

I216 Computational Complexity
and
Discrete Mathematics

by

Prof. Ryuhei Uehara

and

Prof. Atsuko Miyaji

I216 計算量の理論と離散数学

上原隆平、宮地充子

Computational Complexity

- Goal 1:
 - “*Computable Function/Problem/Language/Set*”
- Goal 2:
 - How can you show “*Difficulty of Problem*”
 - There are *intractable* problems even if they are computable!
 - because they require too many resources (time/space)!
 - Technical terms;
 - The class NP, P≠NP conjecture, NP-hardness, reduction

計算量の理論

- ゴール1:
 - “計算可能な関数/問題/言語/集合”
- ゴール2:
 - 「問題の困難さ」を示す方法を学ぶ
 - 計算可能な問題であっても、手におえない場合がある！
 - 計算に必要な資源(時間・領域)が多すぎる時
 - 関連する専門用語;
 - クラスNP, $P \neq NP$ 予想, NP困難性, 還元

5. Computational Complexity

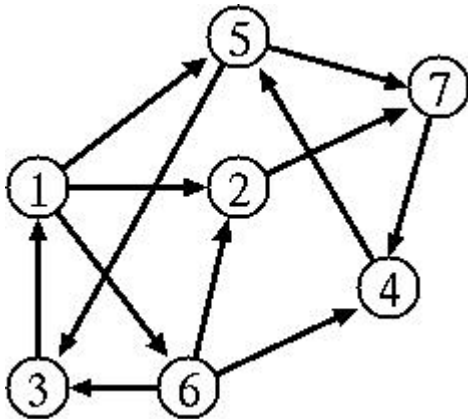
5.3. Class NP

5.3.1. Representative NP problems

- Hamiltonian cycle problem (DHAM)

Input: $\langle G \rangle$: a directed graph G

Question: Does G have a Hamiltonian cycle?



- We can certainly check all possible permutations of n , that counts up to $n! \sim n^n \dots$ 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

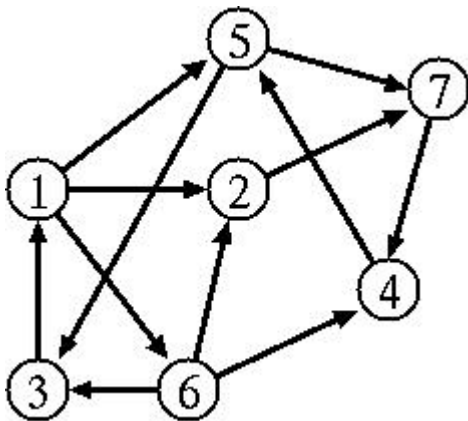
5.3. クラスNP

5.3.1. 代表的なNP問題

- ハミルトン閉路問題 (DHAM)

入力: $\langle G \rangle$: 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 $n! \sim n^n \dots$ 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.

Note: The Hamiltonian Problem can be solved in $O(n^2 2^n)$ time by using smart algorithm based on “Dynamic Programming” technique

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

city

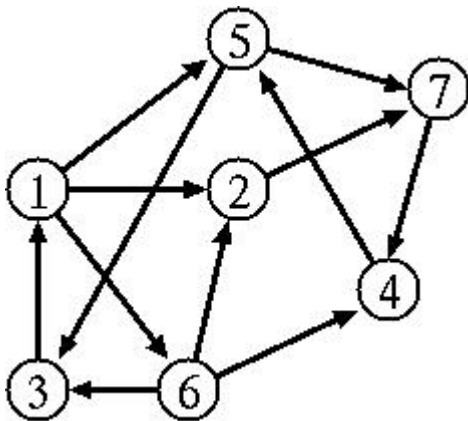
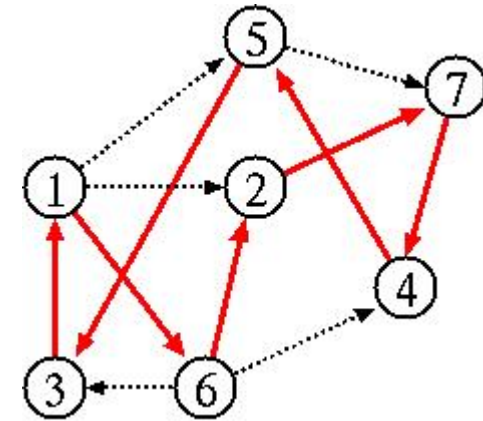
5.3. Class NP

5.3.1. Representative NP problems

- Hamiltonian cycle problem (DHAM)

Input: $\langle G \rangle$: a directed graph G

Question: Does G have a Hamiltonian cycle?



