アナウンス（覚書）
- レポートの締め切り：5月26日
  採点結果は金曜日か土曜日以降，
  メールで聞いてください．
  （紙で提出した人）
- 講義の残り回数：5月26日と6月2日

I216 計算量の理論と離散数学

上原隆平、面和成

I216 Computational Complexity and Discrete Mathematics

by
Prof. Ryuhei Uehara
and
Prof. Kazumasa Omote

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）が多すぎるとき
    - 関連する専門用語;

      クラスNP, P≠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

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

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

- **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 ⇔
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 ⇔
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$x \in L \Leftrightarrow \exists w \in \Sigma^* : |w| \leqq q(|x|)[R(x,w)]$

クラスcoNPの定義
集合LがクラスcoNPに入る ⇔
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$x \in L \Leftrightarrow \forall w \in \Sigma^* : |w| \leqq q(|x|)[R(x,w)]$

5. Computational Complexity

- Observation of the classes

Definition: Class P
Set L is in the class P ⇔
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 ⇔
There exists a poly q and a poly-time computable predicate R such that
for each $x \in \Sigma^*$, $x \in L \Leftrightarrow \exists w \in \Sigma^* : |w| \leqq q(|x|)[R(x,w)]$

Definition: Class coNP
Set L is in the class coNP ⇔
There exists a poly q and a poly-time computable predicate R such that
for each $x \in \Sigma^*$, $x \in L \Leftrightarrow \forall w \in \Sigma^* : |w| \leqq q(|x|)[R(x,w)]$

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$< a_1, a_2, \dots, a_n, b >$

  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$< a_1, a_2, \dots, a_n, b, k >$

  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 $<G, k>$

  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

5.3. Class NP

5.3.3. More representative NP problems

- Knapsack Problem (KNAP)

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

  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 $< a_1, a_2, \dots, a_n, b, k>$

  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 $<G, k>$

  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. Computational Complexity

5.5. Relations in the Complexity Classes

Theorem

(1) NP ⊆ coNP → NP = coNP

(2) coNP ⊆ NP → NP = coNP

(3) NP ≠ coNP → P ≠ NP

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

Proof :

(1) NP ⊆ coNP → NP = coNP

By assumption, it is sufficient to show that coNP ⊆ NP.

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

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

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

$\Leftrightarrow L \in \text{NP}$ (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 ⊆ coNP → NP = coNP

(2) coNP ⊆ 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

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

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

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

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

∴ 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

Note: From (3), proof for NP $\neq$ co-NP 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

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

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

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

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

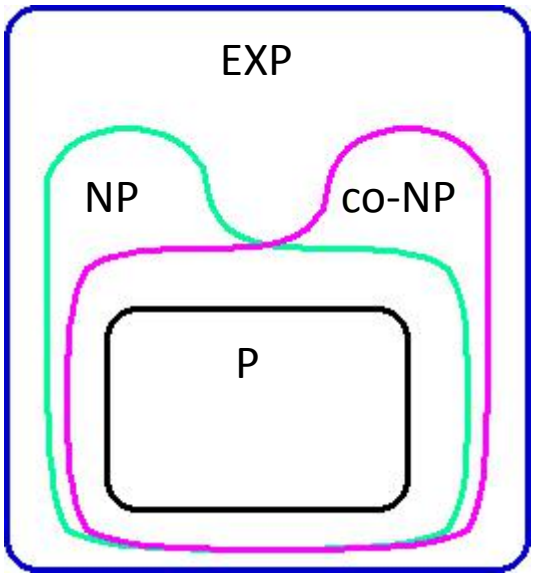
$\therefore$ NP = coNP Q.E.D.

