Introduction to Tree Language Theory

Hitoshi Ohsaki

 National Institute of
Advanced Industrial Science and Technology (AIST)

seminar talk (6/10)

2009

VI. P & NP

TM & k-TM

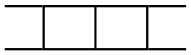
k-TM $(k \geqslant 1)$: MTM with

one control-unit

one read-only tape (called input tape)

k read/write tapes (called working tapes)

k-NTM : non-deterministic k-TM

Cf. TM $\triangleq$ one control-unit  + one read/write tape 

Corollary

TM $=$ k-TM $=$ k-NTM $(k \geqslant 1)$

Proof

The left equivalence follows from the fact that MTM is simulated by TM and the reverse is trivial. Using a similar proof of the previous fact (see the slides of the previous talk), one can show that k-NTM is simulated by NTM and the reverse also holds. Since NTM is simulated by TM, the right equivalence follows. □

Running time & working space

Given k-TM $\mathcal{M} = (\Gamma, \Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$

$\mathsf{time}_{\mathcal{M}}(w)$: the number of moves until $\mathcal{M}$ halts on input w
(called running time of $\mathcal{M}$ on w)

$\mathsf{space}_{\mathcal{M}}(w)$: the number of cells in working tapes which $\mathcal{M}$ visited at least once until $\mathcal{M}$ halts on input w
(called working space of $\mathcal{M}$ on w)

For the length n of input

$T_{\mathcal{M}}(n)$: $\max\{\, \mathsf{time}_{\mathcal{M}}(w) \mid w \in \Sigma^* : |w| = n \,\}$

$S_{\mathcal{M}}(n)$: $\max\{\, \mathsf{space}_{\mathcal{M}}(w) \mid w \in \Sigma^* : |w| = n \,\}$

Question

How should we define T_L (or S_L) for language L ?

Linear speed-up

Given k-TM $\mathcal{M}_1$, there exists $(k+1)$-TM $\mathcal{M}_2$ such that

- $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$
- $T_{\mathcal{M}_2}(n) \leqslant \frac{1}{2}\, T_{\mathcal{M}_1}(n)$ for sufficiently large $n \in \mathbb{N}$

