} \subseteq \boldsymbol{\Delta}_{\mathbf{1}}^{\mathbf{0}}(\mathbf{exp}) \subseteq \boldsymbol{\Sigma}_{\mathbf{1}}^{\mathbf{0}}$ by definition.

Foundations of 2nd order bounded arithmetic 3/3

	2nd order arith.	2nd order BA
1st order objects	elements of $\mathbb{N}$	$\leq p(\lvert x \rvert)$
2nd order objects	$f : \mathbb{N} \to \mathbb{N}$	$f : p(\lvert x \rvert) \to \{0, 1\}$
typical classes of formulas	Σ^1_n	Σ^B_n

(p: polynomial)

Foundations of 2nd order bounded arithmetic 3/3

	2nd order arith.	2nd order BA
1st order objects	elements of $\mathbb{N}$	$\leq p(\lvert x\rvert)$
2nd order objects	$f : \mathbb{N} \to \mathbb{N}$	$f : p(\lvert x\rvert) \to \{0, 1\}$
typical classes of formulas	Σ^1_n	Σ^B_n

(p: polynomial)

Def $\mathbf{V}^n :=$ BASIC $+ \Sigma^B_n$-COMP.

Σ^B_n-COMP: BCA with φ restricted to Σ^B_n.

Thm (Zambella '96)

$f \in \mathrm{FP}^{\Sigma^{\mathrm{P}}_n} \Leftrightarrow f$: Σ^B_{n+1}-definable in $\mathbf{V}^{n+1}$.

Formalising inductive definitions

Def $\forall x, \exists X^{\leq x}, \exists Y^{\leq x}$ s.t. $Y \neq \emptyset$ and

1. $\forall j < x(P_\varphi^\emptyset(j) \leftrightarrow j = 0)$ (i.e. $P_\varphi^\emptyset = \emptyset$)
2. $\forall Z \forall j < |Z|(P_\varphi^{S(Z)}(j) \leftrightarrow \varphi(j, P_\varphi^Z) \wedge j < x)$
3. $\forall j < x(P_\varphi^{X+Y}(j) \leftrightarrow P_\varphi^Y(j))$

(P_φ^X: fresh predicate, S: binary successor $X \mapsto X + 1$)

Recall:

1. $F^0 = \emptyset$
2. $F^{m+1} = F(F^m)$
3. $\exists k \leq 2^{\#S}, \exists l \neq 0$ s.t. $F^{k+l} = F^l$

Formalising inductive definitions

Def $\forall x, \exists X^{\leq x}, \exists Y^{\leq x}$ s.t. $Y \neq \emptyset$ and

1. $\forall j < x(P_\varphi^\emptyset(j) \leftrightarrow j = 0)$ (i.e. $P_\varphi^\emptyset = \emptyset$)
2. $\forall Z \forall j < |Z|(P_\varphi^{S(Z)}(j) \leftrightarrow \varphi(j, P_\varphi^Z) \wedge j < x)$
3. $\forall j < x(P_\varphi^{X+Y}(j) \leftrightarrow P_\varphi^Y(j))$

(P_φ^X: fresh predicate, S: binary successor $X \mapsto X + 1$)

Recall:

1. $F^0 = \emptyset$
2. $F^{m+1} = F(F^m)$
3. $\exists k \leq 2^{\#S}, \exists l \neq 0$ s.t. $F^{k+l} = F^l$

Formalising inductive definitions

Def $\forall x, \exists \boldsymbol{X}^{\leq x}, \exists \mathbf{Y}^{\leq x}$ s.t. $Y \neq \emptyset$ and

1. $\forall j < x(P_\varphi^{\emptyset}(j) \leftrightarrow j = 0)$ (i.e. $P_\varphi^{\emptyset} = \emptyset$)
2. $\forall Z \forall j < |Z|(P_\varphi^{S(Z)}(j) \leftrightarrow \varphi(j, P_\varphi^{Z}) \wedge j < x)$
3. $\forall j < x(P_\varphi^{\mathbf{X}+\mathbf{Y}}(j) \leftrightarrow P_\varphi^{\mathbf{Y}}(j))$

(P_φ^{X}: fresh predicate, S: binary successor $X \mapsto X + 1$)

Recall:

1. $F^0 = \emptyset$
2. $F^{m+1} = F(F^m)$
3. $\exists k \leq 2^{\#S}$, $\exists l \neq 0$ s.t. $F^{k+l} = F^l$

Capturing P and PSPACE

Def $\mathbf{\Sigma_0^B}$-IDEF:

Axiom of inductive definition for $\varphi \in \mathbf{\Sigma_0^B}$.

