I216E Computational Complexity and Discrete Mathematics

by
Prof. Ryuhei Uehara
and
Prof. Eiichiro Fujisaki

I216E 計算量の理論と離散数学

上原隆平&藤崎英一郎

5. Computational Complexity

5.3. Class NP

5.3.*. Nondeterministic computation

Some problems (like 3SAT, DHAM, etc.) have a *common and natural* property;

- once you get a solution, you can check it efficiently
- without solution, it seems to be quite difficult; you may check all possibilities

- Many natural problems have this property in the real problems.
- This property leads us to the notion of "nondeterministic computation"

5.計算量の理論

5.3.クラスNP

5.3.*. 非決定性計算とは

(3SAT, DHAMといった)ある種の問題には，次のような**共通で自然な**性質がある;

- ひとたび解が得られると，その正当性は簡単にチェックできる
- 解を見つけるのは大変そうに思える．可能な場合をしらみつぶしに調べる必要がありそうに見える．

- 現実の自然な問題の多くはこの性質をもつ．
- この性質を表現するのが「非決定性計算」

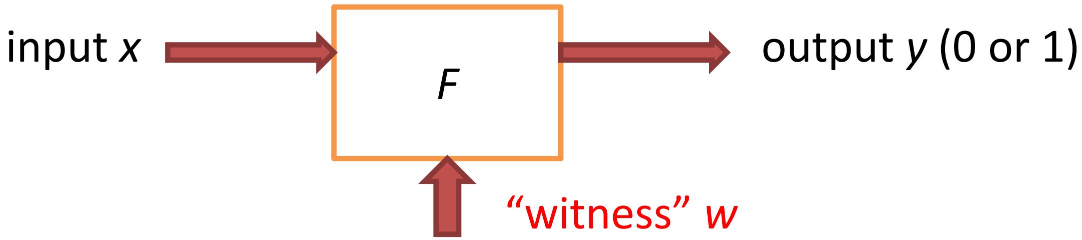
5. Computational Complexity

5.3. Class NP

5.3.*. Nondeterministic computation

"Nondeterministic computation"

- From the viewpoint of Function:



L is called an NP set if there is a function *F* s.t.

1. For each *x*, there is a binary string "witness" *w* s.t.
2. $|w|$ is bounded by a polynomial of $|x|$
3. *F* recognizes $x \in L$ with *w* in polynomial time of $|x|$ and $|w|$

c.f.: NP=**N**ondeterministic **P**olynomial

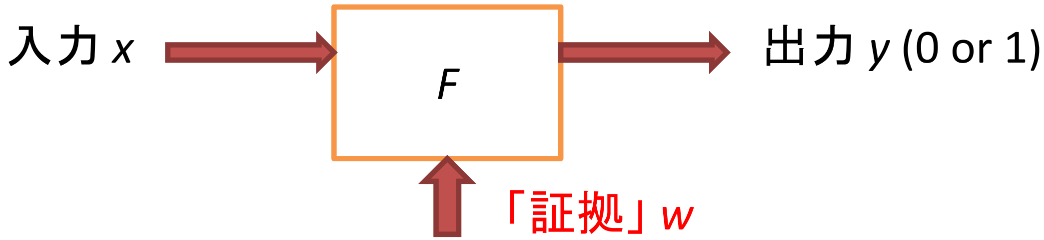
5.計算量の理論

5.3.クラスNP

5.3.*.非決定性計算とは

「非決定性計算」

- 関数の観点からみると:



以下の関数*F*が存在するときL はNP 集合と呼ばれる:

1. 各*x*に対して，2進列の「**証拠**」*w* が存在
2. |*w*| は |*x*|の多項式で上から抑えられる
3. *F* は|*x*|と|*w*|の多項式時間で*w*を使ってx∈*L* を認識する

c.f.: NP=**N**ondeterministic **P**olynomial

5. Computational Complexity

5.3. Class NP

5.3.*. Nondeterministic computation

"Nondeterministic computation"

- From the viewpoint of Logic:

Such a string w is called "witness"

Suppose that we have a polynomial q and polynomial time computable predicate R for a set L such that

$$\text{for each } x \in \Sigma^*, x \in L \leftrightarrow \exists w \in \Sigma^* : |w| \le q(|x|)[R(x,w)]$$

i.e.,

$$L = \{x : \exists w \in \Sigma^* [\, |w| \le q(|x|) \wedge R(x,w)]\}$$

Then, L is called an NP set, and the problem of recognizing L is called an NP problem.

