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

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 an optimal/p-optimal/effectively p-optimal proof system.
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)

\[ p\text{-Acc} \leq \]

**Input:** An NTM $M$ and an $n \in \mathbb{N}$ in unary.

**Parameter:** $|M|$.

**Problem:** Does $M$ accept the empty input in $\leq n$ steps?

Remark. It is unlikely that $p\text{-Acc} \leq \in \mathbb{XP}$ uni.

Equivalent, is there an algorithm that decides $p\text{-Acc} \leq$ in time $n^f(|M|)$ for a function $f: \mathbb{N} \rightarrow \mathbb{N}$?

Theorem (Flum and C., 09) Under some complexity assumption, no such algorithm exists for computable $f$, i.e., $p\text{-Acc} \leq / \in \mathbb{XP}$. 
An open problem in parameterized complexity

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

\[ p\text{-}Acc \leq \]

**Input:** An NTM \( M \) and an \( n \in \mathbb{N} \) in unary.

**Parameter:** \( |M| \).

**Problem:** Does \( M \) accept the empty input in \( \leq n \) steps?

Is \( p\text{-}Acc \leq \in \text{XP}_{\text{uni}} \)?
An open problem in parameterized complexity

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

\[ p\text{-Acc}_{\leq} \]

Input: An NTM \( M \) and an \( n \in \mathbb{N} \) in unary.

Parameter: \( |M| \).

Problem: Does \( M \) accept the empty input in \( \leq n \) steps?

Is \( p\text{-Acc}_{\leq} \in \text{XP}_{\text{uni}} \)? Equivalently, is there an algorithm that decides \( p\text{-Acc}_{\leq} \) in time \( n^{f(|M|)} \) 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)

\[ p\text{-Acc} \leq \]

\begin{itemize}
  \item **Input:** An NTM \( M \) and an \( n \in \mathbb{N} \) in unary.
  \item **Parameter:** \( \|M\| \).
  \item **Problem:** Does \( M \) accept the empty input in \( \leq n \) steps?
\end{itemize}

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

**Remark.** It is unlikely that \( p\text{-Acc} \leq \) is \( W[1]\)-hard.
An open problem in parameterized complexity

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

\[ p\text{-Acc} \leq \]

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

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

**Remark.** It is unlikely that \( p\text{-Acc} \leq \) is \( \text{W}[1] \)-hard.

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

Under some complexity assumption, no such algorithm exists for computable \( f \), i.e., \( p\text{-Acc} \leq \not\in \text{XP} \).
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 \textsc{PTIME}?

Question (Chandra and Harel, 82)
Can we effectively enumerate all \textsc{PTIME}-queries?
Main results

Theorem

The following are equivalent:

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

**Theorem**

The following are equivalent:

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

Theorem

The following are equivalent:

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

Theorem
The following are equivalent:

- There is a \( p \)-optimal propositional proof system.

- Every \textit{slicewise monotone} problem in NP is in XP\textsubscript{uni}. In particular, \( p\text{-Acc}_\leq \in \text{XP}\text{uni} \).
Main results

Theorem

The following are equivalent:

- There is a \textit{\textbf{p-optimal} propositional proof system.}

- Every \textit{\textbf{slice}}wise \textbf{monotone} problem in \textit{\textbf{NP}} is in \textit{\textbf{XP}}\textsubscript{\textbf{uni}}. In particular, \( p\text{-\textbf{Acc}} \leq \in \textbf{XP}\textsubscript{\textbf{uni}}. \)

- \( L\text{\textsubscript{\textbf{inv}}}, \) a logic introduced by \textbf{Andreas Blass} and \textbf{Yuri Gurevich}, captures \textit{\textbf{PTIME}}.
Main results

Theorem

The following are equivalent:

1. There is a optimal propositional proof system.
2. Every slicewise monotone problem in $NP$ is in $co-XNP \cup$. In particular, $p-Acc \leq \in co-XNP \cup$.
3. $L^{\text{inv}}$, a logic introduced by Andreas Blass and Yuri Gurevich, $NP$-captures $PTIME$. 
Main results

Theorem
The following are equivalent:

- There is an 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-XNP_{uni}. In particular, p-ACC_{≤} ∈ co-XNP_{uni}.
Main results

Theorem
The following are equivalent:

- There is an optimal propositional proof system.

