Optimal proof systems, slicewise monotone parameterized problems, and logics for PTIME

Yijia Chen
Shanghai Jiaotong University

August, 2010

(Joint work with **Jörg Flum**, Freiburg)

Flum and *C.* ,

- **On p-optimal proof systems and logics for PTIME**, *ICALP'10*,

- **On slicewise monotone parameterized problems and optimal proof systems for TAUT**, *CSL'10*.

An open problem in proof complexity

An open problem in proof complexity

In proof complexity, we study the length of proofs of propositional tautologies in various proof systems.

An open problem in proof complexity

In proof complexity, we study the length of proofs of propositional tautologies in various proof systems.

RESOLUTION, FREGE SYSTEMS, EXTENDED FREGE SYSTEMS, etc.

An open problem in proof complexity

In proof complexity, we study the length of proofs of propositional tautologies in various proof systems.

Resolution, Frege systems, extended Frege systems, etc.

Question (**Cook** and **Reckhow**, 79; **Krajíček** and **Pudlák**, 89)

Do we have a proof system for TAUT *that can simulate any other proof system with at most polynomial loss of the succinctness of the proofs?*

An open problem in proof complexity

In proof complexity, we study the length of proofs of propositional tautologies in various proof systems.

RESOLUTION, FREGE SYSTEMS, EXTENDED FREGE SYSTEMS, etc.

Question (**Cook** and **Reckhow**, 79; **Krajíček** and **Pudlák**, 89)

Do we have a proof system for TAUT *that can simulate any other proof system with at most polynomial loss of the succinctness of the proofs?*

Depending on the strength of the simulation, we are asking whether there is a optimal/ p-optimal/ effectively p-optimal proof system.

An open problem in parameterized complexity

Question (**Nash**, **Remmel**, and **Vianu**, 05; **Aumann** and **Dombb**, 08)

p-$\textsc{Acc}_{\leq}$

Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input in $\leq n$ steps?

An open problem in parameterized complexity

Question (**Nash**, **Remmel**, and **Vianu**, 05; **Aumann** and **Dombb**, 08)

p-$\textsc{Acc}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input in $\leq n$ steps?

Is p-$\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathsf{uni}}$*?*

An open problem in parameterized complexity

Question (**Nash**, **Remmel**, and **Vianu**, 05; **Aumann** and **Dombb**, 08)

p-$\textsc{Acc}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input in $\leq n$ steps?

Is p-$\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$? Equivalently, is there an algorithm that decides p-$\textsc{Acc}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for a function $f : \mathbb{N} \to \mathbb{N}$?

An open problem in parameterized complexity

Question (**Nash**, **Remmel**, and **Vianu**, 05; **Aumann** and **Dombb**, 08)

p-$\textsc{Acc}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input in $\leq n$ steps?

Is p-$\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathrm{uni}}$? Equivalently, is there an algorithm that decides p-$\textsc{Acc}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for a function $f : \mathbb{N} \to \mathbb{N}$?

Remark. *It is unlikely that p-$\textsc{Acc}_{\leq}$ is* W[1]*-hard.*

An open problem in parameterized complexity

Question (**Nash**, **Remmel**, and **Vianu**, 05; **Aumann** and **Dombb**, 08)

p-$\textsc{Acc}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input in $\leq n$ steps?

Is p-$\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$? Equivalently, is there an algorithm that decides p-$\textsc{Acc}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for a function $f : \mathbb{N} \to \mathbb{N}$?

Remark. *It is unlikely that p-$\textsc{Acc}_{\leq}$ is* W[1]*-hard.*

Theorem (**Flum** and **C.** , 09)

Under some complexity assumption, no such algorithm exists for computable f, i.e., p-$\textsc{Acc}_{\leq} \notin$ XP.

An open problem in finite model theory

Question (**Gurevich**, 88)

Is there a logic capturing PTIME?

