Proof Systems that Take Advice

Olaf Beyersdorff

Institut für Theoretische Informatik
Leibniz-Universität Hannover, Germany

joint work with Johannes Köbler and Sebastian Müller

Proof Systems

Definition (Cook, Reckhow 79)

A proof system for a language L is a function f with $rng(f) = L$.

- correctness: $rng(f) \subseteq L$
- completeness: $L \subseteq rng(f)$
- in addition: proofs should be easy to check,
  i.e. f should be easy to compute.

Proof Systems

Definition (Cook, Reckhow 79)

A proof system for a language L is a function f with $rng(f) = L$.

- correctness: $rng(f) \subseteq L$
- completeness: $L \subseteq rng(f)$
- in addition: proofs should be easy to check, i.e. f should be easy to compute.

Efficiency Requirements

- Cook, Reckhow 79 — f is computable in polynomial time.
- Cook, Krajiček 07 — f is computable by polynomial-size circuits.
- A new model — f is computable in polynomial time using an oracle.

Short Proofs

Definition

A proof system f is polynomially bounded if there exists a polynomial p such that every $x \in rng(f)$ has an f-proof w of length $|w| \leq p(|x|)$.

Short Proofs

Definition

A proof system f is polynomially bounded if there exists a polynomial p such that every $x \in rng(f)$ has an f-proof w of length $|w| \leq p(|x|)$.

Theorem (Cook, Reckhow 79)

A language L has a polynomially bounded proof system in FP *iff* $L \in$ NP*.*

Short Proofs

Definition

A proof system f is polynomially bounded if there exists a polynomial p such that every $x \in rng(f)$ has an f-proof w of length $|w| \leq p(|x|)$.

Theorem (Cook, Reckhow 79)

A language L has a polynomially bounded proof system in FP *iff* $L \in$ NP.

Most important application

- Propositional proof systems for $L = TAUT$.
- $TAUT$ has a polynomially bounded proof system in FP iff NP $=$ coNP.

The Cook-Reckhow Program

Show NP $\neq$ coNP (and hence P $\neq$ NP) by proving superpolynomial lower bounds on the proof length for increasingly stronger propositional proof systems.

The Cook-Reckhow Program

Show NP $\neq$ coNP (and hence P $\neq$ NP) by proving superpolynomial lower bounds on the proof length for increasingly stronger propositional proof systems.

Definition (Krajíček, Pudlák 89)

- A proof system f simulates a proof system g, if for any g-proof w there is an f-proof w' of length $|w'| = |w|^{O(1)}$ such that $f(w') = g(w)$.

The Cook-Reckhow Program

Show NP $\neq$ coNP (and hence P $\neq$ NP) by proving superpolynomial lower bounds on the proof length for increasingly stronger propositional proof systems.

Definition (Krajíček, Pudlák 89)

- A proof system f simulates a proof system g, if for any g-proof w there is an f-proof w' of length $|w'| = |w|^{O(1)}$ such that $f(w') = g(w)$.
- A proof system f for L is optimal if it simulates any proof system for L.

Does TAUT have Optimal Proof Systems?

Question (Krajiček, Pudlák 89)

Does TAUT have an optimal proof system?

Does TAUT have Optimal Proof Systems?

Question (Krajiček, Pudlák 89)

Does TAUT have an optimal proof system?

Some partial answers

- If NE $=$ coNE, then TAUT has optimal proof systems. [Krajiček, Pudlák 89]
- Optimal proof systems for TAUT imply complete sets for promise classes (e.g. NP $\cap$ Sparse, UP, disjoint NP-pairs). [Köbler, Messner, Torán 03]

Proof Systems that Take Advice

Cook & Krajiček (JSL 07) consider non-uniform Frege proofs.

Proof Systems that Take Advice

Cook & Krajiček (JSL 07) consider non-uniform Frege proofs.

Definition (Karp, Lipton 80)

- An advice function is a mapping $h : \mathbb{N} \to \Sigma^*$.
- $h(n)$ is the advice string provided by h for input length n.
- For a language L, $L/h = \{x \mid \langle x, h(|x|)\rangle \in L\}$.

Proof Systems that Take Advice

Cook & Krajiček (JSL 07) consider non-uniform Frege proofs.