Also, the whole set of NP sets is called the class NP.

c.f.: NP=**N**ondeterministic **P**olynomial

5.計算量の理論

5.3.クラスNP

5.3.*.非決定性計算とは

「非決定性計算」

- 論理の視点からみると:

この文字列 w を「証拠」とよぶ

集合Lに対して多項式qと多項式で計算できる述語Rがあり,
以下を満たすとする:

$$\text{for each } x \in \Sigma^*, x \in L \leftrightarrow \exists w \in \Sigma^* :|w| \le q(|x|)[R(x,w)]$$

つまり,
$$L = \{x : \exists w \in \Sigma^* [\, |w| \le q(|x|) \wedge R(x,w)]\}$$

このとき L はNP 集合とよばれ,
Lの認識問題はNP 問題とよばれる.
また NP 集合全体の集合をクラスNPとよぶ.

c.f.: NP=**N**ondeterministic **P**olynomial

5. Computational Complexity

5.3. Class NP

5.3.*. Nondeterministic computation

A "nondeterministic choice" is a kind of parallel computing that generates two branches.

"Nondeterministic computation"

- From the viewpoint of Turing Machine:

  Suppose that Turing machine has "nondeterministic choice" that admits us to two possible choices at the same time; i.e., it has "one of two cases (0) and (1)" statement.

  - A nondeterministic choice allows to assume of two choices and it will be "*true*" if "*at least one of them is true*".

Then, NP problem L can be recognized by a nondeterministic Turing machine in polynomial time.

c.f.: NP=**N**ondeterministic **P**olynomial

5.計算量の理論

5.3.クラスNP

5.3.*.非決定性計算とは

「非決定性選択」はある種の並列計算とみなすこともでき，二つの計算プロセスの生成と考えてもよい．

「非決定性計算」
- チューリングマシンの視点から見ると:

チューリングマシンの「非決定性選択」では，二つの選択肢を「同時に」二つとも選ぶことができる;
つまり「場合(0)と場合(1)のいずれか」という命令がある．

- 非決定性選択は二つの選択肢のうち，「いずれか一方が真」ならば真になる．

このときNP 問題 Lは，非決定性チューリング機械で多項式時間で受理できる問題．

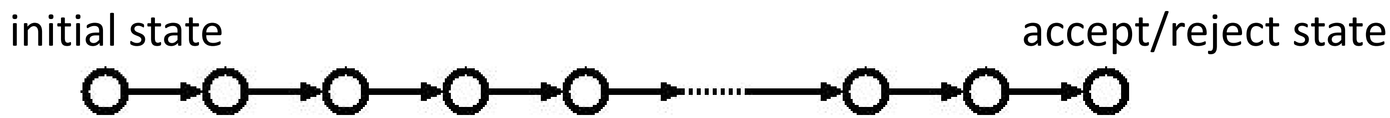
c.f.: NP=**N**ondeterministic **P**olynomial

5. Computational Complexity

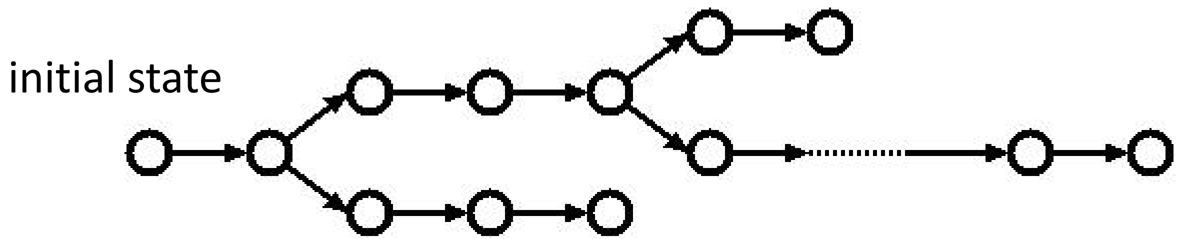
5.3. Class NP

5.3.*. Nondeterministic computation

- From the viewpoint of the computation tree of a Turing Machine:
  - Computation tree of a deterministic Turing machine forms a path;



  - Computation tree of a *nondeterministic* Turing machine forms a *tree*;



- each computation halts in an accept/reject state or loop.
- it accepts if the tree has at least one “accept” in poly-length.