An open problem in finite model theory

Question (**Gurevich**, 88)
Is there a logic capturing PTIME*?*

Question (**Chandra** and **Harel**, 82)
Can we effectively enumerate all PTIME*-queries?*

Main results

Main results

Theorem

The following are equivalent:

Main results

Theorem

The following are equivalent:

- *There is a p-optimal propositional proof system.*

Main results

Theorem

The following are equivalent:

- *There is a p-optimal propositional proof system.*
- *Every slicewise monotone problem in* NP *is in* $\mathrm{XP}_{\mathrm{uni}}$. *In particular,* p-$\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$.

Main results

Theorem

The following are equivalent:

- *There is a p-optimal propositional proof system.*
- *Every slicewise monotone problem in* NP *is in* $\mathrm{XP}_{\mathrm{uni}}$*. In particular,* $p\text{-}\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$.
- L_{inv}*, a logic introduced by* **Andreas Blass** *and* **Yuri Gurevich***, captures PTIME.*

Main results

Theorem

The following are equivalent:

Main results

Theorem

The following are equivalent:

- *There is a optimal propositional proof system.*

Main results

Theorem

The following are equivalent:

- *There is a optimal propositional proof system.*
- *Every slicewise monotone problem in* NP *is in* co-$\mathrm{XNP}_{\mathrm{uni}}$. *In particular,* p-$\textsc{Acc}_{\leq} \in$ co-$\mathrm{XNP}_{\mathrm{uni}}$.

Main results

Theorem

The following are equivalent:

- *There is a optimal propositional proof system.*
- *Every slicewise monotone problem in* NP *is in* co-$\mathrm{XNP}_{\mathrm{uni}}$*. In particular,* $p\text{-}\textsc{Acc}_{\leq} \in \text{co-XNP}_{\mathrm{uni}}$.
- L_{inv}*, a logic introduced by* **Andreas Blass** *and* **Yuri Gurevich**, NP*-captures PTIME.*

Contents

Proof systems

Definition (Cook and Reckhow, 79)

A proof system for TAUT is a surjective function $P : \Sigma^* \to \text{TAUT}$ computable in polynomial time.

Optimal proof systems

Optimal proof systems

Definition

A proof system P for TAUT is optimal, if for every proof system P' for TAUT and all $w' \in \Sigma^*$ there is a $w \in \Sigma^*$ such that $|w| \leq |w'|^{O(1)}$ and

$$P(w) = P'(w').$$

Optimal proof systems

Definition

A proof system P for TAUT is optimal, if for every proof system P' for TAUT and all $w' \in \Sigma^*$ there is a $w \in \Sigma^*$ such that $|w| \leq |w'|^{O(1)}$ and

$$P(w) = P'(w').$$

If in addition, w can be computed from w' in polynomial time, then P is p-optimal.

Optimal proof systems

Definition

A proof system P for TAUT is optimal, if for every proof system P' for TAUT and all $w' \in \Sigma^*$ there is a $w \in \Sigma^*$ such that $|w| \leq |w'|^{O(1)}$ and

$$P(w) = P'(w').$$

If in addition, w can be computed from w' in polynomial time, then P is p-optimal.

Question (**Krajíček** and **Pudlák**, 89)

Is there an optimal proof system for TAUT*?*

Parameterized problems

Parameterized problems

Definition

A parameterized problems (Q, κ) consists of a classical problem $Q \subseteq \Sigma^*$ and a parameterization $\kappa : \Sigma^* \to \mathbb{N}$ computable in polynomial time.

Parameterized problems

Definition

A parameterized problems (Q, κ) consists of a classical problem $Q \subseteq \Sigma^*$ and a parameterization $\kappa : \Sigma^* \to \mathbb{N}$ computable in polynomial time.

Example

p-$\text{ACC}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$.
Problem:	Does $\mathbb{M}$ accept the empty input tape in $\leq n$ steps?

Parameterized problems

Definition

A parameterized problems (Q, κ) consists of a classical problem $Q \subseteq \Sigma^*$ and a parameterization $\kappa : \Sigma^* \to \mathbb{N}$ computable in polynomial time.

Example

p-$\textsc{Acc}_{\leq}$	
Input:	An NTM $\mathbb{M}$ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\|\mathbb{M}\|\|$.
Problem:	Does $\mathbb{M}$ accept the empty input tape in $\leq n$ steps?

