

Chapter 5

Representative Complexity Classes

5.1. Representative time complexity classes

$$\mathcal{P} \equiv \bigcup_{p:\text{polynomial}} \text{TIME}(p(l))$$

$$\mathcal{E} \equiv \bigcup_{c>1} \text{TIME}(2^{cl})$$

$$\mathcal{EXP} \equiv \bigcup_{p:\text{polynomial}} \text{TIME}(2^{p(l)})$$

\mathcal{C} set: set in the complexity class \mathcal{C} .

\mathcal{C} problem: problem of recognizing a \mathcal{C} set.

Problems not in \mathcal{P} are intractable
from the practical viewpoint...

第5章 代表的な計算量クラス

5.1. 代表的な時間計算量クラス

$$\mathcal{P} \equiv \bigcup_{p:\text{多項式}} \text{TIME}(p(l))$$

$$\mathcal{E} \equiv \bigcup_{c>1} \text{TIME}(2^{cl})$$

$$\mathcal{EXP} \equiv \bigcup_{p:\text{多項式}} \text{TIME}(2^{p(l)})$$

\mathcal{C} 集合: 計算量クラス \mathcal{C} に入る集合.

\mathcal{C} 問題: \mathcal{C} 集合の認識問題



ある問題が \mathcal{P} に入っていないなら、
現実的には手に負えない...

Ex.5.3. Problem of evaluating propositional expression (PROP-EVAL)

Input: $\langle F, \langle a_1, a_2, \dots, a_n \rangle \rangle$

F is an extended prop. expression

(a_1, a_2, \dots, a_n) is a truth assignment to F

Question: $F(a_1, a_2, \dots, a_n) = 1$?

	$x \rightarrow y$	$x \leftrightarrow y$
(x, y)	$(\neg x \vee y)$	$((x \rightarrow y) \wedge (y \rightarrow x))$
$(0, 0)$	1	1
$(0, 1)$	1	0
$(1, 0)$	0	0
$(1, 1)$	1	1

例5.3. 命題論理式評価問題(PROP-EVAL)

入力: $\langle F, \langle a_1, a_2, \dots, a_n \rangle \rangle$

F は拡張命題論理式 $\wedge \vee \neg \rightarrow \leftrightarrow$

(a_1, a_2, \dots, a_n) は F に対する真理値割り当て

質問: $F(a_1, a_2, \dots, a_n) = 1$?

	$x \rightarrow y$	$x \leftrightarrow y$
(x, y)	$(\neg x \vee y)$	$((x \rightarrow y) \wedge (y \rightarrow x))$
$(0, 0)$	1	1
$(0, 1)$	1	0
$(1, 0)$	0	0
$(1, 1)$	1	1

Ex.5.3. Problem of evaluating propositional expression (PROP-EVAL)

Input: $\langle F, \langle a_1, a_2, \dots, a_n \rangle \rangle$

F is an extended prop. expression

(a_1, a_2, \dots, a_n) is a truth assignment to F

Question: $F(a_1, a_2, \dots, a_n) = 1$?

Construct a computation tree from a code $\lceil F \rceil$ of ext. prop. expression

It is built in time $O(|\lceil F \rceil|^3)$.

If computation tree is available, we can easily obtain the value

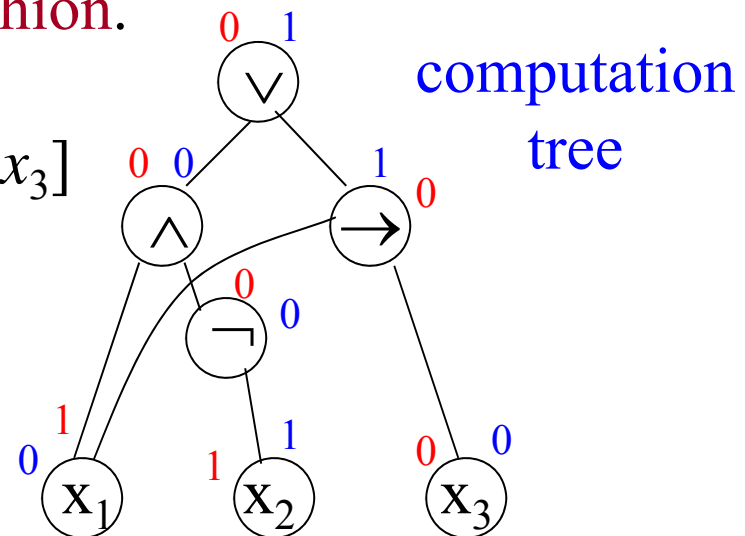
$F(a_1, a_2, \dots, a_n)$ in a **bottom-up fashion**.