Definition (Karp, Lipton 80)

- An advice function is a mapping $h : \mathbb{N} \to \Sigma^*$.
- $h(n)$ is the advice string provided by h for input length n.
- For a language L, $L/h = \{x \mid \langle x, h(|x|)\rangle \in L\}$.
- For a complexity class C and a length bound $k : \mathbb{N} \to \mathbb{N}$, $C/k = \{L/h \mid L \in C, |h(n)| \leq k(n) \text{ for all } n\}$.

Proof Systems that Take Advice

Cook & Krajiček (JSL 07) consider non-uniform Frege proofs.

Definition (Karp, Lipton 80)

- An advice function is a mapping $h : \mathbb{N} \to \Sigma^*$.
- $h(n)$ is the advice string provided by h for input length n.
- For a language L, $L/h = \{x \mid \langle x, h(|x|)\rangle \in L\}$.
- For a complexity class C and a length bound $k : \mathbb{N} \to \mathbb{N}$, $C/k = \{L/h \mid L \in C, |h(n)| \leq k(n) \text{ for all } n\}$.
- $C/\log = \bigcup\{C/k \mid k(n) = O(\log n)\}$.
- $C/\text{poly} = \bigcup\{C/k \mid k(n) = n^{O(1)}\}$.

Proof Systems that Take Advice

Cook & Krajíček (JSL 07) consider non-uniform Frege proofs.

Definition (Karp, Lipton 80)

- An advice function is a mapping $h : \mathbb{N} \to \Sigma^*$.
- $h(n)$ is the advice string provided by h for input length n.
- For a language L, $L/h = \{x \mid \langle x, h(|x|)\rangle \in L\}$.
- For a complexity class C and a length bound $k : \mathbb{N} \to \mathbb{N}$, $C/k = \{L/h \mid L \in C, |h(n)| \leq k(n) \text{ for all } n\}$.
- $C/\log = \bigcup\{C/k \mid k(n) = O(\log n)\}$.
- $C/\text{poly} = \bigcup\{C/k \mid k(n) = n^{O(1)}\}$.

Proposition (Pippenger 79)

$L \in P/\text{poly}$ *iff* L *has poly-size circuits.*

Proof Systems that Take Advice

Cook & Krajiček consider non-uniform proof systems computable in FP/poly.

Proof Systems that Take Advice

Cook & Krajiček consider non-uniform proof systems computable in FP/poly.

Interesting questions on this model

1. Do there exist optimal proof systems with advice?
2. Do there exist polynomially bounded proof systems with advice?
3. Does advice help to shorten propositional proofs?

All Languages have Optimal Proof Systems with Advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 09)

Every language L has an optimal proof system f in $\mathrm{FP}/1$.
In fact, f simulates all proof systems in $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

All Languages have Optimal Proof Systems with Advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 09)

Every language L has an optimal proof system f in $\mathrm{FP}/1$*. In fact, f simulates all proof systems in* $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.

All Languages have Optimal Proof Systems with Advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 09)

Every language L has an optimal proof system f in $\mathrm{FP}/1$. *In fact, f simulates all proof systems in* $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.

All Languages have Optimal Proof Systems with Advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 09)

Every language L has an optimal proof system f in $\mathrm{FP}/1$*.*
In fact, f simulates all proof systems in $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T with advice a only outputs elements from L on inputs of length $|u|$.

All Languages have Optimal Proof Systems with Advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 09)

Every language L has an optimal proof system f in $\mathrm{FP}/1$.
In fact, f simulates all proof systems in $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T with advice a only outputs elements from L on inputs of length $|u|$.
- Now, if $h(|w|) = 1$ and $T(u, a)$ outputs y after at most m steps, then $f(w) = y$. Otherwise, $f(w) = \bot$.

All Languages have Optimal Proof Systems with Advice

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T with advice a only outputs elements from L for inputs of length $|u|$.
- Now, if $h(|w|) = 1$ and $T(u, a)$ outputs y after at most m steps, then $f(w) = y$. Otherwise, $f(w) = \perp$.

