An Introduction to Proof Complexity, Part I.

Pavel Pudlák
Mathematical Institute, Academy of Sciences, Prague
and
Charles University, Prague

Computability in Europe 2009, Heidelberg

computational complexity	*proof complexity*
complexity classes	theories
elementary operations	basic axioms
computations with bounded resources	induction restricted to some formulas
the **P** vs. **NP** problem	the $\Theta_{\mathbf{P}}$ vs. $\Theta_{\mathbf{NP}}$ problem
circuits	propositional proofs

Overview

Part I.

- A lower bound for the propositional Pigeon-Hole Principle.
- Theories and complexity classes.
- Conditional and relativized separation of theories.

Part II.

- Propositional proof systems.
- Feasible interpolation.
- Unprovability of circuit lower bounds.
- Total search problems. (not presented at CiE)
- Feasible incompleteness. (not presented at CiE)

Literature:

- J. Krajíček: Bounded Arithmetic, Propositional Logic and Complexity Theory, 1995
- P. Clote and E. Kranakis: Boolean Functions and Computational Models, Chapter 5, 2002
- S. Cook and P. Nguyen: Foundations of Proof Complexity, to appear, 2009?
- P.P.: Logical Foundations of Mathematics and Complexity Theory, Chapter 6, (manuscript)
- J. Krajíček, lecture notes available on his web-page

A lower bound for propositional Pigeon-Hole Principle

Paradigm: Exponential lower bound on the size of bounded depth circuits computing the parity function.

Theorem (Furst-Saxe-Sipser,Ajtai,Yao,Hastad)

Every bounded depth circuit computing parity has exponential size, i.e., $\forall d \exists \epsilon_d \forall n$ *every boolean circuit of depth* d *computing the parity function of* n *bits has size* $\geq 2^{n^{\epsilon_d}}$.

$\epsilon_d = 1/10d$

Switching Lemma:

1. random restrictions, $\bigwedge \bigvee \mapsto \bigvee \bigwedge$, reduction in the depth of the circuit
2. the parity function is reduced to a parity function on the remaining variables

3.If the circuit is small, we get eventually a k-CNF or k-DNF with $k < n$ which cannot compute the parity function.

A Frege Proof System

propositional variables $p_1, p_2, \ldots$

any complete finite set of connectives. we shall use connectives $\neg$, $\vee$, and $\wedge$.

Any complete finite set of rules.

Example. [Hilbert and Ackermann]

Connectives $\neg, \vee$.

Axiom schemas

1. $\neg(A \vee A) \vee A$
2. $\neg A \vee (A \vee B)$
3. $\neg(A \vee B) \vee (B \vee A)$
4. $\neg(\neg A \vee B) \vee (\neg(C \vee A) \vee (C \vee B))$

Rule

- *From A and $\neg A \vee B$ derive B.*

Every two Frege proof systems polynomially simulate each other.

A Frege Proof System

propositional variables $p_1, p_2, \ldots$
any complete finite set of connectives. we shall use connectives $\neg$, $\vee$, and $\wedge$.
Any complete finite set of rules.

Example. [Hilbert and Ackermann]
Connectives $\neg, \vee$.
Axiom schemas

1. $\neg(A \vee A) \vee A$
2. $\neg A \vee (A \vee B)$
3. $\neg(A \vee B) \vee (B \vee A)$
4. $\neg(\neg A \vee B) \vee (\neg(C \vee A) \vee (C \vee B))$

Rule

- *From A and $\neg A \vee B$ derive B.*

Every two Frege proof systems polynomially simulate each other.

depth of a formula = number of alternations of $\neg$, $\vee$, and $\wedge$.
we shall *abbreviate* by $\bigvee$ and $\bigwedge$ long disjunctions and conjunctions
depth d proofs: proofs that use only formulas of depth d
bounded depth = constant depth

The tautology PHP_n

The depth 3 tautology expressing the principle that there is no 1-1 mapping $F : [n+1] \to [n]$:

$$\neg \left(\bigwedge_{i=1}^{n+1} \bigvee_{j=1}^{n} p_{ij} \right) \vee \bigvee_{j=1}^{n} \bigvee_{1 \le i < i' \le n+1} (p_{ij} \wedge p_{i'j}).$$

Meaning: p_{ij} is true iff $F(i) = j$.
The clause that F is a function is missing.
A weaker version: bijective PHP.

The tautology PHP_n

The depth 3 tautology expressing the principle that there is no 1-1 mapping $F : [n+1] \to [n]$:

$$\neg \left(\bigwedge_{i=1}^{n+1} \bigvee_{j=1}^{n} p_{ij} \right) \vee \bigvee_{j=1}^{n} \bigvee_{1 \le i < i' \le n+1} (p_{ij} \wedge p_{i'j}).$$

Meaning: p_{ij} is true iff $F(i) = j$.
The clause that F is a function is missing.
A weaker version: bijective PHP.