Ex.: $F(x_1, x_2, x_3) = [x_1 \wedge \neg x_2] \vee [x_1 \rightarrow x_3]$

$$F(0, 1, 0) = 1$$

$$F(1, 1, 0) = 0$$

Hence PROP-EVAL $\in \mathcal{P}$



例5.3. 命題論理式評価問題(PROP-EVAL)

入力: $\langle F, \langle a_1, a_2, \dots, a_n \rangle \rangle$

F は拡張命題論理式 $\wedge \vee \neg \rightarrow \leftrightarrow$

(a_1, a_2, \dots, a_n) は F に対する真理値割り当て

質問: $F(a_1, a_2, \dots, a_n) = 1$?

拡張命題論理式 F がコード化されたもの $\lceil F \rceil$ から計算木を作る.

計算木は $O(|\lceil F \rceil|^3)$ 時間で構成できる.

計算木が得られていれば, **ボトムアップ式**で

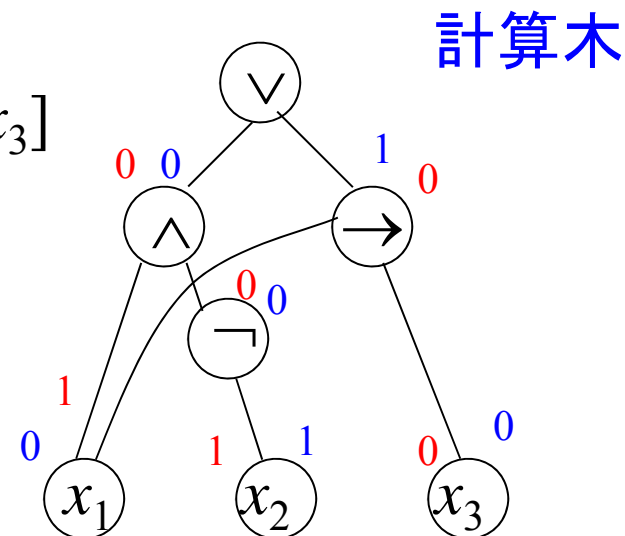
$F(a_1, a_2, \dots, a_n)$ の値は容易に計算可能. 0 1

例: $F(x_1, x_2, x_3) = [x_1 \wedge \neg x_2] \vee [x_1 \rightarrow x_3]$

$$F(0, 1, 0) = 1$$

$$F(1, 1, 0) = 0$$

よって PROP-EVAL $\in \mathcal{P}$



Ex. 5.3. 2-Satisfiability (2SAT)

Input: $\langle F \rangle$ F is 2-conjunctive normal form

Question: Is there any assignment such that $F(a_1, a_2, \dots, a_n) = 1$?

Conjunctive Normal Form (CNF)

$$F = (\bullet \vee \bullet \vee \dots \vee \bullet) \wedge (\bullet \vee \dots \vee \bullet) \wedge \dots \wedge (\dots)$$

- described by \wedge of \vee of literals.

exactly/at most

k SAT

- Each clause contains k literals
- We can define 3SAT, 4SAT similarly.
- SAT consists of any CNF.
- ExSAT consists of any extended propositional expression.

例5.3. 命題論理式充足性問題: 2和積形(2SAT)

入力: $\langle F \rangle$ F は2和積形命題論理式

質問: $F(a_1, a_2, \dots, a_n) = 1$ を満たす割り当てがあるか?

和積形:

$$F = (\bullet \vee \bullet \vee \dots \vee \bullet) \wedge (\bullet \vee \dots \vee \bullet) \wedge \dots \wedge (\dots)$$

- リテラルの論理和の論理積で表現されたもの

ちょうど/たかだか

k 和積形(k SAT)

- 和積形の各論理和が k 個のリテラルを含む

- 3SAT, 4SAT も同様に定義できる。
- SAT: 各論理和のリテラルの個数に制限がないもの
- ExSAT: 入力が拡張命題論理式(\rightarrow や \leftrightarrow も許す)

Ex. 5.4: Graph reachability problem (ST-CON)

Input: $\langle G, s, t \rangle$: an undirected graph G , $1 \leq s, t \leq n (=|G|)$

Question: Does G have a path from s to t ?

- **Cycle** is a path that shares two endpoints.
- **Euler cycle** is a cycle that visits all **edges** once.
- **Hamiltonian cycle** is a cycle that visits all **vertices** once.

Ex. 5.4: Euler cycle problem (DEULER)

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

Question: Does G have an Euler cycle?

Ex. 5.5 Hamiltonian cycle problem (DHAM)

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

Question: Does G have a Hamiltonian cycle?

例5.4: 到達可能性問題(ST-CON)

入力: $\langle G, s, t \rangle$: 無向グラフ G , $1 \leq s, t \leq n (=|G|)$

質問: G 上で s から t への道があるか?

- 閉路とは、始点と終点が同じである路
- オイラー閉路とは、すべての辺を一度ずつ通る閉路
- ハミルトン閉路とは、すべての頂点を一度ずつ通る閉路

例5.4: 一筆書き閉路問題(DEULER)

入力: $\langle G \rangle$: 有向グラフ G

質問: G はオイラー閉路をもつか?