if $\lim_{n\to\infty} T_{\mathcal{M}_1}(n)/n = \infty$

Proof

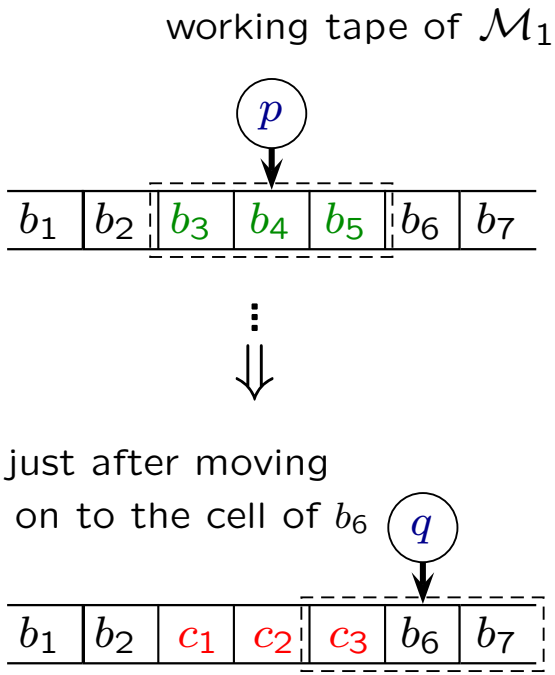
Consider $k = 1$, but the following proof can be generalized to arbitrary $k \geqslant 1$. Let $\mathcal{M}_1 = (\Gamma, \Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$, then define $\mathcal{M}_2 = (\Gamma', \Sigma, Q', \mathsf{q}'_0, Q'_{\mathsf{fin}}, \Delta')$ with $\Gamma' = (\Gamma \times \Gamma \times \Gamma) \cup \Sigma$, $Q' = \{\, (p, i) \mid p \in Q,\ 1 \leqslant i \leqslant 3 \,\} \cup \{\mathsf{q}'_0, \mathsf{q}_{\mathsf{acc}}, \mathsf{q}_{\mathsf{rej}}\}$ and $Q'_{\mathsf{fin}} = \{\, \mathsf{q}_{\mathsf{acc}} \,\}$. Here the blank for $\mathcal{M}_2$ is $(\sharp, \sharp, \sharp)$, and the states q'_0, $\mathsf{q}_{\mathsf{acc}}$, $\mathsf{q}_{\mathsf{rej}}$ are fresh symbols. Transition function Δ' is defined according to $\mathcal{M}_1$'s move on each 3 cells ("chunk" of cells at 2_{i-1}, 2_i, 2_{i+1}). That is, initially, for input $a_0\, a_1 \cdots a_n$, $\mathcal{M}_2$ copies it to (k+1)th working tape in compressed manner, e.g. $(\sharp, a_0, a_1)\,(a_1, a_2, a_3) \cdots (a_{n-1}, a_n, \sharp)$ if n is even ; $(\sharp, a_0, a_1)\,(a_1, a_2, a_3) \cdots (a_{n-2}, a_{n-1}, a_n)$ if n is odd. For this initialization, including the running time to return the tape-head on (k+1)th working tape to the leftmost position, $\mathcal{M}_2$ needs $n + \frac{1}{2}n + c$ moves (c constant). Next, using (k+1)th working tape as the input tape of $\mathcal{M}_2$, $\mathcal{M}_2$ simulates $\mathcal{M}_1$. (proof cont'd)

Proof (cont'd)

For the move of $\mathcal{M}_2$, suppose, for instance, that $\mathcal{M}_1$ with the state p reads y_i on the input tape, and then $\mathcal{M}_1$ moves on the working tape as shown in the figure. If $\mathcal{M}_1$'s tape-head on the input tape stays, say on y_{i+1}, in the same chunk even after the other head moves on to the cell of b_6, Δ' contains the following map :

$$\langle (p,2), (\sharp,\sharp,\sharp), (y_{i-1}, y_i, y_{i+1}), (b_3, b_4, b_5) \rangle \mapsto \langle (q,3), (\sharp,\sharp,\sharp), (y_{i-1}, y_i, y_{i+1}), (c_1, c_2, c_3), \mathsf{S}, \mathsf{R}, \mathsf{S} \rangle$$

On the right-hand side of the above mapping, "S" stands for *stay*. Because $\mathcal{M}_2$'s head on the input tape is no longer needed, it keeps staying on a blank. The head on $(k+1)$th-working tape also stays on the same cell, but the the current state must be changed from $(p,2)$, meaning that $\mathcal{M}_1$'s tape-head reads y_i, to $(q,3)$, meaning that the head reads y_{i+1}. During the computation, $\mathcal{M}_2$ should halt on $\mathsf{q}_{\mathsf{acc}}$ (resp. $\mathsf{q}_{\mathsf{rej}}$) if $\mathcal{M}_1$ accepts (rejects) the input. Moreover, since $\mathcal{M}_1$ is deterministic, $\mathcal{M}_2$ must be deterministic. By construction, the simulation can be done within at most $\frac{1}{2}T_{\mathcal{M}_1}(n)+c'$ (c' constant) moves. Since $\lim_{n\to\infty} T_{\mathcal{M}_1}(n)/n = \infty$, we have that for large $n \in \mathbb{N}$, $T_{\mathcal{M}_2}(n) \leqslant \frac{1}{2}T_{\mathcal{M}_1}(n)$. □