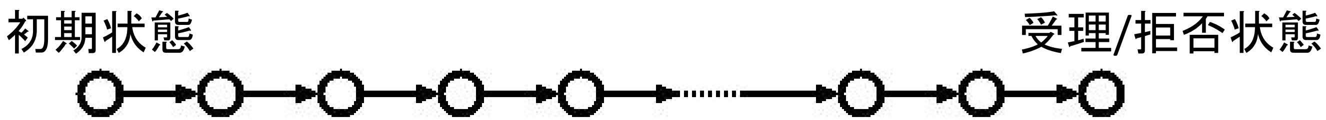
5.計算量の理論

5.3.クラスNP

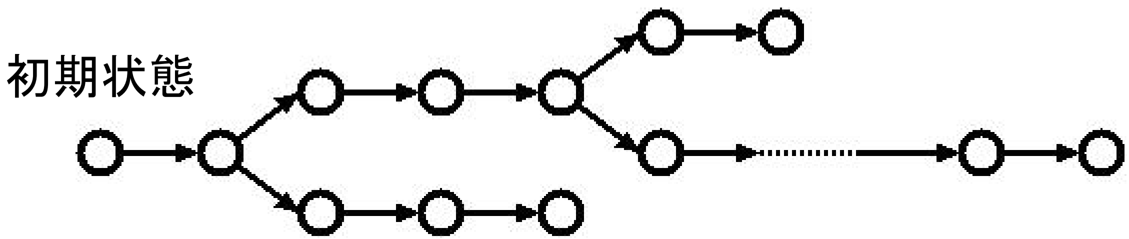
5.3.*.非決定性計算とは

- チューリングマシンの計算木の観点からみると:

  - 決定性のチューリングマシンの計算木はパス(一本道);



  - 非決定性のチューリングマシンの計算木は木;



- 各計算プロセスは受理/拒否状態になるか無限ループ
- 木が多項式長の範囲で受理状態を一つでももてば受理.

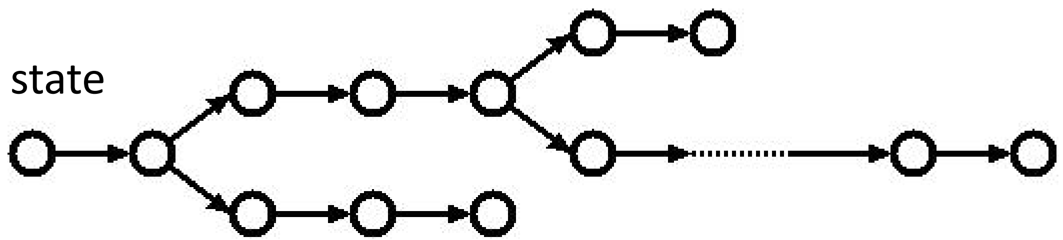
5. Computational Complexity

5.3. Class NP

5.3.*. Nondeterministic computation

The *witness* w gives the right choices

- From the viewpoint of the computation tree of a Turing Machine:
  - Computation tree of a *nondeterministic* Turing machine forms a *tree*;



An NP problem L is recognized by a nondeterministic Turing machine in polynomial time. That is, there is a computation path to an accept state of length polynomial of n.

5.計算量の理論

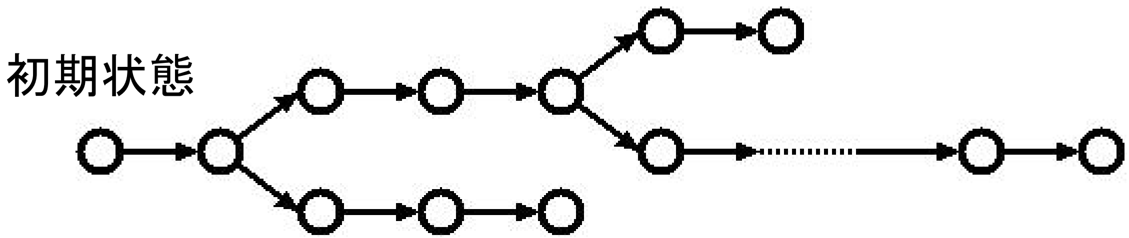
5.3.クラスNP

5.3.*.非決定性計算とは

証拠*w*は正しい選択肢の列を与える

•チューリングマシンの計算木の観点からみると:

•非決定性のチューリングマシンの計算木は木;



• 各計算プロセスは受理/拒否状態になるか無限ループ
• 木が多項式長の範囲で受理状態を一つでももてば受理.

NP 問題 L とは非決定性チューリング機械で
多項式時間で認識できる言語．つまり，受理状態に至る
n の多項式長の計算パスが存在すればよい.

