I482F　実践的アルゴリズム特論
12回目：$\mathcal{NP}$完全性と多項式時間還元

上原隆平
(uehara@jaist.ac.jp)

計算量のクラス

- 集合・問題・言語：
  - アルファベットΣ（典型的には$\Sigma=\{0,1\}$）に対して、全体集合
    $\Sigma^*=\{\varepsilon, 0, 1, 00, 01, 10, 11, 100, 101, 110, 111, 1000, \ldots\}$
    の部分集合のことを<u>言語</u>という。
  - 言語Lに対して、
    与えられた任意のxに対してLに属するかどうか
    を決める問題をその言語の<u>**認識問題**</u>という。

例12.1：

$L_1=\{0, 10, 100, 110, 1000, 1010, 1100, 1110, \ldots\}$

（L_1は偶数の自然数の2進数表記）

$L_2=\{10, 11, 101, 111, 1011, 1101, 10001, \ldots\}$

（L_2は素数の2進数表記）

プログラミング言語モデル

- ここでは
  - プログラミング言語は普通の手続き型言語(たとえばC)
    - 変数
    - 代入文
    - 条件判断(if 文)
    - 制御命令(goto 文)
  - データやプログラムは妥当なコード化がされている
    - アルファベットは$\Sigma=\{0,1\}$
    - 数値データは2進数表現
    - 文字列はASCIIコード

プログラムの実行時間

- あるプログラム P の実行時間は、入力の長さ n に対する関数 $f(n)$ で測る。ただし
  - $f(n)$ は
    - 入力の長さが**高々 n** の**任意の入力** x に対する
    - $P(x)$ の計算時間の**上界**

  を与える関数である。

任意の入力:その長さまでの最悪の入力を考える。

上界:過大評価している場合もある

高々n:nの増加に対して非減少な単調関数としてよい。

計算量のクラス

- 代表的な(＝この授業で出てくる)クラス

定義12.1: クラス$\mathcal{P}$
言語 L がクラス$\mathcal{P}$に入る ⇔
以下を満たすプログラム P と多項式 f が存在:
任意の $x \in \Sigma^*$ に対して $P(x)$ は L の認識問題を $f(|x|)$ 時間で解く

定義12.2: クラス$\mathcal{EXP}$
言語 L がクラス$\mathcal{EXP}$に入る ⇔
以下を満たすプログラム P と指数関数 f が存在:
任意の $x \in \Sigma^*$ に対して $P(x)$ は L の認識問題を $f(|x|)$ 時間で解く

クラス$\mathcal{EXP}$の中の最上位の問題は「手に負えない」

定義12.3: クラス$\mathcal{NP}$
集合 L がクラス$\mathcal{NP}$に入る ⇔
以下を満たすプログラム P と多項式 f と多項式 q が存在:
任意の $x \in \Sigma^*$ に対して
- $\underline{x} \in L$ ならば $\exists w \in \Sigma^*$ $(|w| \leqq q(|x|))$ に対して $P(x,w)$ は L の認識問題を $f(|x|)$ 時間で解く
- $x \notin L$ ならばそのような w は存在しない

計算量のクラス

- 代表的な(＝この授業で出てくる)クラス

定義12.3:クラス$\mathcal{NP}$
集合 L がクラス$\mathcal{NP}$に入る ⇔
以下を満たすプログラム P と多項式 f と多項式 q が存在:
任意の $x \in \Sigma^*$ に対して
- $x \in L$ ならば $\exists w \in \Sigma^*$ $(|w| \leqq q(|x|))$ に対して
  $P(x,w)$ は L の認識問題を $f(|x|)$ 時間で解く
- $x \notin L$ ならばそのような w は存在しない

補注:各 $x \in \Sigma^*$ に対して,上記を満たす $w_x \in \Sigma^*$ 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.

クラス$\mathcal{NP}$とは「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」
という性質をもつクラス

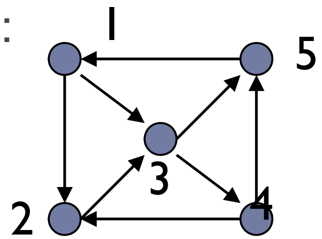
補足:$\mathcal{NP}$=**N**ondeterministic **P**olynomial

$\mathcal{NP}$問題の例

スターリングの近似式 : $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

- ハミルトン閉路問題（DHAM）
  - 入力 : 有向グラフD（頂点は1〜nで番号付けされているとする）
  - 出力 : すべての頂点をちょうど一度づつ訪れる閉路はあるか？

例12.2 :



頂点を訪れる順序は1〜nの順列
<1,2,3,4,5>: ハミルトン閉路
<1,2,3,5,4>: ハミルトン閉路ではない
<1,4,3,2,5>: ハミルトン閉路ではない

全部で約 $n! \sim 2^{n \log n}$ 通りなので、素朴に全部試すと指数時間かかる。

プログラム P :
[入力] グラフ D と 1〜n の順列 p
[出力] p の順に D の頂点を訪れて
　　ハミルトン閉路になっていればYes、そうでなければNo

- プログラム P は $|D|, |p|$ に対する多項式時間アルゴリズム
- $D \in$DHAMならば、$\exists p$が「Dの証拠」の条件を満たす
- $D \notin$DHAMならば、どんなpも「Dの証拠」の条件を満たすことはできない

$\mathcal{NP}$集合であることの意味とは..

