

I216 Computational Complexity  
and  
Discrete Mathematics

by

Prof. Ryuhei Uehara

and

Prof. Atsuko Miyaji

# I216 計算量の理論と離散数学

上原隆平、宮地充子

# 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

# 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

# 5. Computational Complexity

- Observation of the classes

Definition: Class P

Set  $L$  is in the class P  $\Leftrightarrow$

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  $\Leftrightarrow$

There exists a poly  $q$  and a poly-time computable predicate  $R$  such that

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

Definition: Class coNP

Set  $L$  is in the class coNP  $\Leftrightarrow$

There exists a poly  $q$  and a poly-time computable predicate  $R$  such that

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

# 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  $\Leftrightarrow$

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  $\Leftrightarrow$

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

クラスcoNPの定義

集合 $L$ がクラスcoNPに入る  $\Leftrightarrow$

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  $x \in L \Leftrightarrow \forall w \in \Sigma^* : |w| \leq q(|x|) [R(x,w)]$

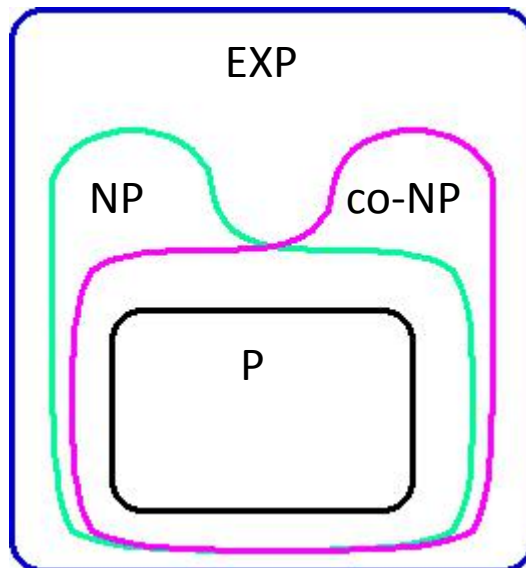
# 5. Computational Complexity

## 5.5. Relations in the Complexity Classes

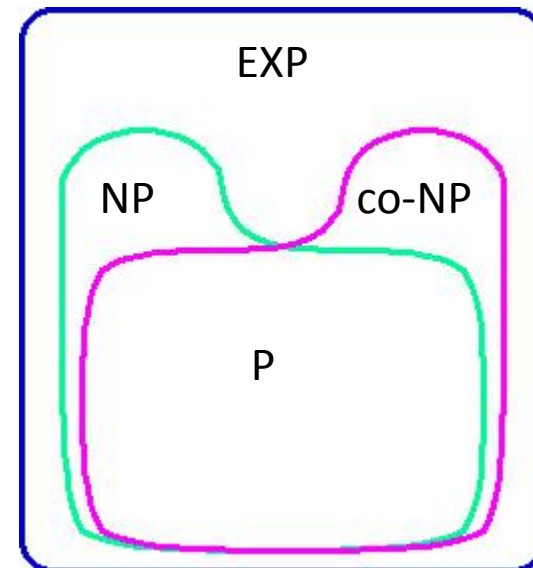
### Theorem

(1)  $P \subseteq NP$ ,  $P \subseteq coNP$  ( $\therefore P \subseteq NP \cap coNP$ )

(2)  $NP \subseteq EXP$ ,  $coNP \subseteq EXP$  ( $\therefore NP \cup coNP \subseteq EXP$ )



or



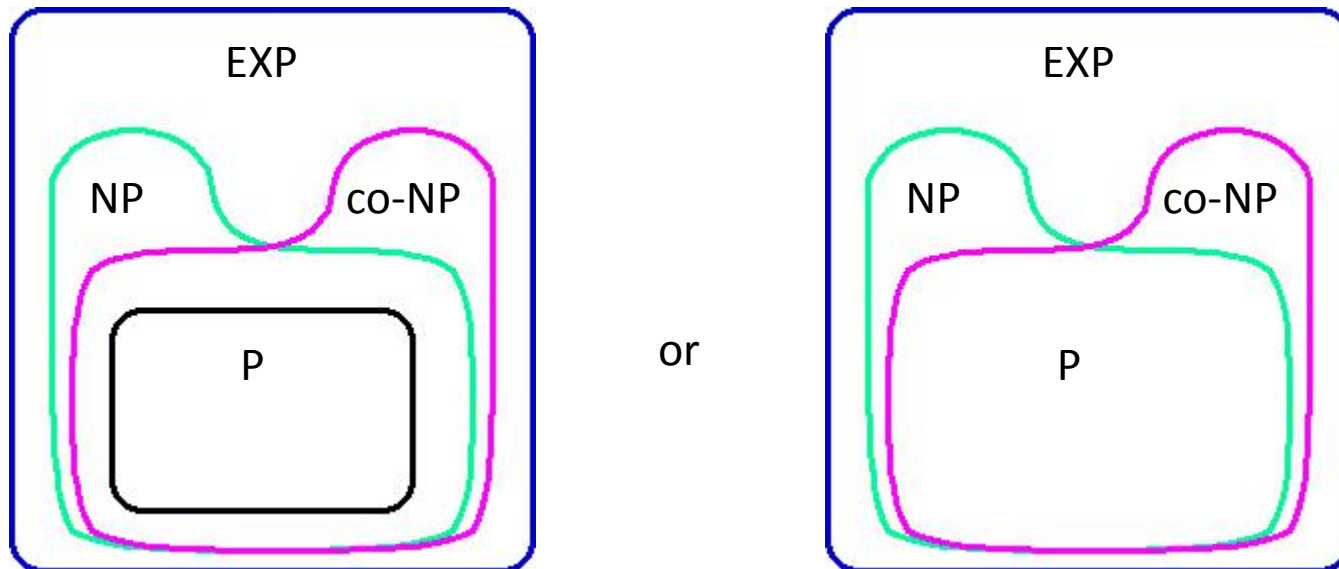
# 5. 計算量の理論

## 5.5. 計算量クラスの関係

定理

(1)  $P \subseteq NP$ ,  $P \subseteq \text{coNP}$  ( $\therefore P \subseteq NP \cap \text{coNP}$ )

(2)  $NP \subseteq \text{EXP}$ ,  $\text{coNP} \subseteq \text{EXP}$  ( $\therefore NP \cup \text{coNP} \subseteq \text{EXP}$ )





# 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  $NP \neq coNP$  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  $L \in coNP$ .

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

# 5. 計算量の理論

## 5.5. 計算量クラスの関係

### 定理

- (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 $NP \neq co-NP$  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.

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

# 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  $NP \neq coNP$  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

$$\begin{aligned} L \in NP &\Leftrightarrow L \in P \quad (P = NP) \\ &\Leftrightarrow \bar{L} \in P \quad (P = coP) \\ &\Leftrightarrow \bar{L} \in NP \quad (P = NP) \\ &\Leftrightarrow L (= \bar{\bar{L}}) \in coNP \quad (\text{Definitions of NP/coNP}) \end{aligned}$$

$\therefore NP = coNP$

Q.E.D.

# 5. 計算量の理論

## 5.5. 計算量クラスの関係

### 定理

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

$$\begin{aligned} L \in NP &\Leftrightarrow L \in P \quad (P = NP) \\ &\Leftrightarrow \bar{L} \in P \quad (P = coP) \\ &\Leftrightarrow L \in NP \quad (P = NP) \\ &\Leftrightarrow L (= \bar{\bar{L}}) \in coNP \quad (NP/coNPの定義より) \end{aligned}$$

$\therefore NP = coNP$

Q.E.D.