5. Computational Complexity

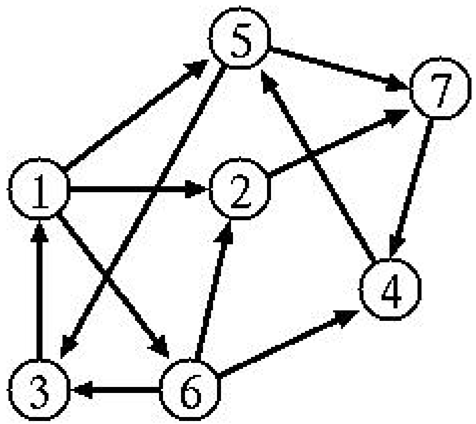
5.3. Class NP

5.3.1. Representative NP problems

- Hamiltonian cycle problem (DHAM)

  Input : <*G*>: a directed graph *G*

  Question : Does *G* have a Hamiltonian cycle?



- We can certainly check all possible permutations of n, that counts up to $n! \sim n^n$…
  it takes exponential time.
- If *G* has a Hamiltonian cycle *C*, and we have it as a witness, we can check that it surely a Hamiltonian cycle.

5.計算量の理論

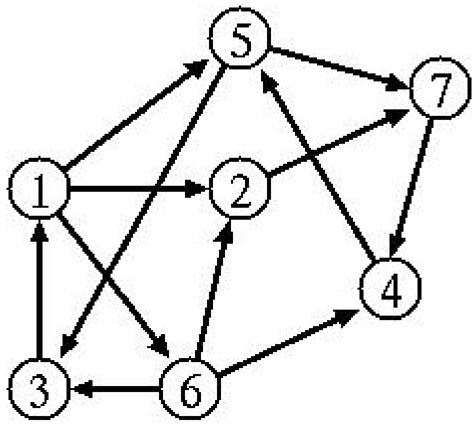
5.3.クラスNP

5.3.1. 代表的なNP問題

- ハミルトン閉路問題 (DHAM)

  入力 : <*G*>: 有向グラフ*G*

  質問 : *G* はハミルトン閉路をもつか?

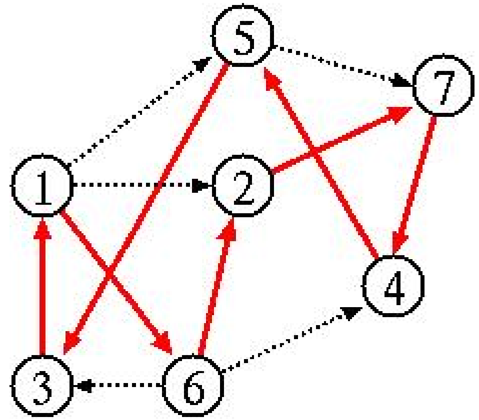


- 原理的には*n*の順列をすべて試せばよいが，可能な組合せの数は最大で$n! \sim n^n$...
  指数時間かかってしまう.
- もし *G* がハミルトン閉路Cをもつならこれを証拠にすれば，効率よくそれをチェックすることができる.

1,6,2,7,4,5,3,1

5. Computational Complexity

5.3. Class NP

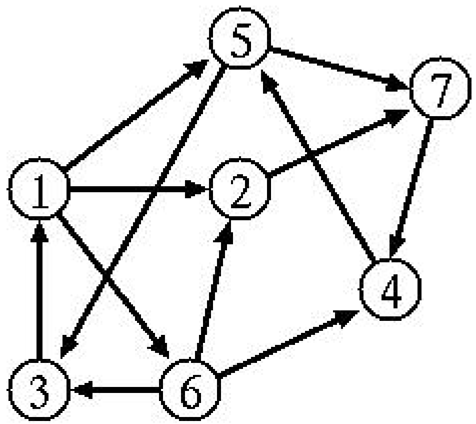


5.3.1. Representative NP problems

- Hamiltonian cycle problem (DHAM)

  Input : <*G*>: a directed graph *G*

  Question : Does *G* have a Hamiltonian cycle?



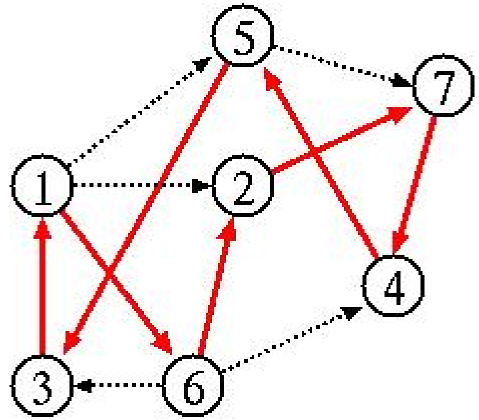
- We can certainly check all possible permutations of n, that counts up to $n! \sim n^n$…
  it takes exponential time.
- If *G* has a Hamiltonian cycle *C*, and we have it as a witness, we can check that it surely a Hamiltonian cycle.