「計算機で解きたい問題」
としては自然な状況

- (とても)直観的な意味：与えられた問題が…
  - 解答を教えてもらえると、自分で多項式時間で確かめられる
  - 自分で解答を見つけるのは指数時間かかりそうに見える
  (本当に指数時間かかるかどうかは100万ドルの賞金問題)

ミレニアム問題
$\mathcal{P}\neq\mathcal{NP}$予想

- もう少し形式的には

多項式時間プログラムPを用いて，$x \in L$? を次のように判定できる.

```
for each  w ∈ Σ^{≤q(|x|)}  do
   if P(x, w)="yes" 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，YesかNoかを判定できる.
ただ，そのような文字列は約 $2^{q(|x|)}$ 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$\mathcal{NP}$集合と考えてよい.
ここから $\mathcal{P}\subseteq\mathcal{NP}\subseteq\mathcal{EXP}$ という包含関係もわかる。

$\mathcal{NP}$問題の例

・**命題論理式充足性問題**(SAT)
入力：n変数の命題論理式F
質問：Fを真にするTrue/Falseの割り当てがあるか？

・**ナップサック問題**(KNAP)
入力：自然数の組$< a_1, a_2, \ldots, a_n, b >$
質問：$\sum_{i \in S} a_i = b$ となる添字の集合 $S \subseteq \{1, \ldots, n\}$ があるか？

・**箱詰め問題**(BIN)
入力：自然数の組$< a_1, a_2, \ldots, a_n, b, k >$
質問：添字の集合$U=\{1, \ldots, n\}$を$U_1, \ldots, U_k$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$< G, k >$
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

多項式時間還元可能性

B を解くプログラムがあれば
それで A を解くことができる

定義12.4:

AとBを任意の集合とする.

(1) 関数 h: A→B: 多項式時間還元(polynomial-time reduction)

$\Longleftrightarrow \begin{cases} \text{(a) } h \text{ は}\Sigma^*\text{から}\Sigma^*\text{への全域的関数} \\ \text{(b) } \forall x \in \Sigma^*[x \in A \leftrightarrow h(x) \in B] \\ \text{(c) } h \text{ は多項式時間計算可能.} \end{cases}$

(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という(polynomial time reducible).
このとき, 次のように書く: $A \leq_m^P B$

直観的な意味:

$A \leq_m^P B$ とすると多項式時間の範囲内では,「Aの難しさ」≦「Bの難しさ」

多項式時間還元の基本性質

B を解くプログラムがあれば
それで A を解くことができる

定理:12.1 $A \le_m^P B$ のとき,

