Computability & Complexity

(lecturer: Moritz Müller)

<u>Content</u>: Optimal algorithms; optimal proof systems & NP versus coNP

SAT	<u>Input</u>: propositional formula α
	<u>Problem</u>: Is α satisfiable?

We know (COOK) SAT is NP-complete

$P = NP \iff$ SAT is ptime decidable

Assume $P \neq NP$. How fast can you solve SAT?

- <u>Clear</u>: $SAT \in E$ time $2^{O(n)}$
- in time $2^{o(n)}$? $\longrightarrow$ <u>No</u>, if ETH holds (exponential time hyp).
- even assuming ETH, you can be fast.
  - exists an alg A_0 for SAT which is ptime on all 2 CNF-s.
  - exists an alg A_1 for SAT ptime on Horn frms.

    CNF, in each disj there is a positive literal,
  - exists an alg A_2 for SAT ptime on frms of bounded-tree width.

    ⋮

- eg: run A_0 and A_1 in parallel: This alg is fast on 2CNF ∪ HORN

<u>but</u> this list is <u>not finite</u>

Q. • Is there A an "optimal algorithm" for SAT ? OPEN

for all alg $\mathbb{B}$ for SAT for all α

$$(\text{time of } \mathbb{A} \text{ on } \alpha) \leq (\text{time of } \mathbb{B} \text{ on } \alpha)^{O(1)}$$

faster than all the others up to a polynom.

Let $\mathbb{A}$ be an algorithm for SAT

Assume $P \neq NP$

Then $\exists \ (x_s)_{s \in \mathbb{N}}$ st (the time of $\mathbb{A}$ on x_s) not $|x_s|^{O(1)}$

Q: Can you compute, in Ptime, such a sequence $(x_s)_s$?

No, if $\mathbb{A}$ is an optimal algorithm.

$\boxed{\text{Else ?}}$

"witnessing failure".

• $P = NP \iff$ there exists a Ptime SAT-solver

i.e. decides SAT
and, additionally, on sat frm α
returns a sat. assignment.

Levin: There exists an "optimal" SAT-solver.

TAUT : Input :	Prop. frm α
Problem :	Is α valid.

TAUT is CoNP-complete.

(Proof systems)

$\mathbb{P}$: Hilbert type proof system.

$\mathbb{P}'$: Gentzen type sequent calculus.

• every $\mathbb{P}'$ proof can be rewritten into $\mathbb{P}$-proof in ptime

It is not true if $\mathbb{P}' =$ Resolution

Q: Does there exist an "optimal" proof system?

In other words: (!!) Does there exist an optimal nondeterministic algorithm for TAUT?

Contents

0. Background: complexity theory

model of computation: multi-tape Turing Maschine (T.M) (deterministic & nondeterm).

alphabet: $\{0,1\}$.

Problems: $Q \subseteq \{0,1\}^*$

$\mathbb{A}, \mathbb{B}, \dots$ ← used for T.M

Def (T.M) (deterministic)

$\mathbb{A} = ($ finite set of states, transition function $\delta)$

- k tapes, each with cells numbered $0, 1, \dots$
  - one side infinite
- cells contain a symbol (0 or 1) or nothing $\square$ "blank"
- 1^{st} cells always contain $\S$

$$\delta : S \times \{0,1,\square,\S\}^k \longrightarrow S' \times \{\S,0,1,\square\}^k \times \{+1, 0, -1\}^k$$

moves: right, stay, left

$$(s, b_1 \dots b_k) \xmapsto{\delta} (s', b_1' \dots b_k', m_1 \dots m_k)$$

A on $x = x_1 \dots x_n \in \{0,1\}^n$,

- starts with state $s_{START} \in S$
- x_i is in cell i on tape 1,
- heads on the first cells.

- the computation stops when $s_{halt} \in S$ is reached
- output: are the bits on tape k up to the first $\square$

A accepts x, if output $= 1$

A rejects x, if output $= 0$

A decides Q, if accepts $x \in Q$ and rejects $x \notin Q$.

(in part, always halts)

- number of steps: $t_A(x)$

$t_A(x) = \infty$ if A on x does not halt.

- A decides Q in ptime, if $t_A(x) \leq poly(|x|)$

$$\boxed{\mathbb{P} := \text{class of all ptime decidable problems.}}$$

Def: nondeterministic TM $A = (S, \delta_0, \delta_1)$

$x \in \{0,1\}^+$

$y \in$

y determines a partial run of A on x

namely: i^{th} step is done with δ_{y_i}

A accepts x if $\exists y \in \{0,1\}^+$

y determines a (complete) accepting run of A on x

$t_A(x) = \min \{ \ell \in \mathbb{N} \mid \exists y \in \{0,1\}^{\ell} :$ y determines an accepting run of A on $x \}$

(convention: $\min \emptyset = \infty$.)

- A accepts Q iff $Q = \{x \in \{0,1\}^* \mid t_A(x) < \infty\}$.
- A accepts Q in ptime iff $t_A(x) \le poly(|x|)$ for all $x \in Q$

$\boxed{NP := \text{class of all ptime acceptable problems}}$

$\boxed{\text{Exercise}}$ Q is c.e. (computably enumerable / recursively enumerable)

$\Updownarrow$

exists (det) A st $Q = \{x \mid t_A(x) < \infty\}$

$\Updownarrow$

exists a nondet A st Q is accepted by A.

Def: A reduction from Q to Q' is

$r : \{0,1\}^* \longrightarrow \{0,1\}^*$ s.t.

for all $x \in \{0,1\}^*$: $\boxed{x \in Q \iff r(x) \in Q'}$

- $\boxed{Q \le_p Q'} \iff$ exists a ptime comp. such reduction

"polynomial time reducible"

- for a complexity class $\mathcal{C} \subseteq \mathcal{P}(\{0,1\}^*)$, $Q \subseteq \{0,1\}^*$ a pb
  - Q is hard for $\mathcal{C}$ (under $\le_p$) iff for all $Q' \in \mathcal{C} : (Q' \le_p Q)$
  - Q is complete for $\mathcal{C}$ iff additionally $Q \in \mathcal{C}$.

Then $\mathcal{C} = \{Q' \mid Q' \le_p Q\}$ provided $\le_p$-closed. (i.e. $Q' \le_p Q \in \mathcal{C} \Rightarrow Q' \in \mathcal{C}$)

$$\boxed{\begin{aligned} coNP &= \text{complements of problems in } NP \\ &= \{ \{0,1\}^* \setminus Q \mid Q \in NP \} \end{aligned}}$$

Note : Q is NP-complete $\Longleftrightarrow$ $\{0,1\}^* \setminus Q$ coNP-complete

(a reduction from one problem to another is also a reduction betw. the complements)

$\boxed{\text{COOK'S THM}}$: SAT is NP-complete.

Corollary : TAUT is coNP-complete.

Pf : it is easy to show TAUT $\equiv_p \{0,1\}^* \setminus$ SAT and apply the note. $\boxed{\Xi}$

Ξ propositional formulas :

- build by using $\neg, \wedge, \vee$ from atoms
- atoms are variables $X, Y, \ldots$, constants $0, 1$.

- assignment A : variables $\longrightarrow \{0,1\}$.
- α is satisfiable if exists A : $A \models \alpha$
- α is valid if forall A : $A \models \alpha$.

- k-CNF $\equiv$ conjunction of disjunction of $\leq k$. literals ($\equiv$ atoms or neg. of atoms)

k-SAT : Input : k-CNF α
Problem : Is α satisfiable ?

$$\begin{cases} \text{is NP complete}, & k \geq 3 \\ \in P, & \text{otw} \end{cases}$$

17.03.2014

$\boxed{0.}$ Background – Complexity theory

– recall (last time): TM A Turing maschine
$t_A(x)$ — time of A on x.
P, NP, c.e. problems;
propositional logic: SAT, TAUT.

$\boxed{\text{Thm (Cook)}}$ SAT is NP-complete

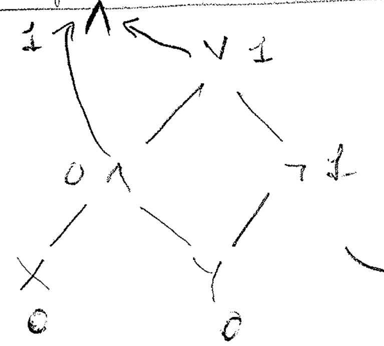
$\boxed{\text{Fundamental Lemma}}$: Let A ptime TM.
Given $1^n = \underbrace{1 \ldots\ldots 1}_{n\text{ times}}$ as an input, one can compute in time $n^{O(1)}$ a circuit $C_n(X_1 \ldots X_n)$ st for all $x = x_1 x_2 \ldots x_n \in \{0,1\}^n$:

$$A \text{ accepts } x \iff C_n(x_1, \ldots, x_n) = 1$$

A <u>circuit</u> C is a directed acyclic graph (V, E) ("gates") with 1 <u>output vertex</u> (fan-out 0), with an <u>ordering</u> $<$ on V, a <u>labeling</u> $\lambda : V \to \{\neg, \wedge, \vee, 0, 1\} \cup Var$
st. $\lambda^{-1}(\neg)$ are vertices with fan-in 1
$\lambda^{-1}(\wedge) \cup \lambda^{-1}(\vee)$ are vertices with fan-in 2
others have fan-in 0 ($\leftarrow$ input vertex labeled as var or constants)

Writing $C(\overline{X})$, means that C has variables $\overline{X}$ in order $<$

Example: computation



$$(\underline{X \wedge Y}) \wedge ((\underline{X \wedge Y}) \vee \neg Y)$$

exponential blow-up.

Proof (sketch) of Cook's Theorem

$Q \in NP$, accepted by nondeterministic TM A in time n^c

$\tilde{Q} := \{ \langle x, y \rangle \mid y \in \{0,1\}^{n^c}$ determines an accepting run of A on x $\}$.