- Every slicewise monotone problem in NP is in co-XNP\textsubscript{uni}. In particular, $p\text{-ACC}_\leq \subseteq \text{co-XNP}_{\text{uni}}$.

- $L_{\text{inv}}$, a logic introduced by Andreas Blass and Yuri Gurevich, NP-captures PTIME.
Contents

Basic notions

Our main results

Some proofs

An application

Connection to the logic $L_{\text{inv}}$
Proof systems

Definition (Cook and Reckhow, 79)

A proof system for TAUT is a surjective function $P : \Sigma^* \rightarrow \text{TAUT}$ computable in polynomial time.
Definition (Cook and Reckhow, 79)
A proof system for TAUT is a surjective function $P : \Sigma^* \rightarrow \text{TAUT}$ computable in polynomial time.
Optimal proof systems

Definition

A proof system $P$ for $\text{TAUT}$ is optimal, if for every proof system $P'$ for $\text{TAUT}$ 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 $\text{TAUT}$?
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?
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

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

Example $p$-Acc
Input:
An NTM $M$ and an $n \in \mathbb{N}$ in unary.

Parameter:
$\|M\|.$

Problem:
Does $M$ accept the empty input tape in $\leq n$ steps?

A key property: If $(M, 100) \in p$-Acc, then $(M, 1000) \in p$-Acc.
Parameterized problems

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

Definition
A parameterized problems $\langle Q, \kappa \rangle$ consists of a classical problem $Q \subseteq \Sigma^*$ and a parameterization $\kappa : \Sigma^* \rightarrow \mathbb{N}$ computable in polynomial time.

Example

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

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

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

Example

\[
\begin{array}{|c|}
\hline
p\text{-Acc}_\leq \\
\hline
\text{Input:} & \text{An NTM } M \text{ and an } n \in \mathbb{N} \text{ in unary.} \\
\text{Parameter:} & |M|. \\
\text{Problem:} & \text{Does } M \text{ accept the empty input tape in } \leq n \text{ steps?} \\
\hline
\end{array}
\]

A key property: If \((M, 100) \in p\text{-Acc}_\leq\), then \((M, 1000) \in p\text{-Acc}_\leq\).
Slicewise monotone problems

Definition

A parameterized problem \((Q, \kappa)\) 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 \(\phi\) and an \(n \in \mathbb{N}\) in unary.

Parameter: \(|\phi|\).

Problem: Does \(\phi\) have a proof of length \(\leq n\)?
Slicewise monotone problems

Definition
A parameterized problem \((Q, \kappa)\) is \text{slicewise monotone} if

Example
\(p\)-Gödel
Input: An FO-sentence \(\phi\) and an \(n \in \mathbb{N}\) in unary.
Parameter: \(\|\phi\|\).
Problem: Does \(\phi\) have a proof of length \(\leq n\)?
Slicewise monotone problems

Definition
A parameterized problem \((Q, \kappa)\) 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 \((Q, \kappa)\) 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., \(\kappa(x, n) = |x|\),

Example
\(p\)-Gödel
Input: An FO-sentence \(\phi\) and an \(n \in \mathbb{N}\) in unary.
Parameter: \(|\phi|\).
Problem: Does \(\phi\) have a proof of length \(\leq n\)?
Slicewise monotone problems

Definition
A parameterized problem $(Q, \kappa)$ 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., $\kappa(x, n) = |x|$, and
- for all $x \in \Sigma^*$ and $n, n' \in \mathbb{N}$ we have
  
  if $(x, n) \in Q$ and $n < n'$, then $(x, n') \in Q$, too.

Example $p$-Gödel

Input: An FO-sentence $\varphi$ and an $n \in \mathbb{N}$ in unary.

Parameter: $\|\varphi\|$.