(1) $B \in \mathcal{P} \rightarrow A \in \mathcal{P}$.
(2) $B \in \mathcal{NP} \rightarrow A \in \mathcal{NP}$.
(3) $B \in \mathcal{EXP} \rightarrow A \in \mathcal{EXP}$.

例12.3:

ONE＝$\{1\}$と定義するとき，クラス$\mathcal{P}$のすべての集合Lについて $L \le_m^P \mathrm{ONE}$ が成り立つ.

(∵)

$$h(x) \equiv \begin{cases} 1, & x \in L\text{のとき} \\ 0, & \text{その他のとき} \end{cases}$$

と定義すると,

(a) hはΣ^*からΣ^*への全域的関数.
(b) $x \in \Sigma^*[x \in L \leftrightarrow h(x) \in \mathrm{ONE}]$
(c) hは多項式時間計算可能（$L \in \mathcal{P} \rightarrow x \in L$の判定も多項式時間内）

多項式時間還元における同値関係

定理12.2: A, B, C: 任意の集合

(1) $A \le_m^P A$

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

定義12.5: $A \equiv_m^P B \leftrightarrow A \le_m^P B \wedge B \le_m^P A$

$\equiv_m^P$ は同値関係

多項式時間還元性のもとで同値な問題群

命題論理式の充足可能性問題の間の関係

2SAT　（命題論理式充足性問題：二和積形式）
3SAT　（命題論理式充足性問題：三和積形式）
SAT　（命題論理式充足性問題）
ExSAT　（拡張命題論理式充足性問題）

$$2\mathrm{SAT} \le_m^P 3\mathrm{SAT}$$

- 高々k個…自明
- ちょうどk個…
  - 同じリテラルを使ってよいなら簡単。
  - だめなら…考えてみよう！

同様に,

$$3\mathrm{SAT} \le_m^P \mathrm{SAT} \le_m^P \mathrm{ExSAT}$$

$$2\mathrm{SAT} \le_m^P 3\mathrm{SAT} \le_m^P \mathrm{SAT} \le_m^P \mathrm{ExSAT}$$

よってここで

$$\mathrm{ExSAT} \le_m^P 3\mathrm{SAT}$$

が示せれば,

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

となる.

ExSATから3SATへの還元

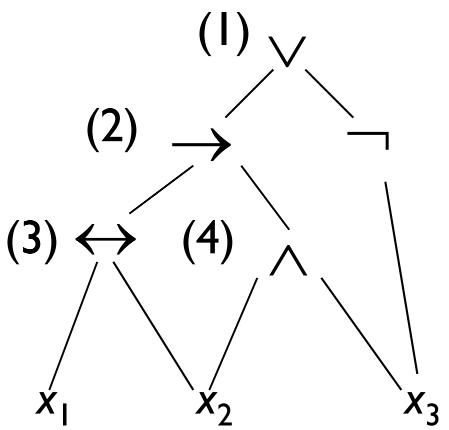
$$E_1(x_1, x_2, x_3) \equiv [[x_1 \leftrightarrow x_2] \rightarrow [x_2 \wedge x_3]] \vee \neg x_3$$

$$F_1(x_1, x_2, x_3) \equiv U_1 \wedge [U_1 \leftrightarrow [U_2 \vee \neg x_3]] \wedge [U_2 \leftrightarrow [U_3 \rightarrow U_4]]$$