- We can certainly check all possible permutations of n , that counts up to $n! \sim n^n \dots$ 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

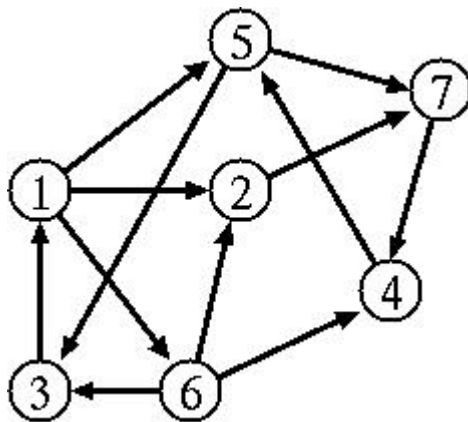
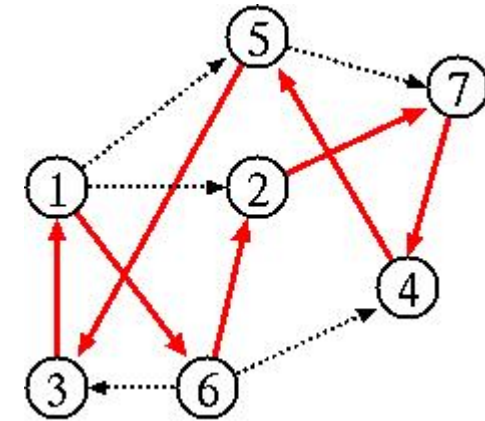
5.3. クラス NP

5.3.1. 代表的な NP 問題

- ハミルトン閉路問題 (DHAM)

入力: $\langle G \rangle$: 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 $n! \sim n^n \dots$ 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, *kSAT*, *ExSAT* (Satisfiability)

Input: $\langle F \rangle$ 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: $\langle F \rangle$ F は和積標準形命題論理式

質問: $F(a_1, a_2, \dots, a_n) = 1$ 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, \dots, 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  $w \in \Sigma^{\leq q(|x|)}$  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 $\langle a_1, a_2, \dots, a_n, b \rangle$

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

- Bin Packing Problem (BIN)

Input: $n+2$ tuple of natural numbers $\langle a_1, a_2, \dots, a_n, b, k \rangle$

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

- Vertex Cover Problem (VC)

Input: pair of undirected graph G and natural number k $\langle G, k \rangle$

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 $\langle a_1, a_2, \dots, a_n, b \rangle$

質問: 添え字の集合 $S \subseteq \{1, \dots, n\}$ 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 $\langle a_1, a_2, \dots, a_n, b, k \rangle$

質問: 添え字の集合 $U = \{1, \dots, n\}$ の分割 U_1, \dots, U_k で $\sum_{i \in U_j} a_i \leq b$ 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 $\langle G, k \rangle$

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) $L \in \text{coNP}$
- (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) $L \in \text{coNP}$

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

[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

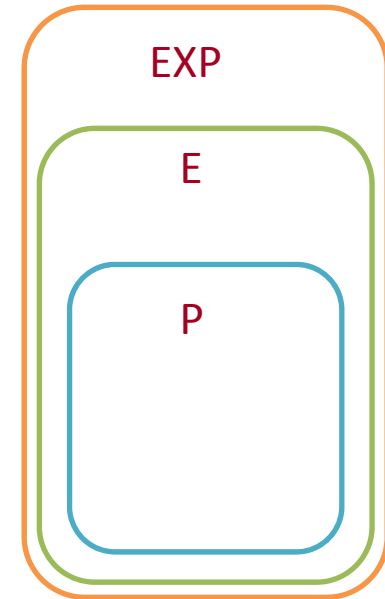
Theorem $P \subseteq E \subseteq EXP$

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 $\text{TIME}(t_1(n)) \subsetneq \text{TIME}(t_2(n))$ 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 $P \subseteq E \subseteq EXP$

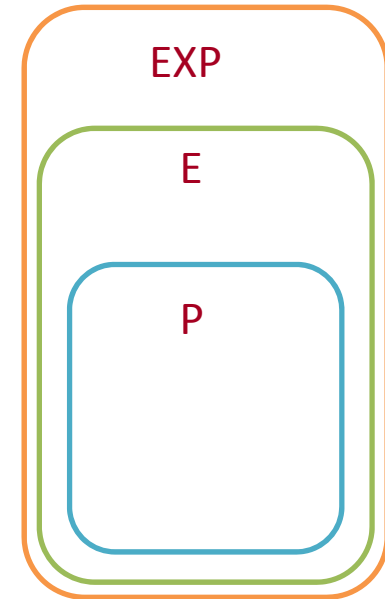
証明: 定義より明らか.

定理 $P \subsetneq E \subsetneq EXP$

証明: 本講義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.

$$TIME(t_1(n)) \subsetneq TIME(t_2(n))$$

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) $P \subseteq NP$, $P \subseteq \text{coNP}$ ($\therefore P \subseteq NP \cap \text{coNP}$)

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

証明(概略):

(1) $P \subseteq NP$ ($P \subseteq \text{coNP}$ 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) $NP \subseteq \text{EXP}$ ($\text{coNP} \subseteq \text{EXP}$ 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) $NP \neq coNP \rightarrow P \neq NP$

Note: From (3), proof for $NP \neq coNP$ is harder than that for $P \neq NP$.

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 $L \in coNP$.

$$\begin{aligned} L \in coNP &\Leftrightarrow \overline{L} \in NP && \text{(by Definition)} \\ &\rightarrow \overline{L} \in coNP && \text{(} NP \subseteq coNP \text{)} \\ &\Leftrightarrow L \in NP && \text{(Definition and } L = \overline{\overline{L}} \text{)} \end{aligned}$$

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.

$$\begin{aligned} L \in coNP &\Leftrightarrow \overline{L} \in NP && \text{(定義より)} \\ &\rightarrow \overline{L} \in coNP && \text{(} NP \subseteq co-NP \text{)} \\ &\Leftrightarrow L \in NP && \text{(定義と } L = \overline{\overline{L}} \text{より)} \end{aligned}$$

5. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem

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

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

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

Note: From (3), proof for $NP \neq coNP$ is harder than that for $P \neq NP$.