Remarks

1. The size of chunk is *not* necessarily 3. When the size is i ($\geqslant 3$), the simulation can be done within $\frac{1}{i-1}\, T_{\mathcal{M}_1}(n)$
2. $\mathcal{M}_2$ requires $|\Sigma|^i$ tape symbols and $(|Q| \times i) + 3$ state symbols
3. If $\lim_{n\to\infty} T_{\mathcal{M}_1}(n)/n^2 = \infty$, one can construct k-TM $\mathcal{M}_2$ that simulates k-TM $\mathcal{M}_1$ (without introducing a new working tape)
4. From the proof, one can see that $S_{\mathcal{M}_2}(w)$ is also improved :

Proposition (Tape compression)

Given k-TM $\mathcal{M}_1$, there exists $(k+1)$-TM $\mathcal{M}_2$ such that

- $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$
- $S_{\mathcal{M}_2}(n) \leqslant \frac{1}{2}\, S_{\mathcal{M}_1}(n)$ for sufficiently large $n \in \mathbb{N}$

Note

If the size of the chunk is i ($i \geqslant 3$), the simulation can be done within $\frac{1}{i-1}\, S_{\mathcal{M}_1}(n)$

DTIME & DSPACE

Let T, S be 1-variable polynomials with positive coefficients

DTIME(T) : languages accepted by k-TM $\mathcal{M}$ such that

$T_{\mathcal{M}}(n) \leqslant T(n)$ for every (sufficiently large) $n \in \mathbb{N}$

DSPACE(S) : languages accepted by k-TM $\mathcal{M}$ such that

$S_{\mathcal{M}}(n) \leqslant S(n)$ for every (sufficiently large) $n \in \mathbb{N}$

Let $\mathcal{P}$ be the set of 1-variable polynomials with positive coefficients

$$\mathrm{P} : \bigcup_{T \in \mathcal{P}} \mathrm{DTIME}(T)$$

$$\mathrm{PSPACE} : \bigcup_{S \in \mathcal{P}} \mathrm{DSPACE}(S)$$

Note

P $\subseteq$ PSPACE

($\because$ Polynomially time-bounded computation requires at most polynomial tape space)

NTIME & NSPACE

Let T, S be 1-variable polynomials with positive coefficients

NTIME(T) : languages accepted by k-NTM $\mathcal{M}$ such that

$T_{\mathcal{M}}(n) \leqslant T(n)$ for every (sufficiently large) $n \in \mathbb{N}$

NSPACE(S) : languages accepted by k-NTM $\mathcal{M}$ such that

$S_{\mathcal{M}}(n) \leqslant S(n)$ for every (sufficiently large) $n \in \mathbb{N}$

Let $\mathcal{P}$ be the set of 1-variable polynomials with positive coefficients

$$\text{NP} : \bigcup_{T \in \mathcal{P}} \text{NTIME}(T)$$

$$\text{NPSPACE} : \bigcup_{S \in \mathcal{P}} \text{NSPACE}(S)$$

Note

P $\subseteq$ NP $\subseteq$ NPSPACE, PSPACE $\subseteq$ NPSPACE

Example of P

The membership problem for CFG in Chomsky normal form :

instance is grammar $\mathcal{G} = (\Sigma, Q, \mathrm{q}_0, Q_{\mathrm{fin}}, \Delta)$, word w in Σ^*

solution is "yes" if $w \in \mathcal{L}(\mathcal{G})$; "no" otherwise

Proof (for the problem being in P)

Let $w = a_1\, a_2 \cdots a_n$ $(n \geqslant 0)$. By assumption of the problem, transition rules in Δ are in the forms of $p \to q\, r$, $p \to a$, $\mathrm{q}_0 \to \varepsilon$ for $p \in Q$, $q, r \in Q - \{q_0\}$, $a \in \Sigma$.