$$\wedge [U_3 \leftrightarrow [x_1 \leftrightarrow x_2]] \wedge [U_4 \leftrightarrow [x_2 \wedge x_3]]$$

F_1の構成方法：E_1の計算木をボトムアップで計算する方法を模倣

このとき，[E_1が充足可能] $\leftrightarrow$[F_1が充足可能]

F_1は三和積形式に直しやすい形になっている．



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

F_1を構成するために，$V_i \rightarrow U_i$とし，V_iの定義式を$\wedge$で結ぶ

ExSATから3SATへの還元

F_1 の構成方法より，
(1)各U_i の値を$V_i(x_1, x_2, x_3)$としない限り，F_1は真にはならない．
(2)各U_iの値を$V_i(x_1, x_2, x_3)$としたとき，$F_1 = E_1$

上の性質が成り立つことは，帰納法を用いるなどして証明可能．
証明は省略．

三和積形式への変換

次の関係を使って展開する：

$$a \to b = \neg a \vee b$$
$$a \leftrightarrow b = (a \to b) \wedge (b \to a) = (\neg a \vee b) \wedge (\neg b \vee a)$$

例：

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

他も同様に展開して整理すると、三和積形式に変形できる．
よって，すべて三和積形式に変形できることがわかる．

多項式時間還元性に基づく困難性・完全性

困難性と完全性の定義と基本的な性質

定義12.5:
計算量クラス$\mathcal{C}$に対し，集合 A が次の条件を満たすとき，
A は($\leq_m^P$ の下で)$\mathcal{C}$-困難という.
(a) $\forall L \in \mathcal{C}\ [L \leq_m^P A]$

クラス$\mathcal{C}$のどの問題と比べても同程度には難しい

定義12.6:
計算量クラス$\mathcal{C}$に対し，$\mathcal{C}$-困難集合 A が $A \in \mathcal{C}$ を満たすとき，
A は($\leq_m^P$の下で)$\mathcal{C}$-完全という.

クラス$\mathcal{C}$の中でもっとも難しい

例12.4: クラス$\mathcal{NP}$の完全集合の例
3SAT, SAT, ExSAT, DHAM, KNAP, BIN,VCなど

多項式時間還元性に基づく困難性・完全性

- 困難性と完全性の定義と基本的な性質

定理12.3: 任意の$\mathcal{C}$-困難集合（含：$\mathcal{C}$-完全集合）Aに対し，

(1) $A \in \mathcal{P} \rightarrow \mathcal{C} \subseteq \mathcal{P}$　　対偶は $\mathcal{C} \not\subset \mathcal{P} \rightarrow A \notin \mathcal{P}$

(2) $A \in \mathcal{NP} \rightarrow \mathcal{C} \subseteq \mathcal{NP}$　　対偶は $\mathcal{C} \not\subset \mathcal{NP} \rightarrow A \notin \mathcal{NP}$

