Simulation of Proof Systems

Bachelor's Thesis

Nils Wisiol

Lehrstuhl für Informatik IV
Institut für Informatik
Julius-Maximilians-Universität Würzburg

04. Juni 2012

Outline

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say *h simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say *h simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Optimal Proof Systems

A $\mathcal{FP}$-function h is called *proof system* for a language L, if $f(\Sigma^*) = L$. If $h(w) = x$ holds, we call w a *h-proof* for x.

Let h and h' be proof systems for a language L, and let p be a polynomial and a let f be a function such that for all h'-proofs w it holds that

$$h(f(w)) = h'(w),$$

where $|f(w)| \leq p(|w|)$ holds. In this case, we say h *simulates* the proof system h'.

If a proof system h for a language L simulates every other proof system for the same language L, we call it *optimal*. Let OPT be the complexity class of all languages possessing a optimal proof system.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path.} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i\rangle)$ be the function defined by

$$f(\langle x, i\rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- $P \subseteq OPT$: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- $NP \subseteq OPT$: For any $L \in NP$, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i\rangle)$ be the function defined by

$$f(\langle x, i\rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- P $\subseteq$ OPT: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- NP $\subseteq$ OPT: For any $L \in$ NP, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i \rangle)$ be the function defined by

$$f(\langle x, i \rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages with Optimal Proof Systems

- $\mathrm{P} \subseteq \mathrm{OPT}$: $L \in P$ has a proof system f: choose $f : \Sigma^* \to L$ with $f(x) = x$ if $x \in L$, otherwise f is undefined. For any other proof system g of L, g itself translates g-proofs into h-proofs in polynomial time.
- $\mathrm{NP} \subseteq \mathrm{OPT}$: For any $L \in \mathrm{NP}$, there is a nondet. TM that accepts L in poly. time. Let $f(\langle x, i\rangle)$ be the function defined by

$$f(\langle x, i\rangle) = \begin{cases} x & \text{if } M \text{ accepts } x \text{ on the } i\text{-th path,} \\ \perp & \text{otherwise.} \end{cases}$$

  f can be calculated in polynomial time and f has range L. Hence, f is a proof system for L. Let g be an arbitrary proof system for L. Then g itself translates g-proofs into f-proofs. Hence, f is an optimal proof system for L.

Languages without Optimal Proof Systems

Theorem

There exists a language $L \in$ co-$NTIME(2^n)$, that does not possess an optimal proof system.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 10^*$
   Let L_i' be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L_i' \in$ co-NTIME(2^n)
3. We will show that for L-proof-systems f_i it holds $L_i' = L_i$. As a consequence, there are only long f_i-proofs for $L_i' \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 10^*$
   Let L_i' be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L_i' \in \text{co-NTIME}(2^n)$
3. We will show that for L-proof-systems f_i it holds $L_i' = L_i$. As a consequence, there are only long f_i-proofs for $L_i' \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 10^*$
   Let L'_i be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L'_i \in \text{co-NTIME}(2^n)$
3. We will show that for L-proof-systems f_i it holds $L'_i = L_i$. As a consequence, there are only long f_i-proofs for $L'_i \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 10^*$
   Let L'_i be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L'_i \in$ co-NTIME(2^n)
3. We will show that for L-proof-systems f_i it holds $L'_i = L_i$. As a consequence, there are only long f_i-proofs for $L'_i \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 10^*$
   Let L_i' be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L_i' \in \text{co-NTIME}(2^n)$
3. We will show that for L-proof-systems f_i it holds $L_i' = L_i$. As a consequence, there are only long f_i-proofs for $L_i' \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Proof Overview

1. Let $f_1, f_2, \ldots$ be an enumeration of all polynomial time functions
2. $L_i = 0^i 1 0^*$
   Let L'_i be the language of all strings L_i without any "short" f_i-proof
   $L = \bigcup_i L'_i \in \text{co-NTIME}(2^n)$
3. We will show that for L-proof-systems f_i it holds $L'_i = L_i$. As a consequence, there are only long f_i-proofs for $L'_i \subset L$
4. This will contradict to the assumption, that f_i is an optimal proof system for L.

Enumerating $\mathcal{FP}$

- Gödel: $M_1, M_2, \ldots$
- We define $M'_1, M'_2, \ldots$ as the M_i with a clock that stops the calculation after $n^i + i$ steps
- Let f_i the function calculated by M_i
- As for unbounded i, the runtime $n^i + i$ is unbounded, we obtain all $\mathcal{FP}$ functions

Kurt Gödel
1906 – 1978

Enumerating $\mathcal{FP}$

- Gödel: $M_1, M_2, \ldots$
- We define $M'_1, M'_2, \ldots$ as the M_i with a clock that stops the calculation after $n^i + i$ steps
- Let f_i the function calculated by M_i
- As for unbounded i, the runtime $n^i + i$ is unbounded, we obtain all $\mathcal{FP}$ functions