例5.5: 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?

It is known that:

- The following problems are in \mathcal{P} :
 - ✓ PROP-EVAL, 2SAT, ST-CON, DEULER

- The following problems are in \mathcal{E} , but...
 - ✓ 3SAT, DHAM



The class \mathcal{NP} between \mathcal{P} and \mathcal{E} ?

以下の事実が知られている:

➤ 以下の問題は \mathcal{P} に属する:

✓ PROP-EVAL, 2SAT, ST-CON, DEULER

➤ 以下の問題は \mathcal{E} に属する、が、、、

✓ 3SAT, DHAM



\mathcal{P} と \mathcal{E} の間(?)のクラス \mathcal{NP}

5.2. Class \mathcal{NP}

Def. 5.2: Suppose that we have a polynomial q and polynomial time computable predicate R for a set L such that

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

$$\text{i.e., } L = \{x : \exists w \in \Sigma^* [|w| \leq q(|x|) \wedge R(x, w)]\} \quad (5.1)$$

Then, L is called an \mathcal{NP} set, and the problem of recognizing L is called an \mathcal{NP} problem.

Also, the whole set of \mathcal{NP} sets is called the **class \mathcal{NP}** .

Note: For each $x \in \Sigma^*$, $w_x \in \Sigma^*$ satisfying the predicate $|w| \leq q(|x|) \wedge R(x, w)$ is called (polynomial) *witness* of x .

Hereafter, we use notation $\exists w \in \Sigma^* : |w| \leq q(|x|) \Rightarrow \exists_q w$

“Given a witness of polynomial length in the input size, we can determine in polynomial time whether it satisfies the condition of a given problem.”

c.f.: $\mathcal{NP} = \underline{\mathbf{N}}$ ondeterministic $\underline{\mathbf{P}}$ olynomial

5.2. クラス \mathcal{NP}

定義5.2: 集合 L に対して次の条件を満たす多項式 q と多項式時間計算可能述語 R が存在したとする.

$$\text{各 } x \in \Sigma^* \text{ で } x \in L \leftrightarrow \exists w \in \Sigma^* : |w| \leq q(|x|) [R(x, w)] \quad (5.1)$$

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

このとき, L を \mathcal{NP} 集合といい, L の認識問題を \mathcal{NP} 問題という.
また, \mathcal{NP} 集合の全体を **クラス \mathcal{NP}** という.

補注: 各 $x \in \Sigma^*$ に対して, 論理式 $|w| \leq q(|x|) \wedge R(x, w)$ を満たす $w_x \in \Sigma^*$ を x の (多項式長の) **証拠** という.

以下では, $\exists w \in \Sigma^* : |w| \leq q(|x|) \Rightarrow \exists_q w$ と略記.

「入力サイズの多項式長の証拠が与えられたとき, これが問題の条件を満たすかどうかを多項式時間で判定できる.」

補足: $\mathcal{NP} = \underline{\mathbf{N}}\text{ondeterministic } \underline{\mathbf{P}}\text{olynomial}$

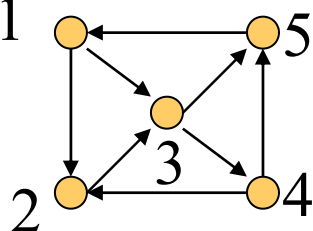
Ex.5.7: Hamilton Cycle Problem (DHAM) $\in \mathcal{NP}$

Assume graph vertices are numbered $1 \sim n$.

Trace on a Hamilton cycle \rightarrow permutation of $1 \sim n$ $\langle l_1, l_2, \dots, l_n \rangle$

This permutation is a **witness** of polynomial length.

Ex.: 1



candidates of witness

(c.f.) There are $n! \sim n^n$ many

$\langle 1, 2, 3, 4, 5 \rangle \rightarrow$ Hamilton cycle \rightarrow witness

$\langle 1, 2, 3, 5, 4 \rangle \rightarrow$ not Hamilton cycle

$\langle 1, 4, 3, 2, 5 \rangle \rightarrow$ not Hamilton cycle

$R_D(x, w) \leftrightarrow [x \text{ is a code of a graph } G(\text{with } n \text{ vertices})]$

$\wedge [w \text{ is a permutation of } 1 \sim n: \langle l_1, l_2, \dots, l_n \rangle]$

$\wedge [w \text{ represents a Hamilton cycle in } G]$

For each $x \in \Sigma^*$ we have

if x is a code of a graph G :

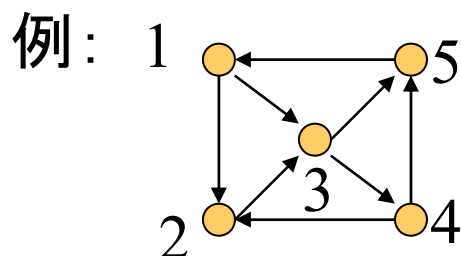
$x \in \text{DHAM} \leftrightarrow \exists w_G (= \langle l_1, \dots, l_n \rangle) [R_D(x, w_G)]$

if x is not a code of any graph: $\forall w [\neg R_D(x, w)]$

例5.7: ハミルトン閉路問題 (DHAM) $\in \mathcal{NP}$

グラフの頂点は $1 \sim n$ と番号づけされていると仮定.