(3) $A \in \mathcal{EXP} \rightarrow \mathcal{C} \subseteq \mathcal{EXP}$　　対偶は $\mathcal{C} \not\subset \mathcal{EXP} \rightarrow A \notin \mathcal{EXP}$

証明：

(1) Bを任意の$\mathcal{C}$集合とすると，Aは$\mathcal{C}$-困難だから，$B \leq_m^P A$

一方，$A \in \mathcal{P}$の仮定より，$B \in \mathcal{P}$

(2), (3), (4)も同様

定理12.3の意味（クラス$\mathcal{NP}$）

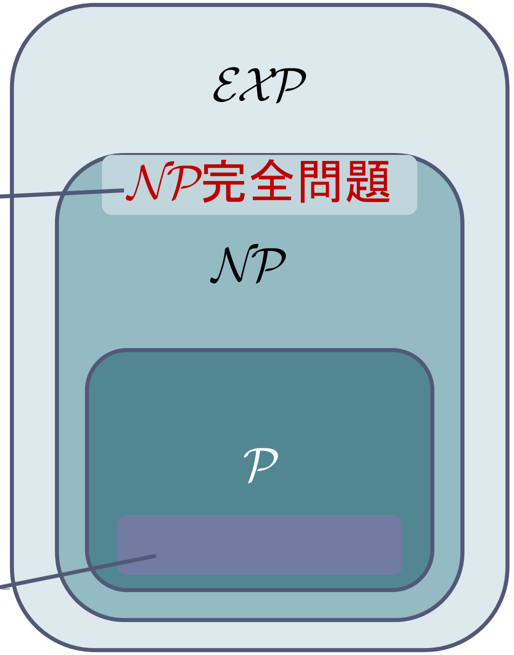
Aを$\mathcal{NP}$-完全集合とする．

定理12.3(1)の対偶より，

$\mathcal{NP} \neq \mathcal{P} \rightarrow A \notin \mathcal{P}$

よって$\mathcal{NP}$-完全集合は$\mathcal{P} \neq \mathcal{NP}$である限り，

多項式時間では認識できない．



問題のサイズが大きくなると手に負えない問題

現実的な時間で解ける問題

多項式時間還元性に基づく困難性・完全性

- 困難性と完全性の定義と基本的な性質

定理12.4: A: 任意の$\mathcal{C}$-完全集合
すべての集合Bに対し,
(1) $A \le_m^P B \rightarrow B$は$\mathcal{C}$-困難.
(2) $A \le_m^P B \wedge B \in \mathcal{C} \rightarrow B$は$\mathcal{C}$-完全.

完全集合が1つでも
見つかれば、そこから
芋づる式に困難性や
完全性を示すことができる。

証明:
定義12.5より, $\forall L \in \mathcal{C}[L \le_m^P A]$
定理12.2(2)より, $L \le_m^P A \wedge A \le_m^P B \rightarrow L \le_m^P B$
したがって, $\forall L \in \mathcal{C}[L \le_m^P B]$
すなわち, Bは$\mathcal{C}$-困難.

完全集合第1号
Cookの定理:3SATはNP完全集合
(証明: Turing Machine の計算プロセスを
すべて論理式で記述しなおす!!)

多項式時間還元性に基づく困難性・完全性

困難性と完全性の証明方法

($\mathcal{NP}$)完全性の証明方法
(I) 定義通りに[すべてのL]について示す
(II) すでに完全であることがわかっている問題を利用する

(I)の例: Cookの定理(SATでTMを模倣)

基本的には…
1. 多項式時間で動く標準プログラムを考えて
2. プログラムの動作を命題論理式で模倣する
→<u>とても大変</u>(手間がかかる)

3SATなどは、
形式が一様なので
扱いやすい

(II)の例: 世の中の$\mathcal{NP}$完全性の証明のほとんど

DHAMは一般のグラフ上で$\mathcal{NP}$完全

DHAMは平面グラフに限定しても$\mathcal{NP}$完全
DHAMは「頂点の次数＝3」に限定しても$\mathcal{NP}$完全
DHAMは2部グラフに限定しても$\mathcal{NP}$完全…

多項式時間還元性に基づく困難性・完全性

困難性と完全性の証明

定理12.5: 以下にあげる集合はすべて$\mathcal{NP}$-完全