All Languages have Optimal Proof Systems with Advice

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T with advice a only outputs elements from L for inputs of length $|u|$.
- Now, if $h(|w|) = 1$ and $T(u, a)$ outputs y after at most m steps, then $f(w) = y$. Otherwise, $f(w) = \bot$.
- If g is a $ps/0$ computed by a p-time transducer T, then f p-simulates g via the FP function $u \mapsto \langle u, 1^T, 1^\varepsilon, 1^{p(|u|)} \rangle$.

All Languages have Optimal Proof Systems with Advice

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^a, 1^m \rangle$ with $u, T, a \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T with advice a only outputs elements from L for inputs of length $|u|$.
- Now, if $h(|w|) = 1$ and $T(u, a)$ outputs y after at most m steps, then $f(w) = y$. Otherwise, $f(w) = \bot$.
- If g is a $ps/0$ computed by a p-time transducer T, then f p-simulates g via the FP function $u \mapsto \langle u, 1^T, 1^\varepsilon, 1^{p(|u|)} \rangle$.
- On the other hand, if T uses logarithmic advice $h(|u|)$, then f simulates g via $u \mapsto \langle u, 1^T, 1^{h(|u|)}, 1^{p(|u|)} \rangle$.

□

Accessing the Advice More Flexibly

Cook & Krajiček also investigate non-uniform proof systems with advice depending on the proof w rather than on $|w|$.

Accessing the Advice More Flexibly

Cook & Krajiček also investigate non-uniform proof systems with advice depending on the proof w rather than on $|w|$.

Definition

Let $k : \mathbb{N} \to \mathbb{N}$ be a length bound.
f is a proof system with k bits of advice (abbr. f is a ps/k), if there exists an advice function $h : \mathbb{N} \to \Sigma^*$ and an advice selector $\ell : \Sigma^* \to \mathbb{N}$ such that

1. the function $w \mapsto 1^{\ell(w)}$ is computable in polynomial time (implying that $\ell(w) \leq p(|w|)$ for some polynomial p),
2. $f(w)$ is computable in polynomial time with advice $h(\ell(w))$, and
3. for all n, the length of the advice $h(n)$ is bounded by $k(n)$.

Accessing the Advice More Flexibly

Cook & Krajiček also investigate non-uniform proof systems with advice depending on the proof w rather than on $|w|$.

Definition

Let $k : \mathbb{N} \to \mathbb{N}$ be a length bound.
f is a proof system with k bits of advice (abbr. f is a ps/k), if there exists an advice function $h : \mathbb{N} \to \Sigma^*$ and an advice selector $\ell : \Sigma^* \to \mathbb{N}$ such that

1. the function $w \mapsto 1^{\ell(w)}$ is computable in polynomial time (implying that $\ell(w) \leq p(|w|)$ for some polynomial p),
2. $f(w)$ is computable in polynomial time with advice $h(\ell(w))$, and
3. for all n, the length of the advice $h(n)$ is bounded by $k(n)$.

For a class F of functions, we denote by ps/F the class of all ps/k with $k \in F$.

Input and Output Advice

Definition

- f is a ps/k with input advice, if ℓ has the form $\ell(w) = |w|$.
- f is a ps/k with output advice, if $\ell(w) = |f(w)|$ for all w in the domain of f.

Input and Output Advice

Definition

- f is a ps/k with input advice, if ℓ has the form $\ell(w) = |w|$.
- f is a ps/k with output advice, if $\ell(w) = |f(w)|$ for all w in the domain of f.

Proposition

Every ps/k is p-equivalent to a ps/k with input advice.

Polynomially Bounded Proof Systems

Question

Which languages have polynomially bounded proof systems with advice?

Polynomially Bounded Proof Systems

Question

Which languages have polynomially bounded proof systems with advice?

Answer

The following table shows which languages are provable by polynomially bounded proof systems with advice.

	input advice	output advice
ps/poly	NP/poly	NP/poly
$ps/\log$	NIC[log, poly]	NP/log
$ps/1$	NIC[log, poly]	NP/1
$ps/0$	NP	

Polynomially Bounded Proof Systems

Languages with polynomially bounded proof systems

	input advice	output advice
ps/poly	NP/poly	NP/poly
$ps/\log$	NIC[log, poly]	NP/log
$ps/1$	NIC[log, poly]	NP/1
$ps/0$	NP	

Strict inclusions between these classes

$$\text{NP} \subsetneq \text{NP}/1 \subsetneq \text{NP}/\log \subsetneq \text{NIC}[\log, \text{poly}] \subsetneq \text{NP}/\text{poly}.$$

Practical Considerations

Observation

1. Advice seems to help when proving tautologies.

Practical Considerations

Observation

1. Advice seems to help when proving tautologies.
2. But: Nonuniform algorithms are not a feasible model of computation in practice.
3. Advice can be arbitrarily complex (even non-recursive).