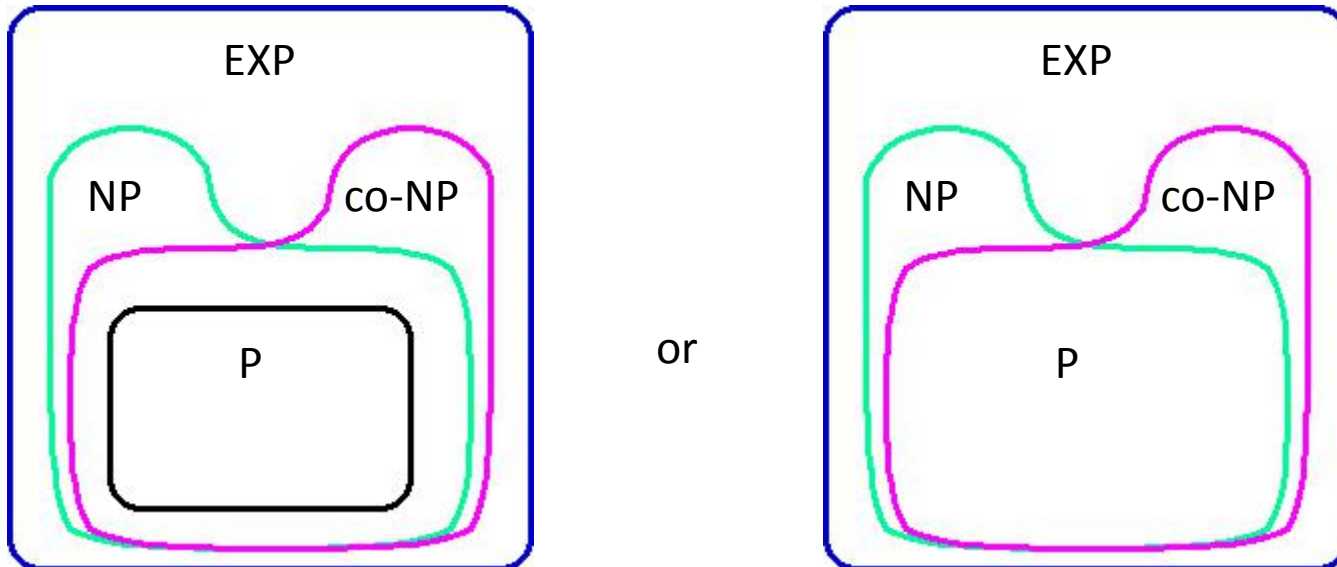
# 5. Computational Complexity

## 5.5. Relations in the Complexity Classes

### Theorem

- (1)  $NP \subseteq coNP \rightarrow NP = coNP$
- (2)  $coNP \subseteq NP \rightarrow NP = coNP$
- (3)  $NP \neq coNP \rightarrow P \neq NP$

We strongly believe that  $P \neq NP$ , and then we have



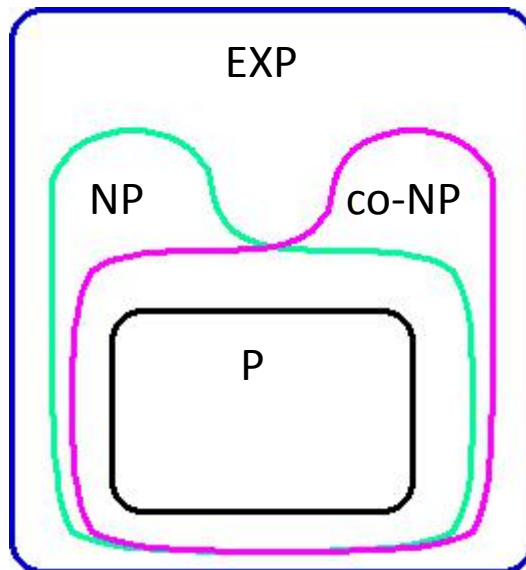
# 5. 計算量の理論

## 5.5. 計算量クラスの関係

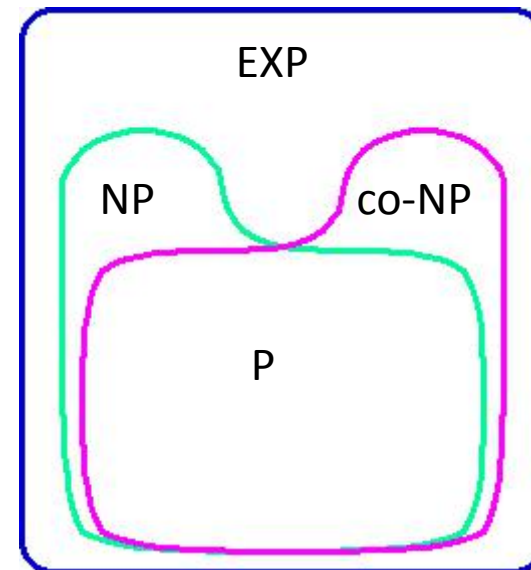
定理

- (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. 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

- $\Leftrightarrow$   $\left\{ \begin{array}{l} \text{(a) } h \text{ is a total function from } \Sigma^* \text{ onto } \Sigma^* \\ \text{(b) } x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B] \\ \text{(c) } h \text{ is polynomial-time computable.} \end{array} \right.$

(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  $A \leq$  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

#### 定義

$A$  と  $B$  を任意の集合とする.

(1) 関数  $h: A \rightarrow B$  が 多項式時間還元 である

$\Leftrightarrow$   $\left\{ \begin{array}{l} \text{(a) } h \text{ は } \Sigma^* \text{ から } \Sigma^* \text{ への全域関数である} \\ \text{(b) } x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B] \\ \text{(c) } h \text{ は多項式時間計算可能である.} \end{array} \right.$

(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  $\leq$   $B$  の難しさ)



## 6. Analysis on Polynomial-Time Computability

### 6.1. Polynomial-time Reducibility

**Theorem**  $A \leq_m^P B$  leads to

- (1)  $B \in P \rightarrow A \in P.$
- (2)  $B \in NP \rightarrow A \in NP.$
- (3)  $B \in \text{co-NP} \rightarrow A \in \text{coNP}.$
- (4)  $B \in \text{EXP} \rightarrow A \in \text{EXP}.$

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

**Ex.:** When we define  $\text{ONE} \equiv \{1\}$ , for each set  $L$  in P we have

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

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  $A \leq_m^P B$  のとき次が成立する

- (1)  $B \in P \rightarrow A \in P.$
- (2)  $B \in NP \rightarrow A \in NP.$
- (3)  $B \in \text{co-NP} \rightarrow A \in \text{coNP}.$
- (4)  $B \in \text{EXP} \rightarrow A \in \text{EXP}.$

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

例:  $\text{ONE} \equiv \{1\}$  と定義すると,  $P$  の各集合  $L$  に対して,

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

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**  $A, B, C$ : 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**  $A, B, C$ 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

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

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

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

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

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

#### 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, 特を示すことはない.

(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**

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

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

Proof (Outline)

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

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

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

証明 (概略)

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

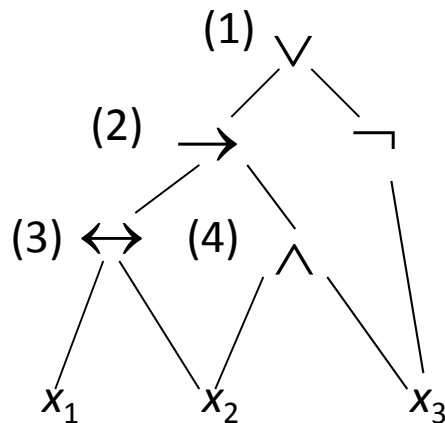
**Theorem (2)**  $3\text{SAT} \equiv_m^P \text{SAT} \equiv_m^P \text{ExSAT}$

Proof (Outline)

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

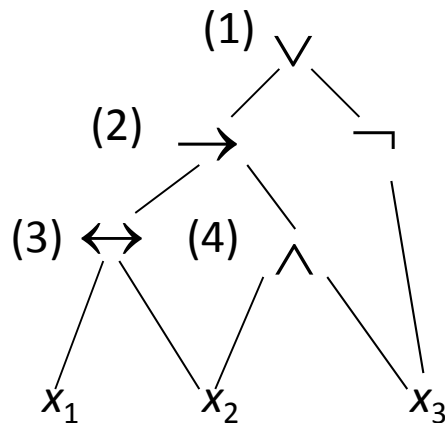
### 6.1. 多項式時間還元可能性

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

証明 (概略)