A key property: If $(\mathbb{M}, 100) \in p\text{-}\textsc{Acc}_{\leq}$, then $(\mathbb{M}, 1000) \in p\text{-}\textsc{Acc}_{\leq}$.

Slicewise monotone problems

Slicewise monotone problems

Definition

A parameterized problem (Q, κ) is slicewise monotone if

Slicewise monotone problems

Definition

A parameterized problem (Q, κ) is slicewise monotone if

- the instances have the form (x, n), where $x \in \Sigma^*$ and $n \in \mathbb{N}$ is given in unary,

Slicewise monotone problems

Definition

A parameterized problem (Q, κ) is slicewise monotone if

- the instances have the form (x, n), where $x \in \Sigma^*$ and $n \in \mathbb{N}$ is given in unary,
- the parameter is $|x|$, i.e., $\kappa(x, n) = |x|$,

Slicewise monotone problems

Definition

A parameterized problem (Q, κ) is slicewise monotone if

- the instances have the form (x, n), where $x \in \Sigma^*$ and $n \in \mathbb{N}$ is given in unary,
- the parameter is $|x|$, i.e., $\kappa(x, n) = |x|$,
- for all $x \in \Sigma^*$ and $n, n' \in \mathbb{N}$ we have

$$\text{if } (x, n) \in Q \text{ and } n < n', \text{ then } (x, n') \in Q, \text{ too.}$$

Slicewise monotone problems

Definition

A parameterized problem (Q, κ) is slicewise monotone if

- the instances have the form (x, n), where $x \in \Sigma^*$ and $n \in \mathbb{N}$ is given in unary,
- the parameter is $|x|$, i.e., $\kappa(x, n) = |x|$,
- for all $x \in \Sigma^*$ and $n, n' \in \mathbb{N}$ we have

$$\text{if } (x, n) \in Q \text{ and } n < n', \text{ then } (x, n') \in Q, \text{ too.}$$

Example

p-GÖDEL

Input:	An FO-sentence φ and an $n \in \mathbb{N}$ in unary.
Parameter:	$\|\varphi\|$.
Problem:	Does φ have a proof of length $\leq n$?

Some uniform parameterized classes

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP}_{\mathrm{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP}_{\mathrm{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.
- $(Q, \kappa) \in \mathrm{XNP}_{\mathrm{uni}}$ if there is a nondeterministic algorithm $\mathbb{A}$ accepting Q such that for some function $f : \mathbb{N} \to \mathbb{N}$ we have $t_{\mathbb{A}}(x) \leq |x|^{f(\kappa(x))}$ for all $x \in Q$.

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP}_{\mathrm{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.
- $(Q, \kappa) \in \mathrm{XNP}_{\mathrm{uni}}$ if there is a nondeterministic algorithm $\mathbb{A}$ accepting Q such that for some function $f : \mathbb{N} \to \mathbb{N}$ we have $t_{\mathbb{A}}(x) \leq |x|^{f(\kappa(x))}$ for all $x \in Q$.

  $t_{\mathbb{A}}(x)$: the number of steps of a shortest accepting run of $\mathbb{A}$ on x if it exists; ∞ otherwise.

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP_{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.
- $(Q, \kappa) \in \mathrm{XNP_{uni}}$ if there is a nondeterministic algorithm $\mathbb{A}$ accepting Q such that for some function $f : \mathbb{N} \to \mathbb{N}$ we have $t_{\mathbb{A}}(x) \leq |x|^{f(\kappa(x))}$ for all $x \in Q$.

  $t_{\mathbb{A}}(x)$: the number of steps of a shortest accepting run of $\mathbb{A}$ on x if it exists; ∞ otherwise.
- $(Q, \kappa) \in \text{co-}\mathrm{XNP_{uni}}$ if its complement $(\Sigma^* \setminus Q, \kappa)$ is in $\mathrm{XNP_{uni}}$.

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP}_{\mathrm{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.
- $(Q, \kappa) \in \mathrm{XNP}_{\mathrm{uni}}$ if there is a nondeterministic algorithm $\mathbb{A}$ accepting Q such that for some function $f : \mathbb{N} \to \mathbb{N}$ we have $t_{\mathbb{A}}(x) \leq |x|^{f(\kappa(x))}$ for all $x \in Q$.