1. If $n = 0$ and $\mathrm{q}_0 \to \varepsilon \in \Delta$, return "yes."
2. Next, execute the following program :

```
for ( i = 1 , i ⩽ n , i++ )
  if ∃ p → a_i in Δ
    T(i,i) := T(i,i) ∪ {p} ;
```

3. Then, execute the program on the right.
4. If $\mathrm{q}_0 \in T(1, n)$, return "yes" ; otherwise return "no."

```
for ( ℓ = 2 , ℓ ⩽ n , ℓ++ )
  for ( i = 1 , i ⩽ n - ℓ + 1 , i++ )
    j := i + ℓ - 2 ;
    for ( k = i , k ⩽ j , k++ )
      if ∃ p → q r in Δ &
         ∃ q ∈ T(i,k) &
         ∃ r ∈ T(k+1,j)
      then T(i,j) := T(i,j) ∪ {p} ;
```

The innermost loop is done in $k_1 \times n$ running time, hence this computation is done in at most $k_2 \times n^3$ (k_1, k_2 constants). This implies that the problem is in DTIME(n^3).□

Example of NP

The bounded halting problem for NTM :

instance is NTM $\mathcal{M} = (\Gamma, \Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$, w in Σ^*, k in $\mathbb{N}$

solution is "yes" if $w \in \mathcal{L}(\mathcal{M})$ within k moves ; "no" otherwise

Proof

Suppose $\langle \Psi(\mathcal{M}), \Psi(w) \rangle$ is the binary code for non-deterministic UTM $\mathcal{M}_{\mathsf{NU}}$, then the question if $w \in \mathcal{L}(\mathcal{M})$ is the membership problem $\langle \Psi(\mathcal{M}), \Psi(w) \rangle \in \mathcal{L}(\mathcal{M}_{\mathsf{NU}})$. Let n be the size of input for this problem, then the total length of the word that represents $\mathcal{M}$ and w with delimiters is at most $2|\Gamma| + 2|Q| + |\Gamma|^2 \times |Q|^2 + |w| + c_1$. Encoding $\mathcal{M}$ and w by Ψ takes $(c_2 \times |\Gamma|^2 \times |Q|^2) + (c_3 \times |w|)$ running time in proportion to n. The simulation of $\mathcal{M}$ takes $(c_4 \times |\Gamma|^2 \times |Q|^2) + c_5$ for each move. In this simulation, at each step, $\mathcal{M}_{\mathsf{NU}}$ tries to find a transition rule which can be applied, and then reflects the result on the tapes. By assumption, the simulation must be stopped within k iterations. Hence, the problem is in NTIME(n). □

Consult books (e.g. [1]) or search on the web for more examples of NP-problems.

---

[1] M.R. Garey & D.S. Johnson: *Computers and Intractability – A Guide to the Theory of NP-completeness*, Freeman, 1979.

Polynomial slow-down (simulation of MTM)

2-TM $\mathcal{M}_1$ can be simulated by TM $\mathcal{M}_2$ in $c \times T_{\mathcal{M}_1}(n)^2$ (c : constant)

Proof

Let Γ be the set of tape symbols of $\mathcal{M}_1$. Similar to the proof of "MTM=TM" in the previous talk on TM, suppose $\diamond$ is a fresh symbol, and then, define (single tape) TM $\mathcal{M}_2$ whose tape symbols are $\Gamma \cup \{\diamond\} \cup \{\overline{a} \mid a \in \Gamma\}$. In this proof, $\mathcal{M}_2$'s single tape is divided into 2 tracks as shown in the figure. The symbols in input tape and working tape are alternately placed on the single tape. The left- and right-end of the sequence are marked by $\diamond$. The locations of $\mathcal{M}_1$'s tape-heads on the input tape and on the working tape are represented by symbols with overline, e.g. $\overline{a}$. Let n be the length of the input. For each move, the simulation takes at most $2 \times T_{\mathcal{M}_1}(n) + 9$ moves, where $8\,(= 2 \times 4)$ moves for adjusting the head and overwriting overlined-symbols to normal symbols, and 1 move for reading $\diamond$. Since the number of iterations is $T_{\mathcal{M}_1}(n)$, the simulation takes $2 \times T_{\mathcal{M}_1}(n)^2 + 9 \times T_{\mathcal{M}_1}(n)$ in total. For the initialization, it takes $c_1 \times n^2$ (c_1 constant). □