Thm 1

Every $\boldsymbol{f} \in$ FP is $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Thm 2

Every $\boldsymbol{f} \in$ FPSPACE is $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Connection to time-complexity

Suppose:

1. A function $f(x)$ is computable in $T(x)$ steps.
2. TAPE^{l} denotes the tape description at the lth step in computing $f(x)$;

   TAPE^{0} = $\boxed{B}\boxed{i_1}\boxed{\cdots}\boxed{i_{|x|}}\boxed{B}\boxed{\cdots}\boxed{B}$

   $(x = i_1 \cdots i_{|x|}$ (input), $i_1, \ldots, i_{|x|} \in \{0, 1\})$

Then

- TAPE$^{T(x)+1}$ = TAPE$^{T(x)}$.
- This gives rise to (finite) inductive definition!

Proof of Theorem 2

Suppose: $f \in$ FPSPACE.

$\exists p$: poly $\begin{cases} f(x) \text{ is computable in } 2^{p(|x|)} \text{steps} \\ |\text{TAPE}^X| \leq p(|x|) \end{cases}$

See: $\text{TAPE}^X \mapsto \text{TAPE}^{X+1}$: Σ_0^B.

By (Σ_0^B-IDEF) $\exists K, \exists L$ s.t. $\text{TAPE}^{K+L} = \text{TAPE}^L$.

See: TAPE^L must be in the accepting state.

So $f(x) = y \Leftrightarrow \exists X^{\leq p(|x|)}, \exists Y^{\leq p(|x|)}$

$\text{TAPE}^{X+Y} = \text{TAPE}^Y \wedge y = \text{output}(\text{TAPE}^Y)$

Hence f is Σ_1^B-definable in $\mathbf{V}^0 + \Sigma_0^B$-IDEF.

Proof of Theorem 2

Suppose: $f \in$ FPSPACE.

$\exists \boldsymbol{p}$: poly $\begin{cases} f(x) \text{ is computable in } 2^{p(|x|)} \text{steps} \\ |\text{TAPE}^{X}| \leq p(|x|) \end{cases}$

See: $\text{TAPE}^{X} \mapsto \text{TAPE}^{X+1}$: Σ_0^B.

By (Σ_0^B-IDEF) $\exists K, \exists L$ s.t. $\text{TAPE}^{K+L} = \text{TAPE}^{L}$.

See: TAPE^{L} must be in the accepting state.

So $f(x) = y \Leftrightarrow \exists X^{\leq p(|x|)}, \exists Y^{\leq p(|x|)}$

$\text{TAPE}^{X+Y} = \text{TAPE}^{Y} \wedge y = \text{output}(\text{TAPE}^{Y})$

Hence f is Σ_1^B-definable in $\mathbf{V}^0 + \Sigma_0^B$-IDEF.

Proof of Theorem 2

Suppose: $f \in$ FPSPACE.

$\exists p$: poly $\begin{cases} f(x) \text{ is computable in } 2^{p(|x|)} \text{steps} \\ |\mathrm{TAPE}^X| \leq p(|x|) \end{cases}$

See: $\mathrm{TAPE}^X \mapsto \mathrm{TAPE}^{X+1}$: Σ_0^B.

By (Σ_0^B-IDEF) $\exists K, \exists L$ s.t. $\mathrm{TAPE}^{K+L} = \mathrm{TAPE}^L$.

See: TAPE^L must be in the accepting state.

So $f(x) = y \Leftrightarrow \exists X^{\leq p(|x|)}, \exists Y^{\leq p(|x|)}$

$\mathrm{TAPE}^{X+Y} = \mathrm{TAPE}^Y \wedge y = \mathrm{output}(\mathrm{TAPE}^Y)$

Hence f is Σ_1^B-definable in $\mathbf{V}^0 + \Sigma_0^B$-IDEF.

Proof of Theorem 2

Suppose: $f \in$ FPSPACE.

$\exists p$: poly $\begin{cases} f(x) \text{ is computable in } 2^{p(|x|)} \text{steps} \\ |\text{TAPE}^{X}| \leq p(|x|) \end{cases}$

See: $\text{TAPE}^{X} \mapsto \text{TAPE}^{X+1}$: Σ_0^{B}.