Test question: Why PHP?

The tautology PHP_n

The depth 3 tautology expressing the principle that there is no 1-1 mapping $F : [n+1] \to [n]$:

$$\neg \left(\bigwedge_{i=1}^{n+1} \bigvee_{j=1}^{n} p_{ij} \right) \vee \bigvee_{j=1}^{n} \bigvee_{1 \le i < i' \le n+1} (p_{ij} \wedge p_{i'j}).$$

Meaning: p_{ij} is true iff $F(i) = j$.
The clause that F is a function is missing.
A weaker version: bijective PHP.

Test question: Why PHP? It is simple, but not trivial.
Parity Principle: There is no partition of $[2n+1]$ into two-element blocks.

Theorem (Ajtai, Krajíček-Pudlák-Woods, Beame-Impagliazzo-Pitassi)

Every bounded depth proof of PHP_n has exponential size, i.e.,
$\forall d \exists \epsilon_d \forall n$ *every depth* d *Frege proof of the bijective* PHP_n *has size* $\geq 2^{n^{\epsilon_d}}$.

$\epsilon_d = 1/6^d$

Basic Intuition: *Truth assignments are 1-1 mappings* $[n+1] \longrightarrow [n]$ *(i.e., matchings).*

Basic Intuition: *Truth assignments are 1-1 mappings* $[n+1] \to [n]$ *(i.e., matchings).*

Let $|D| = n+1$, $|R| = n$, $D \cap R = \emptyset$.
A matching decision tree is a finite tree T with queries:

What is the matching element for i *(or* j*)?*

For a given $i \in D$ or $j \in R$.

Formally:

Definition

A matching decision tree is a finite tree T such that

- all nodes different from leaves are labeled by elements of $D \cup R$;
- leaves are labeled by 0 (false) and 1 (true);
- the edges going out of a node v labeled by $i \in D$ are labeled by all $j \in R$ that do not appear on the path from the root to v;
- the edges going out of a node u labeled by $j \in R$ are labeled by all $i \in D$ that do not appear on the path from the root to u.

Intuition: T defines a boolean function on all matchings $F : D \to R$.

The idea of the lower bound. Given a short proof of bounded depth, transform it into a sequence of shallow decision trees that represent the formulas. Then

1. axioms are represented by trees that accept all assignments,
2. logical rules preserve this property, but
3. PHP_n is represented by the tree that does not accept any assignment—contradiction.

The idea of the lower bound. Given a short proof of bounded depth, transform it into a sequence of shallow decision trees that represent the formulas. Then

1. axioms are represented by trees that accept all assignments,
2. logical rules preserve this property, but
3. PHP_n is represented by the tree that does not accept any assignment—contradiction.

Ajtai's proof (of a weaker lower bound):

finite combinatorial constructions

$\downarrow$

nonstandard model with a 1-1 function $F : [n+1] \longrightarrow n$

$\downarrow$

lower bound on Frege proofs.

The idea of the lower bound. Given a short proof of bounded depth, transform it into a sequence of shallow decision trees that represent the formulas. Then

1. axioms are represented by trees that accept all assignments,
2. logical rules preserve this property, but
3. PHP_n is represented by the tree that does not accept any assignment—contradiction.

A branch b in a matching tree defines a partial matching $\sigma_b \subseteq D \times R$.
A partial matching σ defines a partial truth assignment a_σ as follows

$p_{ij} = 1$, if $(i,j) \in \sigma$
$p_{ij} = 0$, if $(i,j) \notin \sigma$ and $(i \in Dom(\sigma)$ or $j \in Rng(\sigma))$
$p_{ij} = *$ otherwise.

Definition

A matching tree represents a formula ϕ, if for every branch b of T, $\phi|_{\sigma_b} \equiv 0$ or $\equiv 1$, and it equals to the label of the leaf of b.

In order to preserve the rules, trees must be shallow.
Some formulas can only be represented by trees of depth n (the maximal depth); e.g., $\bigoplus_{ij} p_{ij}$.

Therefore, we have to apply random restrictions to make the formulas representable by shallow trees.

[1] Recall: a partial matching σ defines a partial truth assignment a_σ as follows: $p_{ij} = 1$, if $(i,j) \in \sigma$; $p_{ij} = 0$, if $(i,j) \notin \sigma$ and $(i \in Dom(\sigma)$ or $j \in Rng(\sigma))$; $p_{ij} = *$ otherwise.

In order to preserve the rules, trees must be shallow.
Some formulas can only be represented by trees of depth n (the maximal depth); e.g., $\bigoplus_{ij} p_{ij}$.

Therefore, we have to apply random restrictions to make the formulas representable by shallow trees.

Random restriction R_q, $0 < q < 1$:
pick randomly uniformly a partial matching $\rho \subseteq D \times R$, $|\sigma| = \lceil qn \rceil$.