ハミルトン閉路の辿り方 $\rightarrow 1 \sim n$ の順列 $\langle l_1, l_2, \dots, l_n \rangle$
 この順列が多項式長の**証拠**



証拠の候補

(注) 全部で $n! \sim n^n$ 通りある

$\langle 1, 2, 3, 4, 5 \rangle \rightarrow$ ハミルトン閉路 \rightarrow 証拠

$\langle 1, 2, 3, 5, 4 \rangle \rightarrow$ ハミルトン閉路でない

$\langle 1, 4, 3, 2, 5 \rangle \rightarrow$ ハミルトン閉路でない

$R_D(x, w) \leftrightarrow [x$ はあるグラフ $G(n$ 頂点)のコード]

$\wedge [w$ は $1 \sim n$ の順列 $\langle l_1, l_2, \dots, l_n \rangle$]

$\wedge [w$ は G のハミルトン閉路を表している]

すべての $x \in \Sigma^*$ について次の関係が成り立つ.

x があるグラフ G のコードになっているとき:

$x \in \text{DHAM} \leftrightarrow \exists w_G (= \langle l_1, \dots, l_n \rangle) [R_D(x, w_G)]$

x がグラフのコードになっていないとき: $\forall w [\neg R_D(x, w)]$

Ex.5.8: Satisfiability Problem of Prop. Express. (3SAT, SAT, ExSAT)

Goal: ExSAT $\in \mathcal{NP}$

$F(x_1, \dots, x_n)$: arbitrary extended prop. logic. expression

F is satisfiable $\leftrightarrow \exists a_1, \dots, a_n$: each a_i is 0 or 1 [$F(a_1, \dots, a_n) = 1$]

length of a witness q_E

Truth assignment to F is denoted by $\langle a_1, \dots, a_n \rangle$.

\rightarrow its length is $3(n+n+1) = 6n+3 \leq 6\lceil |F| \rceil + 3$

$$q_E(l) = 6l+3$$

predicate R_E

$R_E(x, w) \leftrightarrow [x \text{ is a code of an extended prop. express. } F \text{ (} n \text{ variables)}]$

$\wedge [w \text{ is an assignment to } F : \langle a_1, a_2, \dots, a_n \rangle]$

$\wedge [F(a_1, \dots, a_n) = 1]$

Using a computation tree, the value of $F(a_1, \dots, a_n)$ is computed in polynomial time. Thus, R_E is also computable in polynomial time.

例5.8: 命題論理式充足性問題(3SAT, SAT, ExSATなど)

目標: $\text{ExSAT} \in \mathcal{NP}$

$F(x_1, \dots, x_n)$: 任意の拡張命題論理式

F が充足可能 $\leftrightarrow \exists a_1, \dots, a_n$: 各 a_i は1か0 [$F(a_1, \dots, a_n) = 1$]

証拠の長さ q_E

F への真偽値の割り当てを $\langle a_1, \dots, a_n \rangle$ で表す.

→ 長さは $3(n+n+1) = 6n+3 \leq 6|F| + 3$

$$q_E(l) = 6l+3$$

述語 R_E

$R_E(x, w) \leftrightarrow [x$ はある拡張命題論理式 F (n 変数)のコード]

$\wedge [w$ は F への割り当て $\langle a_1, a_2, \dots, a_n \rangle$]

$\wedge [F(a_1, \dots, a_n) = 1]$

計算木を用いると $F(a_1, \dots, a_n)$ の値は多項式時間で計算可能.
よって, R_E も多項式時間で計算可能.

What does it mean by being an \mathcal{NP} set?

Using q and R satisfying the predicate characterizing an \mathcal{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 $q(|x|)$, then we can accept or reject them. Here note that there are 2 to the $q(|x|)$ (exponentially many) such strings.

We may think that those sets recognizable as above are \mathcal{NP} sets.

*NP*集合であることの意味は何か?

(5.1)を満たす q, R を用いると, $x \in L?$ を次のように判定できる.

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

長さが $q(|x|)$ 以下の文字列をすべて列挙して調べれば, acceptかrejectか判定できる. ただし, そのような文字列は2の $q(|x|)$ 乗個(指数関数)存在することに注意.

上記の計算方式で認識できる集合を*NP*集合と考えてよい.