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

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



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

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

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

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

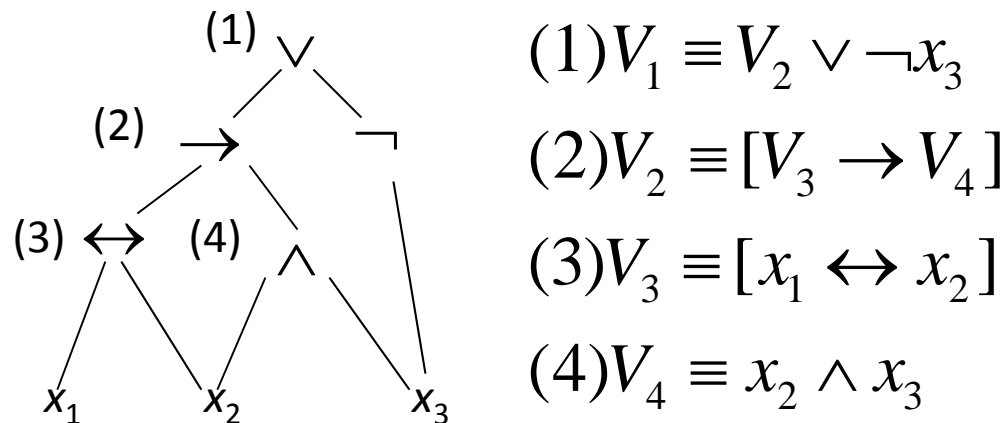
## 6. Analysis on Polynomial-Time Computability

### 6.1. Polynomial-time Reducibility

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

Reduction from ExSAT to 3SAT by an example:

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



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

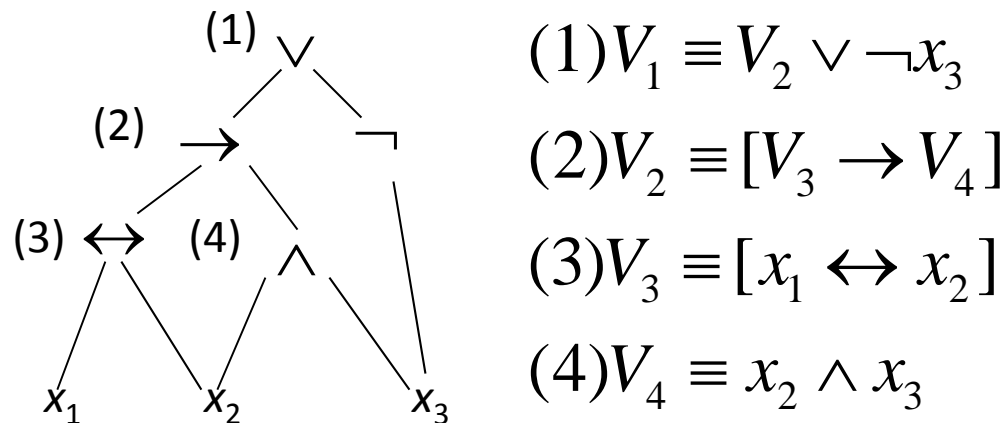
## 6. 多項式時間計算可能性の解析手法

### 6.1. 多項式時間還元可能性

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

ExSAT から 3SAT への還元を例で示す:

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



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

## 6. Analysis on Polynomial-Time Computability

### 6.1. Polynomial-time Reducibility

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

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 construction,  $F()$  is satisfiable iff  $F''()$  is satisfiable.

We show  $F''()$  can be represented by an equivalent  $F'()$  in 3SAT.

$$U_1 \leftrightarrow [U_2 \vee \neg x_3] = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg [U_2 \vee \neg x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_3] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_2 \vee x_2]$$

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

### 6.1. 多項式時間還元可能性

**定理 (2)**  $3SAT \equiv_m^P SAT \equiv_m^P ExSAT$

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

$$U_1 \leftrightarrow [U_2 \vee \neg x_3] = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg [U_2 \vee \neg x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee [\neg U_2 \wedge x_3]] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2] \wedge [U_1 \vee x_3] \\ = [\neg U_1 \vee U_2 \vee \neg x_3] \wedge [U_1 \vee \neg U_2 \vee \neg U_2] \wedge [U_1 \vee x_3 \vee x_3]$$

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

### 6.2. Completeness

#### 6.2.1. Definition and basic properties

##### **Definition**

For a class  $C$ , if a set  $A$  satisfies

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

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

Moreover, if we have

(b)  $A \in C$ ,

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

##### 定義

クラスCに対して, 集合Aが次を満たすとき

$$(a) \forall L \in C [L \leq_m^P A]$$

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

$$(b) A \in C$$

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

**Theorem.** For any C-hard (or C-complete) set  $A$ ,

- |   |  |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$                     | CP: $C \not\subseteq P \rightarrow A \notin P$                     |
| (2) $A \in NP \rightarrow C \subseteq NP$                   | CP: $C \not\subseteq NP \rightarrow A \notin NP$                   |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | CP: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$   | CP: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$   |

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  $A \in P$ , we have  $B \in P$

(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) $A \in P \rightarrow C \subseteq P$                     | 対偶: $C \not\subseteq P \rightarrow A \notin P$                     |
| (2) $A \in NP \rightarrow C \subseteq NP$                   | 対偶: $C \not\subseteq NP \rightarrow A \notin NP$                   |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | 対偶: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$   | 対偶: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$   |

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,

$$(1) A \in P \rightarrow C \subseteq P$$

$$\text{CP: } C \not\subseteq P \rightarrow A \notin P$$

$$(2) A \in \text{NP} \rightarrow C \subseteq \text{NP}$$

$$\text{CP: } C \not\subseteq \text{NP} \rightarrow A \notin \text{NP}$$

$$(3) A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$$

$$\text{CP: } C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$$

$$(4) A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$$

$$\text{CP: } C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$$

**Ex. : Meaning of Theorem for class NP**

Let A be NP-complete set.

By the contraposition of Theorem (1) we have

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

That is, NP-complete sets are NP-sets that cannot be recognized in polynomial time unless  $\text{P} = \text{NP}$ .

## 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**  $C$ 困難(または $C$ 完全)な任意の集合 $A$ に対して,

- |   |  |
|---|--|
| (1) $A \in P \rightarrow C \subseteq P$                     | 対偶: $C \not\subseteq P \rightarrow A \notin P$                     |
| (2) $A \in NP \rightarrow C \subseteq NP$                   | 対偶: $C \not\subseteq NP \rightarrow A \notin NP$                   |
| (3) $A \in \text{coNP} \rightarrow C \subseteq \text{coNP}$ | 対偶: $C \not\subseteq \text{coNP} \rightarrow A \notin \text{coNP}$ |
| (4) $A \in \text{EXP} \rightarrow C \subseteq \text{EXP}$   | 対偶: $C \not\subseteq \text{EXP} \rightarrow A \notin \text{EXP}$   |

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

つまり,  $NP$ 完全集合は $P=NP$ でない限り,

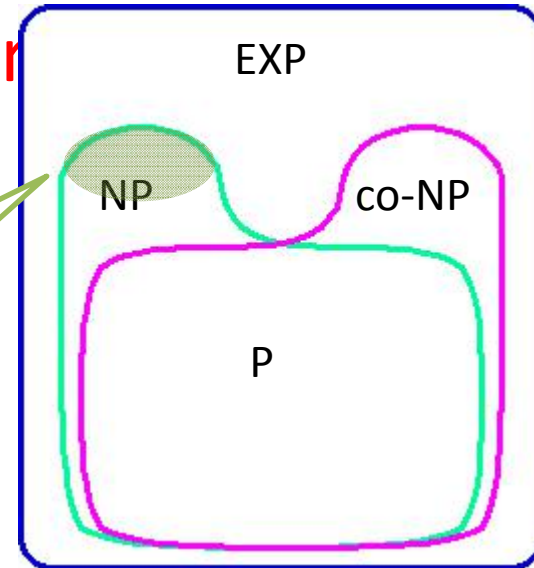
多項式時間では認識できない $NP$ 集合である.