  $t_{\mathbb{A}}(x)$: the number of steps of a shortest accepting run of $\mathbb{A}$ on x if it exists; ∞ otherwise.
- $(Q, \kappa) \in \text{co-}\mathrm{XNP}_{\mathrm{uni}}$ if its complement $(\Sigma^* \setminus Q, \kappa)$ is in $\mathrm{XNP}_{\mathrm{uni}}$.

Trivially p-$\textsc{Acc}_{\leq}$ and p-$\textsc{Gödel}$ are in $\mathrm{XNP}_{\mathrm{uni}}$.

Some uniform parameterized classes

Definition

- $(Q, \kappa) \in \mathrm{XP_{uni}}$ if there is a deterministic algorithm $\mathbb{A}$ deciding $x \in Q$ in time $|x|^{f(\kappa(x))}$ for some function $f : \mathbb{N} \to \mathbb{N}$.
- $(Q, \kappa) \in \mathrm{XNP_{uni}}$ if there is a nondeterministic algorithm $\mathbb{A}$ accepting Q such that for some function $f : \mathbb{N} \to \mathbb{N}$ we have $t_{\mathbb{A}}(x) \leq |x|^{f(\kappa(x))}$ for all $x \in Q$.

  $t_{\mathbb{A}}(x)$: the number of steps of a shortest accepting run of $\mathbb{A}$ on x if it exists; ∞ otherwise.
- $(Q, \kappa) \in \mathrm{co\text{-}XNP_{uni}}$ if its complement $(\Sigma^* \setminus Q, \kappa)$ is in $\mathrm{XNP_{uni}}$.

Trivially p-$\mathrm{ACC}_{\leq}$ and p-GÖDEL are in $\mathrm{XNP_{uni}}$.

Theorem

Let (Q, κ) be slicewise monotone with enumerable Q. Then $(Q, \kappa) \in \mathrm{XNP_{uni}}$.

Our main results

Our main results

Theorem

TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathrm{XP}_{\mathrm{uni}}$.

Our main results

Theorem
TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathrm{XP}_{\mathrm{uni}}$.

Theorem
The following are equivalent:

Our main results

Theorem

TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathrm{XP}_{\mathrm{uni}}$.

Theorem

The following are equivalent:

1. TAUT *has a p-optimal proof system.*

Our main results

Theorem
TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathsf{XP}_{\mathsf{uni}}$.

Theorem
The following are equivalent:

1. TAUT *has a p-optimal proof system.*
2. $p\text{-}\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathsf{uni}}$.

Our main results

Theorem
TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathsf{XP}_{\mathsf{uni}}$.

Theorem
The following are equivalent:

1. TAUT *has a p-optimal proof system.*
2. $p\text{-}\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathsf{uni}}$.
3. $p\text{-}\textsc{Gödel} \in \mathsf{XP}_{\mathsf{uni}}$.

Our main results

Theorem
TAUT *has a p-optimal proof system if and only if every slicewise monotone problem in* NP *is in* $\mathsf{XP}_{\mathsf{uni}}$.

Theorem
The following are equivalent:

1. TAUT *has a p-optimal proof system.*
2. p-$\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathsf{uni}}$.
3. p-$\textsc{Gödel} \in \mathsf{XP}_{\mathsf{uni}}$.
4. *The logic* L_{inv} *captures* PTIME.

Our main results (cont'd)

Our main results (cont'd)

Theorem

TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-$\mathrm{XNP}_{\mathrm{uni}}$*.*

Our main results (cont'd)

Theorem

TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-XNP_{uni}*.*

Theorem

The following are equivalent:

Our main results (cont'd)

Theorem
TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-$\mathrm{XNP}_{\mathrm{uni}}$.

Theorem
The following are equivalent:

1. TAUT *has an optimal proof system.*

Our main results (cont'd)

Theorem

TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-$\mathsf{XNP}_{\mathsf{uni}}$.