Classes related to \mathcal{NP}

Def.5.3.

A set L is called a **co- \mathcal{NP}** set if its complement \bar{L} belongs to \mathcal{NP} .
The whole family of co- \mathcal{NP} sets is called the **class co- \mathcal{NP}** .

Note: It is nonsense to define co- \mathcal{P} since it is equal to \mathcal{P} .

Theorem 5.5. For every set L , the following conditions are equivalent.

(a) $L \in \text{co-}\mathcal{NP}$

(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 .

\mathcal{NP} に関連したクラス

定義5.3. 集合 L は, その補集合 \overline{L} が \mathcal{NP} に属しているとき, **co- \mathcal{NP} 集合** という. また, co- \mathcal{NP} 集合の全体を **クラス co- \mathcal{NP}** という.

補注: co- \mathcal{P} を定義しても \mathcal{P} と同じなので無意味.

定理5.5. すべての集合 L に対し, 次の条件は同値.

- (a) $L \in \text{co-}\mathcal{NP}$
- (b) 集合 L を, 適当な多項式 q と多項式時間計算可能述語 Q を用いて,

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

と表せる.

Ex.5.9: Primality testing

$$\overline{[n] \in \text{PRIME}} \leftrightarrow \exists m : 1 < m < n [n \bmod m = 0]$$

Therefore, for $q_p(n) = n$,

$$R_p(x, w) \leftrightarrow [x \notin \mathbf{N}] \vee [[w \in \mathbf{N}] \wedge [1 < m < n] \wedge [n \bmod m = 0]]$$

(where n and m are natural numbers represented by x and w .

\mathbf{N} is a set of all natural numbers in the binary form)

This definition leads to

for every $x \in \Sigma^*$ we have $x \notin \text{PRIME} \leftrightarrow \exists q_p w [R_p(x, w)]$

This is a witness to $x \notin \text{PRIME}$

Thus, $\overline{\text{PRIME}} \in \mathcal{NP}$, i.e., $\text{PRIME} \in \text{co-}\mathcal{NP}$

In fact, using $Q(x, w) \leftrightarrow \neg R_p(x, w)$, PRIME can be expressed as

$$\text{PRIME} = \{x : \forall q_p w [Q_p(x, w)]\}$$

We can also show that $\text{PRIME} \in \mathcal{NP}$, but its proof is more complex.

例5.9: 素数判定問題

$$\lceil n \rceil \notin \text{PRIME} \leftrightarrow \exists m: 1 < m < n [n \bmod m = 0]$$

したがって, $q_p(n) = n$ とし,

$$R_p(x, w) \leftrightarrow [x \notin \mathbb{N}] \vee [[w \in \mathbb{N}] \wedge [1 < m < n] \wedge [n \bmod m = 0]]$$

(ただし, n, m は各々 x, w が表す自然数,
 \mathbb{N} は自然数の2進表記全体)

と定義すると,

$$\text{すべての } x \in \Sigma^* \text{ に対し, } x \notin \text{PRIME} \leftrightarrow \exists q_p w [R_p(x, w)]$$

これは, $x \notin \text{PRIME}$ に対する証拠

$$\text{よって, } \overline{\text{PRIME}} \in \mathcal{NP}, \text{ i.e., } \text{PRIME} \in \text{co-}\mathcal{NP}$$

実際, $Q(x, w) \leftrightarrow \neg R_p(x, w)$ とすると

$$\text{PRIME} = \{x: \forall q_p w [Q_p(x, w)]\}$$

と表せる.

PRIME $\in \mathcal{NP}$ も示せるが, その証明はもっと複雑.

Examples of \mathcal{NP} problems

- **Composite Number Testing Problem**(COMPOSITE)

input: natural number n

question: Is n composite? (Is it not prime?)

- **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) .

\mathcal{NP} 問題の例

- **合成数判定問題**(COMPOSITE)

入力: 自然数 n

質問: n は合成数か? (素数でないか?)

- **ナップサック問題**(KNAP)

入力: 自然数の組 $\langle a_1, a_2, \dots, a_n, b \rangle$

質問: $\sum_{i \in S} a_i = b$ となる添字の集合 $S \subseteq \{1, \dots, n\}$ があるか?

- **箱詰め問題**(BIN)

入力: 自然数の組 $\langle a_1, a_2, \dots, a_n, b, k \rangle$

質問: 添字の集合 $U = \{1, \dots, n\}$ を U_1, \dots, U_k の k 個に分割し,

各 j で $\sum_{i \in U_j} a_i \leq b$ とすることは可能か?

- **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 に k 頂点の頂点被覆が存在するか?

頂点被覆 S :
どの辺 (u, v) も
 u, v の一方は
 S に含まれる

5.3. Relation in the Complexity Class

Theorem 5.6: $\mathcal{P} \subseteq \mathcal{E} \subseteq \mathcal{EXP}$.