# 6. Analysis on Polynomial-Time Computability

## 6.2. Completeness

### 6.2.1. Definition and basic properties

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

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

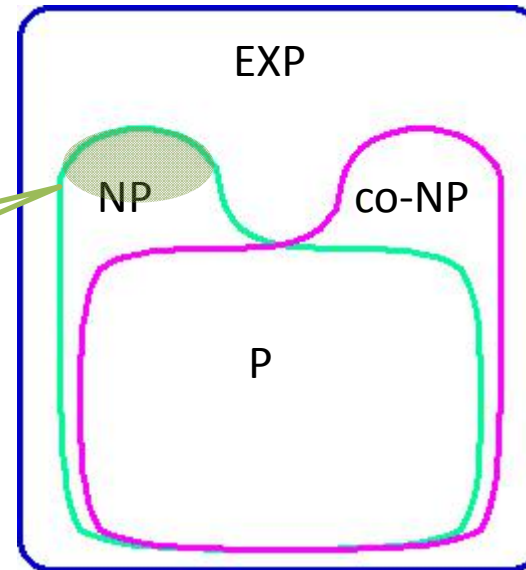
That is, NP-complete sets are NP-sets that cannot be recognized in polynomial time unless  $P = NP$ .

## 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

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： $NP \neq P \rightarrow A \notin P$

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 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  $B \in C \rightarrow B$  is  $C$ -complete.

Proof:

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

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

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

That is,  $B$  is  $C$ -hard.

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

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

証明:

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

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

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

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

**Definition** For a class  $C$ , if a set  $A$  satisfies

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

the set  $A$  is called **C-hard** (under  $\leq_m^P$ ). Moreover, if we have

(b)  $A \in C$ ,

then  $A$  is called **C-complete**.

**Theorem**  $A$ : 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  $B \in C \rightarrow B$  is C-complete.

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

## 6. 多項式時間計算可能性の解析

### 6.2. 完全性

**定義** クラス $C$ に対して, 集合 $A$ が以下を満たすとき

(a)  $\forall L \in C [L \leq_m^P A]$

集合 $A$ は( $\leq_m^P$ のもとで)**C困難**であるという. さらに

(b)  $A \in C$

であれば,  $A$ は**C完全**であるという.

**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)  $A \leq_m^P B \rightarrow B$  は $C$ 困難.

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

ひとたびNP完全問題 $A$ が示されれば, それを用いて他の問題を測ることができる.

# 6. Analysis on Polynomial-Time Computability

## 6.2. Completeness

There are two ways to prove (NP-)completeness:

1. show 'for all L' according to the definition

- Cook's Theorem; he simulated Turing machine by SAT in 1971!

Easy to handle since, e.g., 3SAT has a uniform structure.

Basically...

1. For any program in standard form,
2. simulate it by SAT formulae  
→ pretty complicated and tedious

2. use some known complete problem as a seed

- $3SAT \leq_m^P DHAM$ ,  $3SAT \leq_m^P VC$ , ...
- Thousands of NP-complete problems are reduced from 3SAT!
- E.g., from "DHAM is NP-complete for general graphs", we have
  - DHAM is NP-complete even for planar graphs
  - DHAM is NP-complete even for graphs with max degree=3
  - DHAM is NP-complete even for bipartite graphs...

max  
degree=5

# 6. 多項式時間計算可能性の解析

## 6.2. 完全性

(NP)完全性を示す二つの方法:

### 1. 定義に忠実に「すべての $L'$ 」に対して示す

- クックの定理; 彼は1971年にSATでチューリングマシンのシミュレータを構築した!

例えば3SATは一様な構造を持っているので、扱いやすい。

基本的には...

- 標準形で書かれたプログラムを
- SATの命題論理式で模倣  
→非常に複雑&面倒

### 2. すでに完全性が示されている問題をタネに使う

- $3SAT \leq_m^P DHAM$ ,  $3SAT \leq_m^P VC$ , ...
- 千を超えるNP完全問題が3SATからの還元で示されている!
- 例えば「一般のグラフ上でDHAMはNP完全」という結果から:
  - DHAMは平面グラフ上に限定してもNP完全
  - DHAMは最大次数3に限定してもNP完全
  - DHAMは二部グラフに限定してもNP完全...

最大次数5

# 6. Analysis on Polynomial-Time Computability

## 6.2. Completeness

**Theorem** VC is NP-complete

[Proof] Since  $VC \in NP$ , we show  $3SAT \leq_m^P VC$ .

For given formula  $F(x_1, x_2, \dots, x_n)$ , we construct a pair  $\langle G, k \rangle$  of a graph and an integer in polynomial time such that:

There is an assignment that makes  $F()=1$   
 $\Leftrightarrow G$  has a vertex cover of size  $k$

Construction of  $G$  ( $F$  has  $n$  variables and  $m$  clauses):

1. add vertices  $x_i^+, x_i^-$  and the edge  $(x_i^+, x_i^-)$  for each variable  $x_i$  in  $F$
2. For each clause  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$  in  $F$ , add vertices  $l_{j1}, l_{j2}, l_{j3}$  and three edges  $(l_{j1}, l_{j2}), (l_{j2}, l_{j3}), (l_{j3}, l_{j1})$
3. add the edge  $(l_{j1}, x_i^+)$  if the literal  $l_{j1}$  is  $x_i$ , or add  $(l_{j1}, x_i^-)$  if it is  $\neg x_i$  for each clause  $C_j$
4. let  $k = n + 2m$

## 6. 多項式時間計算可能性の解析

### 6.2. 完全性

定理 VCはNP完全問題

[証明]  $VC \in NP$ なので  $3SAT \leq_m^P VC$ を示せばよい.

与えられた論理式  $F(x_1, x_2, \dots, x_n)$  から以下の条件を満たすグラフと整数の組  $\langle G, k \rangle$  を多項式時間で構成する:

$F()=1$  とする割当てが存在する  
 $\Leftrightarrow G$  が大きさ  $k$  の頂点被覆をもつ

$G$ の構成方法 ( $F$ は  $n$  変数・ $m$  項からなる):

1.  $F$ の各変数  $x_i$  に対して, 頂点  $x_i^+, x_i^-$  と辺  $(x_i^+, x_i^-)$  を追加する
2.  $F$ の各項  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$  に対して, 頂点  $l_{j1}, l_{j2}, l_{j3}$  と3辺  $(l_{j1}, l_{j2}), (l_{j2}, l_{j3}), (l_{j3}, l_{j1})$  を追加する
3. 各項  $C_j$  に対して, リテラル  $l_{j1}$  が  $x_i$  なら辺  $(l_{j1}, x_i^+)$  を,  $\neg x_i$  なら辺  $(l_{j1}, x_i^-)$  を追加する
4.  $k = n + 2m$  とする