5.計算量の理論

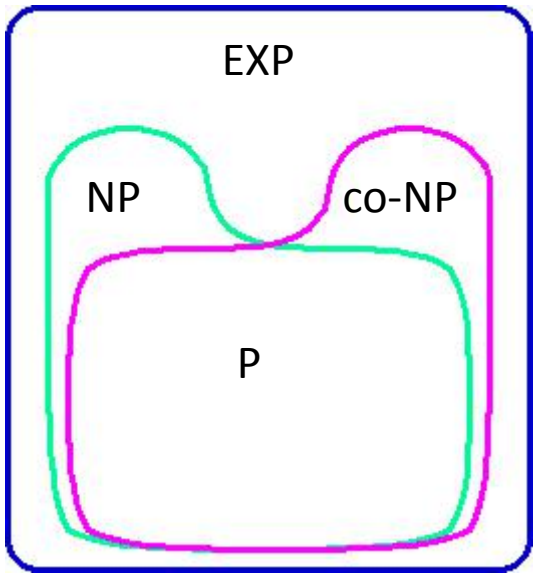
5.5. 計算量クラスの関係

定理
(1) NP ⊆ 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



5. Computational Complexity

5.5. Relations in the Complexity Classes

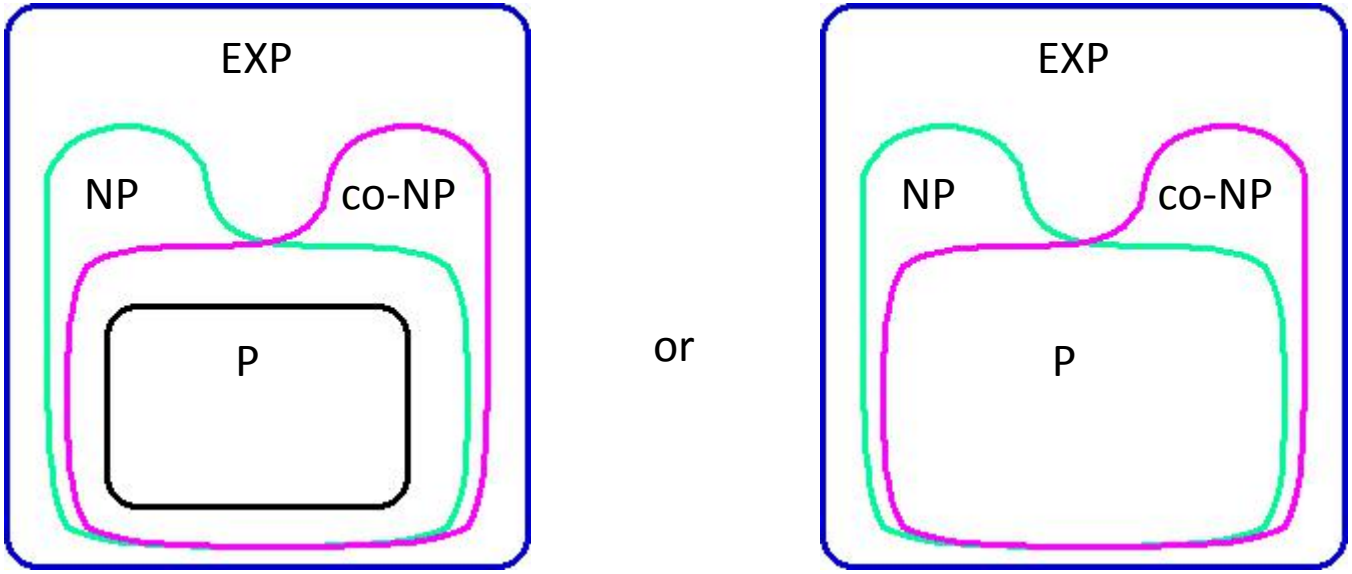
Theorem

(1) NP ⊆ coNP → NP = coNP

(2) coNP ⊆ NP → NP = coNP

(3) NP ≠ coNP → P ≠ NP

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



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 $h: A \to B$ が多項式時間還元である

$\Leftrightarrow$
- (a) h は Σ^* から Σ^* への全域関数である
- (b) $x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B]$
- (c) h は多項式時間計算可能である.

(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≦ B の難しさ)

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

$\Leftrightarrow$
- (a) h is a total function from Σ^* onto Σ^*
- (b) $x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B]$
- (c) h is polynomial-time computable.

(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* $\leqq$ 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

定理　$A \leq_m^P B$ のとき次が成立する

(1) $B \in$ P → $A \in$ P.
(2) $B \in$ NP → $A \in$ NP.
(3) $B \in$ co-NP → $A \in$ coNP.
(4) $B \in$ EXP → $A \in$ EXP.