(where
$\langle x, y \rangle$ encodes the pair
for example, such an encoding could be
$x_1 x_1 x_2 x_2 \dots x_{|x|} x_{|x|} 01 y_1 y_1 y_2 y_2 \dots y_{|y|} y_{|y|}$)

Compute $C(X_1 \dots X_n, Y_1 \dots Y_{n^c})$ such that
for all $x \in \{0,1\}^n$, $y \in \{0,1\}^{n^c}$

$$\boxed{C(x_1 \dots x_n, y_1 \dots y_{n^c}) = 1 \iff \langle x, y \rangle \in \tilde{Q}}$$

$x \in Q \iff C(x, \overline{Y})$ is satisfiable

reduction outputs a fm $\alpha_{C(x,\overline{Y})}(\overline{Y}, \overline{Z})$

$\boxtimes$

Lemma: From a circuit $C(\overline{X})$, one can compute in ptime a fm $\alpha(\overline{X}, \overline{Z})$ st

a) for all assignments A to $\overline{X}$
there is at most one assignment B to $\overline{Z}$
st $A \cup B \models \alpha$

b) if $C(A(X_1), A(X_2) \dots) = 1$, then there is an assignment B to $\overline{Z}$ st.
$A \cup B \models \alpha(\overline{X}, \overline{Z})$

Pf (Sketch)

Let $C = (V, E, <, \lambda)$, $C = C(\overline{X})$ a circuit

Use auxiliary variables. $Z_g : g \in V$

Set $\alpha_C(\overline{X}, \overline{Z})$ to be

$$Z_{\text{output node}} \wedge \bigwedge_{g \in V} \begin{cases} (Z_g \leftrightarrow \lambda(g)) & \text{if } g \text{ is an input} \\ (Z_g \leftrightarrow \neg Z_{g'}) & \text{if } (g', g) \in E \ \& \ \lambda(g) = \neg \\ (Z_g \leftrightarrow Z_{g'} \wedge Z_{g''}) & \text{if } (g, g'), (g, g'') \in E \text{ and } \lambda(g) = \wedge \\ \quad -\text{''}- \quad \vee \quad -\text{''}- & \vee \end{cases}$$

☒

1. Optimal algorithms

Q decidable, nonempty problem.

Defn $\mathbb{A}$ is (almost) optimal algorithm for Q

if $\mathbb{A}$ decides Q and for every $\mathbb{B}$ deciding Q, there exists a polynomial p st for all $x \in \{0,1\}^*$ $(x \in Q)$: $t_{\mathbb{A}}(x) \le p(t_{\mathbb{B}}(x) + |x|)$

the polynomial makes the defn. machine independent

at least to read the input

Proposition: Asm $\mathbb{A}$ is almost optimal for Q. If $\tilde{Q} \subseteq Q$, $\tilde{Q} \in P$, then there exist a polynomial q such that for all $x \in \tilde{Q}$

$$t_{\mathbb{A}}(x) \le q(t_{\mathbb{B}}(x) + |x|)$$

alg deciding Q

Proof: Let $\tilde{Q} \subseteq Q$, $\tilde{Q} \in P$
Let $\tilde{\mathbb{B}}$ decide $\tilde{Q}$ in time q (polynomial

Define $\mathbb{B}$ on x:

(1) if $\tilde{\mathbb{B}}$ accepts x then accept

(2) else run $\mathbb{Q}$ on x

($\mathbb{Q}$ is some arb. alg. deciding Q.

So, $\mathbb{B}$ decides Q.

Then there is a constant $c \in \mathbb{N}$ st

$$t_{\mathbb{A}}(x) \leq \left(t_{\mathbb{B}}(x) + |x|\right)^{c} \qquad \text{for } x \in Q.$$

In part, for $x \in \tilde{Q} \subseteq Q$ then $t_{\mathbb{B}}(x) \leq q(|x|)$

Then $$t_{\mathbb{A}}(x) \leq \left(q(|x|) + |x|\right)^{c}$$

$\boxtimes$.

Example: If $Q \in P$ and $\mathbb{A}$ is a ptime alg for Q then $\mathbb{A}$ is (almost) optimal.

Proposition: There exists a problem $Q \in E \setminus P$ with optimal algorithms

we use: Proposition: There exists a $Q \in E$ ($\setminus P$) which is $\mathrm{TIME}(2^n)$-bi-immune

ie for all infinite $X \subseteq Q$: $X \notin \mathrm{TIME}(2^n)$
& for all infinite $X \subseteq \{0,1\}^* \setminus Q$: $X \notin \mathrm{TIME}(2^n)$

! recall: $E = \bigcup_{c \in \mathbb{N}} \mathrm{TIME}(2^{c \cdot n})$

Pf $\mathbb{A}$ decides Q in time $t_{\mathbb{A}}(x) \leq c \cdot 2^{n \cdot c}$.
Q $\mathrm{TIME}(2^n)$-bi-immune.
then $\mathbb{A}$ is optimal

otherwise, there is $\mathbb{B}$ for Q st for all $i \in \mathbb{N}$
there exists $x_i \in \{0,1\}^*$ st

$$t_{\mathbb{A}}(x_i) > \left(t_{\mathbb{B}}(x_i) + |x_i|\right)^{i}$$

(not almost optimal)

Hence, $$c \cdot 2^{c \cdot |x_i|} > \left(t_{\mathbb{B}}(x_i) + |x_i|\right)^{i}$$

hence: $t_{\mathbb{B}}(x_i) < 2^{|x_i|/2}$ for $i \geq i_0 \in \mathbb{N}$ suitable but $\{x_i : i \geq i_0\} \cap Q$ OR

$\{x_i : i \geq i_0\} \cap (\{0,1\}^* \setminus Q)$ is <u>infinite</u>.

<u>Define $\mathbb{C}$ on x</u> :

compute $2^{|x|/2}$

simulate $\mathbb{B}$ on x for $\leq 2^{|x|/2}$ steps

If simulation <u>halts</u> and <u>accepts</u>, then <u>accept</u>.
(resp rejects)

else <u>reject</u>.

Then $\mathbb{C}$ accepts an infinite subs of Q
(respectively $\{0,1\}^* \setminus Q$)

in time $O(2^{|x|/2}) \leq O(2^{|x|})$ $\boxed{\lightning}$

(? it seems we could delete $\cdot/2$, correct but not needed)

<u>Proposition</u> ~~There~~ exists $Q \in E(\setminus P)$ st

Q does not have an (almost) optimal algorithm.

Indeed, there is a ptime fct
speed : $\{0,1\}^* \to \{0,1\}^*$ st.

for all $\mathbb{A}$ deciding Q,

also speed($\mathbb{A}$) is an alg. deciding Q

and there is no polynomial p st

for all $x \in Q$ $\quad t_{\mathbb{A}}(x) \leq p(t_{speed(\mathbb{A})}(x) + |x|)$

<u>Proof</u> $Q := \{\mathbb{A} \mid \mathbb{A}$ does not accept $\mathbb{A}$ in $\leq 2^{|\mathbb{A}|}$ steps $\}$.

($Q \in E$)

(the proof is very similar to the time hierarchy theorem)
– will be done next time.

24.03.2014

recall: ① Optimal algorithms.

Prop: $\mathbb{A}$ almost optimal alg for Q, $\tilde{Q} \subseteq Q$, $\tilde{Q} \in P \Rightarrow$

$\Rightarrow t_{\mathbb{A}}(x) \leq |x|^{O(1)}$ for all $x \in \tilde{Q}$

Prop: There are $Q \in E \setminus P$ with optimal algorithms.

Prop (*) There are $Q \in E \setminus P$ without almost optimal alg.

Proof (of Prop(*))

(last time): INDEED (Claim) $\exists$ speed: $\{0,1\}^* \longrightarrow \{0,1\}^*$ ptime fct st for all $\mathbb{A}$ for Q, speed($\mathbb{A}$) decides Q and there is no polynomial p st for all $x \in Q$. $t_{\mathbb{A}}(x) \leq p\left(t_{\text{speed}(\mathbb{A})}(x) + |x|\right)$

Claim $\Rightarrow$ Prop(*)

Proof (of Claim)

$Q := \left\{ \mathbb{A} \;/\; \mathbb{A} \text{ does not accept itself in} \leq 2^{|\mathbb{A}|} \text{ steps} \right\}$

Then $Q \in E$.

For $n \in \mathbb{N}$ let $\mathbb{A}_n$:= "$\mathbb{A}$ plus n useless states"

More precisely: $n \longmapsto \mathbb{A}_n$, ptime st:

$$|\mathbb{A}_n| \geq n \quad \& \quad t_{\mathbb{A}} = t_{\mathbb{A}_n} \; \& \; L(\mathbb{A}) = L(\mathbb{A}_n)$$

where $L(\mathbb{A})$ = " the language accepted by $\mathbb{A}$

$$L(\mathbb{A}) = \{ x \in \{0,1\}^* / \mathbb{A} \text{ accepts } x \}.$$

Note: if $\mathbb{B}$ decides Q, then $\mathbb{B}$ does not reject itself.

So, $\mathbb{B} \in Q$. and $t_{\mathbb{B}}(\mathbb{B}) > 2^{|\mathbb{B}|}$

Hence: $t_{\mathbb{A}}(\mathbb{A}_n) = t_{\mathbb{A}_n}(\mathbb{A}_n) > 2^{|\mathbb{A}_n|} \geq 2^n$
if $\mathbb{A}$ decides Q.

speed($\mathbb{A}$) is the algorithm:

on x, test whether $x \in \{\mathbb{A}_i : i \in \mathbb{N}\}$

(is ptime; compute $\mathbb{A}_0, \mathbb{A}_1, \dots, \mathbb{A}_{|x|+1}$

later: to long to be ... to x)

if $\boxed{\text{yes}}$ then accept.

else run $\mathbb{A}$ on x

If $\mathbb{A}$ decides Q, so does speed($\mathbb{A}$)

and $t_{\text{speed}(\mathbb{A})}(\mathbb{A}_n) \leq |\mathbb{A}_n|^{O(1)}$

§2. HARD SEQUENCES

Defn: Let $\mathbb{A}$ decide a problem Q.

$(x_s)_{s \in \mathbb{N}}$ is a hard sequence for $\mathbb{A}$ if

- ($\cdot$) $x_s \in Q$ for all $s \in \mathbb{N}$
- ($\cdot$) $1^s \longmapsto x_s$ is ptime.
- ($\cdot$) $t_{\mathbb{A}}(x_s)$ is not polynomially bounded in s.

Prop $\mathbb{A}$ decides Q. If $(x_s)_s$ is a hard sequence for $\mathbb{A}$ then $\mathbb{A}$ is not almost optimal.

$\boxed{\text{Remark}}$ Prop is CLEAR if $\{x_s : s \in \mathbb{N}\} \in P$ (an earlier proposition)

Hence if $(x_s)_s$ is HONEST, (i.e. $s \leq |x_s|^{O(1)}$)

given x, compute $x_0, x_1, \dots, x_{s^*}$

$s^* = |x|^c$, $c \in \mathbb{N}$ st $s \leq |x_s|^c$

Example: $r_0 := 0$, $r_{i+1} = 2^{r_i}$; $Q = \{1^{r_i} : i \in \mathbb{N}\}$.

Then $Q \in P$.

Let $(x_s)_s$ be a seq st for all $s \in \mathbb{N}$ $\to x_s \in Q$.