Practical Considerations

Observation

1. Advice seems to help when proving tautologies.
2. But: Nonuniform algorithms are not a feasible model of computation in practice.
3. Advice can be arbitrarily complex (even non-recursive).

Question

Can we simplify the advice when proving tautologies?

Practical Considerations

Observation

1. Advice seems to help when proving tautologies.
2. But: Nonuniform algorithms are not a feasible model of computation in practice.
3. Advice can be arbitrarily complex (even non-recursive).

Question

Can we simplify the advice when proving tautologies?

We will see that

1. advice can be substituted by a weak oracle.

Practical Considerations

Observation

1. Advice seems to help when proving tautologies.
2. But: Nonuniform algorithms are not a feasible model of computation in practice.
3. Advice can be arbitrarily complex (even non-recursive).

Question

Can we simplify the advice when proving tautologies?

We will see that

1. advice can be substituted by a weak oracle.
2. we can transfer the advice from the proof to the proven formula.

Substituting Advice by Oracles

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*
2. *Conversely, every propositional proof system computable in polynomial time with access to a sparse* NP*-oracle is simulated by a propositional proof system with logarithmic advice.*

Substituting Advice by Oracles

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*
2. *Conversely, every propositional proof system computable in polynomial time with access to a sparse* NP*-oracle is simulated by a propositional proof system with logarithmic advice.*

Sparse NP-sets are weak

$TAUT \notin \mathrm{NP}^S$ with a sparse NP-oracle S, unless the polynomial hierarchy collapses to its second level. [Kadin 89]

Sketch of Proof

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*

Proof.

- Let f be a proof system computed by M_f with advice function h_f, where $|h_f(n)| \leq c \cdot \log(n)$.

Sketch of Proof

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*

Proof.

- Let f be a proof system computed by M_f with advice function h_f, where $|h_f(n)| \leq c \cdot \log(n)$.
- Oracle $A =_{def} \{\langle 1^n, a\rangle \mid \ a \in \Sigma^{\leq c\cdot\log(n)}$ and $\exists x \in \Sigma^n : \ M_f(x; a) \notin TAUT)\}$.

Sketch of Proof

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*

Proof.

- Let f be a proof system computed by M_f with advice function h_f, where $|h_f(n)| \leq c \cdot \log(n)$.
- Oracle $A =_{def} \{\langle 1^n, a\rangle \mid \ a \in \Sigma^{\leq c\cdot\log(n)}$ and $\exists x \in \Sigma^n : \ M_f(x; a) \notin TAUT)\}$.
- $A \in \mathrm{NP} \cap \mathrm{Sparse}$.

Sketch of Proof

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*

Proof.

- Let f be a proof system computed by M_f with advice function h_f, where $|h_f(n)| \leq c \cdot \log(n)$.
- Oracle $A =_{def} \{\langle 1^n, a\rangle \mid \; a \in \Sigma^{\leq c\cdot\log(n)} \text{ and } \exists x \in \Sigma^n : \; M_f(x; a) \notin TAUT\}$.
- $A \in \mathrm{NP} \cap \mathrm{Sparse}$.
- The proof system $g \in \mathrm{FP}^A$: On input $\langle \pi, \varphi\rangle$ simulate $M_f(\pi; a)$ for every advice $a \in \Sigma^{\leq c\cdot\log(|\pi|)}$ with $\langle 1^{|\pi|}, a\rangle \notin A$.

Sketch of Proof

Theorem

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*

Proof.