1,6,2,7,4,5,3,1

5.計算量の理論

5.3.クラスNP

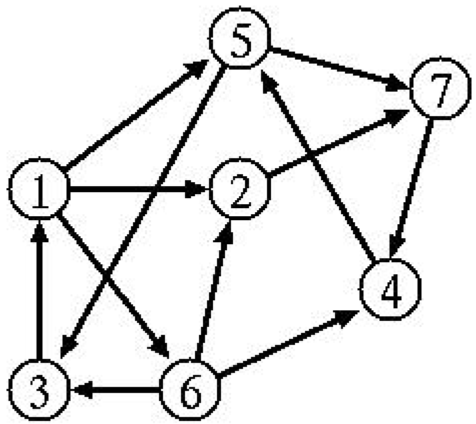
5.3.1.代表的なNP問題



- ハミルトン閉路問題 (DHAM)

入力: <*G*>: 有向グラフ*G*

質問: *G* はハミルトン閉路をもつか?



- 原理的にはnの順列をすべて試せばよいが，可能な組合せの数は最大で$n!\sim n^n$...
  指数時間かかってしまう.
- もし *G* がハミルトン閉路Cをもつならこれを証拠にすれば，効率よくそれをチェックすることができる.

5. Computational Complexity

5.3. Class NP

5.3.1. Representative NP problems

- SAT, *k*SAT, ExSAT (Satisfiability)

  Input : <*F*> *F* is conjunctive normal form

  Question : Any assignment s. t. $F(a_1, a_2, \dots, a_n) = 1$?

- If *F* is satisfiable by an assignment *A*, and we have it as a witness, we can check it in polynomial time by the same way as the PROP_EVAL.

- We can certainly check all possible assignments of $(a_1, a_2, \dots, a_n)$. The assignments are 2^n, that takes exponential time.

5.計算量の理論

5.3.クラスNP

5.3.1.代表的なNP問題

- SAT, *k*SAT, ExSAT (充足可能性)
  入力：<*F*> *F* は和積標準形命題論理式
  質問：$F(a_1, a_2, \ldots, a_n) = 1$となる割当ては存在?

•*F* を充足する割当て *A* があるなら,
それを証拠として使い，PROP_EVALのときと同じ方法で
多項式時間でチェックできる.

- もちろん$(a_1, a_2, \ldots, a_n)$のすべての可能な割当てを
チェックすることはできるが，可能な割当ての個数は
2^n なので，指数時間かかる.

5. Computational Complexity

5.3. Class NP

5.3.2. Another aspect of the NP problems

- What does it mean by being an NP set?
  - Using q and R satisfying the predicate characterizing an NP set, we can determine "$x \in L$?" in the following way.

```
for each w ∈ Σ^{≤q(|x|)} do
  if R(x, w) then accept end-if
end-for;
reject;
```

If we enumerate and check all possible strings of length at most $q(|x|)$, we can accept or reject them.
Here note that there are $2^{q(|x|)}$ (exponentially many) such strings.

We may think that those sets recognizable as above are NP sets.

5.計算量の理論

5.3.クラスNP

5.3.2. NP問題を別の視点から見る

- NP 集合であることの意味は?
  - 命題述語論理によるNP 集合の特徴付けで出てきた q と R を使うと,「$x \in L$?」という質問に次のアルゴリズムで答えることができる.

for each $w \in \Sigma^{\leq q(|x|)}$ do
  if $R(x, w)$ then accept end-if
end-for;
reject;

長さ高々$q(|x|)$のすべての文字列を辞書式に列挙してチェックすれば，受理または拒否を判断できる.
ただし，こうした文字列は$2^{q(|x|)}$ (指数関数的) 通りある.
こうしたアルゴリズムで認識できる集合をNP集合と考えてもよい.

5. Computational Complexity

5.3. Class NP

5.3.3. More representative NP problems

- Knapsack Problem (KNAP)

  Input : n+1 tuple of natural numbers < a_1, a_2, … , a_n , b>

  Question: Is there a set of indices $S \subseteq \{1, \ldots , n\}$ s.t. $\sum_{i \in S} a_i = b$?

- Bin Packing Problem (BIN)

  Input : n+2 tuple of natural numbers < a_1, a_2, … , a_n, b, k>

  Question: Is there a partition of a set of indices $U=\{1, \ldots , n\}$ into $U_1, \ldots , U_k$ such that $\sum_{i \in U_j} a_i \le b$ for each j?

- Vertex Cover Problem (VC)

  Input : pair of undirected graph G and natural number k <G, k>

  Question: Is there a vertex cover of k vertices over G?

Vertex Cover S contains at least one of u and v for each edge $\{u,v\}$.