(1) 3SAT, SAT (ExSATからの還元)
(2) DHAM, VC (3SATからの還元)
(3) KNAP, BIN (3SATからの還元とKNAP $\leq_m^P$ BIN)

(II) $\mathcal{NP}$完全性がわかっている問題からの多項式時間還元:

1. 3SAT $\leq_m^P$ VC
2. DHAM $\leq_m^P$ 頂点の次数が高々5に制限されたDHAM

Vertex Cover: すべての辺の、少なくとも一方の頂点を含む集合
Hamiltonian cycle: すべての頂点を一度ずつ通る閉路

おまけ: DHAMは次数高々3でも$\mathcal{NP}$完全。
高々2だと多項式時間で計算可能。

定理12.5(2) : VC は $\mathcal{NP}$ 完全問題

[証明] VC $\in \mathcal{NP}$ なので、3SAT $\leq_m^P$ VC であることを示せばよい。

論理式 $F(x_1,x_2,\ldots,x_n)$ が与えられたとする。
Fから以下の条件を満たすグラフと自然数の組$\langle G, k \rangle$が
多項式時間で構成できることを示す:

Fを1にする割当が存在する$\Leftrightarrow$$G$がサイズ$k$の頂点被覆を持つ

Gの構成(Fはn変数m項とする):

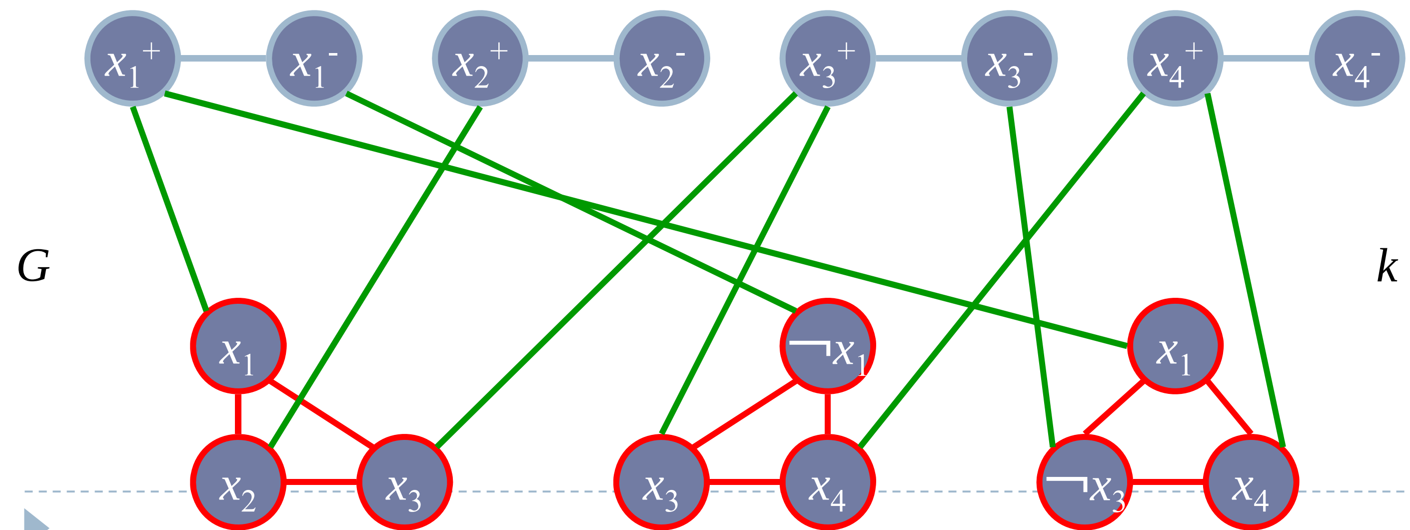
1. Fの各変数 x_i に対し、頂点 x_i^+,x_i^-と、辺(x_i^+,x_i^-)を加える
2. Fの各項$C_j=(l_{i1} \vee l_{i2} \vee l_{i3})$に対し、頂点 l_{i1}, l_{i2}, l_{i3} と辺$(l_{i1},l_{i2}), (l_{i2},l_{i3}), (l_{i3},l_{i1})$を加える
3. 項C_jのリテラル l_{i1} が x_i のときは辺(l_{i1},x_i^+) を、$\neg x_i$ のときは辺(l_{i1},x_i^-) を加える。
4. $k = n+2m$

Fを1にする割当が存在する⇔Gがサイズkの頂点被覆を持つ

Gの構成(Fはn変数m項とする):

1. Fの各変数 x_i に対し、頂点 x_i^+, x_i^-と、辺(x_i^+, x_i^-)を加える
2. Fの各項$C_j=(l_{i1} \vee l_{i2} \vee l_{i3})$に対し、頂点 l_{i1}, l_{i2}, l_{i3} と辺$(l_{i1}, l_{i2}), (l_{i2}, l_{i3}), (l_{i3}, l_{i1})$を加える
3. 項C_jのリテラル l_{i1} が x_i のときは辺(l_{i1}, x_i^+) を、$\neg x_i$ のときは辺(l_{i1}, x_i^-) を加える。
4. $k = n+2m$

例: $F(x_1,x_2,x_3,x_4) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$



$k = 4+2 \times 3=10$

Gの構成は、与えられた F から F のサイズに対する多項式時間で可能 。したがって以下を示せばよい:

Fを1にする割当が存在する⇔Gがサイズkの頂点被覆を持つ

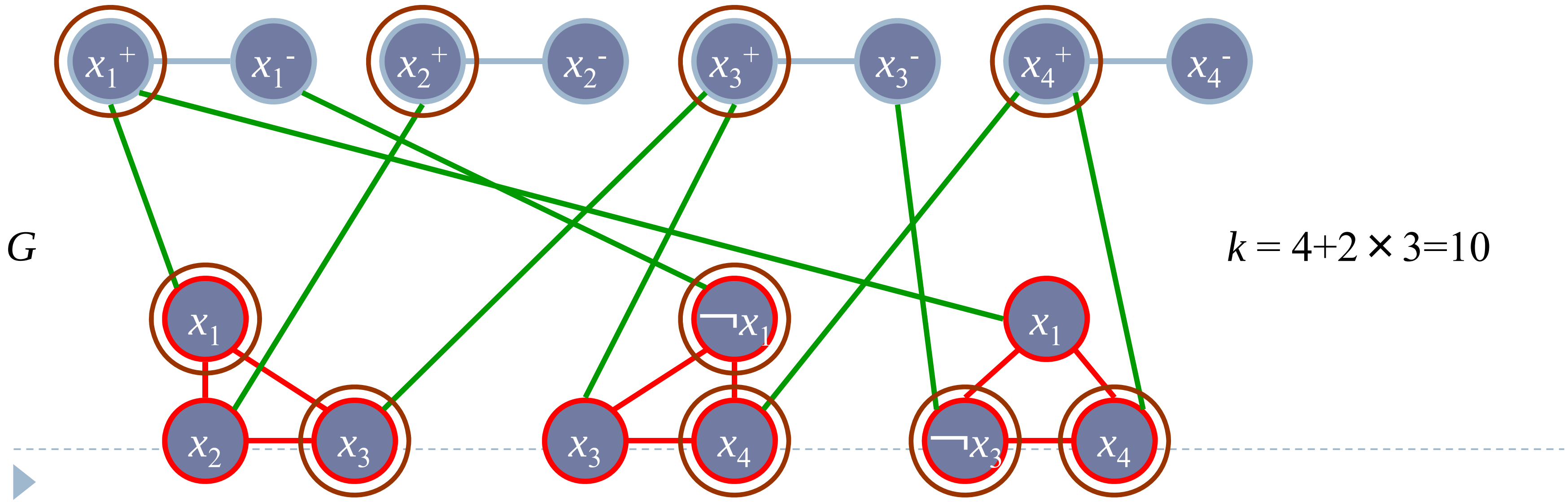
観察:

Gの構成から任意の頂点被覆Sは

- x_i^+, x_i^-のどちらかを含む
- C_jの3頂点中、最低2つ含む