Theorem

The following are equivalent:

1. TAUT *has an optimal proof system.*
2. p-$\textsc{Acc}_{\leq} \in$ co-$\mathsf{XNP}_{\mathsf{uni}}$.

Our main results (cont'd)

Theorem

TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-$\mathsf{XNP}_{\mathsf{uni}}$.

Theorem

The following are equivalent:

1. TAUT *has an optimal proof system.*
2. p-$\textsc{Acc}_{\leq} \in$ co-$\mathsf{XNP}_{\mathsf{uni}}$.
3. p-$\textsc{Gödel} \in$ co-$\mathsf{XNP}_{\mathsf{uni}}$.

Our main results (cont'd)

Theorem

TAUT *has an optimal proof system if and only if every slicewise monotone problem in* NP *is in* co-$\mathrm{XNP}_{\mathrm{uni}}$.

Theorem

The following are equivalent:

1. TAUT *has an optimal proof system.*
2. p-$\textsc{Acc}_{\leq} \in$ co-$\mathrm{XNP}_{\mathrm{uni}}$.
3. p-$\textsc{Gödel} \in$ co-$\mathrm{XNP}_{\mathrm{uni}}$.
4. *The logic* L_{inv} NP-*captures* PTIME.

The proof of one implication

The proof of one implication

Theorem

p-$\mathrm{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT *has a p-optimal proof system.*

A tool

A tool

Theorem (**Sadowski**, 02)

The following statements are equivalent:

1. TAUT *has a p-optimal proof system.*
2. TAUT *has an enumeration of the* P*-easy subsets by* PTIME*-machines.*

A tool

Theorem (**Sadowski**, 02)

The following statements are equivalent:

1. TAUT *has a p-optimal proof system.*
2. TAUT *has an enumeration of the* P*-easy subsets by* PTIME*-machines.*

Definition

An enumeration of the P-easy subsets of TAUT by PTIME-machines is a computable function $M : \mathbb{N} \to \Sigma^*$ such that

$$\{Q_i \mid i \in \mathbb{N} \text{ and } Q_i \text{ is accepted by } M(i) \text{ running in polynomial time}\} = \{Q \mid Q \subseteq \text{TAUT and } Q \in \text{PTIME}\}.$$

Proof

Proof

p-$\textsc{Acc}_{\leq} \in \mathsf{XP}_{\mathsf{uni}}$ implies that TAUT has an p-optimal proof system:

Proof

$p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

Proof

$p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

- An algorithm $\mathbb{A}$ decides $p\text{-}\mathrm{ACC}_{\leq}$ in time

Proof

$p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

- An algorithm $\mathbb{A}$ decides $p\text{-}\mathrm{ACC}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for some function f.

Proof

p-$\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

- An algorithm $\mathbb{A}$ decides p-$\textsc{Acc}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for some function f.
- For a DTM $\mathbb{M}$ let $\mathbb{M}^*$ be an NTM that on the empty input tape
  1. guesses a propositional formula α;
  2. checks whether $\mathbb{M}$ accepts α and rejects if this is not the case;
  3. guesses an assignment and accepts if this assignment does not satisfy α.

Proof

p-$\mathrm{ACC}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

- An algorithm $\mathbb{A}$ decides p-$\mathrm{ACC}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for some function f.
- For a DTM $\mathbb{M}$ let $\mathbb{M}^*$ be an NTM that on the empty input tape
  1. guesses a propositional formula α;
  2. checks whether $\mathbb{M}$ accepts α and rejects if this is not the case;
  3. guesses an assignment and accepts if this assignment does not satisfy α.

  $\mathbb{M}^*$ accepts the empty input tape if and only if $\mathbb{M}$ accepts some α which is not a tautology.

Proof

p-$\textsc{Acc}_{\leq} \in \mathrm{XP}_{\mathrm{uni}}$ implies that TAUT has an p-optimal proof system:

We give an numeration of the P-easy subsets of TAUT by PTIME-machines.

- An algorithm $\mathbb{A}$ decides p-$\textsc{Acc}_{\leq}$ in time $n^{f(\|\mathbb{M}\|)}$ for some function f.
- For a DTM $\mathbb{M}$ let $\mathbb{M}^*$ be an NTM that on the empty input tape
  1. guesses a propositional formula α;
  2. checks whether $\mathbb{M}$ accepts α and rejects if this is not the case;
  3. guesses an assignment and accepts if this assignment does not satisfy α.