Random restrictions are applied to *formulas*, it is a *syntactical* operation defined inductively:

$p_{ij}|_\rho := p_{ij}$ if $a_\rho = *$, and $p_{ij} := a_\rho$ otherwise;[1]
$1 \wedge \phi := \phi$, $0 \wedge \phi = 0$;
$1 \vee \phi := 1$, $0 \vee \phi = \phi$.

[1] Recall: a partial matching σ defines a partial truth assignment a_σ as follows: $p_{ij} = 1$, if $(i,j) \in \sigma$; $p_{ij} = 0$, if $(i,j) \notin \sigma$ and ($i \in Dom(\sigma)$ or $j \in Rng(\sigma)$); $p_{ij} = *$ otherwise.

In order to preserve the rules, trees must be shallow.
Some formulas can only be represented by trees of depth n (the maximal depth); e.g., $\bigoplus_{ij} p_{ij}$.

Therefore, we have to apply random restrictions to make the formulas representable by shallow trees.

Random restriction R_q, $0 < q < 1$:
pick randomly uniformly a partial matching $\rho \subseteq D \times R$, $|\sigma| = \lceil qn \rceil$.

Random restrictions are applied to *formulas*, it is a *syntactical* operation defined inductively:

$p_{ij}|_\rho := p_{ij}$ if $a_\rho = *$, and $p_{ij} := a_\rho$ otherwise;[1]
$1 \wedge \phi := \phi$, $0 \wedge \phi = 0$;
$1 \vee \phi := 1$, $0 \vee \phi = \phi$.

Test question: Why do we do it syntactically?

[1] Recall: a partial matching σ defines a partial truth assignment a_σ as follows: $p_{ij} = 1$, if $(i,j) \in \sigma$; $p_{ij} = 0$, if $(i,j) \notin \sigma$ and $(i \in Dom(\sigma)$ or $j \in Rng(\sigma))$; $p_{ij} = *$ otherwise.

In order to preserve the rules, trees must be shallow.
Some formulas can only be represented by trees of depth n (the maximal depth); e.g., $\bigoplus_{ij} p_{ij}$.

Therefore, we have to apply random restrictions to make the formulas representable by shallow trees.

Random restriction R_q, $0 < q < 1$:
pick randomly uniformly a partial matching $\rho \subseteq D \times R$, $|\sigma| = \lceil qn \rceil$.

Random restrictions are applied to *formulas*, it is a *syntactical* operation defined inductively:

$p_{ij}|_\rho := p_{ij}$ if $a_\rho = *$, and $p_{ij} := a_\rho$ otherwise;[1]
$1 \wedge \phi := \phi$, $0 \wedge \phi = 0$;
$1 \vee \phi := 1$, $0 \vee \phi = \phi$.

Test question: Why do we do it syntactically? Because semantics is trivial—all formulas in a proof are tautologies!

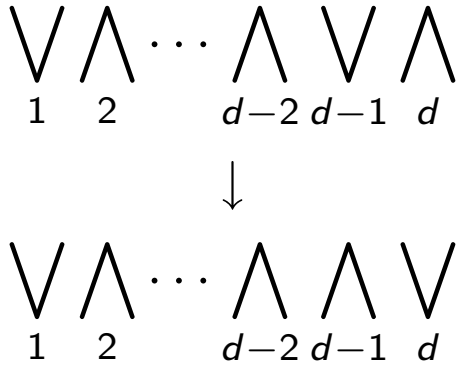
[1] Recall: a partial matching σ defines a partial truth assignment a_σ as follows: $p_{ij} = 1$, if $(i,j) \in \sigma$; $p_{ij} = 0$, if $(i,j) \notin \sigma$ and $(i \in Dom(\sigma)$ or $j \in Rng(\sigma))$; $p_{ij} = *$ otherwise.

k-CNF - a conjunction of disjunctions of size $\leq k$
k-DNF - a disjunction of conjunctions of of size $\leq k$
k-Tree - a tree of depth $\leq k$

a k-Tree can be represented by a k-CNF and a k-DNF

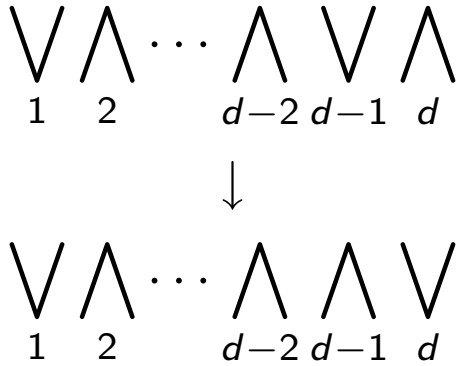
Depth reduction by random restrictions

We want:

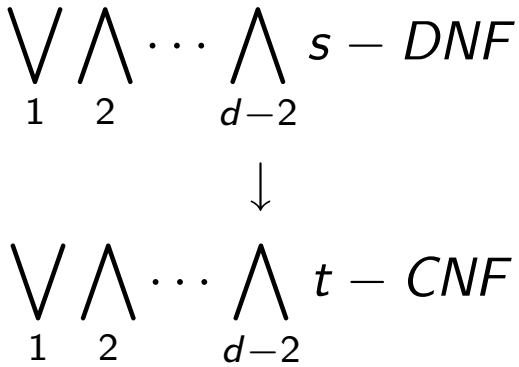


Depth reduction by random restrictions

We want:

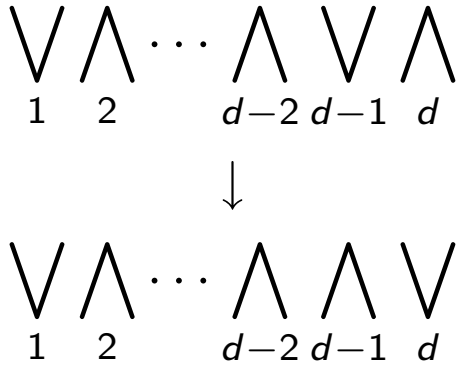


1. original

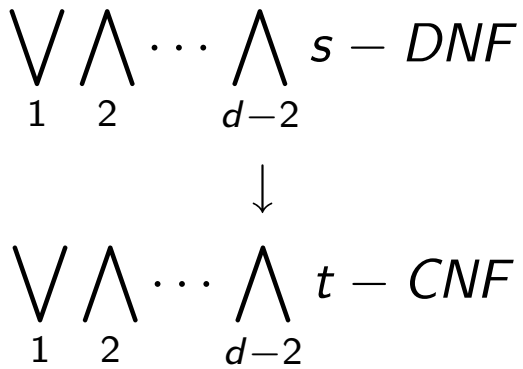


Depth reduction by random restrictions

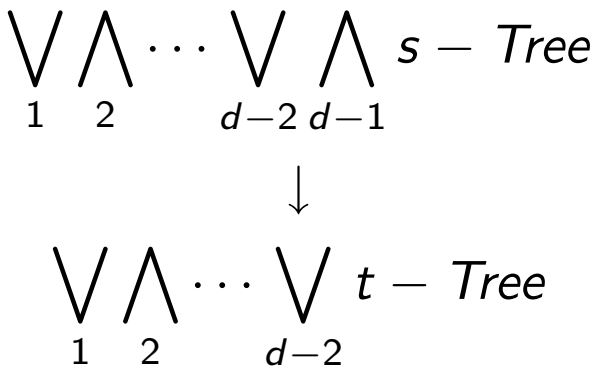
We want:



1. original



2. new way



2. new way in reality:

$$\bigvee_{1}\bigwedge_{2}\cdots\bigwedge_{d-2}\bigvee_{d-1} s-Tree$$

$$\downarrow$$

$$\bigvee_{1}\bigwedge_{2}\cdots\bigwedge_{d-2}\bigvee_{d-1} s-DNF$$

$$\downarrow$$

$$\bigvee_{1}\bigwedge_{2}\cdots\bigwedge_{d-2} s-DNF$$

$$\downarrow \text{(switching lemma)}$$

$$\bigvee_{1}\bigwedge_{2}\cdots\bigwedge_{d-2} t-Tree$$

Lemma (Matching Switching Lemma)

Let $q^4 n^3 \leq 1/10$ and let ϕ be an $s-DNF$. Then for random $\rho \in R_q$, the probability that $\phi|_\rho$ can be represented by a t-Tree is at least

$$1 - (9q^4 n^3 s)^t.$$

[In Håstad's Lemma the formula is $1 - (5qs)^t$.]

The Lemma is applied with
$q = n^{-\alpha}, s = n^{\beta}, t = n^{\gamma}$, $0 < \alpha, \beta, \gamma < 0$
and
$9q^4 n^3 s \leq constant < 1$.
Recall: $\rho \in R_q$ are partial matchings of size $(1-q)n$.

The proof.

1. Håstad's method of conditional probabilities—very difficult to use here.
2. Razborov's counting method.

The proof.

1. Håstad's method of conditional probabilities—very difficult to use here.

2. Razborov's counting method.

Choose a suitable method that transforms an $s - DNF$ into a tree T.
Say that $\rho \in R_q$ is bad, if T obtained from $\phi|_\rho$ has depth $> t$.

$$probability \leq \frac{\#\text{ bad } \rho}{\#\text{ all } \rho}$$

To upper bound the probability one constructs a 1-1 mapping from *bad* ρ*'s* to a set X such that $X \ll \#$ *all* ρ*'s*.

Alternatively: One shows that to determine a *bad* ρ we need substantially fewer bits than to determine a general ρ.