Problem: Does $\varphi$ have a proof of length $\leq n$?
Slicewise monotone problems

Definition
A parameterized problem \((Q, \kappa)\) is \textit{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

\[p\text{-Gödel}\]

\begin{itemize}
  \item \textit{Input:} An FO-sentence \(\varphi\) and an \(n \in \mathbb{N}\) in unary.
  \item \textit{Parameter:} \(||\varphi||\).
  \item \textit{Problem:} Does \(\varphi\) have a proof of length \(\leq n\)?
\end{itemize}
Some uniform parameterized classes

Definition

▶ \((Q, \kappa) \in X_{\text{P uni}}\) if there is a deterministic algorithm \(A\) deciding \(x \in Q\) in time \(|x| f(\kappa(x))\) for some function \(f: \mathbb{N} \rightarrow \mathbb{N}\).

▶ \((Q, \kappa) \in X_{\text{NP uni}}\) if there is a nondeterministic algorithm \(A\) accepting \(Q\) such that for some function \(f: \mathbb{N} \rightarrow \mathbb{N}\) we have \(t_A(x) \leq |x| f(\kappa(x))\) for all \(x \in Q\).

de\(t_A(x): \) the number of steps of a shortest accepting run of \(A\) on \(x\) if it exists; \(\infty\) otherwise.

▶ \((Q, \kappa) \in \text{co-}X_{\text{NP uni}}\) if its complement \((\Sigma^* \setminus Q, \kappa)\) is in \(X_{\text{NP uni}}\).

Trivially \(p\)-Acc and \(p\)-Gödel are in \(X_{\text{NP uni}}\).

Theorem

Let \((Q, \kappa)\) be slicewise monotone with enumerable \(Q\). Then \((Q, \kappa) \in X_{\text{NP uni}}\).
Some uniform parameterized classes

Definition

- \((Q, \kappa) \in \text{XP}_{\text{uni}}\) if there is a deterministic algorithm \(A\) deciding \(x \in Q\) in time \(|x|^{f(\kappa(x))}\) for some function \(f : \mathbb{N} \rightarrow \mathbb{N}\).
Some uniform parameterized classes

**Definition**

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

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

$t_A(x)$: the number of steps of a shortest accepting run of $A$ on $x$ if it exists; $\infty$ otherwise.
Some uniform parameterized classes

Definition

- \((Q, \kappa) \in X_{\text{Puni}}\) if there is a deterministic algorithm \(A\) deciding \(x \in Q\) in time \(|x|^{|f(\kappa(x))|}\) for some function \(f : \mathbb{N} \to \mathbb{N}\).

- \((Q, \kappa) \in X_{\text{Nuni}}\) if there is a nondeterministic algorithm \(A\) accepting \(Q\) such that for some function \(f : \mathbb{N} \to \mathbb{N}\) we have \(t_A(x) \leq |x|^{|f(\kappa(x))|}\) for all \(x \in Q\).

\(t_A(x)\): the number of steps of a shortest accepting run of \(A\) on \(x\) if it exists; \(\infty\) otherwise.
Some uniform parameterized classes

Definition

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

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

  $t_A(x)$: the number of steps of a shortest accepting run of $A$ on $x$ if it exists; $\infty$ otherwise.

- $(Q, \kappa) \in \text{co-XNP}_{\text{uni}}$ if its complement $(\Sigma^* \setminus Q, \kappa)$ is in $\text{XNP}_{\text{uni}}$. 
Some uniform parameterized classes

Definition

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

- \((Q, \kappa) \in \text{XNP}_{\text{uni}}\) if there is a nondeterministic algorithm \(\mathbb{A}\) accepting \(Q\) such that for some function \(f : \mathbb{N} \rightarrow \mathbb{N}\) 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; \(\infty\) otherwise.

- \((Q, \kappa) \in \text{co-XNP}_{\text{uni}}\) if its complement \((\Sigma^* \setminus Q, \kappa)\) is in \(\text{XNP}_{\text{uni}}\).

Trivially \(p\text{-ACC}_\leq\) and \(p\text{-GÖDEL}\) are in \(\text{XNP}_{\text{uni}}\).
Some uniform parameterized classes

Definition

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

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

\(t_A(x)\): the number of steps of a shortest accepting run of \(A\) on \(x\) if it exists; \(\infty\) otherwise.