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

例: ONE≡{1} と定義すると，Pの各集合Lに対して，

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

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 $A \leq_m^P B$ leads to

(1) $B \in$ P → $A \in$ P.
(2) $B \in$ NP → $A \in$ NP.
(3) $B \in$ co-NP → $A \in$ coNP.
(4) $B \in$ EXP → $A \in$ EXP.

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

Ex.: When we define ONE≡{1}, for each set L in P we have

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

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, 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 *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

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

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

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) $2\mathrm{SAT} \leq_m^P 3\mathrm{SAT} \leq_m^P \mathrm{SAT} \leq_m^P \mathrm{ExSAT}$

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

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) $\mathrm{2SAT} \leq_m^P \mathrm{3SAT} \leq_m^P \mathrm{SAT} \leq_m^P \mathrm{ExSAT}$

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

証明 (概略)

(2) (1)より，$\mathrm{ExSAT} \leq_m^P \mathrm{3SAT}$ 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

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

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

Proof (Outline)

(2) It is sufficient to show that $\mathrm{ExSAT} \leq_m^P \mathrm{3SAT}$ 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

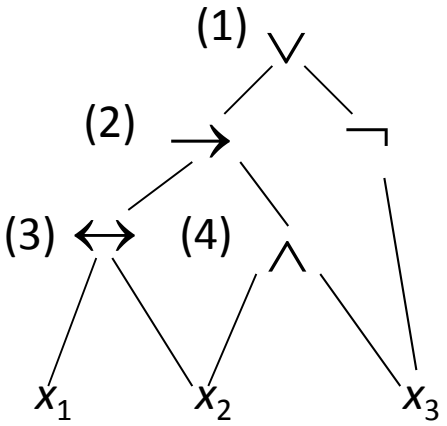
定理 (2) $3\mathrm{SAT} \equiv_m^P \mathrm{SAT} \equiv_m^P \mathrm{ExSAT}$

証明 (概略)

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

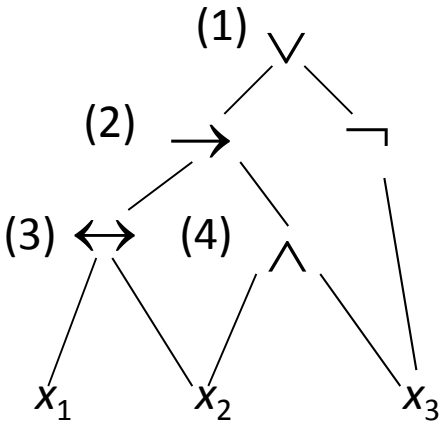
Theorem (2) $3\mathrm{SAT} \equiv_m^P \mathrm{SAT} \equiv_m^P \mathrm{ExSAT}$

Proof (Outline)

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

Reduction from ExSAT to 3SAT by an example:

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



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

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

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

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

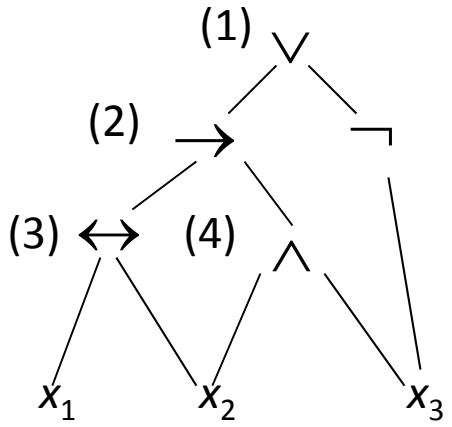
6.多項式時間計算可能性の解析手法

6.1.多項式時間還元可能性

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

ExSAT から 3SAT への還元を例で示す:

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



$$(1) V_1 \equiv V_2 \vee \neg x_3$$
$$(2) V_2 \equiv [V_3 \rightarrow V_4]$$
$$(3) V_3 \equiv [x_1 \leftrightarrow x_2]$$
$$(4) V_4 \equiv x_2 \wedge x_3$$