$\to s \leq |x_s|^c$

$\Rightarrow r_i + 1 \leqslant |x_{r_i+1}|^c$

$\Rightarrow 2^{r_i} \leq |x_{r_i+1}|^c$

$\Rightarrow 2^{r_i/c} \leq |x_{r_i+1}| \Rightarrow |x_s|$ is not $s^{O(1)}$

So, $(x_s)_s$ is $\boxed{\text{not ptime.}}$

A <u>not almost optimal</u> (ie. A not ptime)

Let $x_s := 1^{r_i}$ for $r_i \leq s < r_{i+1}$. ⟵ (⋆): hard seq.

Since A is not ptime $\Rightarrow$ for all $i \in \mathbb{N}$, exists $y_i \in Q$

$$t_A(y_i) \geq |y_i|^i$$

write $y_i = 1^{r_{i_j}} = x_{r_{i_j}}$

$$\Rightarrow t_A(x_{r_{i_j}}) > |r_{i_j}|^i$$

Hence $t_A(x_s)$ is not $s^{O(1)}$, so $(x_s)_s$ is a hard seq for A.

So, A not alm. optimal $\Rightarrow$ A has a hard seq.

Proof of the proposition

(recall: <u>Prop</u>: A decides Q, if $(x_s)_s$ is a hard seq for A then A is not almost optimal)

Let G compute $1^s \longmapsto x_s$.

Diagonalization

G^* on x:

1. $\ell \leftarrow 0$
2. for $s = 0, \ldots, \ell$ do.
3. simulate $(\ell - s + 1)^{th}$ step of G on 1^s
4. if halts, <u>output x</u>, then <u>accept</u>
5. $\ell \leftarrow \ell + 1$
6. goto line 2.

Then $L(G^*) = \{ x_s : s \in \mathbb{N} \}$ and $t_{G^*}(x_s) \leq s^{O(1)}$

Let $\mathbb{A} \parallel G^*$ on x :

run $\mathbb{A}$ and G on x in parallel,

halt when one of them does and

then answer accordingly

Then $t_{\mathbb{A} \parallel G^*}(x_s) \leq s^{O(1)}$

and $\mathbb{A} \parallel G^*$ decides Q.

But $t_{\mathbb{A}}(x_s)$ is not $s^{O(1)}$

as $|x_s| \leq s^{O(1)}$, not $\left(t_{\mathbb{A} \parallel G^*}(x_s) + |x_s|\right)^{O(1)}$

Hence $\mathbb{A}$ is not almost optimal $\blacksquare$

THEOREM: If Q is coNP-complete, then tfae:

(1) Q does not have an almost optimal algorithm

(2) Every algorithm deciding Q has a hard seq.

SOUND(Q)

Input	algorithm $\mathbb{A}$; 1^s for $s \in \mathbb{N}$
Problem	$\{ x \in \{0,1\}^{\leq s} \mid \mathbb{A}$ accepts x in $\leq s$ steps $\} \subseteq Q$?

Lemma 1 (1^{st} sufficient condition)

If $\langle \mathbb{A}, 1^s \rangle \overset{?}{\in}$ SOUND(Q) is decidable in time $s^{f(\mathbb{A})}$ for some function $f: \{0,1\}^* \to \mathbb{N}$,

then Q has an almost optimal algorithm.

Proof: Let $\mathbb{V}$ decide SOUND(Q) in $s^{f(A)}$

Let $\mathbb{Q}$ decide Q.

Let $\mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_2 \ldots$ be an effective enumeration of all algorithms

$\mathbb{A}$ on x :

run $\mathbb{Q}$ on x and, in parallel, do:

for $i \le |x|$, do in parallel :

simulate $\mathbb{A}_i$ on x

if accepts, then

$s \longleftarrow \max\{|x|, t_{\mathbb{A}_i}(x)\}$

if $\mathbb{V}$ accepts $\langle \mathbb{A}_i, 1^s \rangle$

then accept $\quad$ LINE $\circledast$

else never halt

else never halt.

if Q halts first, answer accordingly.

clear: $\mathbb{A}$ decides Q.

$\boxed{\text{to show}}$: $\mathbb{A}$ is almost optimal.

Let $\mathbb{B} = \mathbb{A}_{i_{\mathbb{B}}}$ decide Q.

note: $\langle \mathbb{B}, 1^s \rangle \in \mathrm{SOUND}(Q)$

$\Rightarrow$ $\mathbb{V}$ accepts $\langle \mathbb{B}, 1^s\rangle$ in time $s^{f(\mathbb{B})}$

$\Rightarrow$ for all $x \in Q$; $|x| \ge i_{\mathbb{B}}$: $\mathbb{A}$ accepts in LINE $\circledast$ or earlier.

$$\Rightarrow t_{\mathbb{A}}(x) \le \left(|x| + t_{\mathbb{B}}(x) + t_{\mathbb{V}}\left(\left\langle \mathbb{B}, 1^{\max\{|x|, t_{\mathbb{B}}(x)\}} \right\rangle\right)\right)^{O(1)}$$

$$\le \left(\left(|x| + t_{\mathbb{B}}(x)\right)^{f(\mathbb{B})} \right)^{O(1)}$$

hence $t_{\mathbb{A}}(x) \le \left(|x| + t_{\mathbb{B}}(x)\right)^{O(1)}$ $\quad$ for all $x \in Q$ with $|x| > i_{\mathbb{B}}$

$\blacksquare$

31.03.2014

recall (last time) **Lemma 1**: If $\langle A, 1^s\rangle \overset{?}{\in}$ SOUND(Q) is in time $s^{f(A)}$ for some function $f: \{0,1\}^* \to \mathbb{N}$ then Q has an almost optimal algorithm.

SOUND(Q)

Input: $A, 1^s$ some $s \in \mathbb{N}$
Probl: $\{x \in \{0,1\}^{\leq s} / A \text{ accepts } x \text{ in } \leq s \text{ steps}\} \subseteq Q$?

$\boxed{\text{Lemma 2}}$: Assume SOUND(Q) $\leq_p$ Q.

If $\langle A, 1^s\rangle \overset{?}{\in}$ SOUND(Q) is not decidable in time $s^{f(A)}$ for any function $f: \{0,1\}^* \to \mathbb{N}$ then every algorithm for Q has a hard seq.

Proof: Asm the "if" part

Claim: There is no alg. $\mathbb{W}$ for SOUND(Q) which runs in time $s^{f(A)}$ on inputs $\langle A, 1^s\rangle$ with $L(A) \subseteq Q$

Pf of Claim:

Otw choose $\mathbb{W}$ and f such.

Define $\mathbb{V}$ on $\langle B, 1^s\rangle$:

run $\mathbb{W}$ on $\langle B, 1^s\rangle$. In parallel:

for $r = 0, 1, \dots$
  for all $y \in \{0,1\}^r$
    if B accepts y in $\leq r$ steps & $y \notin Q$
      then if $s < r$ accept
      else reject

if $\mathbb{W}$ halts first, then halt and answer accordingly (always halts)

Then $\mathbb{V}$ decides SOUND(Q).

$$t_{\mathbb{V}}(\langle B, 1^s\rangle) \leq \begin{cases} t_{\mathbb{W}}(\mathbb{B}, 1^s) \xRightarrow{\text{hyp}} s^{f(B)} & \text{if } L(\mathbb{B}) \subseteq Q \\ g(\mathbb{B}) + O(s) & \text{otw.} \end{cases}$$

Contradicting the "if" ↯

∎

Let $\mathbb{B}$ be an algorithm for Q.

Let $\mathbb{R}$ compute a ptime reduction from SOUND(Q) $\leq_p$ Q

Then $\underbrace{\mathbb{B} \circ \mathbb{R}}$ decides SOUND(Q)

↳ (on y, compute $\mathbb{R}(y)$, run $\mathbb{B}$ on $\mathbb{R}(y)$)

$\xRightarrow{\text{Claim}}$ $\exists$ $\mathbb{A}$ with $L(\mathbb{A}) \subseteq Q$ st $t_{\mathbb{B}\circ\mathbb{R}}(\langle \mathbb{A}, 1^s\rangle)$ is not $s^{O(1)}$

But $t_{\mathbb{B}\circ\mathbb{R}}(\langle \mathbb{A}, 1^s\rangle) \leq O\Big(\underbrace{t_{\mathbb{R}}(\langle \mathbb{A}, 1^s\rangle)}_{\leq s^{O(1)}} + t_{\mathbb{B}}(\underbrace{\mathbb{R}(\langle \mathbb{A}, 1^s\rangle)}_{=: x_s})\Big)$

$\Longrightarrow$ $t_B(x_s)$ is <u>not</u> polynomial in s (ie $s^{O(1)}$)

Hence $(x_s)_s$ is a <u>hard sequence</u> for B.

∎

<u>recall</u>: we want to prove the following theorem:

THEOREM: If Q is coNP-complete, then TFAE:

(1) Q does not have an almost optimal algorithm

(2) Every algorithm for Q has a hard sequence

<u>Proof</u>: (2) => (1) by Proposition

$\boxed{(1) \Rightarrow (2)}$

(1) $\overset{(L1)}{\Longrightarrow}$ $\langle \mathbb{A}, 1^s\rangle \overset{?}{\in}$ SOUND(Q) is not in time $s^{f(A)}$

Q is coNP-complete, in part $Q \in$ coNP

$\Downarrow$

SOUND(Q) $\in$ coNP

Q is coNP-hard $\Rightarrow$ $\mathrm{SOUND}(Q) \leq_p Q$

L2 $\Longrightarrow$ (2).

$\blacksquare$

Remark: The proof shows that the theorem holds for Q s.t. $\mathrm{SOUND}(Q) \leq_p Q$.

(e.g. for Π_t^p, PSPACE; E – complete problems)

Lemma If $Q \leq_p Q'$ and every alg for Q has a hard sequence then also every algorithm for Q' has one.

Pf: Let $\mathbb{R}$ compute a reduction $Q \leq_p Q'$

Let $\mathbb{A}'$ decide Q'

Then $\mathbb{A}' \circ \mathbb{R}$ decides Q.

$x'_s := \mathbb{R}(x_s)$ for $(x_s)_s$ hard seq for $\mathbb{A}' \circ \mathbb{R}$

Then $(x'_s)_s$ is a hard seq for $\mathbb{A}'$

Indeed,

$$t_{\mathbb{A}' \circ \mathbb{R}}(x_s) \leq O\Big(\underbrace{t_{\mathbb{R}}(x_s)}_{\leq s^{O(1)}} + t_{\mathbb{A}'}\big(\underbrace{\mathbb{R}(x_s)}_{x'_s}\big)\Big)$$

l.h.s is not $s^{O(1)}$ $\Rightarrow$ hence $t_{\mathbb{A}'}(x'_s)$ is not $s^{O(1)}$!