Given a bad ρ extend it to $\rho' := \rho \cup \sigma$, where σ is a partial matching computed by the first branch b of T of length $> t$. Thus

$$|\rho| = (1-q)n \quad \text{and} \quad |\rho'| > (1-q)n + t$$

$\#$ all $\rho = \binom{n+1}{qn}\binom{n}{qn}((1-q)n)!$

ρ' is an element of a set of size $\leq \binom{n+1}{qn-t}\binom{n}{qn-t}((1-q)n+t)!$
Thus if $\rho \mapsto \rho'$ were 1-1

$$\textit{probability} \leq \frac{\#\text{ bad } \rho}{\#\text{ all } \rho} \leq \frac{\binom{n+1}{qn-t}\binom{n}{qn-t}((1-q)n+t)!}{\binom{n+1}{qn}\binom{n}{qn}((1-q)n)!}$$

Notice that

$$\binom{n+1}{qn-t} \ll \binom{n+1}{qn} \quad \text{and} \quad \binom{n}{qn-t} \ll \binom{n}{qn}$$

$\rho \mapsto \rho'$ is not 1-1, but the number of ρs mapped on a ρ' is small.
Formally, we define $\rho \mapsto (\rho', I)$ where I is some information that suffices to recover ρ from ρ'.

...

Nonstandard semantics for matching trees

Lemma

Let $M \models Tr(\mathbb{N})$ be a countable nonstandard model, $a > \mathbb{N}$ a nonstandard element and $t \in M$ such that $a/t > \mathbb{N}$. Then there exist 1-1 and onto mappings $F : [a] \to [a-1]$ such that for every internal set $X \subseteq [a]$ ($Y \subseteq [a-1]$), of size $\leq t$, $F \cap (X \times [a-1])$ ($F \cap ([a] \times Y)$) is also internal.

In fact, for every partial matching U of size $\leq b$ there exists such an F that extends U.

1. we can evaluate internal matching decision trees of depth $\leq t$ on such mappings;
2. T is satisfied by all such mappings iff all leaves of T are labeled by 1.

An application

Theorem

(Ajtai, 1988) Let $M \models Tr(\mathbb{N})$ be a nonstandard model, $a > \mathbb{N}$ a nonstandard element. Let M_a be M restricted to the interval $[0, a]$. Then it is possible to extend M_a to $M_a[F]$ so that

1. *F is a 1-1 mapping from $[a]$ to $[a-1]$ and*
2. *induction holds for all formulas in $M_a[F]$.*

The same holds true for $M_a^* = \bigcup_{n \in \mathbb{N}} M_{a^n}$.
Using exponential lower bounds we can improve it to $M_{2^{n^{o(1)}}}$.

An application

Theorem

(Ajtai, 1988) Let $M \models Tr(\mathbb{N})$ be a nonstandard model, $a > \mathbb{N}$ a nonstandard element. Let M_a be M restricted to the interval $[0, a]$. Then it is possible to extend M_a to $M_a[F]$ so that

1. *F is a 1-1 mapping from $[a]$ to $[a-1]$ and*
2. *induction holds for all formulas in $M_a[F]$.*

The same holds true for $M_a^* = \bigcup_{n \in \mathbb{N}} M_{a^n}$.
Using exponential lower bounds we can improve it to $M_{2^{n^{o(1)}}}$.

Corollary

PHP for F is not provable in $I\Delta_0[F]$.

($I\Delta_0[F]$ is the theory with induction for bounded formulas in the language of arithmetic extended by the function symbol F.) If the circuit is small, we get eventually a k-CNF or k-DNF with $k < n$ which cannot compute the parity function.

Problem (notoriously open)

Does the schema $I\Delta_0$ prove the schema $PHP\Delta_0$?

By Corollary, it does not do it *uniformly.*

Problem (notoriously open)

Does the schema $I\Delta_0$ prove the schema $PHP\Delta_0$?

By Corollary, it does not do it *uniformly.*

Idea of the proof of the Theorem

By contradiction, suppose that PHP for F is provable from the diagram of M_a and induction. Let P be such a proof. W.l.o.g. assume that F occurs only in atomic formulas of the form $F(x) = y$, where x, y are variables.

Arguing in M,

1. replace $\exists x \leq t\ \phi$ and $\forall x \leq t\ \phi$ by $\bigvee_{x \leq t} \phi$ and $\bigwedge_{x \leq t} \phi$,
2. replace each application of the induction axioms by iterated modus ponens,
3. replace atomic formulas not containing F by the truth constants,
4. replace atomic formulas $F(b) = c$ by propositional variables p_{bc}

Thus we obtain bounded depth Frege proof of PHP_{a-1} of polynomial size.
This is a contradiction, because $M \models Tr(\mathbb{N})$.

q.e.d.

Problem (notoriously open)

Does the schema $I\Delta_0$ prove the schema $PHP\Delta_0$?

