1/12

5.2. クラス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.

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

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

このとき, LをNP集合といい, Lの認識問題をNP問題という.
また, NP集合の全体をクラス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, $\exists w \in \Sigma^* : |w| \leq q(|x|) \quad \Rightarrow \quad \exists_q w$ 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. 」

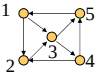
補足: NP=Nondeterministic Polynomial

2/12

例5.7: ハミルトン閉路問題 (DHAM) $\in$ NP

グラフの頂点は1〜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➔ 1〜n の順列$< l_1, l_2, \ldots, l_n >$
この順列が多項式長の証拠

例:



証拠の候補 (注)全部で $n!$〜n^n 通りある
<1,2,3,4,5> ➔ハミルトン閉路 ➔証拠
<1,2,3,5,4> ➔ハミルトン閉路でない
<1,4,3,2,5> ➔ハミルトン閉路でない

$R_D(x, w)$ ↔[xはあるグラフG(n頂点)のコード]
∧ [wは1〜nの順列$< l_1, l_2, \ldots, l_n >$]
∧[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:
$x \in \text{DHAM} \leftrightarrow \exists w_G (=< l_1, \ldots, l_n >)[R_D(x, w_G)]$
xがグラフのコードになっていないとき: $\forall w[\neg R_D(x,w)]$

3/12

例5.8: 命題論理式充足性問題(3SAT, SAT, ExSATなど)
目標: ExSAT $\in$ NP

$F(x_1, \ldots, x_n)$: 任意の拡張命題論理式
Fが充足可能 ↔ $\exists a_1, \ldots, a_n$: 各a_iは1か0 $[F(a_1, \ldots, a_n) = 1]$
証拠の長さq_E
Fへの真偽値の割り当てを$< a_1, \ldots, a_n >$で表す.
→ 長さは $3(n+n+1)=6n+3 \leq 6|\lceil F \rceil| + 3$
$q_E(l) = 6l+3$
述語R_E
$R_E(x, w)$ ↔ [xはある拡張命題論理式F (n変数)のコード]
∧ [wはFへの割り当て$< a_1, a_2, \ldots, a_n >$]
∧ $[F(a_1, \ldots, a_n) = 1]$

計算木を用いると$F(a_1, \ldots, a_n)$ 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.

4/12

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 ∈ Σ^{≤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, そのような文字列は $2^{q(|x|)}$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.

5/12

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NPに属しているとき, **co-NP集合**という. また, co-NP集合の全体を**クラスco-NP**という.

補注: co-P を定義しても 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) $L \in$co-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.

6/12

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,
Nは自然数の2進表記全体)
と定義すると,
すべての $x \in \Sigma^*$に対し, $x \notin \text{PRIME} \leftrightarrow \exists q_p w[R_p(x,w)]$
これは, $x \notin$ PRIMEに対する証拠
よって, $\overline{\text{PRIME}} \in$ NP, i.e., PRIME $\in$ co-NP
実際, $Q(x,w) \leftrightarrow \neg R_p(x,w)$ とすると
PRIME = $\{x : \forall q_p w [Q_p(x,w)]\}$
と表せる.

PRIME $\in$ NPも示せるが, その証明はもっと複雑.

7/12

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$< a_1, a_2, \dots, a_n, b >$
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$< a_1, a_2, \dots, a_n, b, k>$
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 $\sum_{i \in U_j} a_i \le b$ 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$<G, k>$
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

8/12

5.3. 計算量クラス間の関係

定理5.6: P ⊆ E ⊆ EXP.

定義より，明らか．

定理5.7: P ⊊ E ⊊ EXP.

証明:
(1) P ⊊ E.
$t_1(n)=2^n$, $t_2(n)=2^{3n}$とすると，階層定理より，
TIME(2^n) ⊊ TIME(2^{3n})
一方，P ⊆ TIME(2^n) ⊊ TIME(2^{3n}) ⊆ E だから，
P ⊊ E.
(2)も同様．

証明終

9/12

定理5.8.
(1) P ⊆ NP，P ⊆ co−NP (よって，P ⊆ NP ∩ co−NP)
(2) NP ⊆ EXP，co−NP ⊆ EXP (よって，NP ∪ co−NP ⊆ EXP)

証明: (1) P ⊆ NP（P ⊆ co−NP も同様）
L: 任意の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．
$\forall x \in \Sigma^*: [x \in L \leftrightarrow P(x)]$ or $P = \{x: P(x)\}$
$R(x, w) = P(x)$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，NPの定義より，$L \in$ NP i.e., P ⊆ NP.

10/12

(2) NP ⊆ EXP (co-NP ⊆ EXP)
L: 任意の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，
$L = \{x : \exists_q w[R(x,w)]\} = \{x : \exists_q w[\,|w| \le q(|x|) \wedge R(x,w)]\}$
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 ∈ Σ^{≤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$=p(|x| + |w|) \le p(l + q(l))$ ←— 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，$L \in$EXP → NP ⊆ EXP

証明終

11/12

定理5.9.
(1) NP ⊆ co-NP → NP = co-NP
(2) co-NP ⊆ NP → NP = co-NP
(3) NP ≠ co-NP → P ≠ NP.

補注: (3)より，NP ≠ co-NPの証明は，P≠NPの証明より難しい．

証明： (1) NP ⊆ co-NP → NP = co-NP ((2)の証明も同様)
任意の$L \in$co-NPに対して$L \in$ NPが示せれば，co-NP ⊆ NP
が証明できるので，仮定のNP ⊆ co-NPと合わせてNP = co-NP
が言える．
$L \in$ co-NP →$\overline{L} \in$NP (定義5.3より)
→$\overline{L} \in$ co-NP (NP ⊆ co-NPより)
→$L \in$NP (定義5.3と$\overline{\overline{L}}=L$より)

12/12

(3) NP ≠ co-NP → P ≠ NP.

対偶: P = NP → NP = co-NP

P＝NPと仮定すると，すべてのLに対し
$L \in$NP ↔ $L \in$P (P = NP より)
↔ $\overline{L} \in$P (演習問題5.5)
↔ $\overline{L} \in$ NP (P = NP より)
↔ L ($= \overline{\overline{L}}$) ∈ co- NP (定義5.3より)
∴ NP = co- NP

証明終

NP ≠co-NPが正しいと