$\mathcal{M}_1$

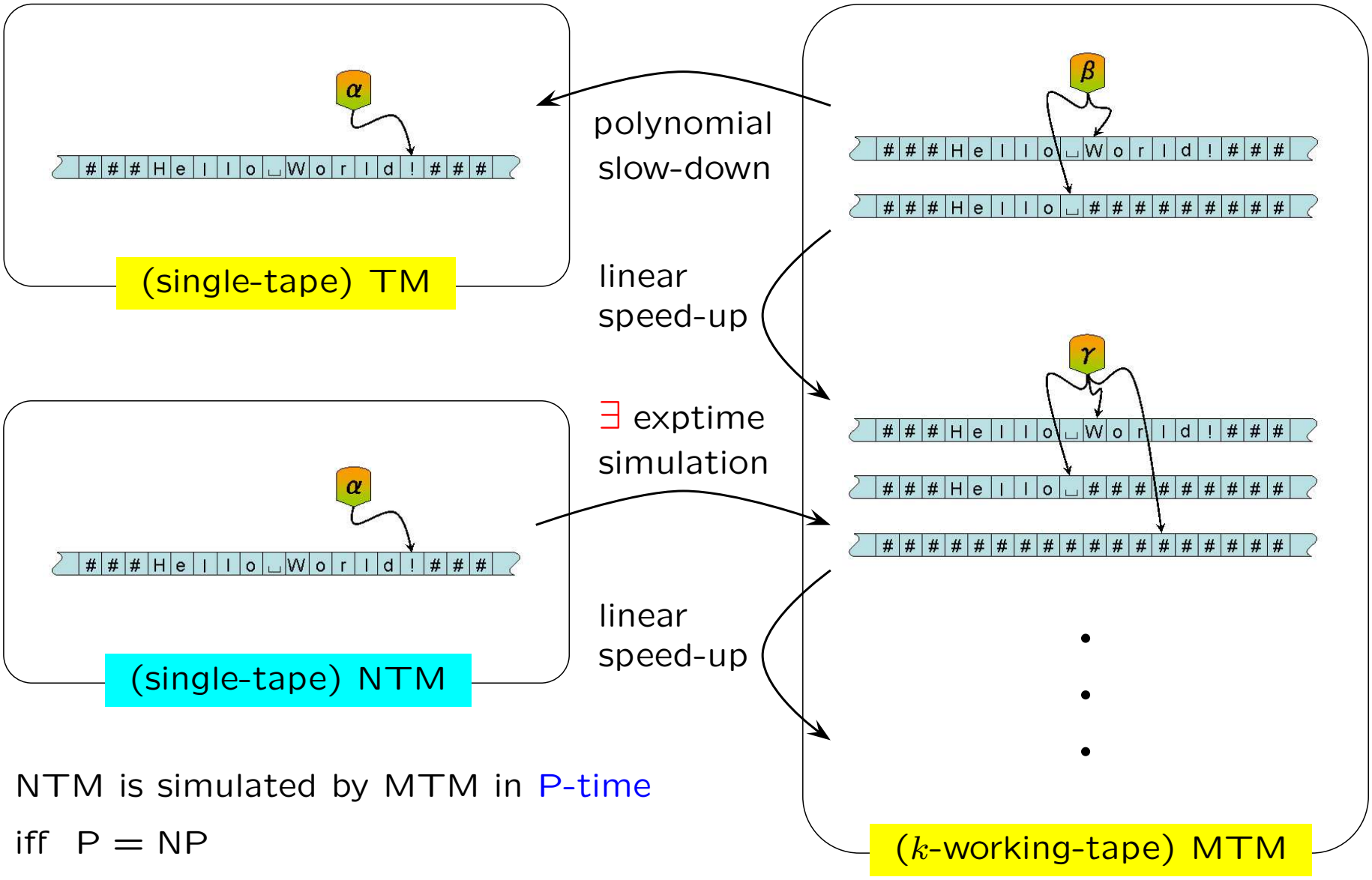
input tape	♯	a	b	c	♯
working tape	♯	f	g	♯	♯

$\mathcal{M}_2$

♯	$\diamond$	a	f	b	g	c	♯	$\diamond$	♯

Note that the above proof can be generalized to the simulation of k-TM ($k \geqslant 2$).