Kurt Gödel
1906 – 1978

Enumerating $\mathcal{FP}$

- Gödel: $M_1, M_2, \ldots$
- We define $M'_1, M'_2, \ldots$ as the M_i with a clock that stops the calculation after $n^i + i$ steps
- Let f_i the function calculated by M_i
- As for unbounded i, the runtime $n^i + i$ is unbounded, we obtain all $\mathcal{FP}$ functions

Kurt Gödel
1906 – 1978

Enumerating $\mathcal{FP}$

- Gödel: $M_1, M_2, \ldots$
- We define $M'_1, M'_2, \ldots$ as the M_i with a clock that stops the calculation after $n^i + i$ steps
- Let f_i the function calculated by M_i
- As for unbounded i, the runtime $n^i + i$ is unbounded, we obtain all $\mathcal{FP}$ functions

Kurt Gödel
1906 – 1978

Enumerating $\mathcal{FP}$

- Gödel: $M_1, M_2, \ldots$
- We define $M'_1, M'_2, \ldots$ as the M_i with a clock that stops the calculation after $n^i + i$ steps
- Let f_i the function calculated by M_i
- As for unbounded i, the runtime $n^i + i$ is unbounded, we obtain all $\mathcal{FP}$ functions

Kurt Gödel
1906 – 1978

Construction of L

- $L_i = 0^i 10^*$
- take $x \in L'_i$, that do not have f_i-proofs

$$L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$$

- union

$$L = \bigcup_{i > 0} L'_i$$

Construction of L

- $L_i = 0^i 10^*$
- take $x \in L_i'$, that do not have f_i-proofs

$$L_i' = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$$

- union

$$L = \bigcup_{i>0} L_i'$$

Construction of L

- $L_i = 0^i 10^*$
- take $x \in L'_i$, that do not have f_i-proofs

$$L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$$

- union

$$L = \bigcup_{i>0} L'_i$$

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME$(2^n) \Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME$(2^n) \Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME$(2^n) \Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME$(2^n) \Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y\in\Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

L is member of co-NTIME(2^n)

- As $L \in$ co-NTIME(2^n) $\Leftrightarrow \overline{L} \in$ NTIME(2^n), we will analyze the complexity of $\overline{L}$.
- $\overline{L} = \overline{\bigcup_{i>0} L'_i} = \bigcap_{i>0} \overline{L'_i}$
- claim: $\bigcap_{i>0} \overline{L'_i} \in$ NTIME(2^n)

$$\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} \left(|y|^{2i} \leq 2^{|x|}\right) \wedge (f_i(y) = x)\right)\}$$

Let x be an arbitrary string.

- Check, if x is member of any L_i: if not, then $x \in \overline{L}$
- Otherwise, choose i^* such that $x \in L_{i^*}$
- $x \in \overline{L'_j}$ for any $j \neq i$
- for any y such that $|y|^{2i} \leq 2^{|x|}$: calculate $f_{i^*}(y)$. If and only if there is a y such that $f_{i^*}(y) = x$, then $x \in \overline{L}$.

A Property for Proof Systems of L

Recall $L_i' = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L_i'} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L_i'$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1z \in L_i$ that is not a member of L_i'
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L_i'$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

A Property for Proof Systems of L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$
and $\overline{L'_i} = \{x \in \Sigma^* : x \notin L_i \vee \left(\exists_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \wedge f_i(y) = x\right)\}$

- any proof system of L is one of the f_i
- claim: for any $f_i(\Sigma^*) = L$ it holds $L_i = L'_i$

- let f_i be a proof system for L
- assume there is a $x = 0^i 1 z \in L_i$ that is not a member of L'_i
- then, there is a y such that $y^{2i} \leq 2^{|x|}$ and $f_i(y) = x$
- Hence, y is a f_i-proof for x. It follows that $x \in L$, and therefore $x \in L'_i$. This is a contradiction.

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds
  $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction □

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L_i' = \{x \in L_i : \forall_{y\in\Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b=0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L_i') \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L_i', such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L_i': $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i 10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L_i' = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i 10^* \in L_i = L_i') \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L_i', such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L_i': $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y\in\Sigma^*} |y|^{2i} \le 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i 10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \le p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \le 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds
  $|y| = |f^*(1x)| \le p(|1x|) \le p(|1x|)^{2i} \le 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L_i' = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i 10^* \in L_i = L_i') \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L_i', such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds
  $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L_i': $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y\in\Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

These results contradict the existence of an optimal proof system for L

Recall $L'_i = \{x \in L_i : \forall_{y \in \Sigma^*} |y|^{2i} \leq 2^{|x|} \implies f_i(y) \neq x\}$.

- Let f_i be an optimal proof system for L
- Let g be defined by
  $$g(bx) = \begin{cases} f_i(x) & (b = 0) \\ x & (b = 1 \text{ and } x = 0^i10^* \in L_i = L'_i) \end{cases}$$
- g is a proof system for L
- as f_i is optimal, there is a f^*, such that $f_i(f^*(x)) = g(x)$. f^* is polynomial bounded: $|f^*(x)| \leq p(|x|)$
- for any x of L'_i, such that $p(|1x|)^{2i} \leq 2^{|x|}$.
- $1x$ is a g-proof for x: $g(1x) = x$
- For $y = f^*(1x)$ it holds
  $|y| = |f^*(1x)| \leq p(|1x|) \leq p(|1x|)^{2i} \leq 2^{|x|}$.
- By Definition of L'_i: $f_i(y) \neq x$, hence $y = f^*(1x)$ is no f_i-proof for x
- This is a contradiction $\square$

Consequences

What do we know about L?