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

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

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

$$L \in NP \Leftrightarrow L \in P \quad (P = NP)$$

$$\Leftrightarrow \bar{L} \in P \quad (P = coP)$$

$$\Leftrightarrow \bar{L} \in NP \quad (P = NP)$$

$$\Leftrightarrow L (= \bar{\bar{L}}) \in coNP \quad (\text{Definitions of NP/coNP})$$

$\therefore NP = coNP$

Q.E.D.

5. 計算量の理論

5.5. 計算量クラスの関係

定理

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

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

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

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

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

以下の対偶を示す: $P = NP \rightarrow NP = coNP$

$P=NP$ と仮定すると, 任意の集合 L に対して以下を得る

$$L \in NP \Leftrightarrow L \in P \quad (P = NP)$$

$$\Leftrightarrow \bar{L} \in P \quad (P = coP)$$

$$\Leftrightarrow L \in NP \quad (P = NP)$$

$$\Leftrightarrow L (= \bar{\bar{L}}) \in coNP \quad (NP/coNPの定義より)$$

$\therefore NP = coNP$

Q.E.D.

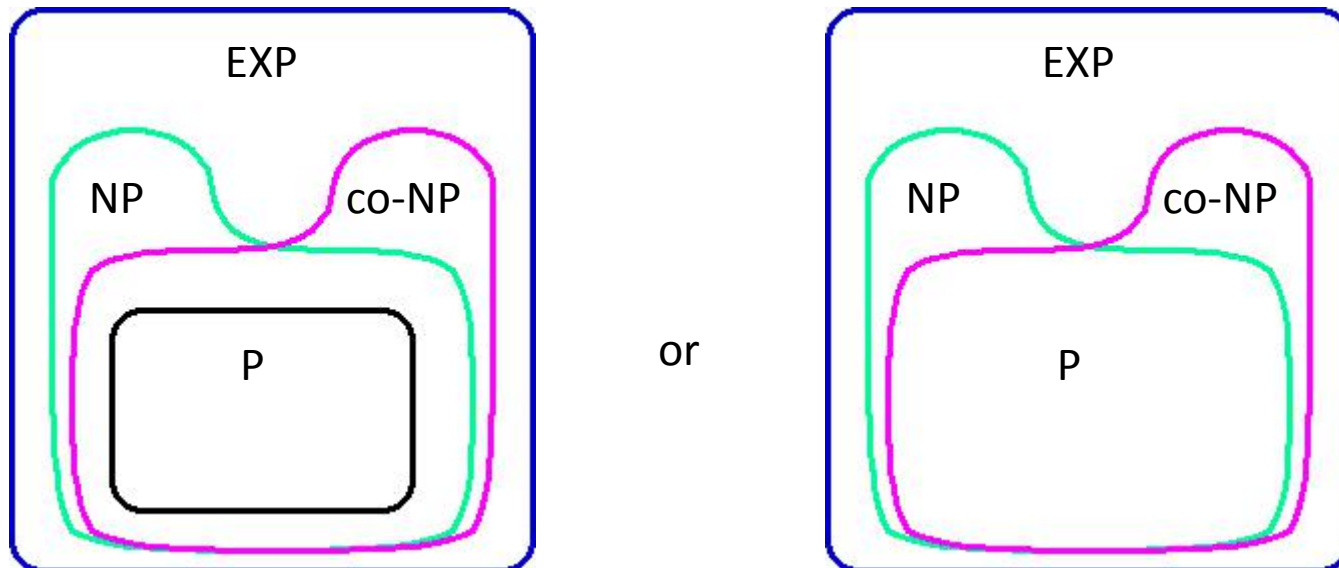
5. Computational Complexity

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 $P \neq NP$, and then we have