Summary of TM & MTM & NTM



NTM is simulated by MTM in P-time

iff P = NP

Reducibility

Given languages L over Σ, M over Γ $(\Sigma \subseteq \Gamma)$

$L \leqslant_{\mathsf{m}} M$: L is many-one reducible to M if

$\exists$ k-TM $\mathcal{M}$ such that for every input $w \in \Sigma^*$,

$w \in L$ if and only if $\mathcal{M}$ halts on w with $w' \in M$

as output on working tape k

$(\exists f$ computable function : $w \in L$ iff $f(w) \in M)$

$L \leqslant^{P}_{\mathsf{m}} M$: L is polynomial-time reducible to M if

$L \leqslant_{\mathsf{m}} M$ & $T_{\mathcal{M}}(n) \leqslant T(n)$ for polynomial T

$(\exists f$ polynomial function : $w \in L$ iff $f(w) \in M)$

$L \leqslant^{\log}_{\mathsf{m}} M$: L is log-space reducible to M

$L \leqslant_{\mathsf{m}} M$ & $S_{\mathcal{M}}(n) \leqslant c \times \log n$ (c : constant)

* kth-tape is write-only and is not taken into account as working space

P- and NP-completeness

Given language L over Σ

L is called NP-hard if for every language $M \in \mathsf{NP}$, $M \leqslant_{\mathsf{m}}^{P} L$

L is called NP-complete if L is NP-hard & $L \in \mathsf{NP}$

L is called P-hard if for every language $M \in \mathsf{P}$, $M \leqslant_{\mathsf{m}}^{\log} L$

L is called P-complete if L is P-hard & $L \in \mathsf{P}$

Note

$\leqslant_{\mathsf{m}}^{\log} \subseteq \leqslant_{\mathsf{m}}^{P}$

$\because$ $\mathsf{DSPACE}(T(n)) \subseteq \mathsf{NSPACE}(T(n)) \subseteq \bigcup_{c>0} \mathsf{DTIME}(2^{c \times T(n)})$ if $T(n) \geqslant \log(n)$

Moreover, the following statements are equivalent :

1. $\mathsf{P} = \mathsf{NP}$ ($1 \Rightarrow 3$ From previous observation)
2. NP-complete problem is in P ($2 \Rightarrow 1$ P, NP are closed under $\leqslant_{\mathsf{m}}^{P}$)
3. P-complete problem is NP-complete ($3 \Rightarrow 2$ Obvious)

P- and NP-complete problems (tricky example)

The bounded halting problem for TM :

instance is TM $\mathcal{M} = (\Gamma, \Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$, w in Σ^*, k in $\mathbb{N}$

solution is "yes" if $w \in \mathcal{L}(\mathcal{M})$ within k moves ; "no" otherwise

This problem is P-complete

Proof

Similar to the *bounded halting problem for NTM*, it is easy to show that this problem is in P. Next, let L be a language over Σ in P. Then, there exists TM $\mathcal{M}_L$ with polynomial $T(n)$ such that $L = \mathcal{L}(\mathcal{M}_L)$ and for every $w \in L$, $\mathcal{M}_L$ halts on w within $T(|w|)$ running time. Take this TM $\mathcal{M}_L$, an (arbitrary) word w from Σ^*, and $k = T(|w|)$, then $w \in L$ if and only if $\mathcal{M}_L$ halts on w within k moves. □

The above example can be modified as an example of NP-complete problems, which means that :