Obvious from the definition.

Theorem 5.7: $\mathcal{P} \subsetneq \mathcal{E} \subsetneq \mathcal{EXP}$.

Hierarchy Thm. (Thm. 4.4):

For any times t_1, t_2 ,

$$\forall c > 0, \forall n [ct_1(n)^2 \leq t_2(n)]$$

$$\rightarrow \text{TIME}(t_1) \subsetneq \text{TIME}(t_2)$$

Proof:

(1) $\mathcal{P} \subsetneq \mathcal{E}$.

For $t_1(n)=2^n, t_2(n)=2^{3n}$, from the hierarchy theorem we have

$$\text{TIME}(2^n) \subsetneq \text{TIME}(2^{3n})$$

On the other hand, since $\mathcal{P} \subseteq \text{TIME}(2^n) \subsetneq \text{TIME}(2^{3n}) \subseteq \mathcal{E}$

$$\mathcal{P} \subsetneq \mathcal{E}.$$

(2) is similar.

Q.E.D.

5.3. 計算量クラス間の関係

定理5.6: $\mathcal{P} \subseteq \mathcal{E} \subseteq \mathcal{EXP}$.

定義より, 明らか.

定理5.7: $\mathcal{P} \subsetneq \mathcal{E} \subsetneq \mathcal{EXP}$.

証明:

(1) $\mathcal{P} \subsetneq \mathcal{E}$.

$t_1(n)=2^n, t_2(n)=2^{3n}$ とすると, 階層定理より,

$$\text{TIME}(2^n) \subsetneq \text{TIME}(2^{3n})$$

一方, $\mathcal{P} \subseteq \text{TIME}(2^n) \subsetneq \text{TIME}(2^{3n}) \subseteq \mathcal{E}$ だから,

$$\mathcal{P} \subsetneq \mathcal{E}.$$

(2)も同様.

階層定理(定理4.4):

任意の制限時間 t_1, t_2 に対し、

$$\forall c > 0, \forall n [c t_1(n)^2 \leq t_2(n)]$$

$$\rightarrow \text{TIME}(t_1) \subsetneq \text{TIME}(t_2)$$

証明終

Theorem 5.8.

(1) $\mathcal{P} \subseteq \mathcal{NP}$, $\mathcal{P} \subseteq \text{co-}\mathcal{NP}$ (thus, $\mathcal{P} \subseteq \mathcal{NP} \cap \text{co-}\mathcal{NP}$)

(2) $\mathcal{NP} \subseteq \mathcal{EXPTIME}$, $\text{co-}\mathcal{NP} \subseteq \mathcal{EXPTIME}$ (thus, $\mathcal{NP} \cup \text{co-}\mathcal{NP} \subseteq \mathcal{EXPTIME}$)

Proof:

(1) $\mathcal{P} \subseteq \mathcal{NP}$ ($\mathcal{P} \subseteq \text{co-}\mathcal{NP}$ is similar)

L : arbitrary \mathcal{P} set

→ L is recognizable in polynomial time

Thus, we have the following description using a polynomial-time computable predicate P :

$$\forall x \in \Sigma^* : [x \in L \leftrightarrow P(x)] \text{ or } P = \{x : P(x)\}$$

We define $R(x, w) = P(x)$ (neglecting the second argument)

→ for any polynomial q ,

$$L = \{x : \exists_q w [R(x, w)]\}$$

Thus, from the definition of \mathcal{NP} , $L \in \mathcal{NP}$ i.e., $\mathcal{P} \subseteq \mathcal{NP}$.

定理5.8.

- (1) $\mathcal{P} \subseteq \mathcal{NP}$, $\mathcal{P} \subseteq \text{co-}\mathcal{NP}$ (よって, $\mathcal{P} \subseteq \mathcal{NP} \cap \text{co-}\mathcal{NP}$)
 (2) $\mathcal{NP} \subseteq \mathcal{EXP}$, $\text{co-}\mathcal{NP} \subseteq \mathcal{EXP}$ (よって, $\mathcal{NP} \cup \text{co-}\mathcal{NP} \subseteq \mathcal{EXP}$)

証明: (1) $\mathcal{P} \subseteq \mathcal{NP}$ ($\mathcal{P} \subseteq \text{co-}\mathcal{NP}$ も同様)

L : 任意の \mathcal{P} 集合

→ L は多項式時間で認識可能

よって, 多項式時間計算可能述語 P を用いて次のように書ける.

$$\forall x \in \Sigma^*: [x \in L \leftrightarrow P(x)] \quad \text{or} \quad P = \{x: P(x)\}$$

$R(x, w) = P(x)$ と定義 (第2引数は無視)

→ 任意の多項式 q について,

$$L = \{x: \exists_{q} w [R(x, w)]\}$$

よって, \mathcal{NP} の定義より, $L \in \mathcal{NP}$ i.e., $\mathcal{P} \subseteq \mathcal{NP}$.