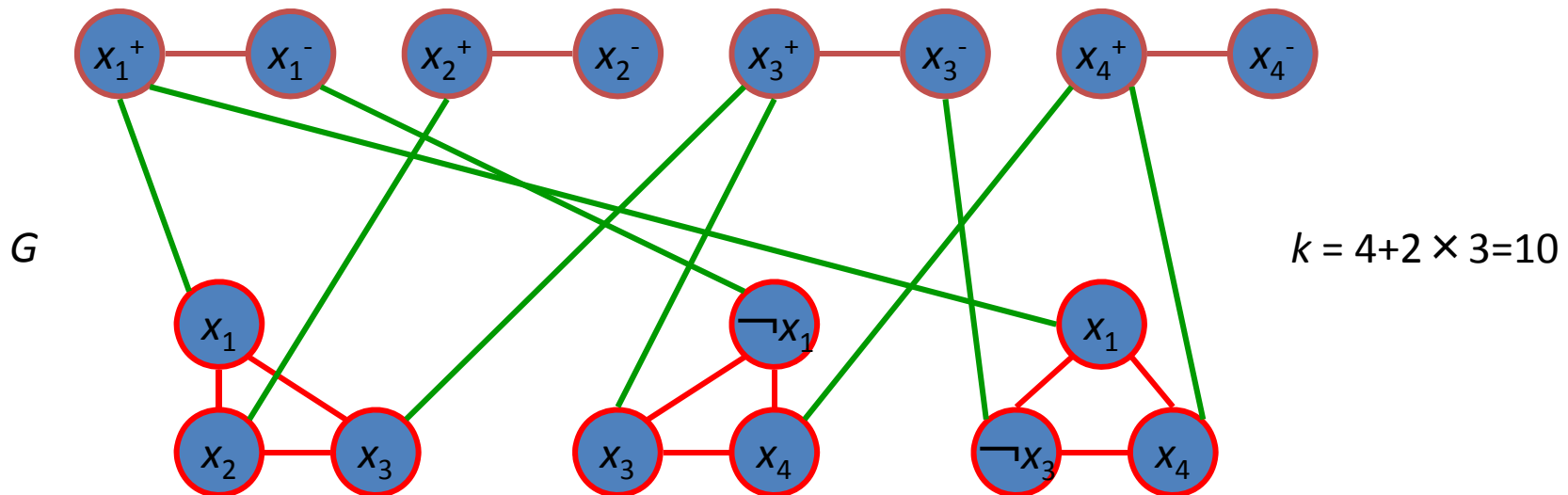
# Theorem VC is NP-complete

There is an assignment that makes  $F()=1$   
 $\Leftrightarrow G$  has a vertex cover of size  $k$

Construction of  $G$  ( $F$  has  $n$  variables and  $m$  clauses):

1. add vertices  $x_i^+, x_i^-$  and the edge  $(x_i^+, x_i^-)$  for each variable  $x_i$  in  $F$
2. For each clause  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$  in  $F$ , add vertices  $l_{j1}, l_{j2}, l_{j3}$  and three edges  $(l_{j1}, l_{j2}), (l_{j2}, l_{j3}), (l_{j3}, l_{j1})$
3. add the edge  $(l_{j1}, x_i^+)$  if the literal  $l_{j1}$  is  $x_i$  or add  $(l_{j1}, x_i^-)$  if it is  $\neg x_i$  for each clause  $C_j$
4. let  $k = n + 2m$

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



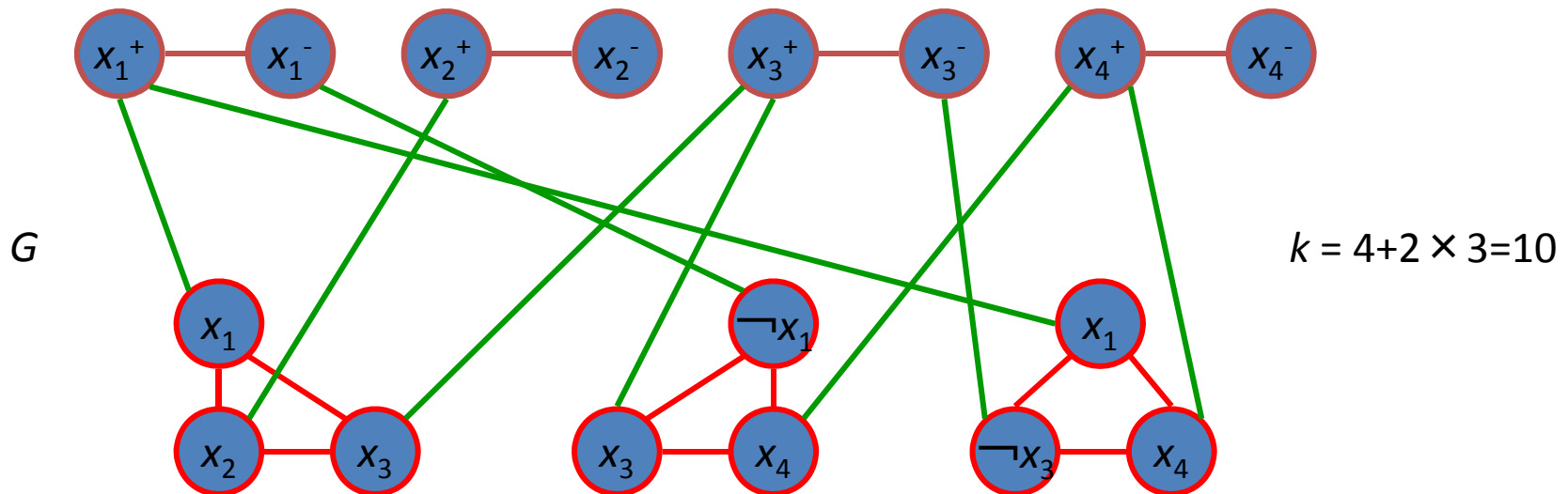
# 定理 VCはNP完全問題

$F()=1$  とする割当てが存在する  
 $\Leftrightarrow G$  が大きさ  $k$  の頂点被覆をもつ

$G$  の構成方法 ( $F$  は  $n$  変数・ $m$  項からなる):

1.  $F$  の各変数  $x_i$  に対して, 頂点  $x_i^+, x_i^-$  と辺  $(x_i^+, x_i^-)$  を追加する
2.  $F$  の各項  $C_j = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$  に対して, 頂点  $l_{i_1}, l_{i_2}, l_{i_3}$  と3辺  $(l_{i_1}, l_{i_2}), (l_{i_2}, l_{i_3}), (l_{i_3}, l_{i_1})$  を追加する
3. 各項  $C_j$  に対して, リテラル  $l_{i_1}$  が  $x_i$  なら辺  $(l_{i_1}, x_i^+)$  を,  $\neg x_i$  なら辺  $(l_{i_1}, x_i^-)$  を追加する
4.  $k = n + 2m$  とする

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





# Theorem VC is NP-complete

It is easy to see that the construction of  $G$  from  $F$  can be done in polynomial time of the size of  $F$ . Hence, we show that...

There is an assignment that makes  $F()=1$   
 $\Leftrightarrow G$  has a vertex cover of size  $k$

**Observation:**

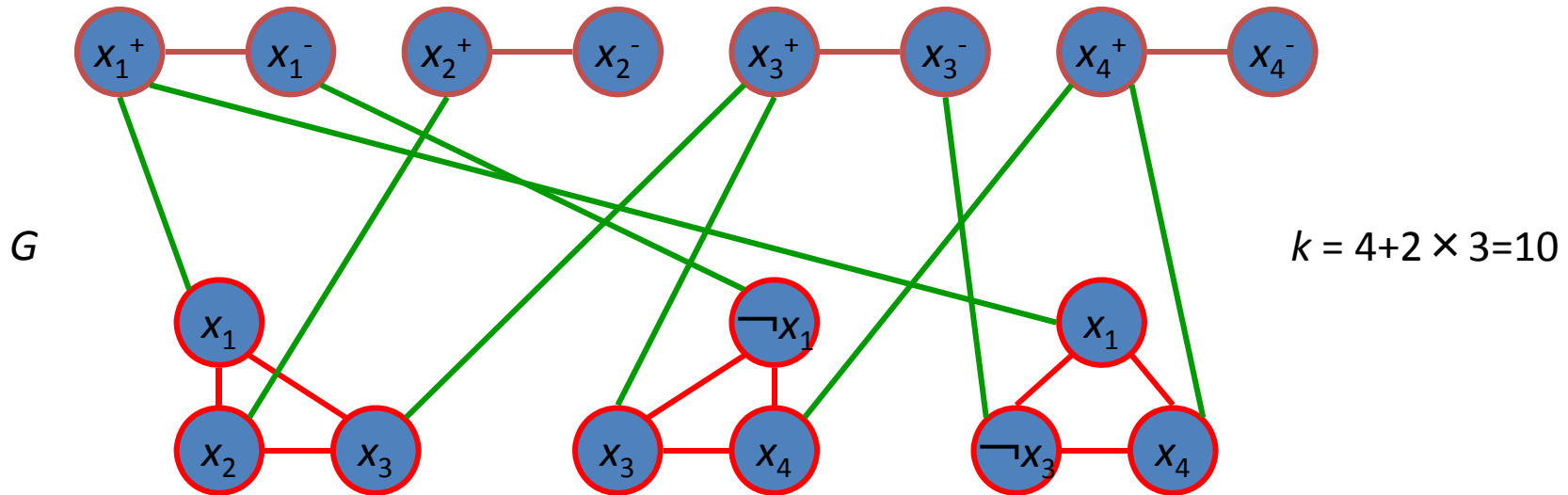
From the construction of  $G$ ,  
 any vertex cover  $S$  should contain

- at least one of  $x_i^+$  or  $x_i^-$
- at least 2 of 3 vertices in  $C_j$

Hence we have  $|S| \geq n+2m = k$ .