よって $|S| \geqq n+2m = k$ である。

例: $F(x_1,x_2,x_3,x_4) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$



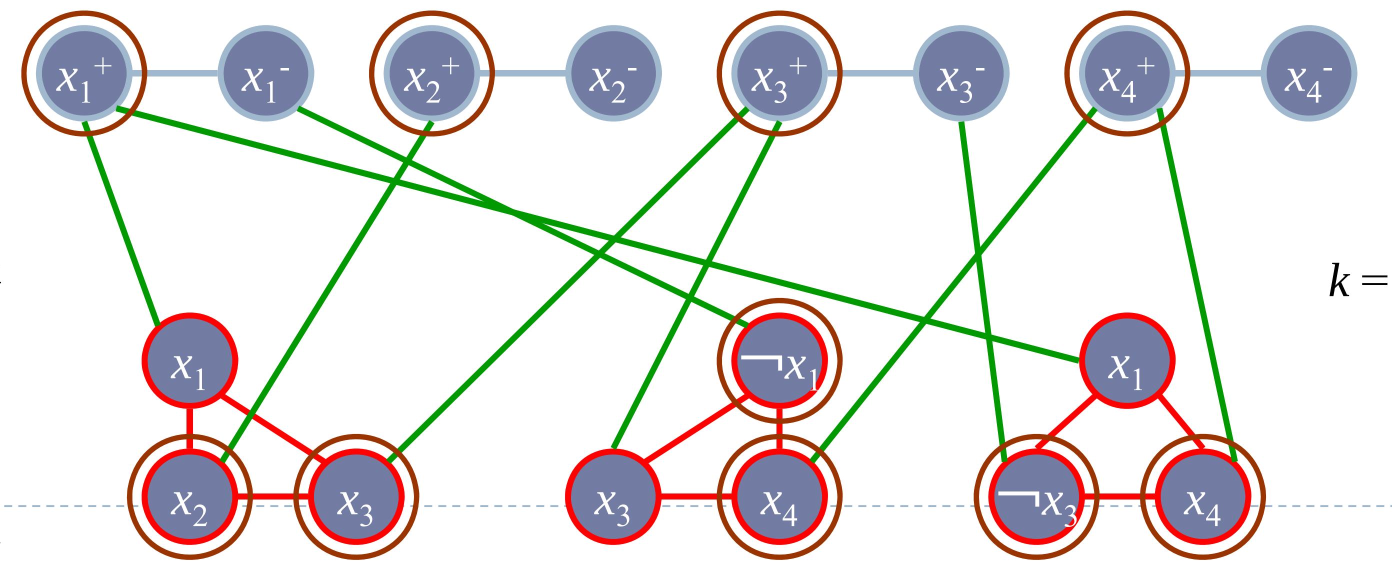
Fを1にする割当が存在する⇒Gがサイズkの頂点被覆を持つ

1. それぞれの変数 x_i が $\begin{cases} x_i=1\text{なら } x_i^+\text{を}S\text{に入れる} \\ x_i=0\text{なら } x_i^-\text{を}S\text{に入れる} \end{cases}$
2. それぞれの項$C_j=(l_{i1},l_{i2},l_{i3})$ は充足されているので、最低1つのリテラル(l_{i1})については変数との間の辺(l_{i1},x_{i1})は x_{i1} によって被覆されている。したがって、それ以外の二つのリテラル(l_{i2},l_{i3})を S に入れる。

⇒ **観察** より、Sはサイズkの頂点被覆になる。

例: $F(x_1,x_2,x_3,x_4) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$

G



$k = 4+2 \times 3=10$

Gがサイズkの頂点被覆を持つ⇒ Fを1にする割当が存在する

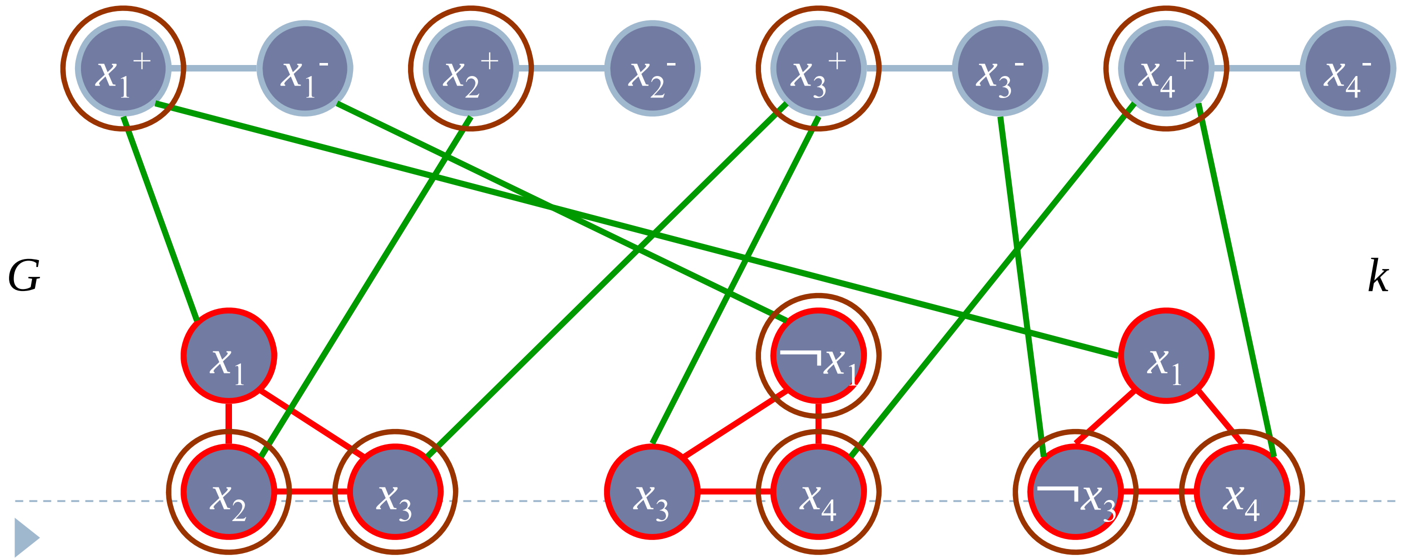
1. **観察** より、被覆Sは項から$2m$個、変数からn個の頂点を含む。

2. さらに各変数x_iについてはx_i^+かx_i^-の一方しか、
各項C_jについてはちょうど2つの頂点しかSに含むことができない。

3. よって各項C_jはSに含まれないリテラルl_iを含むが、
これに付随する辺は他方が被覆されていなければならない。

⇒ $\left(\begin{array}{l} x_i^+ \text{が} S \text{に含まれるなら } x_i=1 \\ x_i^- \text{が} S \text{に含まれるなら } x_i=0 \end{array} \right)$ という割当はFを充足する。

例: $F(x_1,x_2,x_3,x_4) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$

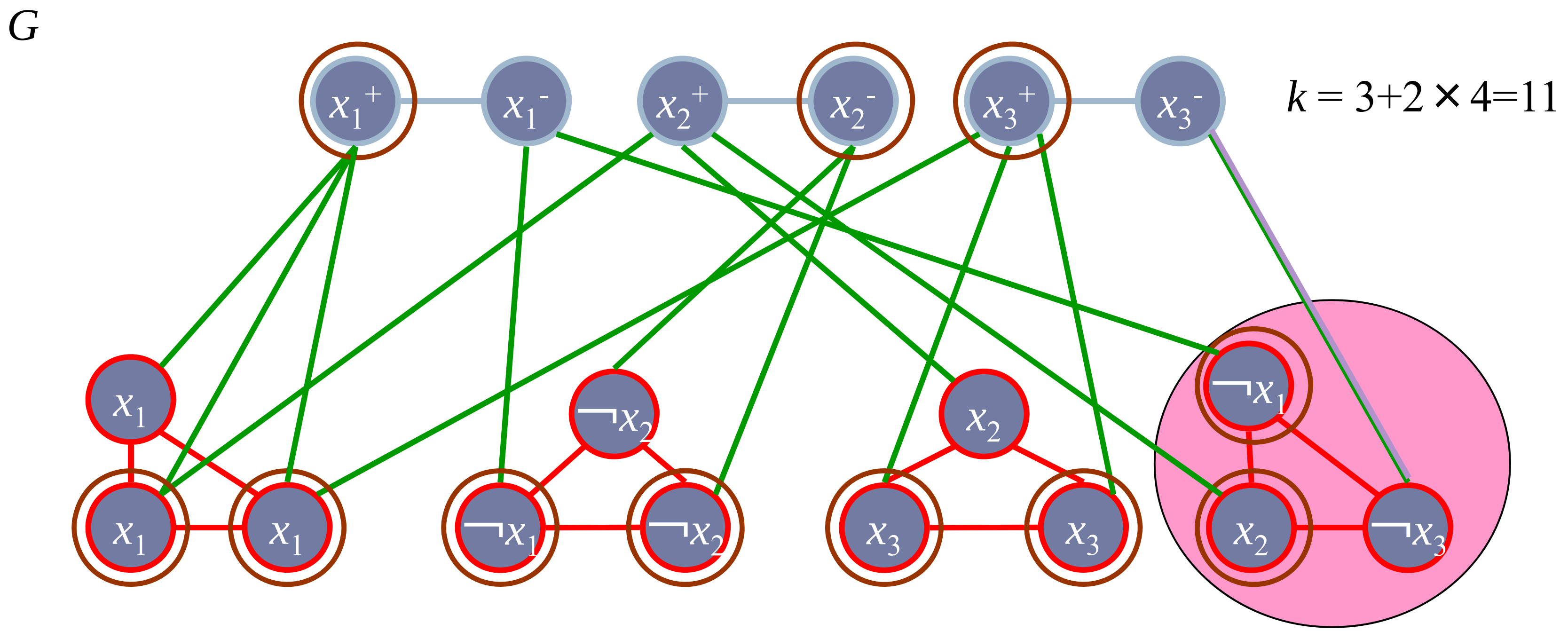


$k = 4+2\times3=10$

QED.

充足できない例:

$$F(x_1,x_2,x_3) = (x_1 \vee x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$



充足できないFでは、どのリテラルも頂点でカバーされていない項が必ず存在する。この項のリテラルは3つとも Vertex Cover に入れざるを得ない。よって Vertex Cover のサイズは k+1以上になる。

定理12.6: 次数高々5の有向グラフ上の DHAM は $\mathcal{NP}$ 完全問題

次数: 頂点に付随する辺の本数

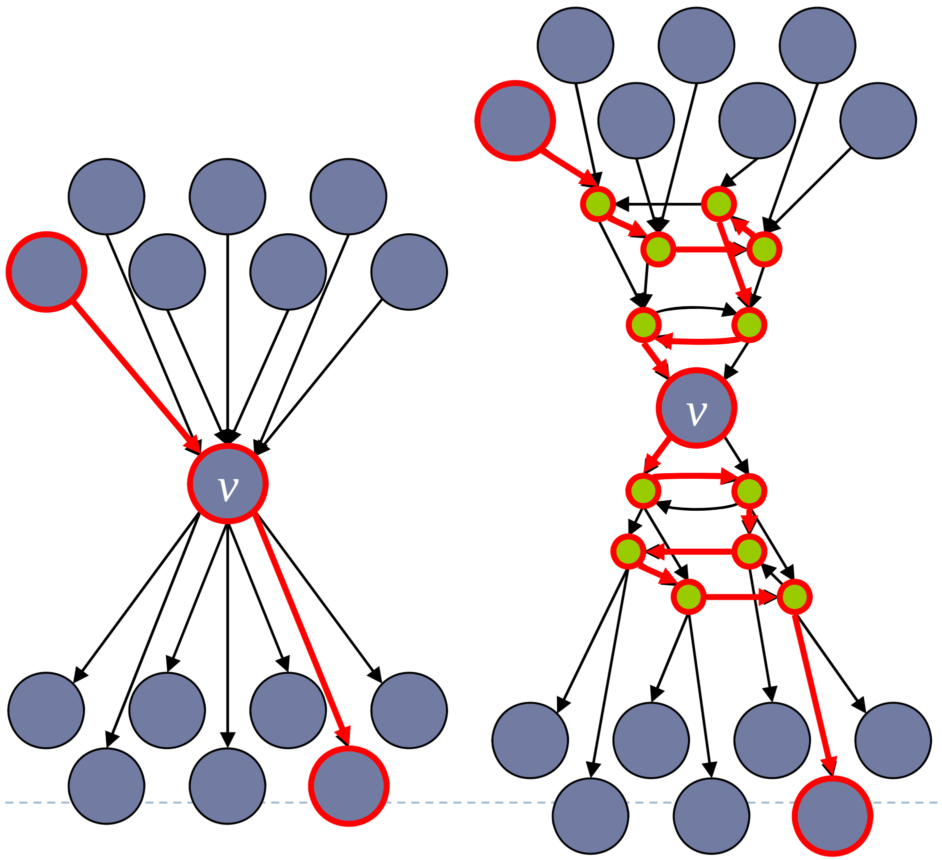
[証明] (上記の問題をDHAM$_{\leqq 5}$と略記する)

DHAM$_{\leqq 5}$ が$\mathcal{NP}$に属するのは、DHAMが$\mathcal{NP}$に属することから自明。したがって完全性を示せばよい。