5.計算量の理論

5.3. クラスNP

5.3.3. 代表的なNP問題再び

- ナップサック問題 (KNAP)

入力: 自然数のn+1個組< a_1, a_2, … , a_n , b>
質問: 添え字の集合S ⊆ {1, … , n} で$\sum_{i\in S} a_i = b$ を満たすものはあるか?

- ビン詰め問題 (BIN)

入力: 自然数のn+2個組< a_1, a_2, … , a_n, b, k>
質問: 添え字の集合 U={1, … , n} の分割U_1, … , U_kで $\sum_{i\in U_j} a_i \le b$ を満たすものはあるか?

- 頂点被覆問題 (VC)

入力: 無向グラフG と自然数kの組 <G, k>
質問: G上に大きさ k の頂点被覆は存在するか?

頂点被覆 S とは，各辺{u,v}に対してu,vの少なくともどちらか一方をふくむ頂点集合

5. Computational Complexity

5.4. Class coNP

Definition

A set L is in coNP if and only if its complement belongs to NP.

Theorem

For every set L, the following conditions are equivalent.

(a) $L \in$ coNP

(b) The set L can be represented as

$$L = \{x : \forall w \in \Sigma^* : \ |w| \leq q(|x|)[Q(x,w)]\}$$

by using some polynomial q and polynomial-time computable predicate Q.

[Note] It is nonsense to define coP since it is equal to P.

5.計算量の理論

5.4. クラスcoNP

定義

集合 *L* が coNP に属する必要十分条件は,
その補集合がNPに属すること.

定理

任意の集合 *L* に対して，以下の二つは同値である.

(a) $L \in \mathrm{coNP}$

(b) *L*は多項式qと多項式時間で計算できる述語Qを使って
次のように書ける: $L = \{x : \forall w \in \Sigma^* : \ |w| \le q(|x|)[Q(x,w)]\}$

[注意] coP はPと同値であることがすぐにわかるので,
定義しても無意味.

5. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem $P \subseteq E \subseteq EXP$

Proof: Obvious from the definition.

Theorem $P \subsetneq E \subsetneq EXP$

Proof: Out of scope in this class...

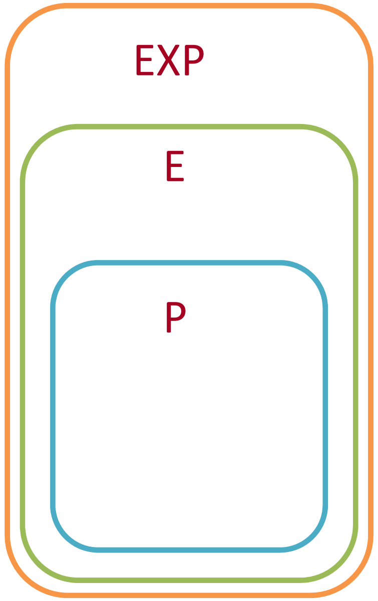
(Brief idea: We can use *diagonalization*

to show a hierarchy theorem that says

$\mathrm{TIME}(t_1(n)) \subsetneq \mathrm{TIME}(t_2(n))$

for, e.g., $t_1(n)^3 = O(t_2(n))$).

We have a *proper* hierarchy



5.計算量の理論

5.5.計算量クラスの関係

定理 P $\subseteq$ E $\subseteq$ EXP

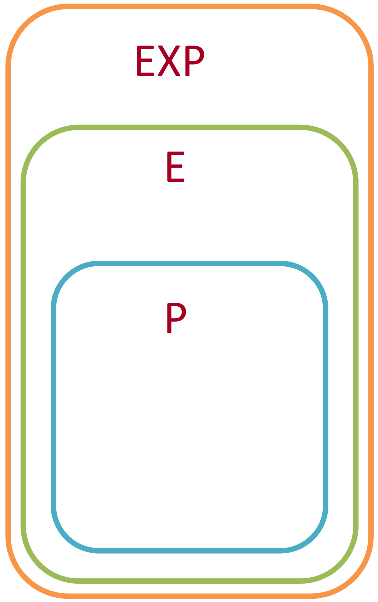
証明: 定義より明らか.

定理 P $\subsetneq$ E $\subsetneq$ EXP

証明: 本書の範囲を超えるので省略.
(アイデアの概略: 対角線論法を巧妙に使うと, 例えば$t_1(n)^3=O(t_2(n))$といった関数に対して次の階層定理を示すことができる.

$$\mathrm{TIME}(t_1(n)) \subsetneq \mathrm{TIME}(t_2(n))$$

*真に異なる*階層構造が成立する



5. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem

(1) $P \subseteq NP$, $P \subseteq coNP$ ($\therefore P \subseteq NP \cap coNP$)

(2) $NP \subseteq EXP$, $coNP \subseteq EXP$ ($\therefore NP \cup coNP \subseteq EXP$)

Proof (Outline):

(1) $P \subseteq NP$ ($P \subseteq coNP$ is similar)

Ignoring the “witness” in the definition of NP, we immediately obtain the definition of P.