(left hand side)

$\blacksquare$

Corollary: If there exists a coNP-complete problem without an almost optimal problem, then no coNP-hard problem has an optimal algorithm.

Pf

Let Q be coNP-complete without almost optimal alg. Let Q' be coNP-hard.

Thm $\Rightarrow$ every alg. for Q has hard sequences.

$Q \leq_p Q' \xrightarrow{\text{Lemma}}$ every alg for Q' has hard seq

$\Downarrow$ Proposition

no alg. for Q' is almost optimal.

$\square$

Def: A decides Q.
$\tilde{Q}$ is a hard set for A if $\tilde{Q} \subseteq Q$; $\tilde{Q} \in P$
and t_A is not polynomial on $\tilde{Q}$.

Remark: We know $\to$ almost optimal alg do not have hard sets

$\to$ $(x_s)_s$ honest hard seq for A then
$\{x_s \mid s \in \mathbb{N}\}$ is a hard set.

Def: Q has padding if there is a function

$\textbf{pad} : (\{0,1\}^*)^2 \longrightarrow \{0,1\}^*$, ptime,

$|pad(x,y)| \geq |x| + |y|$, $pad(x,y) \longmapsto y$ ptime,

$pad(x,y) \in Q \iff x \in Q$.

Prop: Asm Q has padding.
If every alg for Q has hard sequences, then it also has hard sets

Pf: Let pad witness paddability of Q.

Let A decide Q.

Define B on x: $\longrightarrow$ B decides Q.

for $i = 0, 1, \ldots$ do
run A for $\leq i$ steps on each of

$pad(x, 1^0)$, $pad(x, 1^1)$ ..... $pad(x, 1^{(i-1)})$

if any of these computations halt,
then halt and answer accordingly.

B decides Q and for all x, s

$$t_B(x) \le \left(t_A(pad(x, 1^s)) + s \right)^{O(1)}$$

Let $(x_s)_s$ be a hard sequence for $\mathbb{B}$

Then $\left(pad(x_s, 1^s) =: \tilde{x}_s\right)_s$ is a hard seq of $\mathbb{A}$
and $s \le |\tilde{x}_s|$, thus <u>honest</u>

$\boxtimes$

Example Let $Q (\in E)$ be $\begin{cases} \text{infinite} \\ \text{P-immune} \end{cases}$ (ie. without an infinite subs in P)

$\Rightarrow$ no alg for Q has hard sets.

<u>Claim</u>: every infinite Q has an algorithm with a hard seq.

Pf: Let $\mathbb{E}$ enumerate Q : $e_0, e_1, e_2 \dots$
with $|e_0| \le |e_1| \le \dots$

($\exists$ such enum $\Leftrightarrow$ Q is decidable)

Let t_0 be the time until E's first output e_0.

Let $\mathbb{G}$ (generating alg)
on 1^s, output $x_s :=$ last output of E within the first $t_0 + s$ many steps.

Define $\mathbb{A}$ on x:
compute $x_0, x_1 \dots x_{s^*}$
for s^* = minimal with $|x| < |x_{s^*}|$
if $x \notin \{x_s : s < s^*\}$ then reject
else do 2^{s^*} many dummy steps
and then accept

Then

$$t_A(x_s) > 2^{s^*} \ge 2^s \quad \leftarrow \text{is not } s^{O(1)}, \text{ so } (x_s)_s \text{ hard seq.}$$

$\boxtimes$

7.04.2014

Open question: there exists Q st

- every $\mathbb{A}$ for Q has hard seq
- not every $\mathbb{A}$ for Q has hard set

Known: Assume "NP does not have p-measure 0 in E"
Then there exists Q st
- every $\mathbb{A}$ for Q has hard sets
- not every $\mathbb{A}$ for Q has hard seq.

3. LEVIN'S OPTIMAL INVERTER

Def: Let $F: \{0,1\}^* \to \{0,1\}^*$.
An inverter of F is an algorithm $\mathbb{I}$ st for all $y \in im(F)$, $\mathbb{I}$ halts on y and $F(\mathbb{I}(y)) = y$.

Theorem: (Levin, '73)
Let $F: \{0,1\}^* \to \{0,1\}^*$ be computed by $\mathbb{F}$.
There exist a Levin-optimal inverter $\mathbb{O}$ of F
ie. exists $d \in \mathbb{N}$ for all inverters $\mathbb{I}$ of F
exists $c_{\mathbb{I}} \in \mathbb{N}$ for all $y \in im(F)$:

$$t_{\mathbb{O}}(y) \leq c_{\mathbb{I}} \left(t_{\mathbb{I}}(y) + |y| + t_{\mathbb{F}}(\mathbb{I}(y)) \right)^d$$

In particular: if F is ptime then

$$t_{\mathbb{O}}(y) \leq c_{\mathbb{I}} \left(t_{\mathbb{I}}(y) + |y| \right)^d$$

Moreover: $\mathbb{O}$ does not halt on any $y \notin im(F)$

Diagonalization Lemma

Let $\mathcal{D} \neq \emptyset$ be a r.e. set of algorithms.
Then there exists $\mathbb{A}$ st for all $x \in \{0,1\}^*$

(a) $\begin{cases} \bullet\ t_{\mathbb{A}}(x) < \infty \Leftrightarrow \exists\, \mathbb{D} \in \mathcal{D} \quad t_{\mathbb{D}}(x) < \infty \\ \bullet\ t_{\mathbb{A}}(x) < \infty \Rightarrow \exists\, \mathbb{D} \in \mathcal{D} \quad \mathbb{A}(x) = \mathbb{D}(x) \end{cases}$

(b) There is $d \in \mathbb{N}$ st for all $\mathbb{D} \in \mathcal{D}$, there is $c_{\mathbb{D}} \in \mathbb{N}$ for all $x \in \{0,1\}^*$

$$t_{\mathbb{A}}(x) \leq c_{\mathbb{D}} \left(t_{\mathbb{D}}(x) + |x| \right)^d.$$

Moreover, there exists a computable fct mapping $\mathbb{E}$ enumerating $\mathcal{D}$ by some $\mathbb{A}$ with (a) & (b).

Pf Let $\mathbb{E}$ enumerate $\mathcal{D}$

Let $\mathbb{E}_i := \begin{cases} \text{last algorithm output in } i \text{ steps} \\ \text{undefined if there is none} \end{cases}$

$\mathbb{A}$ on x

```
1. l ← 0
2. for i = 0 to l
3.    if E_i defined, then simulate (l - i + 1) th
          step of E_i on x
4.    if simulation halts then halt and output accordingly
5. l ← l+1
6. goto 1
```

clear: $\mathbb{A}$ satisfies (a), computable from $\mathbb{E}$

time in 2-4 is polynomial in $|x| + l$, say

$$\leq c_0 \cdot (l + |x|)^{d_0}$$

verify (b) for $d := d_0 + 1$

let $\mathbb{D} \in \mathcal{D}$, choose $i_{\mathbb{D}}$ minimal st $\mathbb{D} = \mathbb{E}_{i_{\mathbb{D}}}$

Then $\mathbb{A}$ halts in line 4 for $l := i_{\mathbb{D}} + t_{E_{i_{\mathbb{D}}}}(x)$; $i := i_{\mathbb{D}}$

or earlier.

$$t_{\mathbb{A}}(x) \leq O\left(\sum_{l=0}^{i_{\mathbb{D}} + t_{\mathbb{D}}(x)} (l + |x|)^{d_0} \right) \leq$$

$$\leq O\left(\left(i_{\mathbb{D}} + t_{\mathbb{D}}(x) + |x| \right)^{d_0 + 1} \right)$$

$$\leq c_{\mathbb{D}} \cdot \left(t_{\mathbb{D}}(x) + |x| \right)^{d} \qquad \text{for suitable } c_{\mathbb{D}} \in \mathbb{N}$$

$\blacksquare$

Proof of Levin's Theorem

Let $\mathbb{B}$ be an algorithm

Defn $\mathbb{B}^*$ on y :

```
run B on y
if B halts, check F(B(y)) = y
        if so, output B(y)
        else never halt
```

note: $t_{\mathbb{B}^*}(y) \leq O\left(t_{\mathbb{B}}(y) + |y| + t_{\mathbb{F}}(\mathbb{B}(y)) \right)$ (*)

$\mathbb{D} := \{ \mathbb{B}^* \mid \mathbb{B} \text{ algorithm} \}$ is c.e.

Apply Diag. Lemma to get $\mathbb{A} =: \mathbb{O}$

Note : $\mathbb{B}^*$ on y halts only on $y \in \mathrm{im}(F)$

and then with output in $F^{-1}(y)$

by (a) so does $\mathbb{O}$.

Let $\mathbb{I}$ be an inverter of F, $y \in \mathrm{im}(F)$

$\Rightarrow$ $\mathbb{I}^*$ inverter of F & $\mathbb{I}^*(y) = \mathbb{I}(y)$

$\Rightarrow \exists\, \mathbb{D} \in D$: $\mathbb{D}$ halts on y.

$\overset{(a)}{\Rightarrow}$ $\mathbb{O}$ halts on y.

Thus, $\mathbb{D}$ is inverter of $\mathbb{F}$

Moreover,

$$t_{\mathbb{D}}(y) \leq c_{\mathbb{I}^*}\left(t_{\mathbb{I}^*}(y) + |y| + t_{\mathbb{F}}(\mathbb{I}^*(y))\right)^d$$

$$\overset{(*)}{\leq} e_{\mathbb{I}}\left(t_{\mathbb{I}}(y) + |y| + t_{\mathbb{F}}(\mathbb{I}(y))\right)^d$$

for suitable $e_{\mathbb{I}}$

<u>Def</u>: $\mathbb{A}$ is as fast as $\mathbb{B}$ if there is a polynomial p st for all $x \in \{0,1\}^*$: $t_{\mathbb{A}}(x) \leq p(t_{\mathbb{B}}(x) + |x|)$

<u>Rem</u>: (∘) an optimal alg for Q is an alg that decides Q and is as fast as any algorithm deciding Q.

(∘) $\mathbb{D}$ is an inverter that is as fast as any inverter (of $\mathbb{F}$).

<u>Cor</u>: Let Q be a problem. TFAE:

(1) Q has an optimal alg

(2) There is $\emptyset \neq D \subseteq \{\mathbb{A} / \mathbb{A} \text{ decides } Q\}$ c.e. and for all $\mathbb{B}$ deciding Q exists $\mathbb{B}' \in D$ st $\mathbb{B}'$ is as fast as $\mathbb{B}$.