We have no extra vertex!!

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



# 定理 VCはNP完全問題

$F$ から $G$ の構成は、明らかに多項式時間で可能である。

したがって、以下を証明すればよい：

$F()=1$ とする割当てがある  
 $\Leftrightarrow G$  が大きさ $k$ の頂点被覆をもつ

観測：

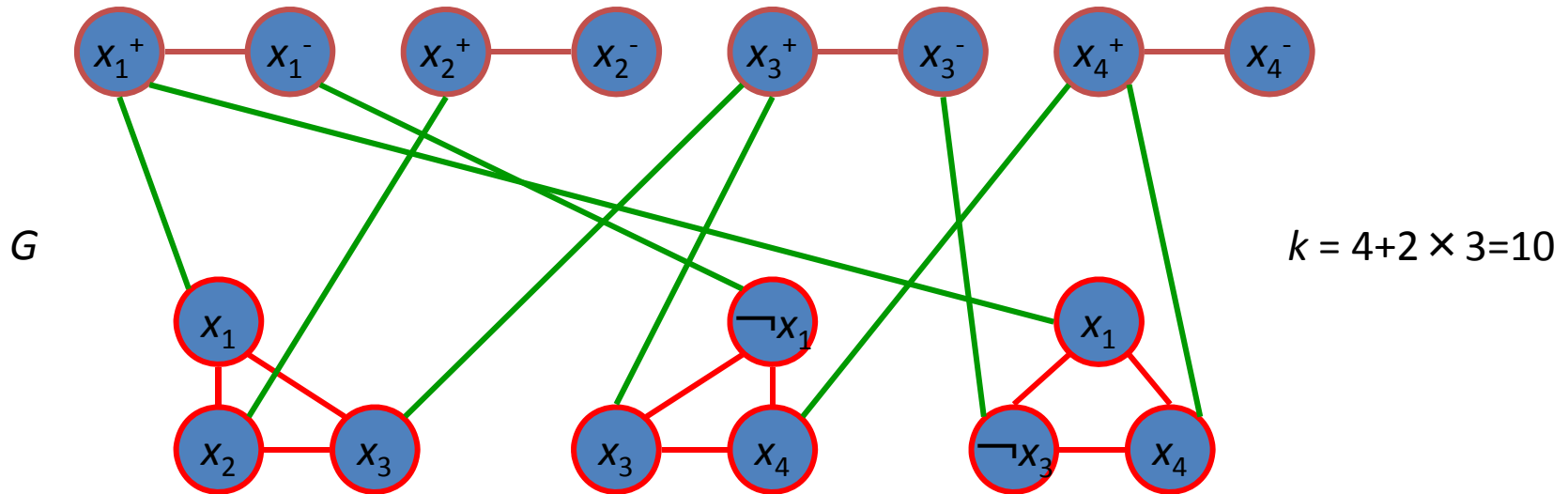
$G$ の構成方法から、頂点被覆  $S$  は  
 以下の頂点を必ず含む

$\left\{ \begin{array}{l} x_i^+ \text{ or } x_i^- \text{ から少なくとも一つ} \\ C_j \text{ の三つの頂点から少なくとも二つ} \end{array} \right.$

よって  $|S| \geq n+2m = k$

余分な頂点は一つもない！

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



# Theorem VC is NP-complete

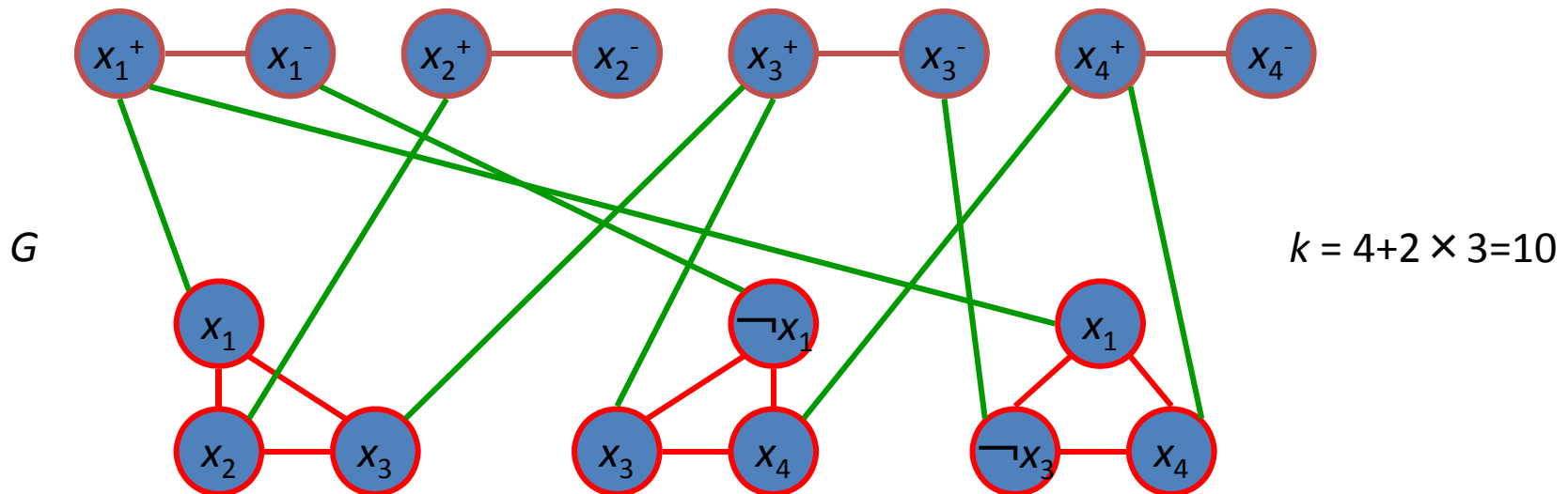
There is an assignment that makes  $F()=1$   
 $\Rightarrow G$  has a vertex cover of size  $k$

1. Put  $\left\{ \begin{array}{l} x_i^+ \text{ if } x_i=1 \\ x_i^- \text{ if } x_i=0 \end{array} \right\}$  into  $S$  for each  $x_i$ .

2. Since each clause  $C_j=(l_{i1}, l_{i2}, l_{i3})$  is satisfied, at least one literal, say  $l_{i1}$ , the edge  $(l_{i1}, x_{i1})$  is covered by the variable  $x_{i1}$ . Therefore, put the remaining literals  $(l_{i2}, l_{i3})$  into  $S$ .

$\Rightarrow$  From the **Observation**  $S$  is a vertex cover of size  $k$ .