The bounded halting problem for NTM is NP-complete

Other NP-complete problems

Satisfiability problem for Boolean formulas [Cook 1971] :

instance is propositional Boolean formula ϕ

solution is "yes" if there is an assignment that satisfies ϕ ;
"no" otherwise

Hamilton circuit in directed graphs :

instance is directed graph G

solution is "yes" if there is a cycle that passes through every vertex in G exactly once ; "no" otherwise

Clique in graphs :

instance is directed graph G, k in $\mathbb{N}$

solution is "yes" if G has a complete sub-graph (each pair of vertices in the sub-graph is connected by an edge) of size k ; "no" otherwise

Savitch's theorem

PSPACE = NPSPACE

Proof

It suffices to show $\supseteq$, because the reverse is trivial. Let L be a language in NPSPACE, then there exists k-NTM $\mathcal{M}_1 = (\Gamma, \Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$ which halts on input w using at most $S(|w|)$ space. Suppose $k = 1$, but this proof can be generalized to any $k \geqslant 1$. We construct below k'-TM $\mathcal{M}_2$ that accepts L and that uses at most $S(|w|)^2$ space. Assume for $\mathcal{M}_2$ that it accepts w when and only when $\mathcal{M}_2$ clean up the working tape (that contains only blanks) and halts with the final state $\mathsf{q}_{\mathsf{fin}}$ and the tape-head on the input tape is placed at leftmost position.

Observation : Since $\mathcal{M}_1$ uses for input w at most $S(|w|)$ space, $w \in L$ if and only if w is accepted within $|\Gamma|^{S(|w|)} \times (|Q| \times S(|w|) \times |w|)$ moves, where $|\Gamma|^{S(|w|)}$ is the number of possible configurations of working tape and $(|Q| \times S(|w|) \times |w|)$ is the possible combinations of state and positions of (two) tape-heads. Hence, $w \in L$ if and only if w is accepted within $2^{c \times S(|w|)}$ moves (c : constant and computable).

This observation brings the idea to define k'-TM $\mathcal{M}_2$ such that $\mathcal{M}_2$ accepts input w if w is accepted by $\mathcal{M}_1$ within $2^{c \times S(|w|)}$ moves ; $\mathcal{M}_2$ rejects w otherwise. One should notice that $\mathcal{M}_2$ does not necessarily simulate $\mathcal{M}_1$ move by move, but it has to determine whether w is accepted by $\mathcal{M}_1$ within $2^{c \times S(|w|)}$ moves. (Proof cont'd)

Proof (cont'd)

Define the procedure reach(α, β, k) on the right : α, β are *configurations* (words of input and working tapes together with the current state and the locations of tape-heads) of $\mathcal{M}_1$, k is a non-negative integer, and C is initially the set of all configurations of $\mathcal{M}_1$ whose length is less than or equals to $S(|w|)+|w|$. Let α_0 and α_{fin} be the initial and final configurations. Given w, by assumption, they are unique. If reach$(\alpha_0, \alpha_{\text{fin}}, 2^{c\times S(|w|)})$ = "yes," there exists a sequence $\alpha_0, \alpha_1, \ldots, \alpha_n (=\alpha_{\text{fin}})$ such that $\mathcal{M}_1$ moves from α_i to α_{i+1}, or $\alpha_i = \alpha_{i+1}$, so w is accepted by $\mathcal{M}_1$. If reach$(\alpha_0, \alpha_{\text{fin}}, 2^{c\times S(|w|)})$ = "no," w is not accepted by $\mathcal{M}_1$. For reach$(\alpha_0, \alpha_{\text{fin}}, 2^{c\times S(|w|)})$, there are recursive calls in the program at most $\log_2 2^{c\times S(|w|)} = c\times S(|w|)$ times. Each stack frame requires at most $2\times S(|w|)$ space. Hence, this program can be realized by k'-TM using $2c\times S(|w|)^2$ space. □

```
reach( α , β , k )
  if  k = 0  &  α = β
      then  return "yes" ;

  if  k = 1  &  M1 moves from α to β in one step
      then  return "yes" ;

  if  k ⩾ 2  then
      while ∃ γ ∈ C do
        if reach( α, γ, ⌈k/2⌉ ) = "yes"  &
           reach( γ, β, ⌊k/2⌋ ) = "yes"
        then  return "yes"  break ;
        else  C := C − {γ} ;
      od
      return "no"
```

