

I238 計算の理論

上原 隆平

2019年I-1期(4-5月)

Announce

- Report 2: 出題は5月23日, 締切は6月4日10:50
(Distributed today; Deadline is 10:50am, June 4.)
- 5月23日のチュートリアルアワーは補講
(Today, I'll give supplemental lecture on 13:30-15:10.)
- 5月28日, 30日は出張のため休講
(Lectures will be cancelled on May 28 and May 30.)

I238 Computation Theory

by

Prof. Ryuhei Uehara

Term I-1, April-May, 2019

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

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

6. 計算量の理論

6.1. 計算時間の評価

6.1.3. 問題の時間計算量

定義6.5: 自然数上の関数 $t(n)$ に対して, 時間計算量 $O(t(n))$ の集合(認識問題)全体の集合を **$O(t(n))$ 時間計算量クラス** とよび, **$\text{TIME}(t(n))$** とかく. こうした関数 $t(n)$ は制限時間と呼ぶ.

6.2. 代表的な計算量クラス

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

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

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

C集合: 計算量クラスCに入る集合.

C問題: C集合の認識問題

ある問題がPに入っていないなら、
現実的には手に負えない...

6. Computational Complexity

6.1. Time Complexity Classes

6.1.3. Time complexity of a problem

Definition 6.5: For a function $t(n)$ over natural numbers, the set of all sets (i.e. recognition problems) with time complexities $O(t(n))$ is called **$O(t(n))$ -time complexity class**, and it is denoted by **$\text{TIME}(t(n))$** . Such a function $t(n)$ is called a time limit.

6.2. Representative time complexity classes

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

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

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

C set: set in the complexity class C.

C problem: problem of recognizing a C set.

Problems not in P are intractable from the practical viewpoint...

6. 計算量の理論

6.3. クラス NP

6.3.1. 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」

6. Computational Complexity

6.3. Class NP

6.3.1. 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”

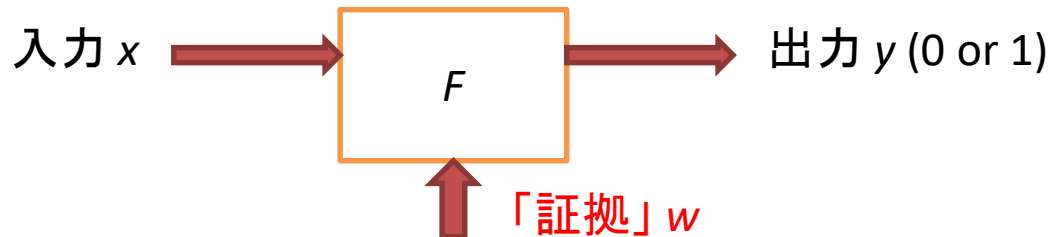
6. 計算量の理論

6.3. クラス NP

6.3.1. 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 $x \in L$ を認識する

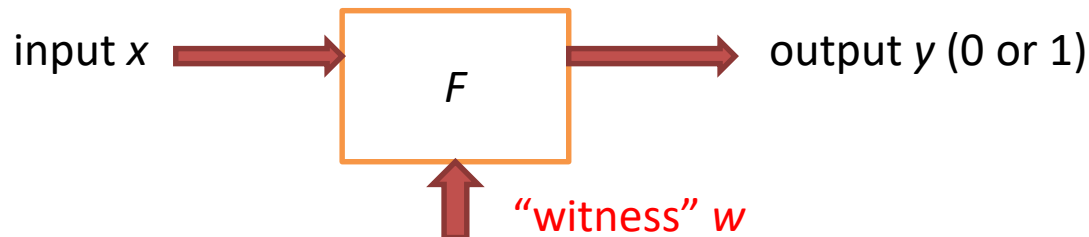
6. Computational Complexity

6.3. Class NP

6.3.1. 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=Nondeterministic Polynomial

6. 計算量の理論

6.3. クラスNP

6.3.1. 非決定性計算とは

「非決定性計算」

- 論理の視点からみると:

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:

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

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

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=Nondeterministic Polynomial