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

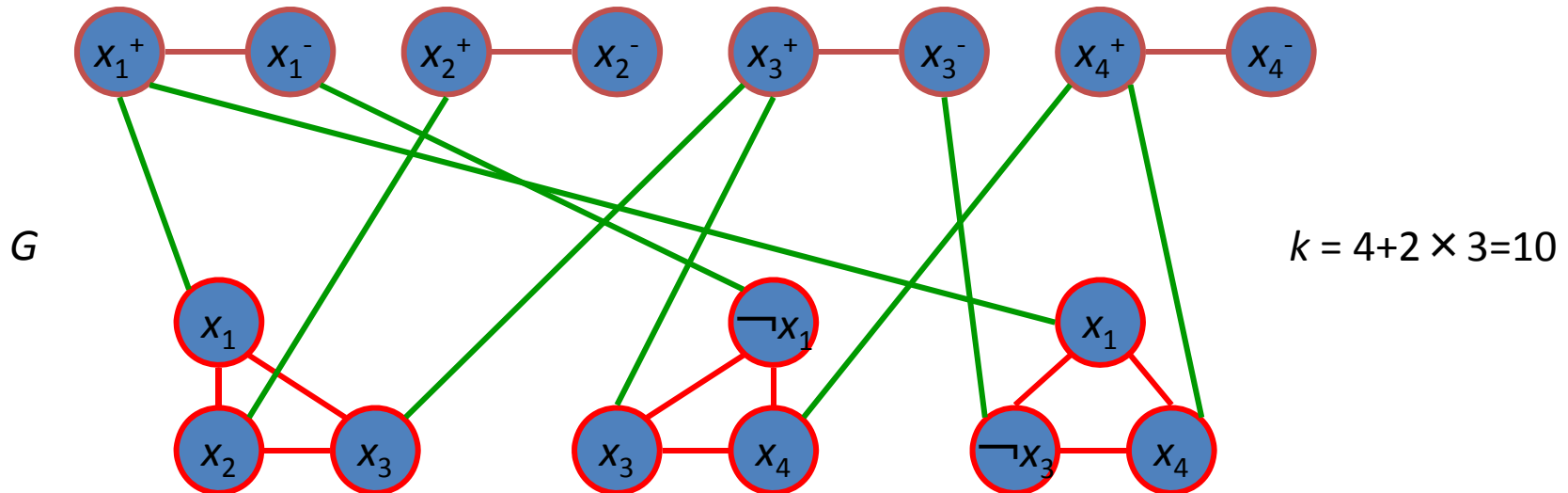


# 定理 VCはNP完全問題

$F()=1$ とする割当てがある  
 $\Rightarrow G$  が大きさ  $k$  の頂点被覆をもつ

1. 各  $x_i$  に対して  $\left\{ \begin{array}{l} x_i=1 \text{ なら } x_i^+ \\ x_i=0 \text{ なら } x_i^- \end{array} \right\}$  を  $S$  に
2. 各項  $C_j=(l_{j1}, l_{j2}, l_{j3})$  は充足されているので, 少なくとも一つのリテラル  $l_{j1}$  に対して辺  $(l_{j1}, x_{i1})$  は変数  $x_{i1}$  で被覆されている. そこで残りの二つのリテラル  $(l_{j2}, l_{j3})$  を  $S$  に  
 $\Rightarrow$  **観測** より  $S$  は大きさ  $k$  の頂点被覆.

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



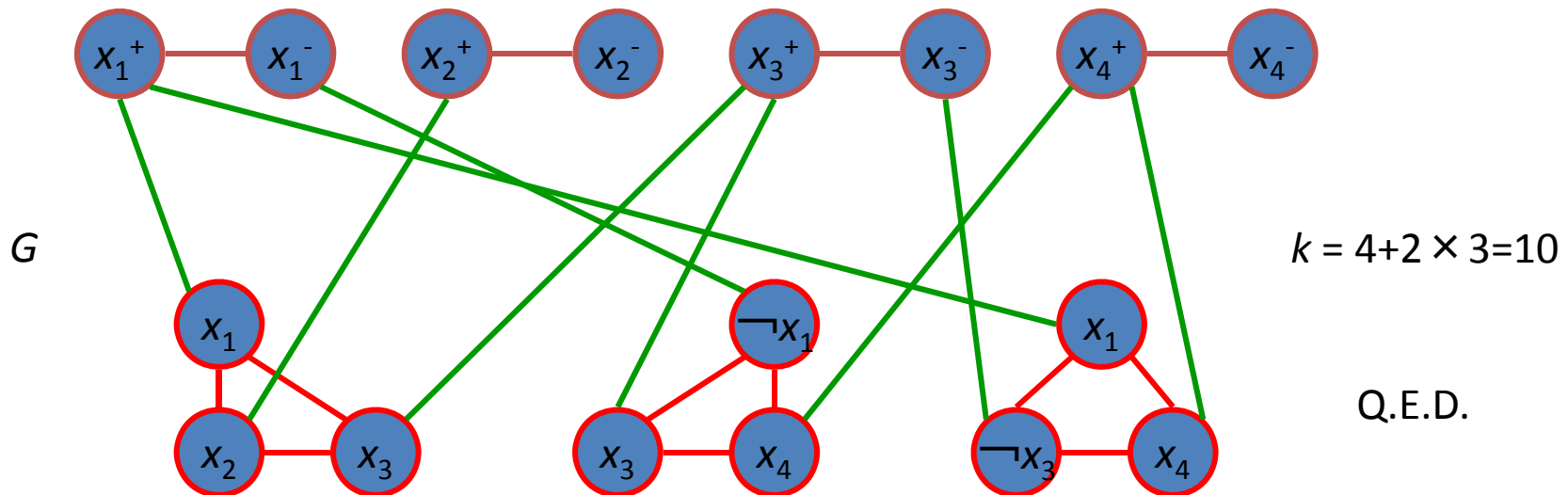
# Theorem VC is NP-complete

If  $G$  has a vertex cover of size  $k$ , there is an assignment that makes  $F()=1$

1. From **Observation**, a cover  $S$  contains  $2m$  vertices from the clauses, and  $n$  vertices from the variables.
2. Thus the cover  $S$  contains exactly one of  $x_i^+$  and  $x_i^-$  and exactly two literals of a clause  $C_j$ .
3. Hence each clause  $C_j$  contains exactly one literal  $l_i$  which is not in  $S$ , and hence incident edge should be covered by a variable vertex.

$\Rightarrow$  The following assignment satisfies  $F: \begin{cases} x_i=1 & \text{if } x_i^+ \text{ in } S \\ x_i=0 & \text{if } x_i^- \text{ in } S \end{cases}$

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



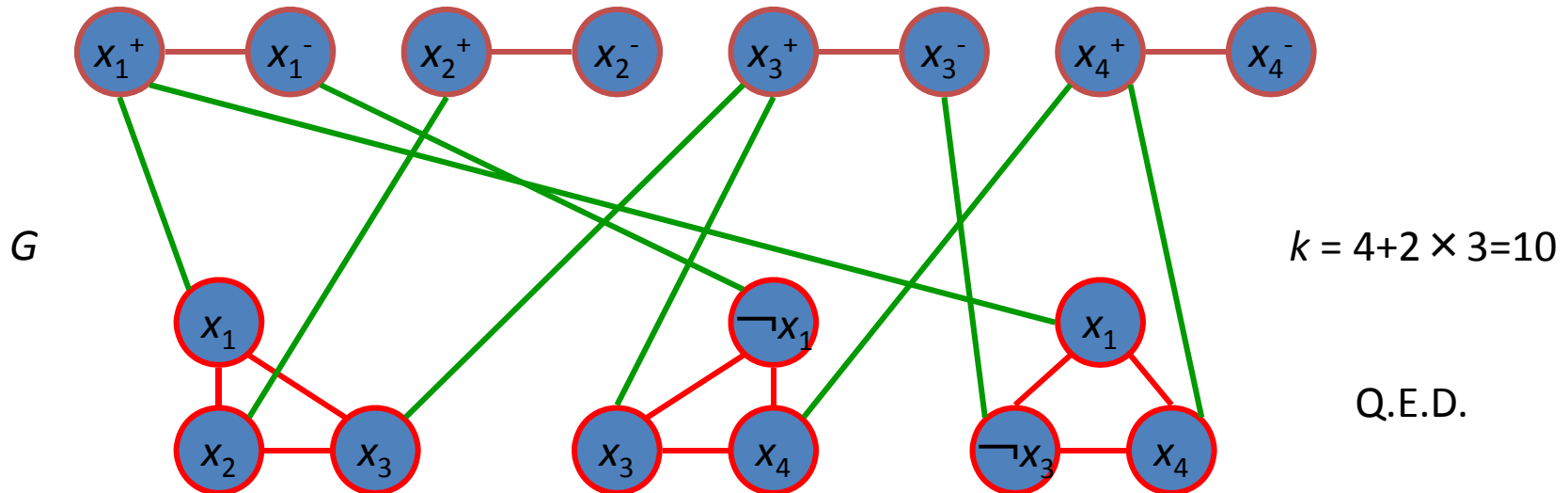
# 定理 VCはNP完全問題

$G$  が大きさ  $k$  の頂点被覆をもつなら,  $F()=1$  とする割当てが存在する

1. 観測 より, 被覆  $S$  は各項から  $2m$  頂点含み, 変数から  $n$  頂点含む.
2. よって被覆  $S$  は  $x_i^+$  と  $x_i^-$  からちょうど一つと, 各項  $C_j$  からちょうど二つのリテラルを含む
3. つまり各項  $C_j$  は  $S$  に含まれないリテラル  $l_i$  をちょうど一つだけ含み, そこにつながる辺は変数頂点で被覆されている.