(2) $NP \subseteq EXP$ ($coNP \subseteq EXP$ is similar)

For the “witness” w of length m, we can check all possible strings of length m in exponential time.

5.計算量の理論

5.5.計算量クラスの関係

定理
(1)$P \subseteq NP$, $P \subseteq coNP$ ($\therefore P \subseteq NP \cap coNP$)
(2)$NP \subseteq EXP$, $coNP \subseteq EXP$ ($\therefore NP \cup coNP \subseteq EXP$)

証明(概略):
(1) $P \subseteq NP$ ($P \subseteq coNP$ も同様)
NPの定義の中の「証拠」を無視すれば，
Pの定義と同値なものが得られる．
(2) $NP \subseteq EXP$ ($coNP \subseteq EXP$ も同様)
長さmのすべての文字列に対して
それが長さmの「証拠」wになるかどうかを
指数時間かけてチェックすればよい．

5. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem
(1) NP $\subseteq$ coNP $\rightarrow$ NP = coNP
(2) coNP $\subseteq$ NP $\rightarrow$ NP = coNP
(3) NP ≠ coNP $\rightarrow$ P ≠ NP

Note: From (3), proof for NP ≠ co-NP is harder than that for P ≠ NP.

Proof :

(1) NP $\subseteq$ coNP $\rightarrow$ NP = coNP

By assumption, it is sufficient to show that coNP $\subseteq$ NP.

We will prove $L \in$ NP for any $L \in$ coNP.

$L \in$ coNP $\Leftrightarrow$ $\overline{L} \in$ NP (by Definition)

$\rightarrow$ $\overline{L} \in$ coNP (NP $\subseteq$ co-NP)

$\Leftrightarrow$ $L \in$ NP (Definition and $L=\overline{\overline{L}}$)

5.計算量の理論

5.5.計算量クラスの関係

定理

(1) $NP \subseteq coNP \rightarrow NP = coNP$

(2) $coNP \subseteq NP \rightarrow NP = coNP$

(3) $NP \neq coNP \rightarrow P \neq NP$

注: (3)より$NP \neq$ co-NP の証明は$P \neq NP$の証明よりも難しい.

証明：

(1) $NP \subseteq coNP \rightarrow NP = coNP$

仮定より $coNP \subseteq NP$を示せばよい.

そこで任意の $L \in coNP$に対して $L \in NP$ を示す.

$$L \in coNP \Leftrightarrow \overline{L} \in NP \quad \text{(定義より)}$$

$$\rightarrow \overline{L} \in coNP \quad (NP \subseteq \text{co-}NP)$$

$$\Leftrightarrow L \in NP \quad (\text{定義と } L=\overline{\overline{L}} \text{ より})$$

5. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem

(1) NP $\subseteq$ coNP $\rightarrow$ NP = coNP

(2) coNP $\subseteq$ NP $\rightarrow$ NP = coNP

(3) NP ≠ coNP $\rightarrow$ P ≠ NP

Note: From (3), proof for NP ≠ co-NP is harder than that for P ≠ NP.

Proof: (3) NP ≠ coNP $\rightarrow$ P ≠ NP

Contraposition : P = NP $\rightarrow$ NP = coNP

If we assume P=NP, for any L we have

$$
\begin{aligned}
L \in \mathrm{NP} &\Leftrightarrow L \in \mathrm{P} \quad (\mathrm{P} = \mathrm{NP}) \\
&\Leftrightarrow \bar{L} \in \mathrm{P} \quad (\mathrm{P} = \mathrm{coP}) \\
&\Leftrightarrow \bar{L} \in \mathrm{NP} \quad (\mathrm{P} = \mathrm{NP}) \\
&\Leftrightarrow L\ (=\bar{\bar{L}}) \in \mathrm{coNP} \quad (\text{Definitions of NP/coNP})
\end{aligned}
$$

∴ NP = coNP Q.E.D.

5.計算量の理論

5.5.計算量クラスの関係

定理

(1) NP $\subseteq$ coNP → NP = coNP

(2) coNP $\subseteq$ NP → NP = coNP

(3) NP ≠ coNP → P ≠ NP

注: (3)よりNP ≠ co-NP の証明はP ≠ NPの証明よりも難しい.