By Corollary, it does not do it *uniformly.*

Idea of the proof of the Theorem

By contradiction, suppose that PHP for F is provable from the diagram of M_a and induction. Let P be such a proof. W.l.o.g. assume that F occurs only in atomic formulas of the form $F(x) = y$, where x, y are variables.

Arguing in M,

1. replace $\exists x \leq t\ \phi$ and $\forall x \leq t\ \phi$ by $\bigvee_{x \leq t} \phi$ and $\bigwedge_{x \leq t} \phi$,
2. replace each application of the induction axioms by iterated modus ponens,
3. replace atomic formulas not containing F by the truth constants,
4. replace atomic formulas $F(b) = c$ by propositional variables p_{bc}

Thus we obtain bounded depth Frege proof of PHP_{a-1} of polynomial size.
This is a contradiction, because $M \models Tr(\mathbb{N})$.

q.e.d.

Questions?

Theories and complexity classes

$\mathcal{C}$ complexity class $\leftrightarrow$ $\Theta_{\mathcal{C}}$ first order theory

- suitable language L
- class of formulas Γ that characterizes the class $\mathcal{C}$
- basic axioms (fixing the interpretation of primitive notions)
- induction for formulas of Γ

Theories and complexity classes

$\mathcal{C}$ complexity class $\leftrightarrow$ $\Theta_{\mathcal{C}}$ first order theory

- suitable language L
- class of formulas Γ that characterizes the class $\mathcal{C}$
- basic axioms (fixing the interpretation of primitive notions)
- induction for formulas of Γ

Examples.

1. $\mathcal{C} = ARITH$, Θ_{ARITH} is Peano Arithmetic.
Basic axioms - Robinson's Arithmetic

Theories and complexity classes

$\mathcal{C}$ complexity class $\leftrightarrow$ $\Theta_{\mathcal{C}}$ first order theory

- suitable language L
- class of formulas Γ that characterizes the class $\mathcal{C}$
- basic axioms (fixing the interpretation of primitive notions)
- induction for formulas of Γ

Examples.

1. $\mathcal{C} = ARITH$, Θ_{ARITH} is Peano Arithmetic.
Basic axioms - Robinson's Arithmetic

2. $\mathcal{C} = PR$, Θ_{PR} is Primitive Recursive Arithmetic.

Theories and complexity classes

$\mathcal{C}$ complexity class $\leftrightarrow$ $\Theta_{\mathcal{C}}$ first order theory

- suitable language L
- class of formulas Γ that characterizes the class $\mathcal{C}$
- basic axioms (fixing the interpretation of primitive notions)
- induction for formulas of Γ

Examples.

1. $\mathcal{C} = ARITH$, Θ_{ARITH} is Peano Arithmetic.
Basic axioms - Robinson's Arithmetic

2. $\mathcal{C} = PR$, Θ_{PR} is Primitive Recursive Arithmetic.

3. $\mathcal{C} = \Delta_0$, Θ_{Δ_0} is $I\Delta_0$, i.e., Peano Arithmetic with induction restricted to Δ_0 formulas (= formulas with bounded quantifiers $\forall x \leq t$ and $\exists x \leq t$, t a term).

Linear Time Hierarchy $= \Delta_0$, thus also $\Theta_{\text{Linear Time Hierarchy}} \equiv I\Delta_0$

4. PH is the Polynomial (Time) Hierarchy, $\bigcup_k \Sigma_k^p$

$\Theta_{PH} \equiv I\Delta_0[x^{\log x}] \equiv I\Delta_0 + \forall x \exists y (y = x^{\log x})$

4. PH is the Polynomial (Time) Hierarchy, $\bigcup_k \Sigma_k^p$

$\Theta_{PH} \equiv I\Delta_0[x^{\log x}] \equiv I\Delta_0 + \forall x \exists y (y = x^{\log x})$

Buss's Bounded Arithmetic

Language $0, S, +, \times, \lfloor x/2 \rfloor, \log x, 2^{\log x . \log y}, \leq$
Classes of bounded formulas Σ_k^b (defines class of sets Σ_k^p).
Basic axioms BASIC

$T_2^k :=$ BASIC + induction for Σ_k^b

For $k \geq 1$, $\Theta_{\Sigma_k^p} \equiv T_2^k$.

In particular, $\Theta_{\mathbf{NP}} \equiv T_2^1$.

But Σ_0^b does not define all sets in $\mathbf{P}$ and T_2^0 is not $\Theta_{\mathbf{P}}$.

How to define $\Theta_{\mathbf{P}}$?

1. Cook: theory PV with a function symbol for every polynomial time algorithm (using the schema of recursion on notation).

2. Jeřábek: extend Buss's language by $\lfloor x/2^y \rfloor$ and add a few axioms to BASIC. Then

$$T_2^0 \equiv \Theta_{\mathbf{P}}$$

in spite of the fact that (probably) the extended Σ_0^b still does not define all sets in $\mathbf{P}$.