- Let f be a proof system computed by M_f with advice function h_f, where $|h_f(n)| \leq c \cdot \log(n)$.
- Oracle $A =_{def} \{\langle 1^n, a\rangle \mid \ a \in \Sigma^{\leq c\cdot\log(n)}$ and $\exists x \in \Sigma^n : \ M_f(x; a) \notin TAUT)\}$.
- $A \in \mathrm{NP} \cap \mathrm{Sparse}$.
- The proof system $g \in \mathrm{FP}^A$: On input $\langle \pi, \varphi\rangle$ simulate $M_f(\pi; a)$ for every advice $a \in \Sigma^{\leq c\cdot\log(|\pi|)}$ with $\langle 1^{|\pi|}, a\rangle \notin A$.
- If M_f outputs φ under any a, then output φ as well. □

Input vs. Output Advice

Input advice seems to be stronger

1. Proofs can be quite long.
2. Formulas of the same size typically require proofs of different size.

Therefore: Input advice yields more advice per formula.

Input vs. Output Advice

Input advice seems to be stronger

1. Proofs can be quite long.
2. Formulas of the same size typically require proofs of different size.

Therefore: Input advice yields more advice per formula.

Results supporting this view

1. There exist optimal proof systems with input advice.
2. Such a result is unlikely to hold for output advice.
3. There exist languages with a polynomially bounded proof system with logarithmic input advice, but without such a system with output advice.

Simplifying Advice

We show that

- If advice helps to prove tautologies, then the advice can be simplified.
- If there exists a proof system with advice with nontrivial upper bounds on the proof lengths, then there is such a proof system with output advice.

Simplifying Advice

We show that

- If advice helps to prove tautologies, then the advice can be simplified.
- If there exists a proof system with advice with nontrivial upper bounds on the proof lengths, then there is such a proof system with output advice.

Theorem

Let $t(n) \in 2^{O(\sqrt{n})}$ and assume that there exists a $t(n)$-bounded propositional proof system f with polylogarithmic input advice. Then there exists an $s(n)$-bounded propositional proof system g with polynomial output advice where $s(n) \in O(t(d \cdot n^2))$ with some fixed constant d independent of f.

Idea of the Proof

Definition

For a proof system f, we say that π is a conjunctive f-proof for a tautology φ if there exist tautologies $\psi_1, \ldots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \cdots \wedge \psi_n$ and φ is among the ψ_i.

Idea of the Proof

Definition

For a proof system f, we say that π is a conjunctive f-proof for a tautology φ if there exist tautologies $\psi_1, \ldots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \cdots \wedge \psi_n$ and φ is among the ψ_i.

Sketch of Proof

- We use a new technique of Buhrman & Hitchcock (CCC'08).

Idea of the Proof

Definition

For a proof system f, we say that π is a conjunctive f-proof for a tautology φ if there exist tautologies $\psi_1, \ldots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \cdots \wedge \psi_n$ and φ is among the ψ_i.

Sketch of Proof

- We use a new technique of Buhrman & Hitchcock (CCC'08).
- Let f be the $t(n)$-bounded proof system with logarithmic input advice.

Idea of the Proof

Definition

For a proof system f, we say that π is a conjunctive f-proof for a tautology φ if there exist tautologies $\psi_1, \ldots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \cdots \wedge \psi_n$ and φ is among the ψ_i.

Sketch of Proof

- We use a new technique of Buhrman & Hitchcock (CCC'08).
- Let f be the $t(n)$-bounded proof system with logarithmic input advice.
- Goal: construct a $t(n)$-bounded proof system g with output advice.

Idea of the Proof

Definition

For a proof system f, we say that π is a conjunctive f-proof for a tautology φ if there exist tautologies $\psi_1, \ldots, \psi_n$ with $|\psi_i| = |\varphi| = n$ such that $f(\pi) = \psi_1 \wedge \cdots \wedge \psi_n$ and φ is among the ψ_i.

Sketch of Proof

- We use a new technique of Buhrman & Hitchcock (CCC'08).
- Let f be the $t(n)$-bounded proof system with logarithmic input advice.
- Goal: construct a $t(n)$-bounded proof system g with output advice.
- Idea: g-proofs will be conjunctive f proofs.

More Details of the Proof

- By counting we show that there exists a number $m_n \leq t(d \cdot n^2)$ such that half of all tautologies of length n have a conjunctive f-proof of length m_n.