- \((Q, \kappa) \in \text{co-XNP}_{\text{uni}}\) if its complement \((\Sigma^* \setminus Q, \kappa)\) is in \(\text{XNP}_{\text{uni}}\).

Trivially \(p\text{-ACC}_\leq\) and \(p\text{-GÖDEL}\) are in \(\text{XNP}_{\text{uni}}\).

Theorem

Let \((Q, \kappa)\) be slicewise monotone with enumerable \(Q\). Then \((Q, \kappa) \in \text{XNP}_{\text{uni}}\).
Our main results

Theorem

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

Theorem

The following are equivalent:

1. TAUT has a p-optimal proof system.
2. \( \mathsf{p-Acc} \subseteq \mathsf{XP} \) uni.
3. \( \mathsf{p-Gödel} \subseteq \mathsf{XP} \) uni.
4. The logic \( \mathsf{L_{inv}} \) captures \( \mathsf{PTIME} \).
Our main results

Theorem
TAUT has a $p$-optimal proof system if and only if every slicewise monotone problem in $NP$ is in $XP_{uni}$. 
Our main results

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

Theorem
The following are equivalent:

1. TAUT has a p-optimal proof system.
2. \( p\text{-Acc} \leq \in \) XP\textsubscript{uni}.
3. \( p\text{-Gödel} \in \) XP\textsubscript{uni}.
4. The logic L\textsubscript{inv} captures PTIME.
Our main results

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

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 XP\textsubscript{uni}.

Theorem
The following are equivalent:
1. TAUT has a p-optimal proof system.
2. $p$-Acc \textsubscript{≤} ∈ XP\textsubscript{uni}.
Our main results

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

**Theorem**
The following are equivalent:
1. TAUT has a p-optimal proof system.
2. \( p\text{-ACC} \subseteq \in \text{XP}_{\text{uni}} \).
3. \( p\text{-GÖDEL} \in \text{XP}_{\text{uni}} \).
Our main results

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

Theorem
The following are equivalent:
1. TAUT has a p-optimal proof system.
2. $p$-ACC\textsubscript{$\leq$} $\in$ XP\textsubscript{uni}.
3. $p$-GÖDEL $\in$ XP\textsubscript{uni}.
4. The logic $L_{\text{inv}}$ captures PTIME.
Our main results (cont’d)

Theorem

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

Theorem

The following are equivalent:

1. TAUT has an optimal proof system.
2. $p\text{-Acc} \leq \in \mathbf{co-XNP}_{uni}$.
3. $p\text{-Gödel} \in \mathbf{co-XNP}_{uni}$.
4. The logic $\mathbf{L}_{inv\mathbf{NP}}$-captures $\mathbf{PTIME}$.
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\textsubscript{uni}.
Our main results (cont’d)

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

Theorem
The following are equivalent:

1. TAUT has an optimal proof system.
2. $p$-Acc $\leq \in \mathbf{co-XNP}_{\text{uni}}$.
3. $p$-Gödel $\in \mathbf{co-XNP}_{\text{uni}}$.
4. The logic $L_{\mathbf{invNP}}$-captures $\mathbf{PTIME}$.
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:
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-XNP\_{uni}.

Theorem
The following are equivalent:

1. TAUT has an optimal proof system.
2. $p$-Acc$\leq$ ∈ co-XNP\_{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<sub>uni</sub>.

**Theorem**

The following are equivalent:

1. TAUT has an optimal proof system.
2. $p$-ACC$\leq$ ∈ co-XNP<sub>uni</sub>.
3. $p$-GÖDEL ∈ co-XNP<sub>uni</sub>. 
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\textsubscript{uni}.