(2) $\mathcal{NP} \subseteq \mathcal{EXP}$ (co- $\mathcal{NP} \subseteq \mathcal{EXP}$)

L : any \mathcal{NP} set

→ There is some polynomial q and polynomial-time computable predicate R such that

$$L = \{x : \exists_q w [R(x, w)]\} = \{x : \exists_q w [|w| \leq q(|x|) \wedge R(x, w)]\}$$

prog L(input x);

begin

for each $w \in \Sigma^{\leq q(|x|)}$ do

if $R(x, w)$ then accept end-if

end-for;

reject

end.

program recognizing L using q
and R

time complexity of the program for an input of length l :

Since R is polynomial-time computable, for some polynomial q

time of $R = p(|x| + |w|) \leq p(l + q(l)) \leftarrow$ polynomial of l

In total, $\{p(l + q(l)) + cq(l)\} 2^{q(l)} + d = O(2^{l+q(l)})$

Hence, $L \in \mathcal{EXP} \rightarrow \mathcal{NP} \subseteq \mathcal{EXP}$

Q.E.D.

(2) $\mathcal{NP} \subseteq \mathcal{EXP}$ ($\text{co-}\mathcal{NP} \subseteq \mathcal{EXP}$)

L : 任意の \mathcal{NP} 集合

→ 多項式 q と多項式時間計算可能述語 R が存在して,

$$L = \{x : \exists_q w [R(x, w)]\} = \{x : \exists_q w [|w| \leq q(|x|) \wedge R(x, w)]\}$$

q と R を用いて, L を認識するプログラムを作る.

```

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

```

長さ l の入力に対するプログラムの時間計算量:

R は多項式時間計算可能だったから, ある多項式 p に対し,

R の計算時間 $= p(|x| + |w|) \leq p(l + q(l)) \leftarrow l$ の多項式
 全体では, $\{p(l+q(l)) + cq(l)\} 2^{q(l)} + d = O(2^{l+q(l)})$

よって, $L \in \mathcal{EXP} \rightarrow \mathcal{NP} \subseteq \mathcal{EXP}$

証明終

Theorem 5.9

- (1) $\mathcal{NP} \subseteq \text{co-}\mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$
- (2) $\text{co-}\mathcal{NP} \subseteq \mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$
- (3) $\mathcal{NP} \neq \text{co-}\mathcal{NP} \rightarrow \mathcal{P} \neq \mathcal{NP}$

Note: from (3) the proof for $\mathcal{NP} \neq \text{co-}\mathcal{NP}$ is harder than that for $\mathcal{P} \neq \mathcal{NP}$.

Proof: (1) $\mathcal{NP} \subseteq \text{co-}\mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$ (proof of (2) is similar)
 Since $\text{co-}\mathcal{NP} \subseteq \mathcal{NP}$ is shown if we prove $L \in \mathcal{NP}$ for any $L \in \text{co-}\mathcal{NP}$
 Combining it with the assumption $\mathcal{NP} \subseteq \text{co-}\mathcal{NP}$, we have
 $\mathcal{NP} = \text{co-}\mathcal{NP}$ and so

$$\begin{aligned}
 L \in \text{co-}\mathcal{NP} &\rightarrow \overline{L} \in \mathcal{NP} && \text{(by Definition 5.3)} \\
 &\rightarrow \overline{\overline{L}} \in \text{co-}\mathcal{NP} && (\mathcal{NP} \subseteq \text{co-}\mathcal{NP}) \\
 &\rightarrow L \in \mathcal{NP} && \text{(Definition 5.3 and } L=\overline{\overline{L}})
 \end{aligned}$$

定理5.9.

- (1) $\mathcal{NP} \subseteq \text{co-}\mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$
- (2) $\text{co-}\mathcal{NP} \subseteq \mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$
- (3) $\mathcal{NP} \neq \text{co-}\mathcal{NP} \rightarrow \mathcal{P} \neq \mathcal{NP}$.

補注: (3)より, $\mathcal{NP} \neq \text{co-}\mathcal{NP}$ の証明は, $\mathcal{P} \neq \mathcal{NP}$ の証明より難しい.