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

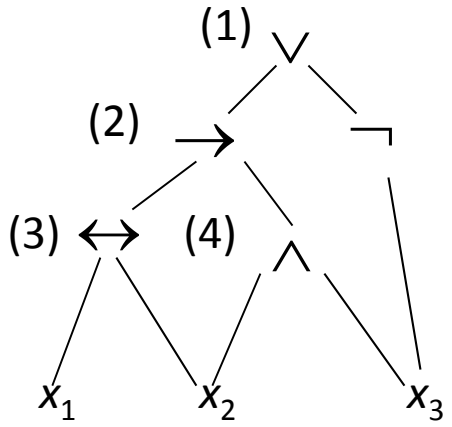
6. Analysis on Polynomial-Time Computability

6.1. Polynomial-time Reducibility

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

Reduction from ExSAT to 3SAT by an example:

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



$$(1) V_1 \equiv V_2 \vee \neg x_3$$
$$(2) V_2 \equiv [V_3 \rightarrow V_4]$$
$$(3) V_3 \equiv [x_1 \leftrightarrow x_2]$$
$$(4) V_4 \equiv x_2 \wedge x_3$$

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

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) $3\mathrm{SAT} \equiv_m^P \mathrm{SAT} \equiv_m^P \mathrm{ExSAT}$

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⇔*F''*()は充足可能.
F''() をこれと同値な3SATの要素 *F'*() で表現する.

$$\begin{aligned} U_1 \leftrightarrow [U_2 \vee \neg x_3] &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg [U_2 \vee \neg x_2]] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_2]] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_2] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_2 \vee x_2] \end{aligned}$$

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

6.1. Polynomial-time Reducibility

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

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 constriction, *F*() is satisfiable iff *F''*() is satisfiable.
We show *F''*() can be represented by an equivalent *F'*() in 3SAT.

$$\begin{aligned} U_1 \leftrightarrow [U_2 \vee \neg x_3] &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg [U_2 \vee \neg x_2]] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_2]] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_2] \\ &= [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_2 \vee x_2] \end{aligned}$$

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

6.2.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) $\forall L \in \mathrm{C}\ [L \leq_m^P A]$

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) $A \in \mathrm{C}$

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

Definition

For a class C, if a set A satisfies

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

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

Moreover, if we have

(b) $A \in \mathrm{C}$,

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 → C $\subseteq P$　　対偶: C$\not\subset$ P → $A \notin$P

(2) $A \in$ NP → C $\subseteq NP$　　対偶: C$\not\subset$NP → $A \notin$NP

(3) $A \in$ coNP → C $\subseteq coNP$　　対偶: C$\not\subset$ coNP → $A \notin$ coNP

(4) $A \in$ EXP → C $\subseteq EXP$　　対偶: C$\not\subset$EXP → $A \notin$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) $A \in \mathrm{P} \rightarrow \mathrm{C} \subseteq P$ CP: $\mathrm{C} \not\subset \mathrm{P} \rightarrow A \notin \mathrm{P}$

(2) $A \in \mathrm{NP} \rightarrow \mathrm{C} \subseteq NP$ CP: $\mathrm{C} \not\subset \mathrm{NP} \rightarrow A \notin \mathrm{NP}$

(3) $A \in \mathrm{coNP} \rightarrow \mathrm{C} \subseteq coNP$ CP: $\mathrm{C} \not\subset \mathrm{coNP} \rightarrow A \notin \mathrm{coNP}$

(4) $A \in \mathrm{EXP} \rightarrow \mathrm{C} \subseteq EXP$ CP: $\mathrm{C} \not\subset \mathrm{EXP} \rightarrow A \notin \mathrm{EXP}$

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 $A \in \mathrm{P}$, we have $B \in \mathrm{P}$

(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 → C $\subseteq P$　　対偶: C$\not\subset$ P → $A \notin$ P

(2) $A \in$ NP → C $\subseteq NP$　　対偶: C$\not\subset$NP → $A \notin$ NP

(3) $A \in$ coNP → C $\subseteq coNP$　対偶: C$\not\subset$ coNP → $A \notin$ coNP

(4) $A \in$ EXP → C $\subseteq EXP$　　対偶: C$\not\subset$EXP → $A \notin$ 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≠P ➔ $A \notin$ 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. For any C-hard (or C-complete) set A,

(1) $A \in P \rightarrow C \subseteq P$ CP: $C \not\subset P \rightarrow A \notin P$

(2) $A \in NP \rightarrow C \subseteq NP$ CP: $C \not\subset NP \rightarrow A \notin NP$

(3) $A \in coNP \rightarrow C \subseteq coNP$ CP: $C \not\subset coNP \rightarrow A \notin coNP$

(4) $A \in EXP \rightarrow C \subseteq EXP$ CP: $C \not\subset EXP \rightarrow A \notin EXP$

Ex. : Meaning of Theorem for class NP

Let A be NP-complete set.