この文字列 w を「証拠」とよぶ

6. Computational Complexity

6.3. Class NP

6.3.1. Nondeterministic computation

“Nondeterministic computation”

- From the viewpoint of Logic:

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

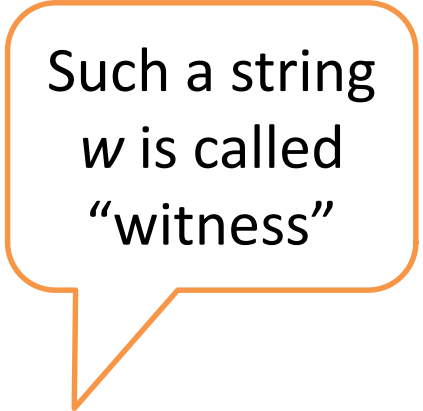
for each $x \in \Sigma^*$, $x \in L \leftrightarrow \exists w \in \Sigma^* : |w| \leq q(|x|) [R(x, w)]$

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

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 = Nondeterministic Polynomial



Such a string w is called “witness”

6. 計算量の理論

6.3. クラス NP

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

6.3.1. 非決定性計算とは

「非決定性計算」

- チューリング機械の視点から見ると:

チューリングマシンの「非決定性選択」では、複数の選択肢を「同時に」すべて選ぶことができる；つまり「場合(0)と場合(1)のいずれか」という命令がある。

- 非決定性選択は複数の選択肢のうち、「どれかが真」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。

6. Computational Complexity

6.3. Class NP

6.3.1. Nondeterministic computation

“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.

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

6. 計算量の理論

6.3. クラス NP

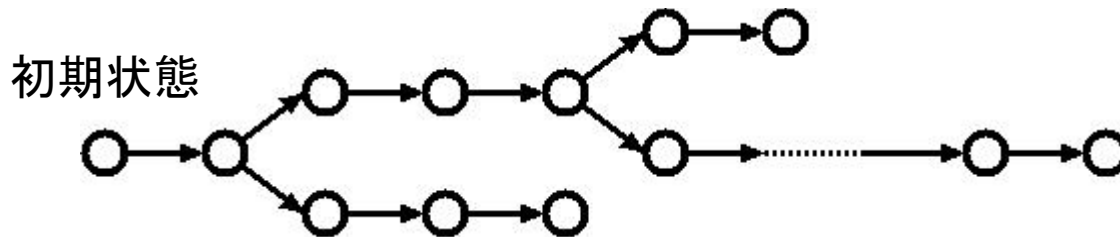
6.3.1. 非決定性計算とは

• チューリング機械の計算木の観点からみると:

• 決定性のチューリング機械の計算木はパス(一本道);



• 非決定性のチューリング機械の計算木は木;



- 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.

6. Computational Complexity

6.3. Class NP

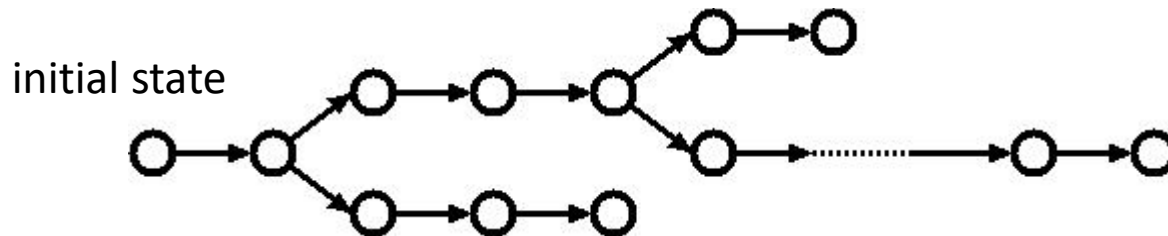
6.3.1. Nondeterministic computation

- 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.