DHAM $\leq_m^P$ DHAM$_{\leqq 5}$を示す。

アイデア:

次数14の頂点v(左)の(入ってくる辺集合)と(出ていく辺集合)を右図の`gadget’で置き換える

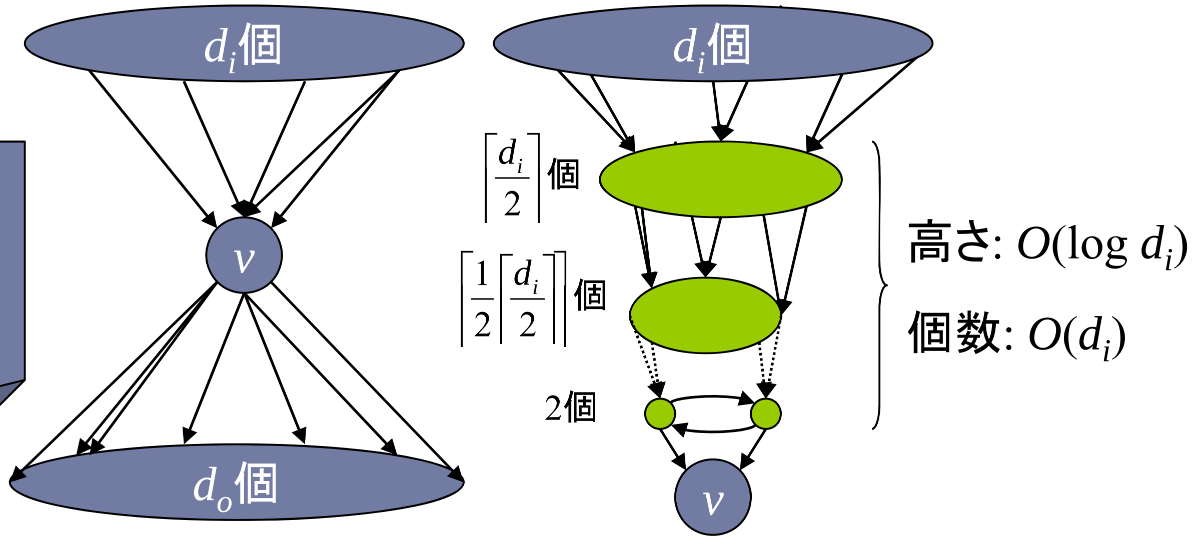
左図でvを1度だけ通る閉路と右図でvを1度だけ通る閉路は対応する。



定理12.6: 次数高々5の有向グラフ上の DHAM は $\mathcal{NP}$ 完全問題

アイデア:

ポイント:
- 各閉路は上から下
- 各頂点は次数≦5



[証明(概要)]

与えられたグラフGの次数が6以上のそれぞれの頂点に入る辺と出る辺を上記の gadget で置き換える。

1. 元のグラフGがn頂点m辺であったなら、gadget で置き換えたあとのグラフG'は$O(n+m)$頂点 $O(m)$辺となる。したがって上記の還元はGの大きさの多項式時間で可能。
2. またG'のすべての頂点は次数はたかだか5である。
3. Gがハミルトン閉路をもつ⇔G'がハミルトン閉路を持つ

QED.

おまけ(Addition)

> 多くの自然な問題は
> - 多項式時間で解けるか
> - *NP*困難か
>
> のどちらかである場合が多い(?)

- Ryuhei Uehara, Shigeki Iwata:
  Generalized Hi-Q is NP-complete,
  The Transactions of the IEICE, E73, p.270-273, 1990.
- Peisen Zhang, Huitao Sheng, Ryuhei Uehara:
  A Double Classification Tree Search Algorithm for Index SNP Selection, *BMC Bioinformatics*, 5:89, 2004.
- Sachio Teramoto, Erik D. Demaine, Ryuhei Uehara:
  Voronoi Game on Graphs and Its Complexity,
  2nd IEEE Symp. on Computational Intelligence and Games, p.265-271, 2006.
- Ryuhei Uehara, Sachio Teramoto:
  Computational Complexity of a Pop-up Book,
  4th International Conference on Origami in Science, Mathematics, and Education, 2006.
- Ryuhei Uehara:
  Simple Geometrical Intersection Graphs,
  3rd Workshop on Algorithms and Computation,
  Lecture Notes in Computer Science, Vol. 4921, p.25-33, 2008.

参考文献

- $\mathcal{NP}$完全性にまつわる話は「計算量の理論」と呼ばれる分野
- JAISTの講義ではI216「計算の理論と離散数学」の中の1/2くらいで扱っている
  - 「計算理論の基礎」シプサ著、太田・田中・阿部・植田・藤岡・渡辺訳、共立出版
  - 「オートマトン・言語理論・計算論」ホップクロフト・ウルマン・モトワニ著、野崎・町田・高橋・山崎訳、サイエンス社
  - 「計算可能性・計算の複雑さ入門」渡辺治著、近代科学社