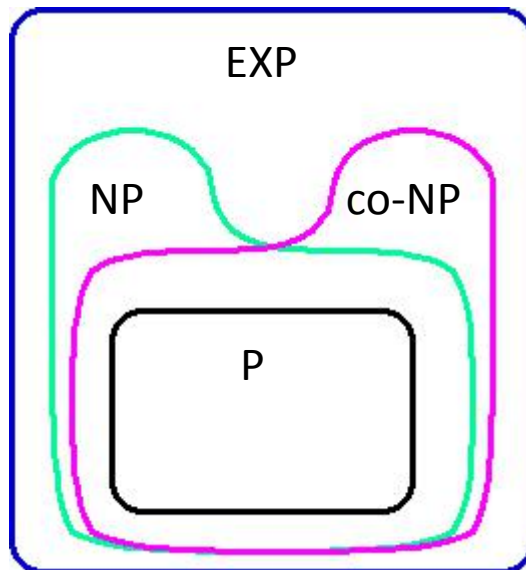
5. 計算量の理論

5.5. 計算量クラスの関係

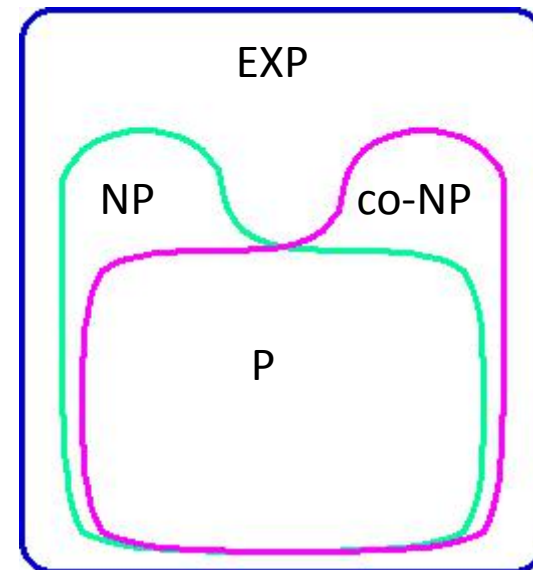
定理

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

$P \neq NP$ 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



5. Computational Complexity

- Observation of the classes

Definition: Class P

Set L is in the class P \Leftrightarrow

There exists a poly-time computable predicate R such that

for each $x \in \Sigma^*$, $x \in L \Leftrightarrow R(x)$

Definition: Class NP

Set L is in the class NP \Leftrightarrow

There exists a poly q and a poly-time computable predicate R such that

for each $x \in \Sigma^*$, $x \in L \Leftrightarrow \exists w \in \Sigma^* : |w| \leq q(|x|)[R(x,w)]$

Definition: Class coNP

Set L is in the class coNP \Leftrightarrow

There exists a poly q and a poly-time computable predicate R such that

for each $x \in \Sigma^*$, $x \in L \Leftrightarrow \forall w \in \Sigma^* : |w| \leq q(|x|)[R(x,w)]$

5. 計算量の理論

• 計算量クラスの定義を概観すると...

クラスPの定義

集合 L がクラスPに入る \Leftrightarrow

以下を満たす多項式時間計算可能述語 R が存在:

各 $x \in \Sigma^*$ で $x \in L \Leftrightarrow R(x)$

クラスNPの定義

集合 L がクラスNPに入る \Leftrightarrow

以下を満たす多項式 q と多項式時間計算可能述語 R が存在:

各 $x \in \Sigma^*$ で $x \in L \Leftrightarrow \exists w \in \Sigma^* : |w| \leq q(|x|) [R(x,w)]$

クラスcoNPの定義

集合 L がクラスcoNPに入る \Leftrightarrow

以下を満たす多項式 q と多項式時間計算可能述語 R が存在:

各 $x \in \Sigma^*$ で $x \in L \Leftrightarrow \forall w \in \Sigma^* : |w| \leq q(|x|) [R(x,w)]$

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Definition

Let A and B be arbitrary sets.

(1) function $h: A \rightarrow B$: polynomial-time reduction

- $$\Leftrightarrow \left\{ \begin{array}{l} \text{(a) } h \text{ is a total function from } \Sigma^* \text{ onto } \Sigma^* \\ \text{(b) } x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B] \\ \text{(c) } h \text{ is polynomial-time computable.} \end{array} \right.$$

(2) When there is a poly-time reduction from A to B , we say A is polynomial-time reducible to B .

Then, we denote by

$$A \leq_m^P B$$

(...within polynomial time, hardness of $A \leq$ that of B)

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定義

A と B を任意の集合とする.

(1) 関数 $h: A \rightarrow B$ が 多項式時間還元 である

\Leftrightarrow $\left\{ \begin{array}{l} \text{(a) } h \text{ は } \Sigma^* \text{ から } \Sigma^* \text{ への全域関数である} \\ \text{(b) } x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B] \\ \text{(c) } h \text{ は多項式時間計算可能である.} \end{array} \right.$

(2) A から B への多項式時間還元が存在するとき

A は B へ多項式時間還元可能 であるといい,

$A \leq_m^P B$ とかく.

(...多項式時間程度の差を無視すれば, A の難しさ \leq B の難しさ)

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem $A \leq_m^P B$ leads to

- (1) $B \in P \rightarrow A \in P$.
- (2) $B \in NP \rightarrow A \in NP$.
- (3) $B \in \text{co-NP} \rightarrow A \in \text{coNP}$.
- (4) $B \in \text{EXP} \rightarrow A \in \text{EXP}$.

Note: class E is exceptional. Generally, $B \in E \rightarrow A \in E$ is not true.

Ex.: When we define $\text{ONE} \equiv \{1\}$, for each set L in P we have

$$L \leq_m^P \text{ONE}$$

if we define $h(x) \equiv \begin{cases} 1, & \text{if } x \in L \\ 0, & \text{otherwise} \end{cases}$

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理 $A \leq_m^P B$ のとき次が成立する

- (1) $B \in P \rightarrow A \in P.$
- (2) $B \in NP \rightarrow A \in NP.$
- (3) $B \in \text{co-NP} \rightarrow A \in \text{coNP}.$
- (4) $B \in \text{EXP} \rightarrow A \in \text{EXP}.$

注意: クラス E は例外. 一般に $B \in E \rightarrow A \in E$ は成立しない.

例: $\text{ONE} \equiv \{1\}$ と定義すると, P の各集合 L に対して,

$$L \leq_m^P \text{ONE}$$

である. ここで $h(x) \equiv \begin{cases} 1, & \text{if } x \in L \\ 0, & \text{otherwise} \end{cases}$

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem A, B, C : arbitrary sets

$$(1) A \leq_m^P A$$

$$(2) A \leq_m^P B \wedge B \leq_m^P C \rightarrow A \leq_m^P C$$

Definition

$$A \equiv_m^P B \leftrightarrow A \leq_m^P B \wedge B \leq_m^P A$$

\equiv_m^P is an equivalence relation.