6. 計算量の理論

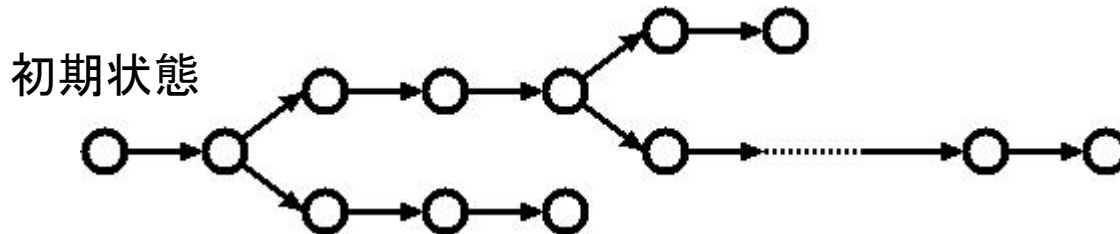
6.3. クラス NP

証拠 w は正しい選択枝の列を与える

6.3.1. 非決定性計算とは

• チューリング機械の計算木の観点からみると:

• 非決定性のチューリング機械の計算木は木;



- 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.

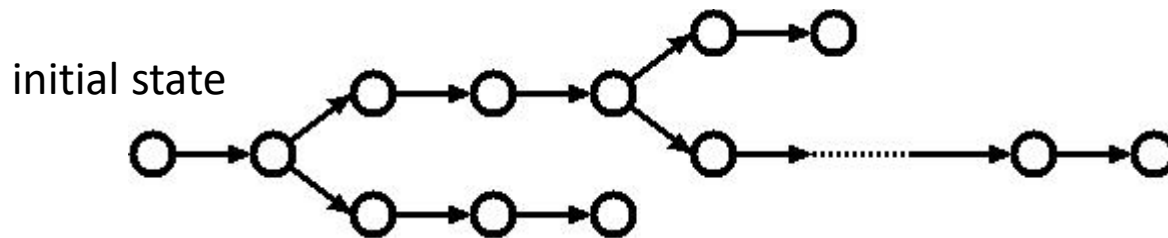
6. Computational Complexity

6.3. Class NP

6.3.1. 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*;



each computation halts in an accept/reject state

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 .

6. 計算量の理論

6.3. クラス NP

6.3.1. 非決定性チューリング機械の形式的定義

決定性チューリング機械の形式的定義(復習):

チューリング機械とは7つ組 $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ で,

Q は状態集合(有限集合)

Σ は空白記号 b を含まない入力アルファベット(有限集合)

Γ はテープアルファベットで $b \in \Gamma$ かつ $\Sigma \subseteq \Gamma$ (有限集合)

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ は遷移関数

$q_0 \in Q$ は初期状態

$q_a \in Q$ は受理状態

$q_r \in Q$ は拒否状態で, $q_a \neq q_r$

6. Computational Complexity

6.3. Class NP

6.3.1. Formal definition of nondeterministic Turing Machine

Formal definition of deterministic Turing machine (again):

A Turing machine is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, where

Q is the set of states,

Σ is the input alphabet not containing the blank b ,

Γ is the tape alphabet, where $b \in \Gamma$ and $\Sigma \subseteq \Gamma$,

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function,

$q_0 \in Q$ is the start state,

$q_a \in Q$ is the accept state, and

$q_r \in Q$ is the reject state, where $q_a \neq q_r$

6. 計算量の理論

6.3. クラス NP

決定性チューリング機械:

– 状態遷移は:

- 関数 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ により決まる.

非決定性チューリング機械:

– 状態遷移は:

- 関数 $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$ により決まる.

ある状態における「次の状態」が複数あってよい.

6. Computational Complexity

6.3. Class NP

Deterministic Turing machine:

A transition is:

- Determined by the function $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

Nondeterministic Turing machine:

– A transition is :

- Determined by the function $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$

Many “next states” can exist for each state.

6. 計算量の理論

6.3. クラス NP

決定性チューリング機械:

– 計算とは:

- 初期状態 q_0 から始まり,
- 与えられたテープ上の文字列で状態遷移し,
- q_a か q_r に到達したら停止する.
- q_a で停止したら入力を受理, q_r で停止したら入力を拒否と考える.