How to define $\Theta_{\mathbf{P}}$?

1. Cook: theory PV with a function symbol for every polynomial time algorithm (using the schema of recursion on notation).

2. Jeřábek: extend Buss's language by $\lfloor x/2^y \rfloor$ and add a few axioms to BASIC. Then

$$T_2^0 \equiv \Theta_{\mathbf{P}}$$

in spite of the fact that (probably) the extended Σ_0^b still does not define all sets in **P**.

$\Theta_{\mathbf{P}}$ versus $\Theta_{\mathbf{NP}}$

Theorem (Krajíček-Pudlák-Takeuti 1991)

If $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}}$, then $\Sigma_2^p = \Pi_2^p$.

If $\Theta_{\mathbf{P}} \vdash \mathbf{P} = \mathbf{NP}$, then $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}}$.

How to define $\Theta_{\mathbf{P}}$?

1. Cook: theory PV with a function symbol for every polynomial time algorithm (using the schema of recursion on notation).

2. Jeřábek: extend Buss's language by $\lfloor x/2^y \rfloor$ and add a few axioms to BASIC. Then

$$T_2^0 \equiv \Theta_{\mathbf{P}}$$

in spite of the fact that (probably) the extended Σ_0^b still does not define all sets in **P**.

$\Theta_{\mathbf{P}}$ versus $\Theta_{\mathbf{NP}}$

Theorem (Krajíček-Pudlák-Takeuti 1991)

If $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}}$, then $\Sigma_2^p = \Pi_2^p$.

If $\Theta_{\mathbf{P}} \vdash \mathbf{P} = \mathbf{NP}$, then $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}}$.

Problem (central in proof complexity)

Prove $\Theta_{\mathbf{P}} \not\equiv \Theta_{\mathbf{NP}}$.

Witnessing functions

FP the class of polynomial time computable functions.

Theorem (Buss 1984)

1. Suppose $\Theta_{\mathbf{P}} \vdash \forall x \exists y\ \phi(x, y)$, *where* $\phi \in \Sigma_1^b$. *Then there exists* $f \in \mathbf{FP}$ *such that*

$$\mathbb{N} \models \forall x\ \phi(x, f(x)). \tag{1}$$

2. If $f \in \mathcal{FP}$ *then there exists a formula* $\phi(x, y)$ *such that* $\Theta_{\mathbf{P}} \vdash \forall x \exists y\ \phi(x, y)$ *and* (1).

We say that **FP** are witnessing functions for $\Theta_{\mathbf{P}}$.
Thus, e.g., T_2^0 is associated with **FP**.

Witnessing functions

FP the class of polynomial time computable functions.

Theorem (Buss 1984)

1. Suppose $\Theta_{\mathbf{P}} \vdash \forall x \exists y\ \phi(x, y)$, *where* $\phi \in \Sigma_1^b$. *Then there exists* $f \in \mathbf{FP}$ *such that*

$$\mathbb{N} \models \forall x\ \phi(x, f(x)). \tag{1}$$

2. If $f \in \mathcal{FP}$ *then there exists a formula* $\phi(x, y)$ *such that* $\Theta_{\mathbf{P}} \vdash \forall x \exists y\ \phi(x, y)$ *and* (1).

We say that **FP** are witnessing functions for $\Theta_{\mathbf{P}}$.
Thus, e.g., T_2^0 is associated with **FP**.

Cook's approach to $\mathcal{C} \leftrightarrow \Theta_{\mathcal{C}}$

Say that a function is in a complexity class $\mathcal{C}$ if *the bit graph* of f is in $\mathcal{C}$.

A theory T *is associated with a complexity class* $\mathcal{C}$, if the functions (with the bit graph) in $\mathcal{C}$ are witnessing functions of T.

Second order theories

1. $w \in \{0,1\}^n \leftrightarrow$ binary representation of a number $m < 2^n$
2. $w \in \{0,1\}^n \leftrightarrow X \subseteq [0, n-1]$

As usual, induction for sets is the same, the schema of comprehension depends on the complexity class.

This gives more flexibility to define theories associated with various complexity classes.

The witnessing theorem for conditional separations of $\Theta_{\mathbf{P}}$ from $\Theta_{\mathbf{NP}}$

$$\Theta_{\mathbf{NP}} = \Theta_{\mathbf{P}} + \forall x \exists y (y = \max_{x; x \leq a} f(a, x)),$$

where f is polynomial time computable function such that $f(a, x) \leq a$.
The parameter a will be omitted from f.

$f(b) = \max_x f(x) \;\Leftrightarrow\; \forall y \leq a (f(b) \geq f(y))$

The witnessing theorem for conditional separations of $\Theta_{\mathbf{P}}$ from $\Theta_{\mathbf{NP}}$

$$\Theta_{\mathbf{NP}} = \Theta_{\mathbf{P}} + \forall x \exists y (y = \max_{x; x \leq a} f(a, x)),$$

where f is polynomial time computable function such that $f(a, x) \leq a$.
The parameter a will be omitted from f.