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理 A, B, C を任意の集合とする.

$$(1) A \leq_m^P A$$

$$(2) A \leq_m^P B \wedge B \leq_m^P C \rightarrow A \leq_m^P C$$

定義

$$A \equiv_m^P B \leftrightarrow A \leq_m^P B \wedge B \leq_m^P A$$

\equiv_m^P は同値関係.

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem

$$(1) 2\text{SAT} \leq_m^P 3\text{SAT} \leq_m^P \text{SAT} \leq_m^P \text{ExSAT}$$

$$(2) 3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$$

Proof

(1) we have some proofs depending on definition:

- (a) each instance of 2SAT is also in 3SAT if the definition is “at most 3 literals in a clause”.
- (b) each clause $(x \vee y)$ can be replaced by $(x \vee y \vee y)$.
- (c) each clause $(x \vee y)$ can be replaced by $(x \vee y \vee z) \wedge (x \vee y \vee \bar{z})$.

In any case, they are poly-time reduction, and the original formula is satisfiable iff so is the resulting formula.

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理

$$(1) 2SAT \leq_m^P 3SAT \leq_m^P SAT \leq_m^P \text{ExSAT}$$

$$(2) 3SAT \equiv_m^P SAT \equiv_m^P \text{ExSAT}$$

証明

(1) 定義によっていくつかの証明が考えられる:

(a) 定義が「各項に高々3リテラル」の場合は, 2SATの入力は3SATの入力としても有効なので, 特に示すことはない.

(b) 各項 $(x \vee y)$ を単に $(x \vee y \vee y)$ で置き換えればよい.

(c) 各項 $(x \vee y)$ に対して新しい変数を導入して

$$(x \vee y \vee z) \wedge (x \vee y \vee \bar{z}).$$

と置き換えてもよい.

どの場合も多項式時間還元で, 元の論理式が充足可能である必要十分条件は, 新しい式が充足可能であること.

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem

$$(1) \text{ 2SAT} \leq_m^P \text{ 3SAT} \leq_m^P \text{ SAT} \leq_m^P \text{ ExSAT}$$

$$(2) \text{ 3SAT} \equiv_m^P \text{ SAT} \equiv_m^P \text{ ExSAT}$$

Proof (Outline)

(2) It is sufficient to show that $\text{ExSAT} \leq_m^P \text{ 3SAT}$ by (1).

Strategy:

For any given F in ExSAT, we construct another F' in 3SAT such that F is satisfiable iff F' is satisfiable.

To do that, we first construct the computation tree of F , and construct F' that represents the computation process of F .

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理

$$(1) 2SAT \leq_m^P 3SAT \leq_m^P SAT \leq_m^P \text{ExSAT}$$

$$(2) 3SAT \equiv_m^P SAT \equiv_m^P \text{ExSAT}$$

証明 (概略)

(2) (1)より, $\text{ExSAT} \leq_m^P 3SAT$ が成立することを示せばよい.

基本戦略:

ExSATの式 F が与えられたら, それに基づいて3SATの式 F' を構成する. ただしここで F が充足可能である必要十分条件が F' が充足可能であるようにする. そのために, まず F の計算木を構築し, 次に F の計算手順を表現する論理式 F' を構築する.

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

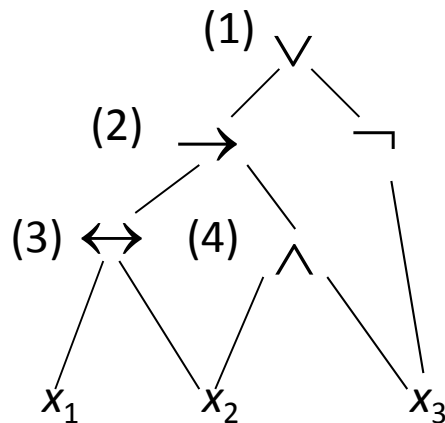
Theorem (2) $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$

Proof (Outline)

(2) It is sufficient to show that $\text{ExSAT} \leq_m^P 3\text{SAT}$ by (1).

Reduction from ExSAT to 3SAT by an example:

$$F(x_1, x_2, x_3) \equiv [[x_1 \leftrightarrow x_2] \rightarrow [x_2 \wedge x_3]] \vee \neg x_3$$



$$(1) V_1 \equiv V_2 \vee \neg x_3$$

$$(2) V_2 \equiv [V_3 \rightarrow V_4]$$

$$(3) V_3 \equiv [x_1 \leftrightarrow x_2]$$

$$(4) V_4 \equiv x_2 \wedge x_3$$

6. 多項式時間計算可能性の解析手法

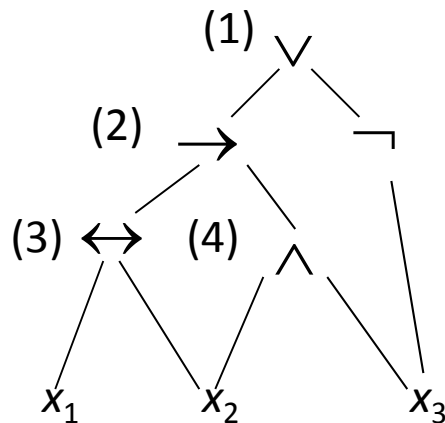
6.1. 多項式時間還元可能性

定理 (2) $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$

証明 (概略)