証明: (1) $\mathcal{NP} \subseteq \text{co-}\mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$ ((2)の証明も同様)
 任意の $L \in \text{co-}\mathcal{NP}$ に対して $L \in \mathcal{NP}$ が示せれば, $\text{co-}\mathcal{NP} \subseteq \mathcal{NP}$
 が証明できるので, 仮定の $\mathcal{NP} \subseteq \text{co-}\mathcal{NP}$ と合わせて $\mathcal{NP} = \text{co-}\mathcal{NP}$
 が言える.

$$\begin{aligned}
 L \in \text{co-}\mathcal{NP} &\rightarrow \overline{L} \in \mathcal{NP} && \text{(定義5.3より)} \\
 &\rightarrow \overline{\overline{L}} \in \text{co-}\mathcal{NP} && (\mathcal{NP} \subseteq \text{co-}\mathcal{NP} \text{より}) \\
 &\rightarrow L \in \mathcal{NP} && \text{(定義5.3と } L = \overline{\overline{L}} \text{より)}
 \end{aligned}$$

$$(3) \mathcal{NP} \neq \text{co-}\mathcal{NP} \rightarrow \mathcal{P} \neq \mathcal{NP}$$

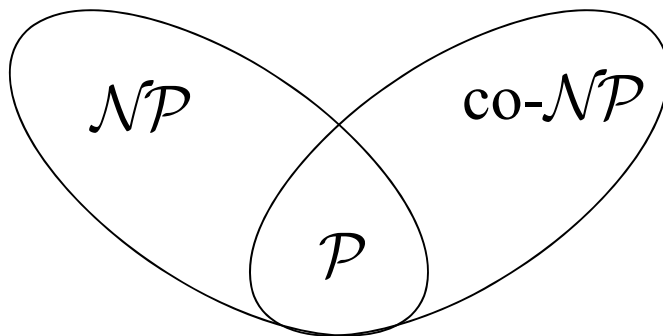
$$\text{Contraposition: } \mathcal{P} = \mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$$

If we assume $\mathcal{P} = \mathcal{NP}$, for any L we have

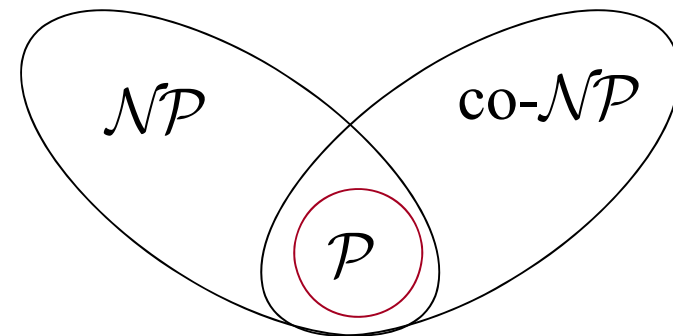
$$\begin{aligned} L \in \mathcal{NP} &\leftrightarrow \underline{L} \in \mathcal{P} && (\mathcal{P} = \mathcal{NP}) \\ &\leftrightarrow \overline{L} \in \mathcal{P} && (\text{Exercise 5.5}) \\ &\leftrightarrow \overline{\overline{L}} \in \underline{\mathcal{NP}} && (\mathcal{P} = \mathcal{NP}) \\ &\leftrightarrow L (= \overline{\overline{L}}) \in \text{co-}\mathcal{NP} && (\text{Definition 5.3}) \\ &\therefore \mathcal{NP} = \text{co-}\mathcal{NP} \end{aligned}$$

Q.E.D.

If $\mathcal{NP} \neq \text{co-}\mathcal{NP}$ is true,



or



(3) $\mathcal{NP} \neq \text{co-}\mathcal{NP} \rightarrow \mathcal{P} \neq \mathcal{NP}$.

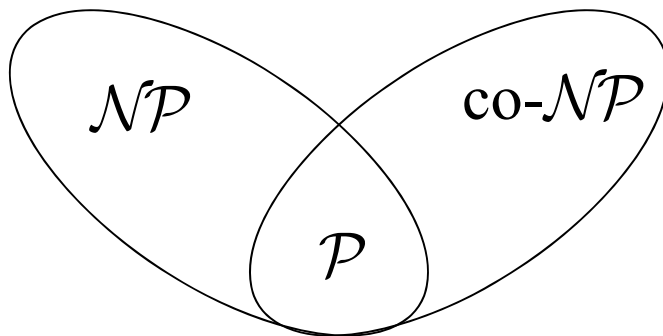
対偶: $\mathcal{P} = \mathcal{NP} \rightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$

$\mathcal{P} = \mathcal{NP}$ と仮定すると, すべての L に対し

$$\begin{aligned}
 L \in \mathcal{NP} &\leftrightarrow \underline{L} \in \mathcal{P} && (\mathcal{P} = \mathcal{NP} \text{ より}) \\
 &\leftrightarrow \overline{L} \in \mathcal{P} && (\text{演習問題5.5}) \\
 &\leftrightarrow \overline{\overline{L}} \in \underline{\mathcal{NP}} && (\mathcal{P} = \mathcal{NP} \text{ より}) \\
 &\leftrightarrow L (= \overline{\overline{L}}) \in \text{co-}\mathcal{NP} && (\text{定義5.3より}) \\
 &\therefore \mathcal{NP} = \text{co-}\mathcal{NP}
 \end{aligned}$$

証明終

$\mathcal{NP} \neq \text{co-}\mathcal{NP}$ が正しいと



or