証明: (3) NP ≠ coNP → P ≠ NP

以下の対偶を示す: P = NP → NP = coNP

P=NPと仮定すると, 任意の集合 L に対して以下を得る

$$
\begin{aligned}
L \in \mathrm{NP} &\Leftrightarrow L \in \mathrm{P} \quad (\mathrm{P} = \mathrm{NP}) \\
&\Leftrightarrow \overline{L} \in \mathrm{P} \quad (\mathrm{P} = \mathrm{coP}) \\
&\Leftrightarrow \overline{L} \in \mathrm{NP} \quad (\mathrm{P} = \mathrm{NP}) \\
&\Leftrightarrow L\ (=\overline{\overline{L}}) \in \mathrm{coNP} \quad (\text{NP/coNPの定義より})
\end{aligned}
$$

∴ NP = coNP　　Q.E.D.

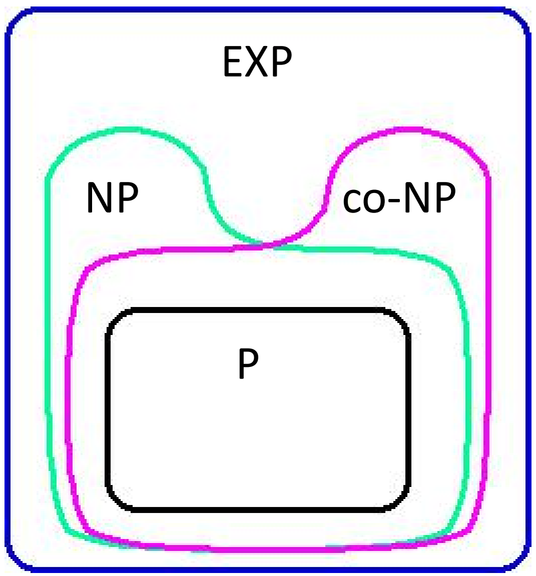
5. Computational Complexity

5.5. Relations in the Complexity Classes

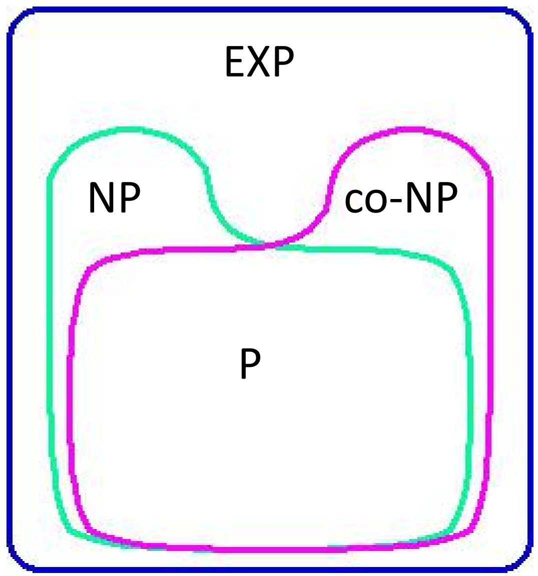
Theorem

(1) $NP \subseteq coNP \rightarrow NP = coNP$
(2) $coNP \subseteq NP \rightarrow NP = coNP$
(3) $NP \neq coNP \rightarrow P \neq NP$

We strongly believe that P ≠ NP, and then we have



or

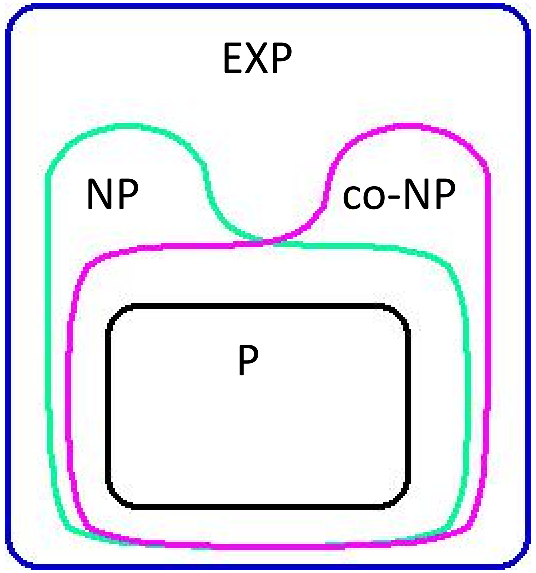


5.計算量の理論

5.5. 計算量クラスの関係

定理
(1) NP ⊆ coNP → NP = coNP
(2) coNP ⊆ NP → NP = coNP
(3) NP ≠ coNP → P ≠ NP

P≠ NPが成立すると強く信じられているので，
以下の構造になっていると予想される．



または