(2) $\text{ExSAT} \leq_m^P 3\text{SAT}$ が成立することを示せばよい.
ExSAT から 3SAT への還元を例で示す:

$$F(x_1, x_2, x_3) \equiv [[x_1 \leftrightarrow x_2] \rightarrow [x_2 \wedge x_3]] \vee \neg x_3$$



$$(1) V_1 \equiv V_2 \vee \neg x_3$$

$$(2) V_2 \equiv [V_3 \rightarrow V_4]$$

$$(3) V_3 \equiv [x_1 \leftrightarrow x_2]$$

$$(4) V_4 \equiv x_2 \wedge x_3$$

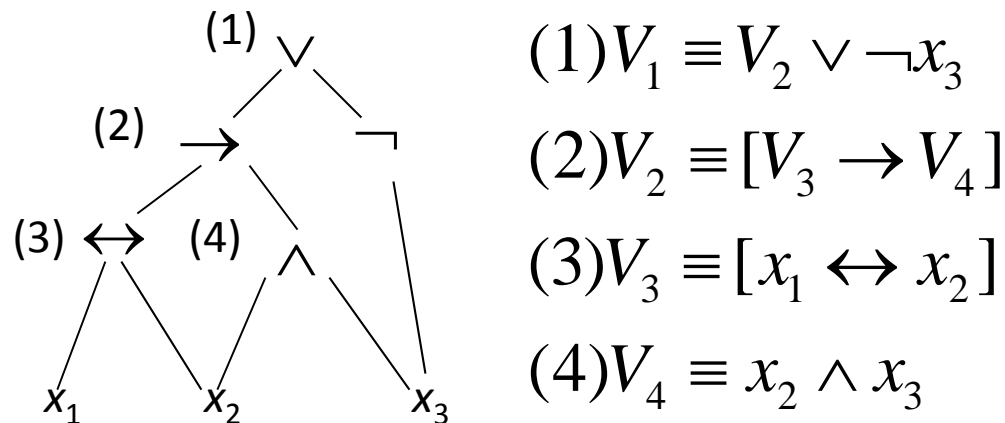
6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem (2) $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$

Reduction from ExSAT to 3SAT by an example:

$$F(x_1, x_2, x_3) \equiv [[x_1 \leftrightarrow x_2] \rightarrow [x_2 \wedge x_3]] \vee \neg x_3$$



$$F''(x_1, x_2, x_3) \equiv U_1 \wedge [U_1 \leftrightarrow [U_2 \vee \neg x_3]] \wedge [U_2 \leftrightarrow [U_3 \rightarrow U_4]] \\ \wedge [U_3 \leftrightarrow [x_1 \leftrightarrow x_2]] \wedge [U_4 \leftrightarrow [x_2 \wedge x_3]]$$

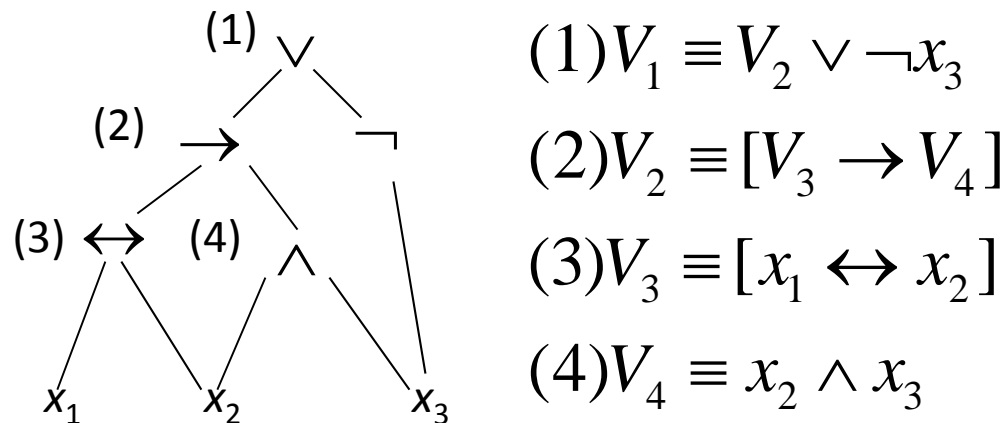
6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理 (2) $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{EXSAT}$

EXSAT から 3SAT への還元を例で示す:

$$F(x_1, x_2, x_3) \equiv [[x_1 \leftrightarrow x_2] \rightarrow [x_2 \wedge x_3]] \vee \neg x_3$$



$$F''(x_1, x_2, x_3) \equiv U_1 \wedge [U_1 \leftrightarrow [U_2 \vee \neg x_3]] \wedge [U_2 \leftrightarrow [U_3 \rightarrow U_4]] \\ \wedge [U_3 \leftrightarrow [x_1 \leftrightarrow x_2]] \wedge [U_4 \leftrightarrow [x_2 \wedge x_3]]$$

6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

Theorem (2) $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$

Reduction from ExSAT to 3SAT by an example:

$$F''(x_1, x_2, x_3) \equiv U_1 \wedge [U_1 \leftrightarrow [U_2 \vee \neg x_3]] \wedge [U_2 \leftrightarrow [U_3 \rightarrow U_4]] \\ \wedge [U_3 \leftrightarrow [x_1 \leftrightarrow x_2]] \wedge [U_4 \leftrightarrow [x_2 \wedge x_3]]$$

Then, by construction, $F()$ is satisfiable iff $F''()$ is satisfiable.