<u>Pf</u>: 1) ⟹ 2)

Let $\mathbb{A}^*$ be an optimal alg for Q

Set $D := \{\mathbb{A}^*\}$

2) ⟹ 1): choose $\mathbb{A}$ for D according to Diag. Lemma.

a) ⟹ $\mathbb{A}$ decides Q. Let $\mathbb{B}$ decide Q.
Choose $\mathbb{B}'$ acc. to (2). Let $x \in \{0,1\}^*$!

b) ⟹ $t_{\mathbb{A}}(x) \leq c_{\mathbb{B}}\left(t_{\mathbb{B}'}(x) + |x|\right)^d$

$\leq q\left(t_{\mathbb{B}}(x) + |x|\right)$ for suitable q.

(given as Exercise) Prop: Let Q be a problem. Then $\{\mathbb{A} \mid \mathbb{A} \text{ decides } Q\}$ is not r.e.

Pf: Otw let $\mathbb{E}$ be an enumeration.

Let $\mathbb{B}$ be arbitrary

$\mathbb{B}_0$ on x

run $\mathbb{A}_0$ on x
if x codes a complete run of $\mathbb{B}$ on λ
then invert $\mathbb{A}_0$'s answer

Then $\mathbb{B}_0$ decides $Q \iff \mathbb{B}$ doesn't halt on λ

$\mathbb{B}_1$ on x

for all $y \in \{0,1\}^*$ (in $<_{lex}$-order)
if y codes a complete run of $\mathbb{B}$ on λ
then run $\mathbb{A}_0$ on x and answer accordingly.

Then $\mathbb{B}_1$ decides $Q \iff \mathbb{B}$ halts on λ.

decide $\{\mathbb{B} \mid \mathbb{B} \text{ halts on } \lambda\}$:

run $\mathbb{E}$
if $\mathbb{E}$ outputs $\mathbb{B}_0$ then reject
if $\mathbb{E}$ outputs $\mathbb{B}_1$ then accept

(given as Exercise) Prop: Let $F : \{0,1\}^* \xrightarrow[\text{ptime}]{\text{injective, honest}} \{0,1\}^*$ and $\mathbb{O}$ be a Levin optimal inverter of F. TFAE:

(1) F is a worst-case one-way fct
i.e. F^{-1} is not ptime.

(2) $\mathbb{O}$ is not ptime on $\mathrm{im}(F)$

Theorem (Levin '73)

Let $F: \{0,1\}^* \to \{0,1\}^*$ be ptime

There exists a <u>Levin-optimal</u> inverter $\mathbb{O}$ of F, ie exists $d \in \mathbb{N}$ st for every inverter $\mathbb{I}$ of F there is $c_{\mathbb{I}} \in \mathbb{N}$ st for all $y \in im(F)$ $\quad t_{\mathbb{O}}(y) \le c_{\mathbb{I}}\left(t_{\mathbb{I}}(y) + |y|\right)^d$

Exercise:

Asm F is ptime, honest, injective

Let $\mathbb{O}$ be a Levin-optimal inverter for F. Then TFAE:

(1) F is worst-case one-way

(2) $\mathbb{O}$ is not ptime on $im(F)$

§ 4 SAT-solvers

Def A <u>SAT-solver</u> is an algorithm that, given a propositional frm α:

- halts rejecting if $\alpha \notin SAT$
- outputs an assignment A (to at least the variables of α) st $A \models \alpha$, <u>otw</u>

Theorem: There exists an <u>almost optimal</u> SAT-solver A^*

i.e. A^* is a SAT-solver <u>and</u> for every SAT-solver $\mathbb{B}$, there is a polynomial $p_{(\mathbb{B})}$ st for all $\alpha \in SAT$:

$$t_{A^*}(\alpha) \le p\left(t_{\mathbb{B}}(\alpha) + |\alpha|\right)$$

Pf: Define

$$F_{SAT}(x) = \begin{cases} \alpha & \text{if } x = \langle \alpha, A\rangle \text{ st } \alpha \text{ is a frm and } A \models \alpha \\ 1 & \text{else} \end{cases}$$

(1: or any other fixed satisf. frm.)

Then $\boxed{im(F_{SAT}) = SAT}$

Let $\mathbb{O}$ be a Levin-optimal inverter for F_{SAT} and A_0 an arbitrary alg. deciding SAT

A^* on α: run A_0 and $\mathbb{O}$ in parallel on α

if A_0 rejects then halt and reject

if $\mathbb{O}$ outputs $\langle \alpha, A \rangle$, then output A

(we assume $\mathbb{O}$ does not halt on α outside SAT.)

(0) if $\alpha \notin SAT$ then A^* rejects α

(1) if $\alpha \in SAT$ then A^* does not reject and $\mathbb{O}$ halts and outputs F_{SAT}-preimage of α, that is a string of the form $\langle \alpha, A \rangle$ with $A \models \alpha$. then A^* outputs A

Thus, A^* is a SAT-solver

Let $\mathbb{B}$ be a SAT-solver, $\alpha \in SAT$

Then $\mathbb{B}$ "is" an F_{SAT}-inverter, more precisely $\mathbb{B}'(\alpha) := \langle \alpha, \mathbb{B}(\alpha) \rangle$ is an F_{SAT}-inverter.

Lemma $\Longrightarrow$ $t_{\mathbb{O}}(\alpha) \leq c_{\mathbb{B}'} \left(t_{\mathbb{B}'}(\alpha) + |\alpha| \right)^d$

$$\leq \left(t_{\mathbb{B}}(\alpha) + |\alpha| \right)^{O(1)}$$

$\Rightarrow$ claim

because $t_{A^*}(\alpha) \leq O\left(t_{\mathbb{O}}(\alpha) \right)$

Proposition Let A decide SAT Then there is a SAT-solver A^{inv} st for all $\alpha \in SAT$:

("self reducibility" in theoretische Informatik)

$$t_{A^{inv}}(\alpha) \leq O\left(|\alpha| \cdot \max_{\substack{|\beta| \leq |\alpha| \\ \beta \in SAT}} t_A(\beta) \right)$$

Pf: A^{inv} on $\alpha = \alpha(X_1, X_2 \ldots X_n)$ (arb prop. fm)

1. run A on α
2. if A rejects α then reject

3. $\beta \leftarrow \alpha$, $A \leftarrow \emptyset$. 2/28.04.2014

4. for $i = 1, \dots, n$ do

5. $\beta_0 \leftarrow \beta \frac{0}{X_i}$ (i.e. replace var X_i in β by the constant 0)

6. $\beta_1 \leftarrow \beta \frac{1}{X_i}$

7. run $\mathbb{A}$ on β_0, β_1 in paralell.

8. if $\mathbb{A}$ accepts β_0 first then $\beta \leftarrow \beta_0$, $A \leftarrow A \cup \{(X_i, 0)\}$

9. else $\beta \leftarrow \beta_1$, $A \leftarrow A \cup \{(X_i, 1)\}$

10. output $\langle \alpha, A \rangle$

Then $\mathbb{A}^{inv}$ is a SAT-solver

Asm $\left| \gamma \frac{b}{X} \right| \le |\gamma|$ for frm γ, $b \in \{0,1\}$, X var.

Line 7 is executed only if at least one of β_0, β_1 is satisfiable.

in time: $\le \max \{ t_{\mathbb{A}}(\beta) \mid |\beta| \le |\alpha|, \beta \in SAT \}$

also bounds the time in line 1. (if $\alpha \in SAT$)

this is executed $\le n \le |\alpha|$ many times.

$\blacksquare$

(Schnorr, '74?) (Verbitsky, '76?)

Theorem: There exists a length-optimeal algorithm $\mathbb{A}_{SAT}$ for SAT.

i.e. for every $\mathbb{B}$ deciding SAT there exists a polynomial p st for all satisfiable frm α ($\alpha \in SAT$)

$$t_{\mathbb{A}_{SAT}}(\alpha) \le p\left(|\alpha| \cdot \max_{\substack{\beta \in SAT \\ |\beta| \le |\alpha|}} t_{\mathbb{B}}(\beta)\right)$$

Pf: Let $\mathbb{A}^*$ be an almost optimal SAT-solver

$\mathbb{A}_{SAT}$ on α:

runs $\mathbb{A}^*$ on α and $\mathbb{A}_0$ on α in paralell
($\mathbb{A}_0$ fixed decision procedure for SAT)

if $\mathbb{A}^*$ outputs a satisfying assignment for α, ACCEPT

if $\mathbb{A}_0$ rejects, reject

Let $\alpha \in SAT$

$$t_{A_{SAT}}(\alpha) \le O\left(t_{A^*}(\alpha) + |\alpha|\right)$$

Let $\mathbb{B}$ decide SAT.

$\Rightarrow B^{inv}$ is a SAT-solver and

$$t_{B^{inv}}(\alpha) \le O\left(|\alpha| \cdot \max_{\substack{\beta \in SAT \\ |\beta| \le |\alpha|}} t_{\mathbb{B}}(\beta)\right).$$

We know $\mathbb{A}^*$ is an <u>almost optimal</u> SAT-solver

$$\Rightarrow \quad t_{\mathbb{A}^*}(\alpha) \le q\left(t_{B^{inv}}(\alpha) + |\alpha|\right)$$

$$t_{A_{SAT}}(\alpha) \le O\left(t_{A^*}(\alpha) + |\alpha|\right)$$

$$\le O\left(q\left(t_{B^{inv}}(\alpha) + |\alpha|\right) + |\alpha|\right) \le$$

$$\le \left(|\alpha| + \max_{\beta}\left(t_{\mathbb{B}}(\beta)\right)\right)^{O(1)}$$

$\blacksquare$

Note: A_{SAT} is explicitly written down.

Theorem $\boxed{P = NP \iff A_{SAT} \text{ is ptime on } \alpha \in SAT}$

Proof: "$\Rightarrow$"

$\mathbb{A}$ decides SAT in time $q(n)$

for all $\alpha \in SAT$

$$t_{A_{SAT}}(\alpha) \le p\left(t_A(\alpha) + |\alpha|\right)$$

for some p.

<u>This is ptime</u> $\le p\left(q(|\alpha|) + |\alpha|\right)$

"$\Leftarrow$" Run A_{SAT} on α for $p(|\alpha|)$ steps. 3/28.04.2014.

and reject if this didn't halt.

(where p polynomial witnessing r.h.s.).