非決定性チューリング機械:

– 計算とは:

- 初期状態 q_0 から始まり,
- 与えられたテープ上の文字列で状態の集合に遷移し,
- 遷移した状態集合が q_a を含んでいたら受理して停止する.
 - 上記以外の場合は定義する必要がない(モデルによって色々だが, どれも同じ)

6. Computational Complexity

6.3. Class NP

Deterministic Turing machine :

- A deterministic computation:
 - It starts from the start configuration q_0 ,
 - makes transitions on a given tape string, and
 - halts when it reach to q_a or q_r .
 - It **accepts** on q_a , or **rejects** on q_r .

Nondeterministic Turing machine :

- A nondeterministic computation :
 - It starts from the start configuration q_0 ,
 - makes transitions on a given tape string, and
 - halts when it reach to the set of states that contains q_a or q_r .
 - It **accepts** on q_a , or **rejects** on q_r .
 - It is not essential the case that the set contains both

6. 計算量の理論

6.3. クラスNP

クラスNPの非決定性チューリング機械による定義:

- 集合LがクラスNPに入る必要十分条件は, ある非決定性チューリング機械 M と多項式 q が存在し, 以下を満たすとき:

$x \in L \Leftrightarrow M$ が高々 $q(|x|)$ 時間で x を受理する

6. Computational Complexity

6.3. Class NP

Definition of the class NP by using nondeterministic Turing machine:

- A set L is in the class NP if and only if there are a nondeterministic Turing machine M and a polynomial q such that:

$$x \in L \Leftrightarrow M \text{ accepts } x \text{ at most } q(|x|) \text{ time.}$$

6. 計算量の理論

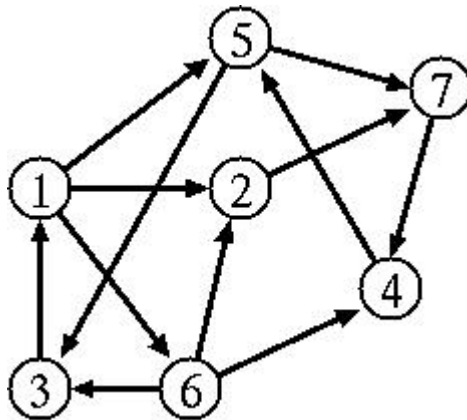
6.3. クラス NP

6.3.2. 代表的な NP 問題

問題 6.5: 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?



- 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.

6. Computational Complexity

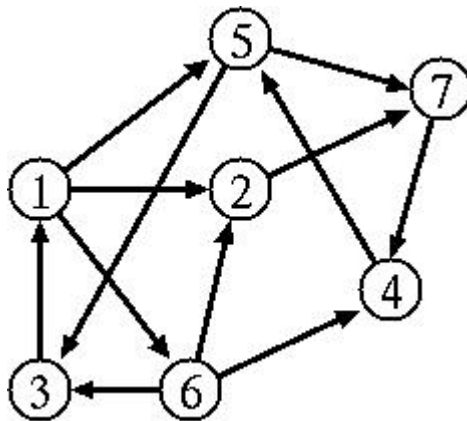
6.3. Class NP

6.3.2. Representative NP problems

Problem 6.5: 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.

6. 計算量の理論

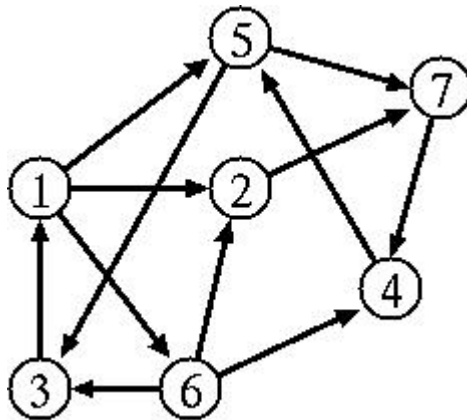
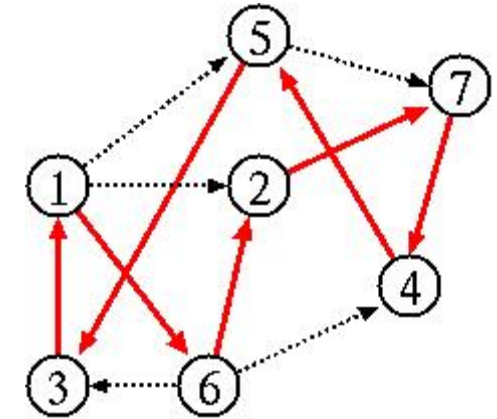
6.3. クラス NP

6.3.2. 代表的な NP 問題

問題6.5: 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?