We show $F''()$ can be represented by an equivalent $F'()$ in 3SAT.

$$U_1 \leftrightarrow [U_2 \vee \neg x_3] = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg[U_2 \vee \neg x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_3] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_2 \vee x_2]$$

The other cases are similar, and $F'()$ is in 3SAT.

6. 多項式時間計算可能性の解析手法

6.1. 多項式時間還元可能性

定理 (2) $3SAT \equiv_m^P SAT \equiv_m^P ExSAT$

ExSAT から 3SAT への還元を例で示す:

$$F''(x_1, x_2, x_3) \equiv U_1 \wedge [U_1 \leftrightarrow [U_2 \vee \neg x_3]] \wedge [U_2 \leftrightarrow [U_3 \rightarrow U_4]] \\ \wedge [U_3 \leftrightarrow [x_1 \leftrightarrow x_2]] \wedge [U_4 \leftrightarrow [x_2 \wedge x_3]]$$

このとき構成から, $F()$ は充足可能 $\Leftrightarrow F''()$ は充足可能.
 $F''()$ をこれと同値な 3SAT の要素 $F'()$ で表現する.

$$U_1 \leftrightarrow [U_2 \vee \neg x_3] = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg [U_2 \vee \neg x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_3] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_3 \vee x_3]$$

他のケースも同様に変形でき, $F'()$ は 3SAT の要素となる.

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic properties

Definition

For a class C , if a set A satisfies

(a) $\forall L \in C [L \leq_m^P A]$,

the set A is called **C-hard** (under \leq_m^P).

Moreover, if we have

(b) $A \in C$,

then A is called **C-complete**.

Ex. Examples of NP-complete sets

3SAT, SAT, ExSAT, DHAM, KNAP, BIN, VC, etc.

6. 多項式時間計算可能性の解析手法

6.2. 完全性

6.2.1. 定義と基本性質

定義

クラスCに対して, 集合Aが次を満たすとき

$$(a) \forall L \in C [L \leq_m^P A]$$

集合Aは(\leq_m^P のもとで) **C困難**であるという.

さらに次を満たすなら

$$(b) A \in C$$

Aは**C完全**であるという.

例. NP完全集合の例

3SAT, SAT, ExSAT, DHAM, KNAP, BIN, VC など

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic properties

Theorem. For any C-hard (or C-complete) set A ,

- | | |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$ | CP: $C \not\subseteq P \rightarrow A \notin P$ |
| (2) $A \in NP \rightarrow C \subseteq NP$ | CP: $C \not\subseteq NP \rightarrow A \notin NP$ |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | CP: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$ | CP: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$ |

Proof: CP: contraposition

(1) Let B be any C-set. Then, since A is C-hard,

$B \leq_m^P A$ and by the assumption $A \in P$, we have $B \in P$

(2), (3), (4) are similar.

6. 多項式時間計算可能性の解析手法

6.2. 完全性

6.2.1. 定義と基本性質

定理 C 困難(または C 完全)な任意の集合 A に対して,

- | | |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$ | 対偶: $C \not\subseteq P \rightarrow A \notin P$ |
| (2) $A \in NP \rightarrow C \subseteq NP$ | 対偶: $C \not\subseteq NP \rightarrow A \notin NP$ |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | 対偶: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$ | 対偶: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$ |

証明:

(1) 任意の C 集合を B とする. A が C 困難であることから,

$B \leq_m^P A$ であり, $A \in P$ という仮定より $B \in P$ をえる.

(2), (3), (4) も同様.

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic properties

Theorem. For any C-hard (or C-complete) set A,

- | | |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$ | CP: $C \not\subseteq P \rightarrow A \notin P$ |
| (2) $A \in NP \rightarrow C \subseteq NP$ | CP: $C \not\subseteq NP \rightarrow A \notin NP$ |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | CP: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$ | CP: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$ |

Ex. : Meaning of Theorem for class NP

Let A be NP-complete set.

By the contraposition of Theorem (1) we have

$$NP \neq P \rightarrow A \notin P$$

That is, NP-complete sets are NP-sets that cannot be recognized in polynomial time unless $P = NP$.

6. 多項式時間計算可能性の解析手法

6.2. 完全性

6.2.1. 定義と基本性質

定理 C 困難(または C 完全)な任意の集合 A に対して,

- | | |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$ | 対偶: $C \not\subseteq P \rightarrow A \notin P$ |
| (2) $A \in NP \rightarrow C \subseteq NP$ | 対偶: $C \not\subseteq NP \rightarrow A \notin NP$ |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | 対偶: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$ | 対偶: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$ |

例: クラス NP に関する定理の意味するところ

NP 完全集合を A とする.