  $\mathbb{M}^*$ accepts the empty input tape if and only if $\mathbb{M}$ accepts some α which is not a tautology.

  For every $n \in \mathbb{N}$, if $\mathbb{M}^*$ does not accept the empty input tape in at most $n^{O(1)}$ steps, i.e., $(\mathbb{M}^*, n^{O(1)}) \notin p$-$\textsc{Acc}_{\leq}$, then every formula α with $|\alpha| \leq n$ which $\mathbb{M}$ accepts in polynomial time is a tautology.

Proof (cont'd)

Proof (cont'd)

Proof (cont'd)

- A DTM $\mathbb{M}$ is clocked if $\mathbb{M}$ contains a natural number $\mathrm{time}(\mathbb{M})$ such that $n^{\mathrm{time}(\mathbb{M})}$ is a bound for the running time of $\mathbb{M}$ on inputs of length n.

Proof (cont'd)

- A DTM $\mathbb{M}$ is clocked if $\mathbb{M}$ contains a natural number $\mathrm{time}(\mathbb{M})$ such that $n^{\mathrm{time}(\mathbb{M})}$ is a bound for the running time of $\mathbb{M}$ on inputs of length n.

- For a clocked DTM $\mathbb{M}$ let $\mathbb{M}^+$ be a DTM that on input α accepts if and only if (i) and (ii) hold:
  (i) $\mathbb{M}$ accepts α;
  (ii) $(\mathbb{M}^*, |\alpha|^{\mathrm{time}(\mathbb{M})+4}) \notin p\text{-}\mathrm{ACC}_{\leq}$.

Proof (cont'd)

- A DTM $\mathbb{M}$ is clocked if $\mathbb{M}$ contains a natural number $\mathrm{time}(\mathbb{M})$ such that $n^{\mathrm{time}(\mathbb{M})}$ is a bound for the running time of $\mathbb{M}$ on inputs of length n.

- For a clocked DTM $\mathbb{M}$ let $\mathbb{M}^+$ be a DTM that on input α accepts if and only if (i) and (ii) hold:
  (i) $\mathbb{M}$ accepts α;
  (ii) $(\mathbb{M}^*, |\alpha|^{\mathrm{time}(\mathbb{M})+4}) \notin p\text{-}\mathrm{ACC}_{\leq}$.

  $\mathbb{M}^+$ checks (i) by simulating $\mathbb{M}$ and (ii) by simulating $\mathbb{A}$, hence run in time polynomial in $|\alpha|$.

Proof (cont'd)

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $\mathbb{M}$ be a clocked DTM. We prove that $\mathbb{M}^+$ accepts a (P-easy) subset of TAUT.

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $\mathbb{M}$ be a clocked DTM. We prove that $\mathbb{M}^+$ accepts a (P-easy) subset of TAUT.

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $\mathbb{M}$ be a clocked DTM. We prove that $\mathbb{M}^+$ accepts a (P-easy) subset of TAUT.
If $\mathbb{M}^+$ accepts α, then, by (i), $\mathbb{M}$ accepts α and by (ii), $(\mathbb{M}^*, |\alpha|^{\mathrm{time}(\mathbb{M})+4}) \notin p\text{-}\mathrm{ACC}_{\leq}$.

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $\mathbb{M}$ be a clocked DTM. We prove that $\mathbb{M}^+$ accepts a (P-easy) subset of TAUT.
If $\mathbb{M}^+$ accepts α, then, by (i), $\mathbb{M}$ accepts α and by (ii),
$(\mathbb{M}^*, |\alpha|^{\mathrm{time}(\mathbb{M})+4}) \notin p\text{-}\mathrm{ACC}_{\leq}$.
Therefore, by definition of $\mathbb{M}^*$, every assignment satisfies α and hence
$\alpha \in$ TAUT.