More Details of the Proof

- By counting we show that there exists a number $m_n \leq t(d \cdot n^2)$ such that half of all tautologies of length n have a conjunctive f-proof of length m_n.
- Iterating this argument, we get proof lengths $m_{n,1}, \ldots, m_{n,p(n)}$ such that every $\varphi \in TAUT^n$ has a conjunctive f-proof of length $\ell \in \{m_{n,1}, \ldots, m_{n,p(n)}\}$.

More Details of the Proof

- By counting we show that there exists a number $m_n \leq t(d \cdot n^2)$ such that half of all tautologies of length n have a conjunctive f-proof of length m_n.
- Iterating this argument, we get proof lengths $m_{n,1}, \ldots, m_{n,p(n)}$ such that every $\varphi \in TAUT^n$ has a conjunctive f-proof of length $\ell \in \{m_{n,1}, \ldots, m_{n,p(n)}\}$.
- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)}) \rangle$$

where $h_f(m)$ is the advice for f on length m.

More Details of the Proof

- By counting we show that there exists a number $m_n \leq t(d \cdot n^2)$ such that half of all tautologies of length n have a conjunctive f-proof of length m_n.
- Iterating this argument, we get proof lengths $m_{n,1}, \ldots, m_{n,p(n)}$ such that every $\varphi \in TAUT^n$ has a conjunctive f-proof of length $\ell \in \{m_{n,1}, \ldots, m_{n,p(n)}\}$.
- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)}) \rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i \rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in TAUT^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.

More Details of the Proof

- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)})\rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in \mathit{TAUT}^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.

More Details of the Proof

- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)})\rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in \mathit{TAUT}^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.
- g is computed by a machine M_g as follows:
  - Input: $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$

More Details of the Proof

- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)})\rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in TAUT^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.
- g is computed by a machine M_g as follows:
  - Input: $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$
  - M_g uses its advice to check whether $|\pi| = m_{n,i}$.

More Details of the Proof

- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)})\rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in \mathit{TAUT}^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.
- g is computed by a machine M_g as follows:
  - Input: $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$
  - M_g uses its advice to check whether $|\pi| = m_{n,i}$.
  - If affirmative, M_g simulates $M_f(\pi)$ using advice $h_f(m_{n,i})$.

More Details of the Proof

- The advice for g for length n will be the sequence

$$\langle m_{n,1}, h_f(m_{n,1}), \ldots, m_{n,p(n)}, h_f(m_{n,p(n)})\rangle$$

  where $h_f(m)$ is the advice for f on length m.
- A g-proof is of the form $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$ where
  - $\varphi, \psi_1, \ldots, \psi_n \in \mathit{TAUT}^n$ and $\varphi \in \{\psi_1, \ldots, \psi_n\}$,
  - π is an f-proof of $\bigwedge_{i=1}^{n} \psi_i$, and
  - i is a number $\leq p(n)$.
- g is computed by a machine M_g as follows:
  - Input: $\langle \varphi, \psi_1, \ldots, \psi_n, \pi, i\rangle$
  - M_g uses its advice to check whether $|\pi| = m_{n,i}$.
  - If affirmative, M_g simulates $M_f(\pi)$ using advice $h_f(m_{n,i})$.
  - If M_f outputs $\bigwedge_{i=1}^{n} \psi_i$, then M_g outputs φ, otherwise the output is undefined. □

Conclusion

Question

Does advice help to prove propositional tautologies?

Conclusion

Question

Does advice help to prove propositional tautologies?

Partial answers

- There exists an optimal proof system with one bit of advice.

Conclusion

Question

Does advice help to prove propositional tautologies?

Partial answers

- There exists an optimal proof system with one bit of advice.
- The set of all languages with polynomially bounded proof systems strictly increases when using advice.

Conclusion

Question

Does advice help to prove propositional tautologies?

Partial answers

- There exists an optimal proof system with one bit of advice.
- The set of all languages with polynomially bounded proof systems strictly increases when using advice.
- We can use weak oracles instead of advice.

Conclusion

Question

Does advice help to prove propositional tautologies?

Partial answers

- There exists an optimal proof system with one bit of advice.
- The set of all languages with polynomially bounded proof systems strictly increases when using advice.
- We can use weak oracles instead of advice.
- If advice helps (i.e. under advice there exist non-trivial upper bounds on the proof size), then we can simplify the advice from input to output advice.