Q: $\boxed{\text{Is } A_{SAT} \text{ almost optimal?}}$

$\boxed{5.05.2014}$

Lemma (reminder)

There is a ptime function computing from a circuit $C(\bar{x})$ a propositional frm $\alpha^c(\bar{x}, \bar{u})$ st

for all assignments A for $\bar{X}$

a) $\exists^{\leq 1}$ assignment B for $\bar{Z}$ st $A \cup B \models \alpha^c$

b) if $C(A(\bar{X})) = 1$, then there is B for $\bar{Z}$ St $A \cup B \models \alpha^c$

Moreover there is a ptime function W st

$$C(A(\bar{X})) = 1 \Rightarrow A \cup W(A) \models \alpha^c$$

Thm: Asm $P \neq NP$.

Let $\mathbb{A}$ be a SAT-solver, $c \in \mathbb{N}$. Then there is a sequence $(\alpha_n)_{n \in \mathbb{N}}$ st:

- $1^n \mapsto \alpha_n$ is ptime
- $|\alpha_n| \geq n \quad \forall n$
- $t_{\mathbb{A}}(\alpha_n) \geq |\alpha_n|^c$, $\alpha_n \in SAT \quad \exists^\infty n$

Pf.: choose q st runs of lgth m of $\mathbb{A}$ on input of lgth $n \leq m$ have code of lgth $q(m)$

FAIL$^{\mathbb{A}}$

Input $x\ y\ z$

Problem: there is n, m st $|x| = |y| = n$ & $|z| = q(n^c)$

x propositional frm, $y \models x$

z non-halting lgth n^c run of $\mathbb{A}$ on x

$\hookrightarrow$ this is in P.

Choose circuits $C_n(\bar{X}, \bar{Y}, \bar{Z})$ st f.a. $x, y \in \{0,1\}^n$
$z \in \{0,1\}^{q(n)}$

$xyz \in \mathrm{FAIL}^{A} \iff C_n(xyz) = 1$

Choose formulas $\alpha^{C_n}(\bar{X}, \bar{Y}, \bar{Z}, \bar{U})$: lemma

wlog: $|\alpha^{C_n}| \geq n$

given 1^n, compute α_n as follows:

run $\mathbb{A}$ on α^{C_n} for $|\alpha^{C_n}|^c$ steps

if simulation halts accepting with output A

then output $\alpha_n := A(\bar{X})$

else output $\alpha_n := \alpha^{C_n}$

$P \neq NP \Rightarrow \exists^\infty n \;\; \exists x, y \in \{0,1\}^n \;\; \exists z \in \{0,1\}^{q(n)}$

$xyz \in \mathrm{FAIL}$.

for each such n: $\alpha^{C_n} \in SAT$

$= \alpha_n$ if $t_{\mathbb{A}}(\alpha^{C_n}) > |\alpha^{C_n}|^c$

else $\alpha_n = A(\bar{X})$ for A a satisfying assignment of α^{C_n}

$\Rightarrow \alpha_n \in SAT, \; t_{\mathbb{A}}(\alpha_n) > |\alpha_n|^c$

$\boxtimes$

Thm: Asm $\mathbb{P} \neq \mathbb{NP}$.

Let $\mathbb{A}$ be a SAT-solver, $c \in \mathbb{N}$. Then there is a seq $(\langle \alpha_n, A_n \rangle)_{n \in \mathbb{N}}$ st

$1^n \longmapsto \langle \alpha_n, A_n \rangle$ is ptime

$|\alpha_n| = n \;\; \forall n$

$\exists^\infty n \;\; A_n \models \alpha_n \;\&\; t_{\mathbb{A}}(\alpha_n) > n^c$.

Pf: asm frm. encodings end with 1

and asm q is as above.

FAIL2

Input: 1^k, $x, y \in \{0,1\}^n$, $z \in \{0,1\}^{q(n)}$

Problem: $x = \alpha 00\ldots0$ for some frm α st $n^{1/k} < |\alpha| < n$, $k \leq n$
& $y \models \alpha$
& z non-halting run of $\mathbb{A}$ on α of length $|\alpha|^c$.

circuit $C_{n,k}(\overline{W}, \overline{X}, \overline{Y}, \overline{Z})$ accepting precisely those $1^k xyz$ which are in FAIL 2.

formulas $\alpha^{C_{n,k}}(\overline{W}, \overline{X}, \overline{Y}, \overline{Z}, \overline{U})$: Lem

$|\alpha^{C_{n,k}}| \leq n^d$ suitable α

wlog $= n^d$ suitable d.

$\beta_n := \alpha^{C_{k,n}}(\overline{1}, \overline{X}^n \overline{Y}^n \overline{Z}^n, \overline{U}^n)$

for k large enough such that $n^d < (n-1)^d$.

1st case $\exists^\infty n$: A on β_n outputs an assignment in $|\beta_n|^c$ steps

given n, compute α_n as follows:

run A on β_m for $m = n, n+1, \dots n^k$

each for $|\beta_m|^c$ steps.

if this output assignment A' with

$A'(\overline{X}^m) = \alpha 0^{m-n}$ for true α

then output $\langle \alpha, A'(\overline{Y}^m) \rangle$

else output a ______ $\langle \gamma, B \rangle$

which $B \models \gamma$; $|\gamma| = n$.

2nd case: ex. $n_0 \in \mathbb{N}$ st for all $n \geq n_0$

A does not accept β_n in $|\beta_n|^c$ steps

choose $n_1 \geq n_0$ minimal st $\beta_{n_1} \in SAT$

$[P \neq NP \Rightarrow \exists^\infty n \ \beta_n \in SAT]$

$\boxed{\text{Claim 1}}$: $\forall m \geq n_1$: $\beta_m \in SAT$

Pf: as n_1 is minimal, there is $\alpha \in SAT$, $t_A(\alpha) > n_1^c$, $|\alpha| = n_1$

$\Rightarrow \beta_{n_1+1} \dots \beta_{(n_1-1)^k} \in SAT$

esp. $\beta_{n_1^d} \in SAT$,

because $t_A(\beta_{n_1^{d^i}}) = \infty$ & $|\beta_{n_1^{d^i}}| = n_1^{d^{i+1}}$

hence $\beta_{n_1^{d^{i-1}}} \in SAT$ $\forall i \geq 1$.

lgth $n_1^{d^i}$

$\Rightarrow \beta_{n_1^{d^i}+1}, \ldots, \beta_{(n_1^{d^i}-1)^k} \in SAT$

this implies the claim 1

$\boxed{\text{Claim 2}}$: there is a ptime fct $D^?$ mapping 1^n to a satisfying assignment of β_n (for all $n_1 \geq n$)

Pf: compute m st $n^{1/k} < m^d \leq n$

suffices to compute $A(\bar{X}^n)$, $A(\bar{Y}^n)$, $A(\bar{Z}^n)$

1). $A(\bar{X}^n) := \beta_m 0^{n-m^d}$

2). $A(\bar{Y}^n) :=$ sat. assignment of β_m

3) $A(\bar{Z}^n) :=$ lgth $|\beta_m|^c$ run of A on β_m.

from this, get in ptime an ass B for $\bar{U}^m$ st

$$A \cup B \models \beta_m(\bar{X}, \bar{Y}, \bar{Z}, \bar{U}^m)$$

computation in 2) is by recursion!

once m drops below n_1^k use brute force.

the rest is clear

$\blacksquare$

Prop: Asm $NP \cap coNP \neq P$. Then A_{SAT} is not almost optimal.

Pf:

Assume A_{SAT} is almost optimal
Let $Q \in NP \cap coNP$. TO show: $Q \in P$.

Choose $R_0, R_1 \subseteq (\{0,1\}^*)^2$ in P, poly. p_0, p_1 st for all $x \in \{0,1\}^*$

a) $x \notin Q \Rightarrow \exists y \in \{0,1\}^{p_0(|x|)} : (x,y) \in R_0$

b) $x \in Q \Rightarrow \exists y \in \{0,1\}^{p_1(|x|)} \; (x,y) \in R_1$

circuits $C_n^b(\bar{X}\,\bar{Y})$ st $(x,y) \in R_b \Leftrightarrow C_n^b(x,y) = 1$

frms. $\alpha^{C_n^b}(X_1 .. X_n, Y_1^b \dots Y_{p_b(n)}^b, U_1 \dots U_{q_b(n)})$ for suitable polynomials q_b.

define $\beta_x := \alpha^{C_n^0}(x, \bar{Y}^0, \bar{U}^0) \vee \alpha^{C_n^1}(x, \bar{Y}^1, \bar{U}^1)$

Then for all $x \in \{0,1\}^n$: $\beta_x \in SAT$

$\gamma_x := \beta_x \wedge (1 \vee x_1 \vee \dots \vee x_n)$

for Boolean constants $x_i \in \{0,1\}$

is satisfiable for all x

and $\{\gamma_x \,/\, x \in \{0,1\}^*\} \in P$

[given γ, check it has the form $(\gamma_0 \vee \gamma_1) \wedge (1 \vee x_1 \vee \dots \vee x_n)$
then check $\gamma_b = \alpha^{C_n^b}(x, \bar{Y}^b, \bar{U}^b)$]

Prop $\overset{\text{alm}}{\underset{\text{opt}}{\Rightarrow}}$ $t_{A_{SAT}}(\gamma_x) \leq |\gamma_x|^{O(1)}$
$\leq |x|^{O(1)}$

Recall: A_{SAT} on α runs an optimal sat solver on α: A^*

$\Rightarrow t_{A^*}(\gamma_x) \leq |x|^{O(1)}$

output A of A^* on γ_x sat. γ_x

$\Rightarrow A \models \alpha^{C^0_{|x|}}(x, \dots)$ or $A \models \alpha^{C^1_{|x|}}(x, \dots)$, but not both.

But $\chi_Q(x) =$ the $b \in \{0,1\}$ st. $A \models \alpha^{C^b_{|x|}}(x, \dots)$

V : PROOF SYSTEMS

A. Propositional proof systems.

Def: A <u>propositional proof system</u> is a ptime $F : \{0,1\}^* \to \{0,1\}^*$ with $\mathrm{im}(F) = \mathrm{TAUT}$.

"X is an F-proof of α"

<u>Example</u>: A Frege system $\mathcal{F}$ is a finite set of <u>rules</u> $\boxed{\frac{\alpha_0, \ldots, \alpha_{k-1}}{\alpha_k}}$ (tuples of propositional fmls.) st.

- Sound $\models \bigwedge_{i<k} \alpha_i \to \alpha_k$
- implicationally complete:
  if $\Gamma \models \alpha$ then there is an $\mathcal{F}$-proof of α from Γ.