**Theorem**
The following are equivalent:
1. TAUT has an optimal proof system.
2. \( p\text{-Acc} \leq \in \text{co-XNP}_{\text{uni}} \).
3. \( p\text{-Gödel} \in \text{co-XNP}_{\text{uni}} \).
4. The logic \( L_{\text{inv}} \) NP-captures PTIME.
The proof of one implication
The proof of one implication

Theorem
\[ p\text{-Acc}_\leq \in X\text{P}_{\text{uni}} \text{ implies that TAUT has a } p\text{-optimal proof system.} \]
Theorem (Sadowski, 02)

The following statements are equivalent:

1. \( \text{TAUT} \) has a p-optimal proof system.
2. \( \text{TAUT} \) has an enumeration of the \( \mathcal{P} \)-easy subsets by \( \mathcal{PTIME} \)-machines.

**Definition**

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

\[
\{ Q_i \mid i \in \mathbb{N} \text{ and } Q_i \text{ is accepted by } M(i) \text{ running in polynomial time} \} = \{ Q \mid Q \subseteq \text{TAUT} \text{ and } Q \in \mathcal{PTIME} \}.
\]
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.*
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 $M : \mathbb{N} \rightarrow \Sigma^*$ such that

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

We give an numeration of the P-easy subsets of TAUT by PTIME-machines. An algorithm $A$ decides $p$-Acc in time $n^f(\|M\|)$ for some function $f$. For a DTM $M$ let $M^*$ be an NTM that on the empty input tape
1. guesses a propositional formula $\alpha$;
2. checks whether $M$ accepts $\alpha$ and rejects if this is not the case;
3. guesses an assignment and accepts if this assignment does not satisfy $\alpha$.

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

\( p\text{-Acc} \in XP_{uni} \) implies that TAUT has an \( p\text{-optimal} \) proof system:
Proof

$p\text{-}\text{Acc} \leq \in \text{XP_{uni}}$ implies that TAUT has an p-optimal proof system:

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

\( p\text{-Acc}_\leq \in \mathbf{X}_{\text{uni}} \) 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 \( A \) decides \( p\text{-Acc}_\leq \) in time

\[
\text{for some function } f(n) \text{.}
\]

- For a DTM \( M \) let \( M^* \) be an NDTM that on the empty input tape

1. guesses a propositional formula \( \alpha \);
2. checks whether \( M \) accepts \( \alpha \) and rejects if this is not the case;
3. guesses an assignment and accepts if this assignment does not satisfy \( \alpha \).

\( M^* \) accepts the empty input tape if and only if \( M \) accepts some \( \alpha \) which is not a tautology.

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

\[ p\text{-Acc}_\leq \in XP_{uni} \text{ implies that TAUT has an p-optimal proof system:} \]

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

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

\( p\text{-Acc}_\leq \in \text{XP}_{\text{uni}} \) 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 \( A \) decides \( p\text{-Acc}_\leq \) in time \( n^{f(\|M\|)} \) for some function \( f \).

- For a DTM \( M \) let \( M^* \) be an NTM that on the empty input tape
  1. guesses a propositional formula \( \alpha \);
  2. checks whether \( M \) accepts \( \alpha \) and rejects if this is not the case;
  3. guesses an assignment and accepts if this assignment does not satisfy \( \alpha \).
Proof

$p\text{-Acc}_\leq \in XP_{\text{uni}}$ implies that TAUT has a $p$-optimal proof system:

We give an numeration of the $P$-easy subsets of TAUT by $\text{PTIME}$-machines.

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

- For a DTM $M$ let $M^*$ be an $\text{NTM}$ that on the empty input tape
  1. guesses a propositional formula $\alpha$;
  2. checks whether $M$ accepts $\alpha$ and rejects if this is not the case;
  3. guesses an assignment and accepts if this assignment does not satisfy $\alpha$.

$M^*$ accepts the empty input tape if and only if $M$ accepts some $\alpha$ which is not a tautology.
Proof

$p\text{-Acc}_{\leq} \in XP_{\text{uni}}$ implies that TAUT has a p-optimal proof system:

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

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

- For a DTM $M$ let $M^*$ be an NTM that on the empty input tape

1. guesses a propositional formula $\alpha$;
2. checks whether $M$ accepts $\alpha$ and rejects if this is not the case;
3. guesses an assignment and accepts if this assignment does not satisfy $\alpha$.