By the contraposition of Theorem (1) we have

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

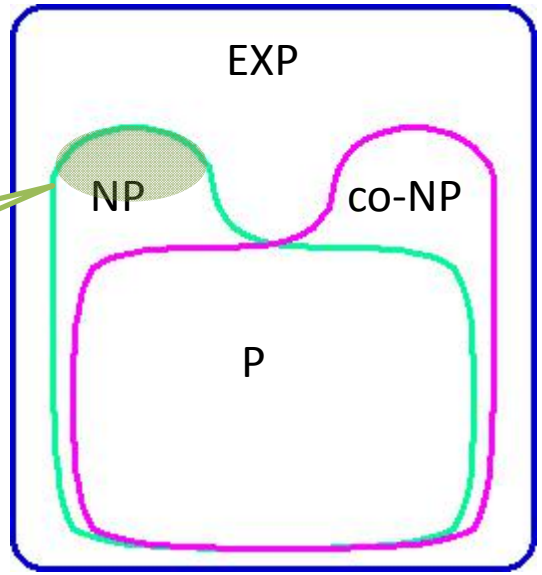
That is, NP-complete sets are NP-sets that cannot be recognized in polynomial time unless P = NP.

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

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≠P ➔ $A \notin$ P

つまり，NP完全集合はP=NPでない限り，

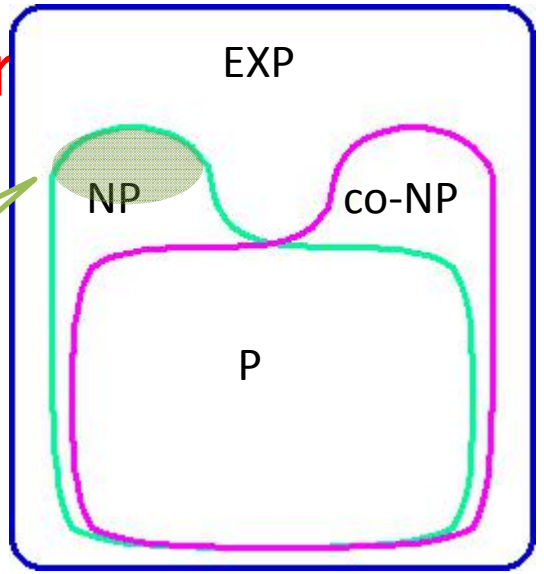
多項式時間では認識できないNP集合である.

6. Analysis on Polynomial-Time Computability

6.2. Completeness

6.2.1. Definition and basic proper

NP-complete problems form the most difficult problems in the class NP.



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　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) $A \leq_m^P B$ →B は C困難.

(2) $A \leq_m^P B$ かつ $B \in$ C → B は C完全.

証明:

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

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

よって, $\forall L \in \mathrm{C}[L \leq_m^P B]$

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.

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 $B \in \mathrm{C} \rightarrow B$ is C-complete.

Proof :

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

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

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

That is, *B* is C-hard.

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

残りの予定 (Schedule)

- 5/26(Thu):多項式時間還元性
- 6/2(Thu): 多項式時間還元性に基づく完全性
  - アンケート
  - オフィスアワーに試験. 以下参照.
- 6/2(Thu): 試験(Exam)
  - 40点満点
  - 選択肢(Choices); 5月26日に多数決で決めましょう.
    - 電子デバイス以外何でも (Anything without electricity (w/o cell/ipad/…))
    - 教科書/スライド/ノート (Textbooks, copy of slides, and hand written notes)
    - スライドのコピー/手書きノート/筆記用具のみ(Copy of slides, hand-written note, and pens/pencils)
    - 手書きノートと筆記用具のみ(Hand-written note and pens/pencils)
    - 筆記用具のみ(Only pens and pencils)