Theorem

If $L \subseteq 0^*10^*$ has no optimal proof system, then there is a polynomial time equivalent $T \in TALLY$, that does not possess an optimal proof system.

Corollary

Let $u : \mathbb{N} \to \mathbb{N}$ be a strictly increasing polynomial time function. Then there is a language $L \in co\text{-}NTIME(2^n)$, that has no optimal proof system. Additionally, every length of a string in L is in the range of u.

Theorem

For every language L that possesses no optimal proof system there is a polynomial time equivalent, many-one-mitotic language, that does not possess an optimal proof system.

Consequences

What do we know about L?

Theorem

If $L \subseteq 0^*10^*$ has no optimal proof system, then there is a polynomial time equivalent $T \in TALLY$, that does not possess an optimal proof system.

Corollary

Let $u : \mathbb{N} \rightarrow \mathbb{N}$ be a strictly increasing polynomial time function. Then there is a language $L \in co\text{-}NTIME(2^n)$, that has no optimal proof system. Additionally, every length of a string in L is in the range of u.

Theorem

For every language L that possesses no optimal proof system there is a polynomial time equivalent, many-one-mitotic language, that does not possess an optimal proof system.

Consequences

What do we know about L?

Theorem

If $L \subseteq 0^*10^*$ has no optimal proof system, then there is a polynomial time equivalent $T \in TALLY$, that does not possess an optimal proof system.

Corollary

Let $u : \mathbb{N} \to \mathbb{N}$ be a strictly increasing polynomial time function. Then there is a language $L \in$ co-$NTIME(2^n)$, that has no optimal proof system. Additionally, every length of a string in L is in the range of u.

Theorem

For every language L that possesses no optimal proof system there is a polynomial time equivalent, many-one-mitotic language, that does not possess an optimal proof system.

Consequences

What do we know about L?

Theorem

If $L \subseteq 0^*10^*$ has no optimal proof system, then there is a polynomial time equivalent $T \in TALLY$, that does not possess an optimal proof system.

Corollary

Let $u : \mathbb{N} \to \mathbb{N}$ be a strictly increasing polynomial time function. Then there is a language $L \in$ co-$NTIME(2^n)$, that has no optimal proof system. Additionally, every length of a string in L is in the range of u.

Theorem

For every language L that possesses no optimal proof system there is a polynomial time equivalent, many-one-mitotic language, that does not possess an optimal proof system.

Consequences

What do we know about L?

Theorem

If $L \subseteq 0^*10^*$ has no optimal proof system, then there is a polynomial time equivalent $T \in TALLY$, that does not possess an optimal proof system.

Corollary

Let $u : \mathbb{N} \to \mathbb{N}$ be a strictly increasing polynomial time function. Then there is a language $L \in$ co-$NTIME(2^n)$, that has no optimal proof system. Additionally, every length of a string in L is in the range of u.

Theorem

For every language L that possesses no optimal proof system there is a polynomial time equivalent, many-one-mitotic language, that does not possess an optimal proof system.

Literature I

[AB09] Sanjeev Arora and Boaz Barak, *Computational complexity: A modern approach*, 1st ed., Cambridge University Press, New York, NY, USA, 2009.

[Coo71] Stephen A. Cook, *The complexity of theorem-proving procedures*, Proceedings of the third annual ACM symposium on Theory of computing (New York, NY, USA), STOC '71, ACM, 1971, pp. 151–158.

[CR79] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic **44** (1979), 36–50.

[For09] Lance Fortnow, *The status of the P versus NP problem*, Commun. ACM **52** (2009), no. 9, 78–86.

Literature II

[KM00] Johannes Köbler and Jochen Messner, *Is the standard proof system for SAT p-optimal?*, Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (London, UK, UK), FST TCS 2000, Springer-Verlag, 2000, pp. 361–372.

[KMT03] Johannes Köbler, Jochen Messner, and Jacobo Torán, *Optimal proof systems imply complete sets for promise classes*, Inf. Comput. **184** (2003), no. 1, 71–92.

[KP89] Jan Krajícek and Pavel Pudlák, *Propositional proof systems, the consistency of first order theories and the complexity of computations*, J. Symb. Log. **54** (1989), no. 3, 1063–1079.

Literature III

[Mes99] Jochen Messner, *On optimal algorithms and optimal proof systems*, Proceedings of the 16th annual conference on Theoretical aspects of computer science (Berlin, Heidelberg), STACS'99, Springer-Verlag, 1999, pp. 541–550.

[Pap94] Christos H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.

Images Sources I

- Kurt Gödel: Wikipedia, `de.wikipedia.org`