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

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

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

For a clocked DTM $M$, let $M^+$ be a DTM that on input $\alpha$ accepts if and only if (i) and (ii) hold:

(i) $M$ accepts $\alpha$;
(ii) $(M^*, |\alpha| \times \text{time}(M) + 4) / \in \text{p-Acc} \leq$. 

$M^+$ checks (i) by simulating $M$ and (ii) by simulating $A$, hence run in time polynomial in $|\alpha|$. 
Proof (cont’d)

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

- For a clocked DTM $M$, let $M^+$ be a DTM that on input $\alpha$ accepts if and only if (i) and (ii) hold:
  
  (i) $M$ accepts $\alpha$;

  (ii) $(M^*, |\alpha| + 4) / \in \text{p-Acc} \leq M^+$.

$M^+$ checks (i) by simulating $M$ and (ii) by simulating $A$, hence run in time polynomial in $|\alpha|$. 
- A DTM \( M \) is \textit{clocked} if \( M \) contains a natural number \( \text{time}(M) \) such that \( n^{\text{time}(M)} \) is a bound for the running time of \( M \) on inputs of length \( n \).
- A DTM $M$ is **clocked** if $M$ contains a natural number $\text{time}(M)$ such that $n^{\text{time}(M)}$ is a bound for the running time of $M$ on inputs of length $n$.

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

Proof (cont’d)
Proof (cont’d)

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

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

$M^+$ checks (i) by simulating $M$ and (ii) by simulating $A$, hence run in time polynomial in $|\alpha|$.
Proof (cont’d)

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

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

Therefore, by definition of $M^*$, every assignment satisfies $\alpha$ and hence $\alpha \in \text{TAUT}$.

Now let $Q \subseteq \text{TAUT}$ be a P-easy subset of TAUT and let $M$ be a clocked machine deciding $Q$. Then $M^+$ accepts $Q$. □
We show that $M^+$, where $M$ 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.
We show that $M^+$, where $M$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

First let $M$ be a clocked DTM. We prove that $M^+$ accepts a (P-easy) subset of TAUT.
We show that $M^+$, where $M$ ranges over all clocked machines, yields an enumeration of all P-easy subsets of TAUT by NP-machines.

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

If $M^+$ accepts $\alpha$, then, by (i), $M$ accepts $\alpha$ and by (ii),

$$(M^*, |\alpha|^\text{time}(M)+4) \notin p\text{-ACC}_\leq.$$
Proof (cont’d)

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

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

If $\mathcal{M}^+$ accepts $\alpha$, then, by (i), $\mathcal{M}$ accepts $\alpha$ and by (ii),

$$(\mathcal{M}^*, |\alpha|^{\text{time}(\mathcal{M})+4}) \notin \text{p-Acc}_{\leq}.$$

Therefore, by definition of $\mathcal{M}^*$, every assignment satisfies $\alpha$ and hence $\alpha \in \text{TAUT}$. 

Now let $\mathcal{Q} \subseteq \text{TAUT}$ be a P-easy subset of $\text{TAUT}$ and let $\mathcal{M}$ be a clocked machine deciding $\mathcal{Q}$. Then $\mathcal{M}^+$ accepts $\mathcal{Q}$. □
Proof (cont’d)

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

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

If $M^+$ accepts $\alpha$, then, by (i), $M$ accepts $\alpha$ and by (ii),

$$(M^*, |\alpha|^{\text{time}(M) + 4}) \notin p\text{-Acc}_\leq.$$

Therefore, by definition of $M^*$, every assignment satisfies $\alpha$ and hence $\alpha \in TAUT$.