$f(b) = \max_x f(x) \;\Leftrightarrow\; \forall y \leq a (f(b) \geq f(y))$

Theorem

If $\Theta_{\mathbf{P}} = \Theta_{\mathbf{NP}}$ then $\exists n \; \exists$ polynomial time computable functions $g_1, \ldots g_n$ such that $\forall y_1, \ldots, y_n \leq a$

$$g_1(a) \leq a \wedge g(a, y_1) \leq a \wedge \cdots \wedge g(a, y_1, \ldots, y_{n-1}) \leq a$$

and

$$f(g_1(a)) \geq f(y_1) \vee f(g_2(a, y_1)) \geq f(y_2) \vee \cdots \vee f(g_n(a, y_1, \ldots, y_{n-1})) \geq f(y_n).$$

$=$ interactive computation of $\max_x f(x)$ using counterexamples.
If $\Sigma_2^p \neq \Pi_2^p$, then such a computation is not possible, hence $\Theta_{\mathbf{P}} \neq \Theta_{\mathbf{NP}}$.

$f(a, g_1(a)) \geq f(a, y_1) \vee f(a, g_2(a, y_1)) \geq f(a, y_2) \vee \ldots$
$\vee f(a, g_n(a, y_1, \ldots, y_{n-1})) \geq f(a, y_n)$
is obtained from Herbrand's theorem applied to

$$\forall a \exists x \forall y \ f(a, x) \geq f(a, y).$$

Theorem (Herbrand's Theorem for prefix $\forall\exists\forall$)

$$\forall a \exists x \forall y \ \phi(a, x, y)$$

is provable in the predicate calculus iff

$$\phi(a, t_1(a), y_1) \vee \phi(a, t_2(a, y_1), y_2) \vee \cdots \vee \phi(a, t_n(a, y_1, \ldots, y_{n-1}), y_n)$$

is provable in the propositional calculus for some terms $t_1, \ldots, t_n$ with only variables displayed.

Relativized separations

In complexity theory we have, e.g., oracles A, B such that $\mathbf{P}^A = \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{NP}^B$ [Baker-Gill-Solovay 1975].

Definition

Let Γ be a set of formulas defined by a syntactical condition that ignores atomic formulas. Let T be a theory axiomatized by a schema (of induction, PHP, etc.) for the class of formulas Γ. Then the relativized theory $T[R]$ is defined by extending the class of formulas Γ to $\Gamma[R]$, R a new predicate symbol, and extending the schema of T to the schema for all formulas in $\Gamma[R]$. $T[R]$ does not contain any specific axioms about R.

Relativized separations

In complexity theory we have, e.g., oracles A, B such that $\mathbf{P}^A = \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{NP}^B$ [Baker-Gill-Solovay 1975].

Definition

Let Γ be a set of formulas defined by a syntactical condition that ignores atomic formulas. Let T be a theory axiomatized by a schema (of induction, PHP, etc.) for the class of formulas Γ. Then the relativized theory $T[R]$ is defined by extending the class of formulas Γ to $\Gamma[R]$, R a new predicate symbol, and extending the schema of T to the schema for all formulas in $\Gamma[R]$. $T[R]$ does not contain any specific axioms about R.

Theorem

$$PHP\Delta_0[R] \not\equiv I\Delta_0[R].$$

Theorem

$$\Theta_{\mathbf{P}}[R] \not\equiv \Theta_{\mathbf{NP}}[R].$$

Proof:
The proof of $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}} \Rightarrow \Sigma_2^p = \Pi_2^p$ relativizes. Use an oracle A such that $\Sigma_2^{p,A} = \Pi_2^{p,A}$ whose existence is a consequence of Håstad's lower bounds.

□

Theorem

$$\Theta_{\mathbf{P}}[R] \not\equiv \Theta_{\mathbf{NP}}[R].$$

Proof:
The proof of $\Theta_{\mathbf{P}} \equiv \Theta_{\mathbf{NP}} \Rightarrow \Sigma_2^p = \Pi_2^p$ relativizes. Use an oracle A such that $\Sigma_2^{p,A} = \Pi_2^{p,A}$ whose existence is a consequence of Håstad's lower bounds.

□

Theorem

For all $k \geq 0$,

1. *If $\Theta_{\Sigma_k^b} \equiv \Theta_{\Sigma_{k+1}^b}$, then $\Sigma_{k+2}^p = \Pi_{k+2}^p$.*
2. *$\Theta_{\Sigma_k^b}[R] \not\equiv \Theta_{\Sigma_{k+1}^b}[R]$.*

Note: The oracles separating Σ_{k+2}^p from Π_{k+2}^p are constructed using lower bounds on bounded depth boolean circuits [Håstad 1989]. These bounds were proved using random restrictions.