定理(1)の対偶より: $NP \neq P \rightarrow A \notin P$

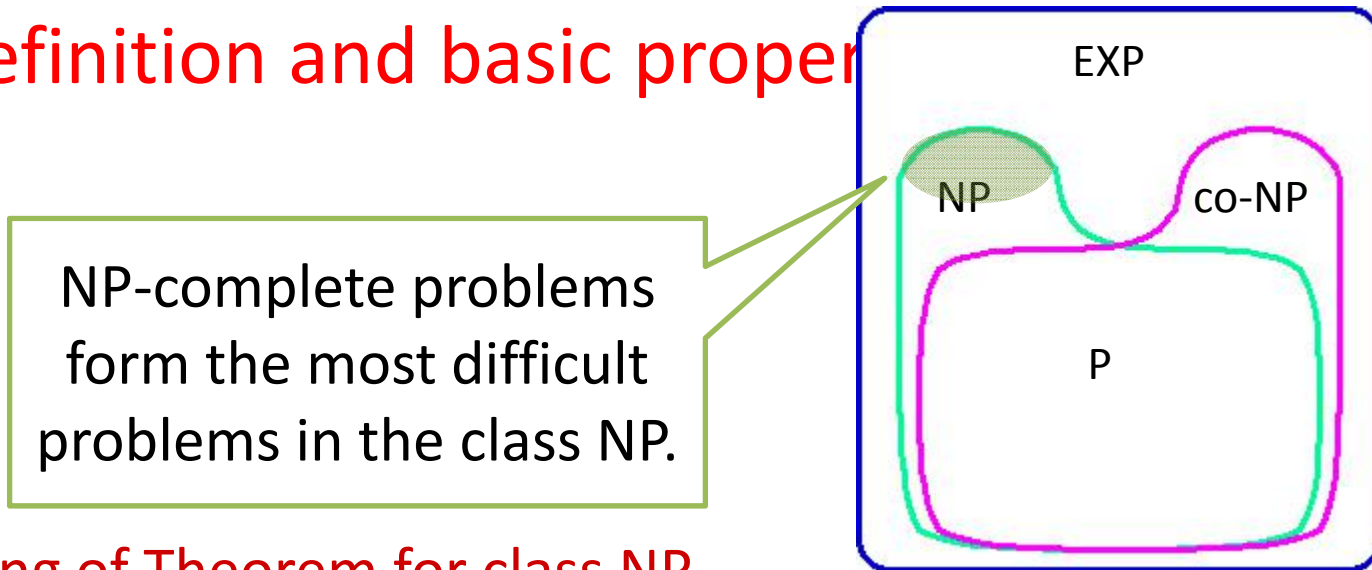
つまり, NP 完全集合は $P=NP$ でない限り,

多項式時間では認識できない NP 集合である.

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic properties



Ex. : Meaning of Theorem for class NP

Let A be NP-complete set.

By the contraposition of Theorem (1) we have

$$NP \neq P \rightarrow A \notin P$$

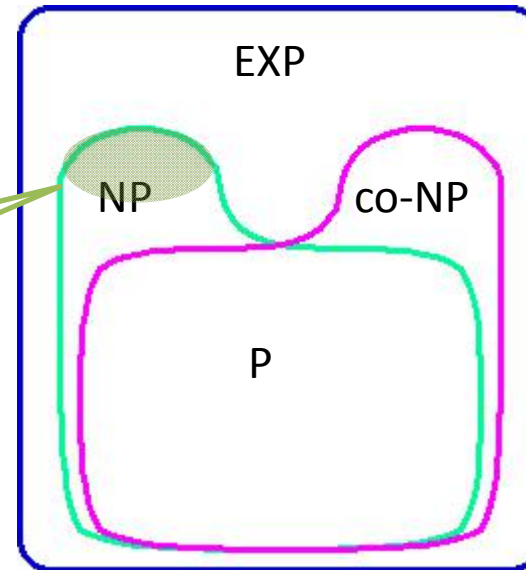
That is, NP-complete sets are NP-sets that cannot be recognized in polynomial time unless $P = NP$.

6. 多項式時間計算可能性の解析手法

6.2. 完全性

6.2.1. 定義と基本性質

NP完全問題とは、クラスNPの中で最も難しい問題群を構成しているといえる。



例：クラスNPに関する定理の意味するところ

NP完全集合をAとする。

定理(1)の対偶より： $NP \neq P \rightarrow A \notin P$

つまり、NP完全集合は $P=NP$ でない限り、

多項式時間では認識できないNP集合である。

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic properties

Theorem 6.4. A : any C -complete set

For any set B we have

(1) $A \leq_m^P B \rightarrow B$ is C -hard.

(2) $A \leq_m^P B$ and $B \in C \rightarrow B$ is C -complete.

Proof:

By definition, $\forall L \in C [L \leq_m^P A]$

By Theorem, $L \leq_m^P A \wedge A \leq_m^P B \rightarrow L \leq_m^P B$

Therefore, $\forall L \in C [L \leq_m^P B]$

That is, B is C -hard.

Once you have an NP-complete problem A , it can be used to measure to the other problems

6. 多項式時間計算可能性の解析手法

6.2. 完全性

6.2.1. 定義と基本性質

定理 A : 任意の C 完全集合
任意の集合 B に対して以下が成立
(1) $A \leq_m^P B \rightarrow B$ は C 困難.
(2) $A \leq_m^P B$ かつ $B \in C \rightarrow B$ は C 完全.

証明:

定義より, $\forall L \in C [L \leq_m^P A]$

定理より, $L \leq_m^P A \wedge A \leq_m^P B \rightarrow L \leq_m^P B$

よって, $\forall L \in C [L \leq_m^P B]$

つまり B は C 困難.

ひとたび NP 完全問題 A が得られたら、これを使って他の問題の困難性を「測定」できる。