Proof (cont'd)

We show that $\mathbb{M}^+$, where $\mathbb{M}$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $\mathbb{M}$ be a clocked DTM. We prove that $\mathbb{M}^+$ accepts a (P-easy) subset of TAUT.
If $\mathbb{M}^+$ accepts α, then, by (i), $\mathbb{M}$ accepts α and by (ii), $(\mathbb{M}^*, |\alpha|^{\mathrm{time}(\mathbb{M})+4}) \notin p\text{-}\mathrm{ACC}_{\leq}$.
Therefore, by definition of $\mathbb{M}^*$, every assignment satisfies α and hence $\alpha \in$ TAUT.

Now let $Q \subseteq$ TAUT be a P-easy subset of TAUT and let $\mathbb{M}$ be a clocked machine deciding Q. Then $\mathbb{M}^+$ accepts Q. □

An application

An application

Definition

A proof system P is effectively p-optimal if for every proof system P' for TAUT, there exists a polynomial time computable function $T : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$ we have

$$P(T(w)) = P'(w).$$

Moreover, we can compute such a T from P'.

An application

Definition

A proof system P is effectively p-optimal if for every proof system P' for TAUT, there exists a polynomial time computable function $T : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$ we have

$$P(T(w)) = P'(w).$$

Moreover, we can compute such a T from P'.

Definition (**C.** and **Flum**, 09)

$\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{NP}[\mathrm{TC}^{\log \mathrm{TC}}]$ means that for every time constructible and increasing function $h : \mathbb{N} \to \mathbb{N}$ we have

$$\mathrm{NTIME}(h^{O(1)}) \not\subseteq \mathrm{DTIME}(h^{O(\log h)}).$$

An application

Definition
A proof system P is effectively p-optimal if for every proof system P' for TAUT, there exists a polynomial time computable function $T : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$ we have

$$P(T(w)) = P'(w).$$

Moreover, we can compute such a T from P'.

Definition (C. and Flum, 09)
$\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{NP}[\mathrm{TC}^{\log \mathrm{TC}}]$ means that for every time constructible and increasing function $h : \mathbb{N} \to \mathbb{N}$ we have

$$\mathrm{NTIME}(h^{O(1)}) \not\subseteq \mathrm{DTIME}(h^{O(\log h)}).$$

Theorem
If $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{NP}[\mathrm{TC}^{\log \mathrm{TC}}]$ *holds, then there is no effectively p-optimal proof system for* TAUT*.*

Logics for PTIME

Logics for PTIME

Definition

A logic $\mathscr{L}$ captures PTIME if:

Logics for PTIME

Definition

A logic $\mathscr{L}$ captures PTIME if:

- for every class K of structures

Logics for PTIME

Definition

A logic $\mathscr{L}$ captures PTIME if:

- for every class K of structures (over the same vocabulary and closed under isomorphisms)

$$K \in \text{PTIME} \iff K = \text{Mod}(\varphi) = \{\mathcal{A} \mid \mathcal{A} \models \varphi\} \text{ for some } \mathscr{L}\text{-sentence } \varphi;$$

Logics for PTIME

Definition

A logic $\mathscr{L}$ captures PTIME if:

- for every class K of structures (over the same vocabulary and closed under isomorphisms)

$$K \in \text{PTIME} \iff K = \text{Mod}(\varphi) = \{\mathcal{A} \mid \mathcal{A} \models \varphi\} \text{ for some } \mathscr{L}\text{-sentence } \varphi;$$

- There exists an algorithm $\mathbb{M}$ deciding $\mathcal{A} \models \varphi$ in time $\|\mathcal{A}\|^{f(|\varphi|)}$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Logics for PTIME

Definition

A logic $\mathscr{L}$ captures PTIME if:

- for every class K of structures (over the same vocabulary and closed under isomorphisms)

$$K \in \text{PTIME} \iff K = \text{Mod}(\varphi) = \{\mathcal{A} \mid \mathcal{A} \models \varphi\} \text{ for some } \mathscr{L}\text{-sentence } \varphi;$$