Corollary P ⊆ NP ⊆ PSPACE = NPSPACE

Exercise

1. Show that if languages $A, B \in \mathsf{P}$, then $A \cup B$, $A \cap B$, $(A)^{\mathsf{C}}$, $A \cdot B \in \mathsf{P}$.
2. Show that if languages $A, B \in \mathsf{NP}$, then $A \cup B$, $A \cap B$, $A \cdot B \in \mathsf{NP}$.
3. Show that for all languages A, B, C over Σ, $A \leqslant_{\mathsf{m}}^{P} A$ (reflexivity), $A \leqslant_{\mathsf{m}}^{P} B \ \& \ B \leqslant_{\mathsf{m}}^{P} C \Rightarrow A \leqslant_{\mathsf{m}}^{P} C$ (transitivity), $A \leqslant_{\mathsf{m}}^{P} B \Rightarrow (A)^{\mathsf{c}} \leqslant_{\mathsf{m}}^{P} (B)^{\mathsf{c}}$.
4. Which of the following statements hold? Explain the reason why.
   (a) $A \leqslant_{\mathsf{m}}^{P} B \ \ \& \ \ B \in \mathsf{P} \ \Rightarrow \ A \in \mathsf{P}$
   (b) $A \leqslant_{\mathsf{m}}^{P} B \ \ \& \ \ B \in \mathsf{NP} \ \Rightarrow \ A \in \mathsf{NP}$
   (c) $A \leqslant_{\mathsf{m}}^{P} B \ \ \& \ \ B \in \mathsf{PSPACE} \ \Rightarrow \ A \in \mathsf{PSPACE}$
   (d) $A \leqslant_{\mathsf{m}}^{P} B \Rightarrow A \leqslant_{\mathsf{m}}^{P} (B)^{\mathsf{c}}$
5. Select one of the problems on page 16 (or find another example), and then show that this problem is NP-complete.
6. Show that $\mathsf{P} = \mathsf{NP}$ if and only if a finite language is NP-complete.
7. Show that the following problem is NP-complete : Given a finite set U, subsets $S_1, \cdots, S_n \, (\subseteq U)$ and an integer k, the question if there exist k subsets $S_{i_1}, \cdots, S_{i_k}$ such that $S_{i_1} \cup \cdots \cup S_{i_k} = U$.

Appendix : What if P = NP ...

Most of researchers believe $P \neq NP$ today. But if this long-standing open question "$P = NP$?" is positively solved, what happens ? :

It turns out

- NP = coNP, P = PH (the union of all complexity classes in the polynomial hierarchy), and thus, polynomial hierarchy is collapsed

Furthermore

- SSL, RSA, PGP are no longer secure infrastructures, so
- E-commerce business is exposed to serious menace of security flaws

On the other hand

- better predictions of weather, earthquakes and other natural phenomena are established
- mathematicians could be replaced by efficient theorem-discovering programs (Gödel 1956).

Additionally

- one who has solved the question first receives the prize of $1M from CMI [1]

---

[1] *P vs NP: Millennium Prize Problems*, Clay Mathematics Institute of Cambridge, USA, May 2000. Information available at `http://www.claymath.org/millennium/`

National Institute of Advanced Industrial Science and Technology (AIST) – Senri-site, AIST Kansai.

Office: Shin-Senri Nishi 1–2–14 (MSK bldg. 5th floor), Toyonaka, Osaka 560–0083, Japan

URL: `http://staff.aist.go.jp/hitoshi.ohsaki/`