An <u>$\mathcal{F}$ proof of α from Γ</u> is a finite seq $(\beta_0 \ldots \beta_n)$ of formulas st for all $i < n$

- $\beta_i \in \Gamma$ or
- there is a rule $\frac{\alpha_0 \ldots \alpha_{k-1}}{\alpha_k}$ in $\mathcal{F}$
  and a substitution $\sigma : \mathrm{Vars} \xrightarrow[\text{part}]{} \mathrm{Fmls}$.
  and $j_0 \ldots j_{k-1} < i$ for all $\nu < k$
  $\beta_\nu = \alpha_\nu^\sigma$, $\beta_i = \alpha_k^\sigma$

here, α^σ is obtained from α by simultaneously replacing X by $\sigma(X)$ if $X \in \mathrm{dom}(\sigma)$.

An <u>$\mathcal{F}$-proof of α</u> is an $\mathcal{F}$-proof of α from $\emptyset$.

$\mathcal{F}$ gives a prop. proof system $F_{\mathcal{F}} : \{0,1\}^* \to \mathrm{TAUT}$ given by

$$F_{\mathcal{F}}(X) = \begin{cases} \alpha & \text{if } X \text{ is } \mathcal{F} \text{ proof of } \alpha \\ 1 & \text{else.} \end{cases}$$

Def. A prop proof system F is polynomially bounded

iff exists p st for all $\alpha \in TAUT$ $\exists$ F-proof x of α

st $|x| \leq p(|\alpha|)$

Prop (Cook–Reckhoff) $NP = coNP \iff \exists$ a poly bd. prop. proof system

Pf: Let F be a poly bd system.

Define a nondet. ptime algorithm A on frm α to do:

$$\boxed{\begin{array}{l} \text{guess } x \in \{0,1\}^{\leq p(|\alpha|)} \\ \text{check } F(x) = \alpha \\ \text{if "yes" then accept} \\ \text{if "no" then reject} \end{array}}$$

Then A is ptime and $L(A) = TAUT$

$\Rightarrow TAUT \in NP \Rightarrow NP = coNP$

Asm $NP = coNP$

choose a nondet. ptime A accepting $TAUT$

define $F : \{0,1\}^* \to \{0,1\}^*$

$$F(x) = \begin{cases} \alpha & \text{if } x \text{ is an accepting run of } A \text{ on } \alpha \\ 1 & \text{else.} \end{cases}$$

ptime, with image $TAUT$ $\Rightarrow$ F is a prop. proof system

if $\alpha \in TAUT \Rightarrow A$ accepts α

Let x be the run of A on α. Then $F(x) = \alpha$

Since A is ptime, there is a poly. p. st $|x| \leq p(|\alpha|)$

Hence F is polynomially bounded.

$\boxtimes$

We can compare strength of systems via:

<u>Def</u>. Let F_0, F_1 be prop proof systems.

A p-simulation of F_1 in F_0 is a ptime $T: \{0,1\}^* \to \{0,1\}^*$ s.t. for all x: $F_1(X) = F_0(T(X))$ (write $F_1 \leq_p F_0$ if such a simulation exists)

<u>Prop</u>: If $F_1 \leq_p F_0$ and F_1 is polyn. bounded, then so is F_0

<u>Proof</u>: Let c be st every $\alpha \in TAUT$ has F_1-proof of lgth $\leq |\alpha|^c$

Let d be st T can be computed in time n^d

Then if x is a length $\leq |\alpha|^c$ F_1-proof of α

then $T(x)$ is F_0-proof of α of lgth $\leq (|\alpha|^c)^d$

$\blacksquare$

<u>Example</u>: Let $\mathcal{F}_0, \mathcal{F}_1$ be Frege systems. Then $F_{\mathcal{F}_1} \leq_p F_{\mathcal{F}_0}$

Pf: Let $R := \dfrac{\alpha_0 \ldots \alpha_{k-1}}{\alpha_k}$ rule in $\mathcal{F}_1$

$\mathcal{F}_1$ sound

$\mathcal{F}_0$ impl. complete $\Rightarrow \exists$ a $\mathcal{F}_0$-proof π_R of α_k from $\{\alpha_0 \ldots \alpha_{k-1}\}$

<u>note</u>: if σ is a substitution π $\mathcal{F}_0$ proof

π^σ is a $\mathcal{F}_0$-pf of α_k^σ from $\{\alpha_0^\sigma, \ldots \alpha_{k-1}^\sigma\}$

(note $\pi^\sigma = (\beta_0^\sigma \ldots \beta_{n-1}^\sigma)$ if $\pi = (\beta_0 \ldots \beta_{n-1})$)

let $s = \max\{|\sigma(X)| \,/\, X$ appears in $\alpha_0 \ldots \alpha_k\}$

then $\pi^\sigma \leq |\pi| \cdot s$.

Let $\pi = (\beta_0 \ldots \beta_{n-1})$ $\mathcal{F}_1$-proof of α

If β_i is obtained by rule R_i, substituting σ_i

replace β_i by $\pi_{R_i}^{\sigma_i}$ (size $\leq |\pi_{R_i}| \cdot |\pi|$

This gives an $\mathcal{F}_0$-proof $\tilde{\pi}$ of α of size $\left(\max_{R \in \mathcal{F}_1} |\pi_R|\right) \cdot |\pi|^2 =$

$= O(|\pi|^2)$

Clearly $\pi \to \tilde{\pi}$ is ptime. Hence $F_{\mathcal{F}_1} \leq_p F_{\mathcal{F}_0}$.

Embarrassing open question:

$\boxed{\text{Let } \mathcal{F} \text{ be a Frege system. Is } F_{\mathcal{F}} \text{ poly. bd?}}$

Exercise: $\mathrm{Sat}_n(X_1 \dots X_n, Z_1 \dots Z_n, \bar{V})$

Let $\mathrm{proof}^{\mathbb{F}}_{n,m}(X_1 \dots X_n, Y_1, \dots Y_m, \bar{\mathcal{U}})$ be formulas.

st for all $z, x \in \{0,1\}^n$, $y \in \{0,1\}^m$:

(o) $\mathrm{proof}^{\mathbb{F}}_{m,m}(x, y, \bar{\mathcal{U}}) \in SAT \iff \mathbb{F}(y) = x$

size $\leq t_{\mathbb{F}}(m)^{100}$

(oo) $\mathrm{unsat}_n(x, z, \bar{V}) \in SAT \iff x$ is a fml, y assign. sat $(\neg x)$

Moreover $(\mathbb{F}, n, m) \longmapsto \mathrm{proof}^{\mathbb{F}}_{n,m}$ ptime

$n \longrightarrow \mathrm{unsat}_n$

with images in P

and there are ptime u, v st.

$\mathrm{proof}^{F}_{n,m}(x, y, \bar{\mathcal{U}}) \in SAT \Rightarrow \mathrm{proof}^{F}_{n,m}(x, y, u(x,y))$

$\mathrm{unsat}_n(x, z, \bar{V}) \in SAT \Rightarrow \mathrm{unsat}^{F}_n(x, z, v(x,y))$

F prop proof system

Lemma: (clear) $\mathrm{sound}^{F}_{n,m}(X_1 \dots X_n, Y_1 \dots Y_m, \bar{U}, \bar{V}) :=$

$:= \mathrm{proof}_{n,m}(\bar{X}, \bar{Y}, \bar{\mathcal{U}}) \to \neg \mathrm{unsat}_n(\bar{X}, \bar{Z}, \bar{V}))$

Then $\{\mathrm{sound}^{F}_{n,m} \mid n, m \in \mathbb{N}\} \subseteq TAUT, \in P.$

<u>Def</u>: F prop proof system, F is <u>decent</u> iff the following tasks are ptime:

(D_1): from an F-proof π of a frm α and a subst σ with $\mathrm{dom}(\sigma) \subseteq \mathrm{vars}(\alpha)$
$\mathrm{im}(\sigma) \subseteq \{0,1\}$
Boolean const
construct an F proof π' of α^{σ}.

(D_2). from a free sentence α (no vars)
construct an F proof of α

(D_3) from proofs of $(\alpha \to \beta)$ and α construct a proof of β

(D_4) from a proof of $\neg\, \mathrm{unsat}_{|\alpha|}(\ulcorner \alpha \urcorner, \bar{Z}, \bar{V})$ construct a proof of α.

<u>Thm</u>: Let F_0, F_1 pps, F_0 decent.
Asm there is a ptime funct. mapping (n, m) to an F_0-pf of $\mathrm{sound}^{F_1}_{n,m}$. Then $F_1 \leq_p F_0$

<u>Example</u>: Frege syst. are decent for a careful choice of unsat_n frmls to ensure (D_4)

<u>Pf</u>: Let π be an F_1-pf of α

1) compute an F_0 - pf of $\mathrm{sound}^{F_1}_{n,m}(\bar{X}, \bar{Y}, \bar{U})$ (Ass)
for $n = |\ulcorner \alpha \urcorner|$, $m = |\ulcorner \pi \urcorner|$

2) Compute an F_0 - pf of (D_1)

$\mathrm{proof}_{n,m}(\ulcorner \alpha \urcorner, \ulcorner \pi \urcorner, u(\ulcorner \alpha \urcorner, \ulcorner \pi \urcorner)) \to \neg\, \mathrm{unsat}_n(\ulcorner \alpha \urcorner, \bar{Z}, \bar{V})$

3) compute F_0 - pf of (D_2)

$\mathrm{proof}_{n,m}($ ______ $)$

4). compute an F_0-pf of (D3)

$\neg \text{unsat}_n(\ulcorner \alpha \urcorner, \bar{Z}, \bar{V})$

5) compute F_0-pf of α (D4)

$\boxtimes$

Def: A prop pf system is p-optimal $\Leftrightarrow$ it p-sim. every other pps.

Open: Do p-opt pps exist?

Open: is a (all) Frege syst p-optimal?

26.05.2014:

Recall Prop frms unsat_n

$\text{proof}^F_{n,m}$

computable in time $t_F(m)^{100}$

$\text{unsat}_n(x, z, \bar{u}) \in SAT \Leftrightarrow$ "$y \models \neg x$".

$\text{proof}^F_{n,m}(x, y, \bar{V}) \in SAT \Leftrightarrow F(y) = x$

$\text{Sound}^F_{n,m} :\equiv \left(\text{proof}^F(\bar{X}, \bar{Y}, \bar{U}) \longrightarrow \neg\, \text{unsat}_n(\bar{X}, \bar{Z}, \bar{V})\right)$

Def: A p.p.s F is p-optimal if it p-simulates all others.

Remark: $\mathcal{F}$ plus the rule $\frac{\alpha}{\alpha^\sigma}$ σ substitution is not known to be simulated by $\mathcal{F}$ ($\mathcal{F}$ is Frege syst).

Thm: p-optimal p.p.s exist $\Leftrightarrow$ almost optimal algorithms for TAUT exist.