- There exists an algorithm $\mathbb{M}$ deciding $\mathcal{A} \models \varphi$ in time $\|\mathcal{A}\|^{f(|\varphi|)}$ for some function $f : \mathbb{N} \to \mathbb{N}$. Equivalently,

$$\Big(\{(\mathcal{A}, \varphi) \mid \mathcal{A} \models \varphi \text{ with } \varphi \in \mathscr{L}\}, \kappa((\mathcal{A}, \varphi) := |\varphi|)\Big) \in \text{XP}_{\text{uni}}.$$

Definition

A logic $\mathscr{L}$ captures PTIME if:

- for every class K of structures (over the same vocabulary and closed under isomorphisms)

$$K \in \mathrm{PTIME} \iff K = \mathrm{Mod}(\varphi) = \{\mathcal{A} \mid \mathcal{A} \models \varphi\} \text{ for some } \mathscr{L}\text{-sentence } \varphi;$$

- There exists an algorithm $\mathbb{M}$ deciding $\mathcal{A} \models \varphi$ in time $\|\mathcal{A}\|^{f(|\varphi|)}$ for some function $f : \mathbb{N} \to \mathbb{N}$. Equivalently,

$$\Big(\{(\mathcal{A}, \varphi) \mid \mathcal{A} \models \varphi \text{ with } \varphi \in \mathscr{L}\}, \kappa((\mathcal{A}, \varphi) := |\varphi|)\Big) \in \mathrm{XP}_{\mathrm{uni}}.$$

Conjecture (Gurevich, 88)

There is no logic capturing PTIME.

The logic L_{inv}

The logic L_{inv}

For every vocabulary τ we let $\tau_< := \tau \,\dot{\cup}\, \{<\}$.

The logic L_{inv}

For every vocabulary τ we let $\tau_< := \tau \,\dot\cup\, \{<\}$.

Definition

Let φ be a sentence of least fixed-point logic (LFP) over $\tau_<$ and $m \in \mathbb{N}$.
φ is $\leq m$-invariant if for all τ-structures $\mathcal{A}$ with $|A| \leq m$ we have

$$(\mathcal{A}, <_1) \models_{\text{LFP}} \varphi \iff (\mathcal{A}, <_2) \models_{\text{LFP}} \varphi.$$

for all orderings $<_1$ and $<_2$ on A.

The logic L_{inv}

For every vocabulary τ we let $\tau_< := \tau \,\dot{\cup}\, \{<\}$.

Definition

Let φ be a sentence of least fixed-point logic (LFP) over $\tau_<$ and $m \in \mathbb{N}$.
φ is $\leq m$-invariant if for all τ-structures $\mathcal{A}$ with $|A| \leq m$ we have

$$(\mathcal{A}, <_1) \models_{\text{LFP}} \varphi \iff (\mathcal{A}, <_2) \models_{\text{LFP}} \varphi.$$

for all orderings $<_1$ and $<_2$ on A.

Definition (Blass and Gurevich, 88)

Let $L_{\text{inv}}[\tau] = \text{LFP}[\tau_<]$.

The logic L_{inv}

For every vocabulary τ we let $\tau_< := \tau \,\dot\cup\, \{<\}$.

Definition

Let φ be a sentence of least fixed-point logic (LFP) over $\tau_<$ and $m \in \mathbb{N}$.
φ is $\leq m$-invariant if for all τ-structures $\mathcal{A}$ with $|A| \leq m$ we have

$$(\mathcal{A}, <_1) \models_{\mathrm{LFP}} \varphi \iff (\mathcal{A}, <_2) \models_{\mathrm{LFP}} \varphi.$$

for all orderings $<_1$ and $<_2$ on A.

Definition (**Blass** and **Gurevich**, 88)

Let $L_{\mathrm{inv}}[\tau] = \mathrm{LFP}[\tau_<]$.

Then for every $\varphi \in L_{\leq}[\tau]$ and τ-structure $\mathcal{A}$:

$$\mathcal{A} \models_{L_{\mathrm{inv}}} \varphi \iff \Big(\varphi \text{ is } \leq |A|\text{-invariant and } (\mathcal{A}, <) \models_{\mathrm{LFP}} \varphi \text{ for some ordering } < \text{ on } A\Big).$$

Theorem
TAUT *has a p-optimal proof system if and only if* L_{inv} *captures* PTIME.

Thank You!