- 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.

6. Computational Complexity

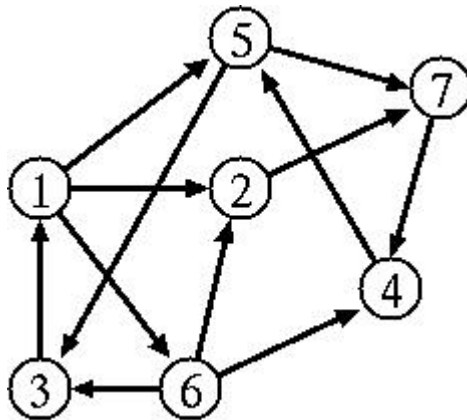
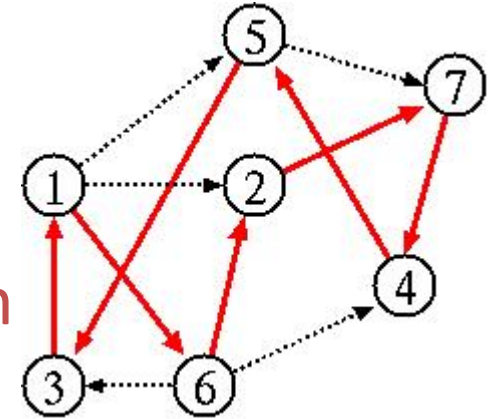
6.3. Class NP

6.3.2. Representative NP problems

Problem 6.5: Hamiltonian cycle problem

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.

6. 計算量の理論

6.3. クラス NP

6.3.2. 代表的な NP 問題

問題6.2: SAT, kSAT, 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.

6. Computational Complexity

6.3. Class NP

6.3.2. Representative NP problems

Problem 6.2: 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.

6. 計算量の理論

6.3. クラス NP

6.3.3. 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。

6. Computational Complexity

6.3. Class NP

6.3.3. 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.

6. 計算量の理論

6.3. クラスNP

6.3.4. 代表的なNP問題再び

問題6.6: 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?

問題6.7: 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?

問題6.8: 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

6. Computational Complexity

6.3. Class NP

6.3.4. More representative NP problems

Problem 6.6: 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. $\sum_{i \in S} a_i = b$?

Problem 6.7: 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 ?

Problem 6.8: 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 $\{u, v\}$.

6. 計算量の理論

6.4. クラス coNP

定義6.7:

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6.6:

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。

6. Computational Complexity

6.4. Class coNP

Definition 6.7:

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

Theorem 6.6:

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.

6. 計算量の理論

6.5. 計算量クラスの関係

定理6.8: $P \subseteq E \subseteq EXP$

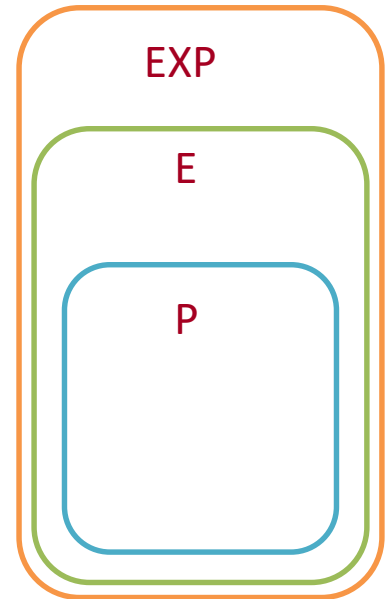
証明: 定義より明らか.