Pf: "$\Rightarrow$" (holds more generally, not only for TAUT)

"$\Leftarrow$" Let A_{opt} be an almost optimal algorithm for TAUT.

Let F be a pps, computed by $\mathbb{F}$

$\Rightarrow \{ \mathrm{sound}^{\mathbb{F}}_{n,m} \mid n,m \} \subseteq \mathrm{TAUT}, \in P$

$\Rightarrow t_{A_{opt}}$ is ptime on $\mathrm{sound}^{\mathbb{F}}_{n,m}$

$\boxed{\text{Claim}}$: Every pps F is p-simulated by a time cn^2-computable pps $F^>$; $c \in \mathbb{N}$ fixed constant

Pf of claim: ← padding

Let F be computable in time n^{d_F}, $d_F \in \mathbb{N}$

Defn $F^>$ to map $\langle Y, 1^{|x|^{d_F}} \rangle$ to $F(X)$

and other strings to 1.

Define F_{opt} by the following algorithm:

check: input has the form $\langle \mathbb{F}, y, 1^t \rangle$

for $\mathbb{F}$ an alg, y an arbitrary string

[if not then output 1]

fixed TAUTOLOGY

simulate $\mathbb{F}$ on y for $c \cdot |y|^2$ steps.

if the simulation does not halt, output 1

othw: let $x := \mathbb{F}(y)$.

check x is a fml.

compute the fml $|\mathrm{sound}^{\mathbb{F}}_{|x|,|y|}|$

for $(|y|^2)^{100}$-many steps.

if this computation does not halt, output 1

otw run A_{opt} on $\mathrm{sound}^{\mathbb{F}}_{|x|\,|y|}$ for $\le t$ steps.

if A_{opt} accepts, then output x

otw. output 1

F_{opt} outputs either $\bot$ or x.

It outputs x only if $\text{sound}^{\mathbb{F}}_{|x|,|y|} \in \text{TAUT}$.

and $\mathbb{F}(y) = x$

$\Rightarrow$ $\text{proof}^{\mathbb{F}}_{|x|,|y|}(x, \bar{z}, u(x,y))$ is <u>true</u>

$\Rightarrow$ $\neg\,\text{unsat}_{|x|}(x, \bar{Z}, \bar{V}) \in \text{TAUT}$

$\Rightarrow$ $x \in \text{TAUT}$

Hence: $\boxed{\text{im}(F_{opt}) \subseteq \text{TAUT}}$

It <u>suffices to show</u> that $\boxed{F \leq_p F_{opt}}$ for F a $c \cdot n^2$ time comput. pps.

say computed via $\mathbb{F}$.

$y \longmapsto \langle \mathbb{F}, y, 1^t \rangle$ for $t := |y|^{d_{\mathbb{F}}}$ where $d_{\mathbb{F}}$ is st.

$t_{A_{opt}}$ on $\text{sound}^{\mathbb{F}}_{n,m}$ is bounded

by $m^{d_{\mathbb{F}}}$

$\blacksquare$

<u>Prop</u> A p-optimal pps is polynomially bounded iff there exists a poly. bd. pps.

<u>Pf</u> "$\Rightarrow$" ✓

"$\Leftarrow$" F poly. bd. pps

let F_{opt} be p-optimal pps. ~~Then~~

$F \leq_p F_{opt}$ $\underset{(\text{Prop})}{\Rightarrow}$ F_{opt} is poly. bd.

$\blacksquare$

$\boxed{\text{B: } \underline{\text{General proof systems}}}$

Assume $Q \neq \emptyset$, decidable.

<u>Def</u>: A <u>proof system</u> for Q is a ptime surjection F onto Q.

If $F(y) = x$ then y is an <u>F-proof of x</u>

F is <u>polynomially bounded</u> if there is a poly. p st for all $x \in Q$ $\exists y \in \{0,1\}^{\leq p(|x|)}$: $F(y) = x$.

A <u>p-simulation of F in F'</u> where F and F' are proof syst of Q is a ptime T st $F(y) = F'(T(y))$ for all $y \in \{0,1\}^*$

Then write $F \leq_p F'$

F is <u>p-optimal</u> $\iff$ $F' \leq_p F$ for all proof systems F' for Q

$\boxed{\text{Prop}}$: (a). There exists a polynomially bounded proof system for $Q \iff Q \in NP$

(b) Let F be a p-optimal proof system for Q

Then $Q \in NP \iff$ there exists a poly bdd. proof system for Q

$\iff F$ is polyn. bounded.

$\square$ (the proof for $Q = TAUT$ works!)

Exercise!

$\boxed{\text{THM}}$: Let F_{opt} be a p-optimal proof system for Q.

Let $\mathbb{O}$ be an optimal inverter of F_{opt} which does not halt on $y \notin \text{im}(F_{opt}) = Q$.

Then $\mathbb{O}^{dec}$ is an almost optimal alg. for Q

the <u>decision</u> procedure

$\mathbb{O}^{dec}$ on y: runs $\mathbb{O}$; $\mathbb{A}_0$ on y in parallel

if $\mathbb{O}$ halts then accept

if $\mathbb{A}_0$ rejects then reject.

where $\mathbb{A}_0$ is an algorithm deciding Q.

2.06.2014

(Survey: Chen & Flum!) on the Webpage

recall

$Q \subseteq \{0,1\}^*$, $\neq \emptyset$, decidable

$F : \{0,1\}^* \to \{0,1\}^*$ ptime

F proof system for Q : $\mathrm{im}(F) = Q$

Theorem: Let F_{opt} be a p-optimal proof system for Q.
Let $\mathbb{O}$ be an optimal inverter of F_{opt}.
Then $\mathbb{O}^{dec}$ is an almost optimal algorithm for Q.

Proof: Let B decide Q.

Define a proof system F_B for Q as follows:

$$F_B(x) = \begin{cases} y & \text{if } x \text{ is a run of } B \text{ on } y, \text{ accepting} \\ y_0 & \text{else}, \text{ where } y_0 \in Q \text{ arbitrary, fixed.} \end{cases}$$

$\Rightarrow F_B \leq_p F_{opt}$ via T

Define an inverter $\mathbb{I}$ of F_{opt}:

on y, compute $T(\text{run of } B \text{ on } y)$

$$\Rightarrow F_{opt}(\mathbb{I}(y)) = F_{opt}(T(x)), \; x \text{ is the run of } B \text{ on } y.$$
$$= F_B(x)$$
$$= y. \quad \text{so, indeed an inverter.}$$

$\mathbb{O}$ is optimal $\Longrightarrow t_{\mathbb{O}}(y) \leq (\underbrace{t_{\mathbb{I}}(y)}_{\leq (t_B(y))^{O(1)}} + |y|)^{O(1)}$

(for $y \in Q$)

$$\Rightarrow t_{\mathbb{O}^{dec}}(y) \leq O(t_{\mathbb{O}}(y))$$
$$\leq (t_B(y) + |y|)^{O(1)}$$

$\boxtimes$

Def: $Q \subseteq \{0,1\}^*$ is <u>paddable</u> $\Leftrightarrow$ there is a ptime, injective

$$pad : (\{0,1\}^*)^2 \longrightarrow \{0,1\}^* \text{ st}$$

for all $x, y \in \{0,1\}^*$

a) $pad(x,y) \in Q \Leftrightarrow x \in Q$

b) $|pad(x,y)| \geq |x| + |y|$

c) partial fct $pad(x,y) \mapsto x$, $pad(x,y) \mapsto y$ are ptime

Theorem: Q paddable. Then Q has an almost optimal algorithm iff Q has a p-optimal proof system.

Proof: "$\Leftarrow$" — just the previous theorem

"$\Longrightarrow$" As for propositional proof systems we see that every proof system for Q is p-simulated by a proof system for Q which is computable in time $c\, n^2$ (where $c \in \mathbb{N}$ is a suitable constant)

$\mathbb{F}$ is computable as follows:

Fix $y_0 \in Q$.

On input y, check that y has the form

$$y = \langle \mathbb{F}, z, 1^t \rangle \quad \text{for TM } \mathbb{F}, \text{ string } z, \ t \in \mathbb{N}$$

if not, then output y_0

otw, simulate $\mathbb{F}$ on z for $c \cdot |z|^2$ many steps.

if the simulation does not halt then output y_0

otw. let $x := \mathbb{F}(y)$

check "$\mathbb{F}$ is o.k."

if so, output x

else output y_0

"$\mathbb{F}$ is o.k." $\equiv$ $\mathbb{A}_{opt}$ on $\boxed{pad(x, z)}$ accepts in $\leq t$ steps.

where $\mathbb{A}_{opt}$ is an almost optimal alg for Q

Note: – either y_0 or x is output.

$y_0 \in Q$ and x is output only if $\boxed{\mathbb{A}_{opt} \text{ accepts } pad(x, z)} \Rightarrow$

$\Rightarrow pad(x, z) \in Q \Rightarrow x \in Q$

Thus $im(\mathbb{F}) \subseteq Q$.

Sts: F p-simulates every other proof system F' for Q.

Let F' be computed by $\mathbb{F}'$

Then $\{ pad(x,z) \mid F'(z) = x \} \subseteq Q$.

$$\left(\text{if } F'(z) = x \implies x \in Q \implies pad(x,z) \in Q \right)$$

$$\{ \quad \text{---}\,\text{''}\,\text{---} \quad \} \in P$$

(given u, check $u \in im(pad)$

if so, compute x, z st $pad(x,z) = u$

check $F'(z) = x$

Since $\mathbb{A}_{opt}$ is almost optimal $\implies \mathbb{A}_{opt}(pad(x,z)) \leq |z|^{d_{F'}}$

p-simulation: $\quad z \longmapsto \langle \mathbb{F}', z, 1^{|z|^{d_{F'}}} \rangle$

works if F' is computable in time $c \cdot n^2$ $\square$

C. Hard sequences for proof systems

Def: Let F be a proof system for Q.

$(x_s)_{s \in \mathbb{N}}$ is a hard sequence for F iff.

a) $1^s \longmapsto x_s$ is ptime

b) $\forall s \in \mathbb{N}: \; x_s \in Q$

c) there is no ptime function π st

$$\forall s \in \mathbb{N} \quad F(\pi(1^s)) = x_s.$$

Ex: If a hard seq for F exists then F is not p-optimal.

(a solution is hidden in the proof of the next Thm)

THM: If every algorithm for Q has hard sequences, then every proof system for Q has hard sequences, namely, if $\mathbb{O}$ is an optimal inverter of F for Q and $(x_s)_{s \in \mathbb{N}}$ is a hard sequence for the alg. $\mathbb{O}^{dec}$, then $(x_s)_{s \in \mathbb{N}}$ is a hard sequence for F.