Now let $Q \subseteq TAUT$ be a P-easy subset of TAUT and let $M$ be a clocked machine deciding $Q$. Then $M^+$ 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 $T : \Sigma^* \rightarrow \Sigma^*$ 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 $T : \Sigma^* \rightarrow \Sigma^*$ 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)
$NP[TC] \not\subseteq NP[TC^{\log TC}]$ means that for every time constructible and increasing function $h : \mathbb{N} \rightarrow \mathbb{N}$ we have

$$NTIME(h^{O(1)}) \not\subseteq DTIME(h^{O(\log h)}).$$
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 $T : \Sigma^* \rightarrow \Sigma^*$ 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)**

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

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

**Theorem**

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

Definition

A logic $L$ captures PTIME if:

▶ for every class $K$ of structures (over the same vocabulary and closed under isomorphisms)
   $K \in \text{PTIME} \iff K = \text{Mod}(\varphi) = \{ A | A \models \varphi \}$ for some $L$-sentence $\varphi$;

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

Equivalently, $\left( \{ (A, \varphi) | A \models \varphi \text{ with } \varphi \in L \} , \kappa(\varphi) = |\varphi| \right) \in \text{XP uni}$.

Conjecture (Gurevich, 88)

There is no logic capturing PTIME.
Definition
A logic $L$ captures PTIME if:

1. For every class $K$ of structures (over the same vocabulary and closed under isomorphisms) $K \in \text{PTIME} \iff K = \text{Mod}(\phi) = \{A| |A| = \phi\}$ for some $L$-sentence $\phi$;
2. There exists an algorithm $M$ deciding $|A| = \phi$ in time $\|A\| f(|\phi|)$ for some function $f: \mathbb{N} \rightarrow \mathbb{N}$.

Equivalently, $\left(\{(A,\phi)| |A| = \phi\text{ with } \phi \in L\}, \kappa((A,\phi) := |\phi|)\right) \in \text{XP uni}$.

Conjecture (Gurevich, 88)
There is no logic capturing PTIME.
Logics for PTIME

Definition
A logic $\mathcal{L}$ captures PTIME if:

- for every class $K$ of structures
Definition

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

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

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

Definition

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

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

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

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

Conjecture (Gurevich, 88)

There is no logic capturing PTIME.
Logics for PTIME

Definition

A logic \( \mathcal{L} \) captures PTIME if:

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

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

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

\[
\left( \{ (A, \varphi) \mid A \models \varphi \text{ with } \varphi \in \mathcal{L} \}, \kappa((A, \varphi) := |\varphi|) \right) \in \text{XP}_\text{uni}.
\]
Logics for PTIME

Definition
A logic $\mathcal{L}$ captures PTIME if:

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

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

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

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

Conjecture (Gurevich, 88)
There is no logic capturing PTIME.
The logic $L_{inv}$
The logic $L_{inv}$

For every vocabulary $\tau$ we let $\tau< := \tau \cup \{<\}$. 
The logic $L_{\text{inv}}$

For every vocabulary $\tau$ we let $\tau_{\prec} := \tau \cup \{\prec\}$.

**Definition**

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

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

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

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

Let $L_{\text{inv}}[\tau] = \text{LFP}[\tau_{\prec}]$. Then for every $\varphi \in L_{\leq}[\tau]$ and $\tau$-structure $A$:

$A \models L_{\text{inv}} \varphi \iff (\varphi \text{ is } \leq_{|A|}-\text{invariant} \text{ and } (A, <) \models \text{LFP } \varphi \text{ for some ordering } < \text{ on } A).$
The logic $L_{inv}$

For every vocabulary $\tau$ we let $\tau_\prec := \tau \cup \{\prec\}$.

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

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

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

**Definition** ([Blass and Gurevich, 88])
Let $L_{inv}[\tau] = \text{LFP}[\tau_\prec]$. 
The logic $L_{\text{inv}}$

For every vocabulary $\tau$ we let $\tau^< := \tau \cup \{<\}$.

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

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

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

**Definition (Blass and Gurevich, 88)**
Let $L_{\text{inv}}[\tau] = \text{LFP}[\tau^<]$.

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

$$A \models_{L_{\text{inv}}} \varphi \iff \left( \varphi \text{ is } \leq |A|\text{-invariant} \right.$$ 

$$\left. \quad \text{and } (A, <) \models_{\text{LFP}} \varphi \text{ for some ordering } < \text{ on } A \right).$$
Theorem
TAUT has a $p$-optimal proof system if and only if $L_{\text{inv}}$ captures PTIME.
Thank You!