定理6.9: $P \subsetneq E \subsetneq EXP$

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.

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

真に異なる階層構造
が成立する



6. Computational Complexity

6.5. Relations in the Complexity Classes

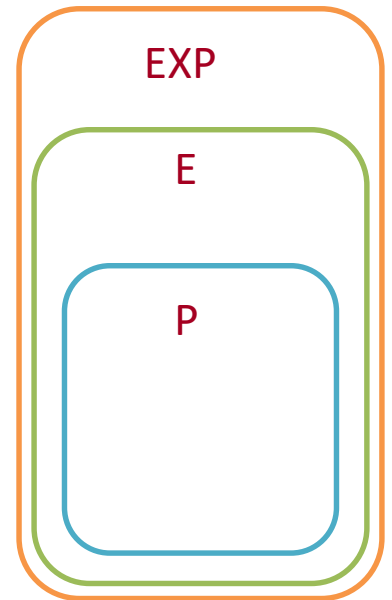
Theorem 6.8: $P \subseteq E \subseteq EXP$

Proof: Obvious from the definition.

Theorem 6.9: $P \subsetneq E \subsetneq EXP$

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



6. 計算量の理論

6.5. 計算量クラスの関係

定理6.10:

(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。

6. Computational Complexity

6.5. Relations in the Complexity Classes

Theorem 6.10:

(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.

6. 計算量の理論

6.5. 計算量クラスの関係

定理6.11:

- (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:

(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.

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

6. Computational Complexity

6.5. Relations in the Complexity Classes

Theorem 6.11:

(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 $L \in coNP$.

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

6. 計算量の理論

6.5. 計算量クラスの関係

定理6.11:

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

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

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

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

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

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

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

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

$\therefore NP = coNP$

Q.E.D.

6. Computational Complexity

6.5. Relations in the Complexity Classes

Theorem 6.11:

(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.

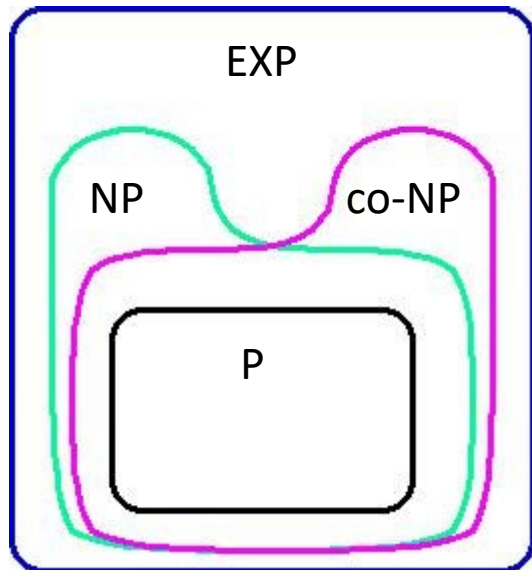
6. 計算量の理論

6.5. 計算量クラスの関係

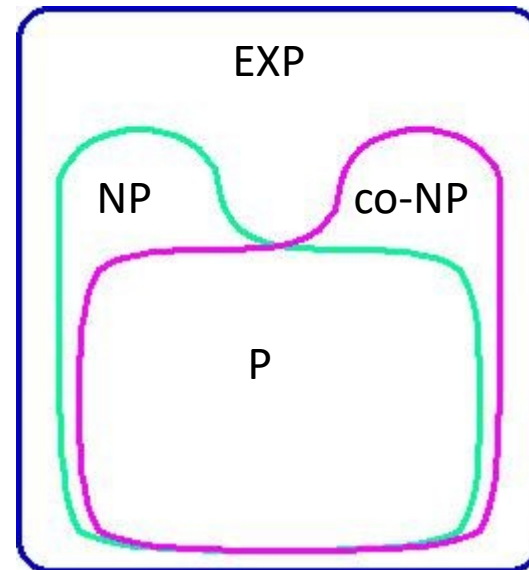
定理6.11

- (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



6. Computational Complexity

6.5. Relations in the Complexity Classes

Theorem 6.11:

(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