By (Σ_0^{B}-IDEF) $\exists K, \exists L$ s.t. $\text{TAPE}^{K+L} = \text{TAPE}^{L}$.

See: TAPE^{L} must be in the accepting state.

So $f(x) = y \Leftrightarrow \exists X^{\leq p(|x|)}, \exists Y^{\leq p(|x|)}$

$$\text{TAPE}^{X+Y} = \text{TAPE}^{Y} \wedge y = \text{output}(\text{TAPE}^{Y})$$

Hence f is Σ_1^{B}-definable in $\mathbf{V}^0 + \Sigma_0^{\mathrm{B}}$-IDEF.

Inflationary inductive definition

Can Theorem 1 be sharpen?:

Thm 1 Every $\boldsymbol{f} \in$ FP is $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Inflationary inductive definition

Can Theorem 1 be sharpen?:

Thm 1 Every $f \in$ FP is $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Def An operator F is inflationary if $X \subseteq F(X)$.

Note: Inflationary inductive definition can be reduced monotone one over FOL. (Gurevich-Shelah '86)

Inflationary inductive definition

Can Theorem 1 be sharpen?:

Thm 1 Every $f \in \mathrm{FP}$ is $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Def An operator F is inflationary if $X \subseteq F(X)$.

Note: Inflationary inductive definition can be reduced monotone one over FOL. (Gurevich-Shelah '86)

We can define:

Def $\mathbf{\Sigma_0^B}$-IIDEF: a restriction of $\mathbf{\Sigma_0^B}$-IDEF to inflationary inductive definition.

Results

Thm 1 (sharpened) $f \in$ FP if and only if $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IIDEF.

($\Longleftarrow$) Reduce $\mathbf{\Sigma_0^B}$-IIDEF to $\mathbf{V^0} + \mathbf{\Sigma_1^B}$-IND $= \mathbf{V^1}$.

Recall:

Thm (Zambella '96)

$f \in$ FP $\Leftrightarrow$ f: $\mathbf{\Sigma_1^B}$-definable in $\mathbf{V^1}$.

Conjecture

Conjecture $\mathbf{\Sigma}_{\mathbf{0}}^{\mathrm{B}}$-IDEF can be reduced to $\mathbf{W}_{\mathbf{1}}^{\mathbf{1}}$.
($\mathbf{W}_{\mathbf{1}}^{\mathbf{1}}$: 3rd order extension of $\mathbf{V}^{\mathbf{1}}$)

Thm (Skelley '06)
$\boldsymbol{f} \in$ FPSPACE $\Leftrightarrow$ $\boldsymbol{f}$ is $\mathbf{\Sigma}_{\mathbf{1}}^{\boldsymbol{\mathcal{B}}}$-definable in $\mathbf{W}_{\mathbf{1}}^{\mathbf{1}}$.
($\mathbf{\Sigma}_{\mathbf{1}}^{\boldsymbol{\mathcal{B}}}$: 3rd order extension of $\mathbf{\Sigma}_{\mathbf{1}}^{\mathrm{B}}$)

Conjecture

Conjecture $\mathbf{\Sigma_0^B}$-IDEF can be reduced to $\mathbf{W_1^1}$.
($\mathbf{W_1^1}$: 3rd order extension of $\mathbf{V^1}$)

Thm (Skelley '06)
$f \in$ FPSPACE $\Leftrightarrow f$ is $\mathbf{\Sigma_1^{\mathcal{B}}}$-definable in $\mathbf{W_1^1}$.
($\mathbf{\Sigma_1^{\mathcal{B}}}$: 3rd order extension of $\mathbf{\Sigma_1^B}$)

Corollary of Conjecture
$f \in$ FPSPACE $\Leftrightarrow f$ is $\mathbf{\Sigma_1^B}$-definable in
$\mathbf{V^0} + \mathbf{\Sigma_0^B}$-IDEF.

Conclusion

- Finite model-theoretic characterisations of P and PSPACE can be reformulated by inductive definitions in bounded arithmetic.
- P vs. PSPACE can be reduced to inflationary vs. non inflationary inductive definitions.
- PSPACE can be discussed about without using 3rd order notions.
  - $\mathbf{V}^1$ (2nd order) corresponds to P.
  - $\mathbf{W}^1_1$ (3rd order) corresponds to PSPACE.

Thank you for your attention!

Speaker is generously supported by the John Templeton Foundation.