⇒ 以下の条件を満たす割当ては  $F$  を充足する:  $\left[ \begin{array}{l} x_i^+ \in S \text{ なら } x_i=1 \\ x_i^- \in S \text{ なら } x_i=0 \end{array} \right]$

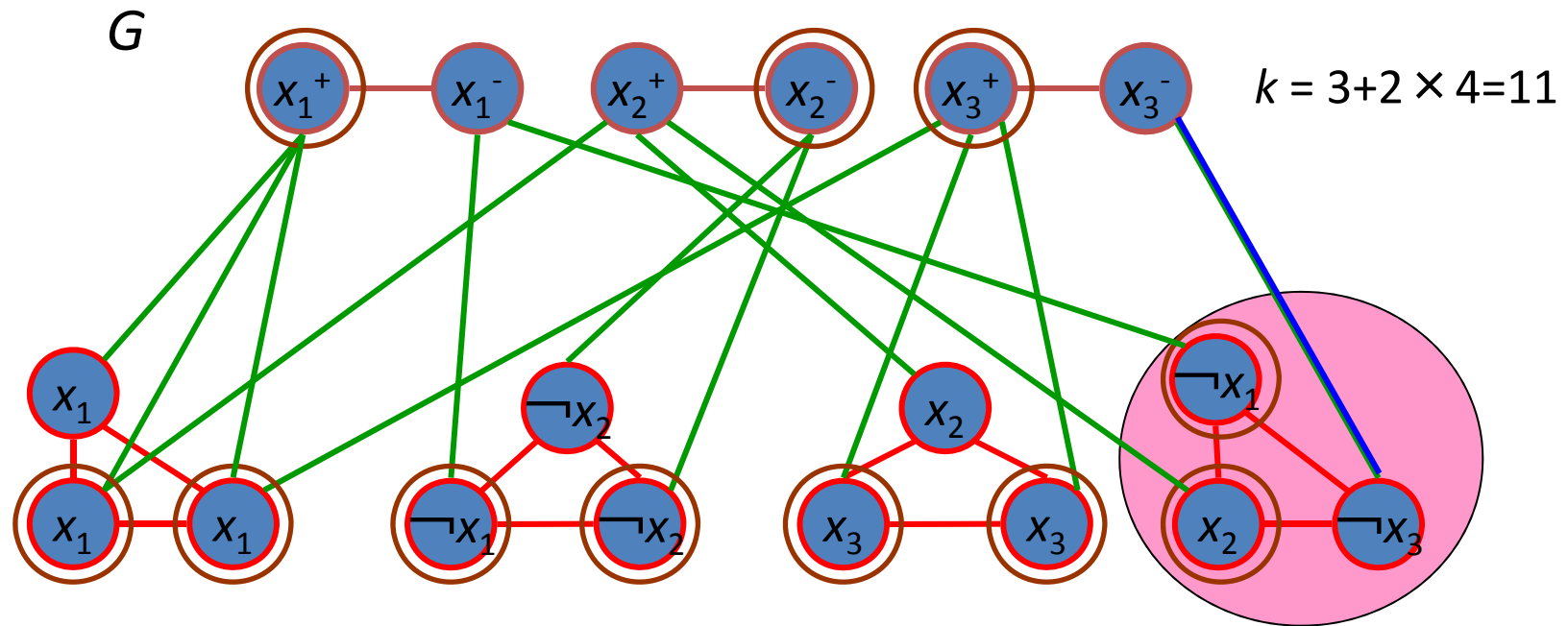
例:  $F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$



# Theorem VC is NP-complete... Addition

What happens if the formula is not satisfiable?

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

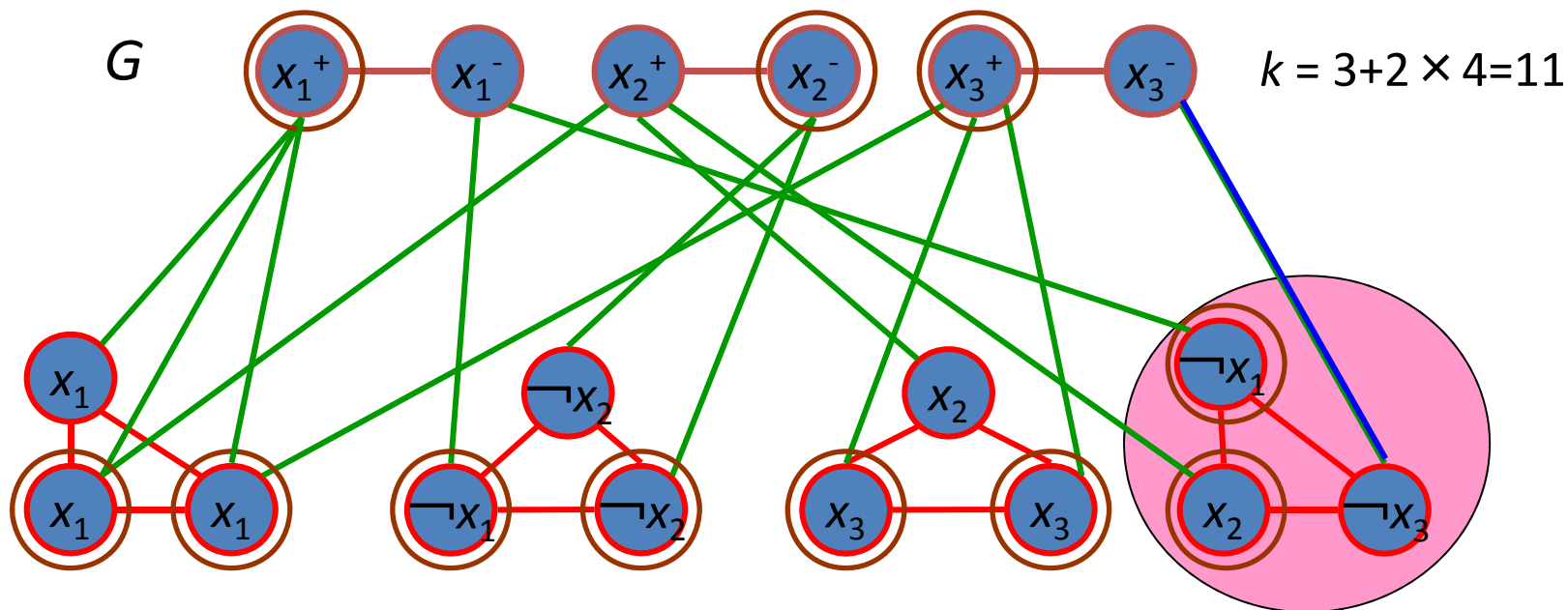


If  $F$  is unsatisfiable, it contains at least one clause s. t. each literal is *not* covered by a vertex. So, Vertex Cover should contain *three* literals in the clause. Hence any vertex cover has size at least  $k+1$ .

# 定理 VCはNP完全問題... 補足

命題論理式が充足可能でないときにはどうなるのか?

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



$F$ が充足可能でないときには、ある項において頂どのリテラルも変数側の頂点によって被覆されない。よって頂点被覆集合はこの項のリテラルを三つともふくまなければならない。したがって頂点被覆集合は少なくとも大きさ $k+1$ になる。



# 6. Analysis on Polynomial-Time Computability

## 6.2. Completeness

### Theorem

DHAM is NP-complete even if maximum degree=5.

[Proof]

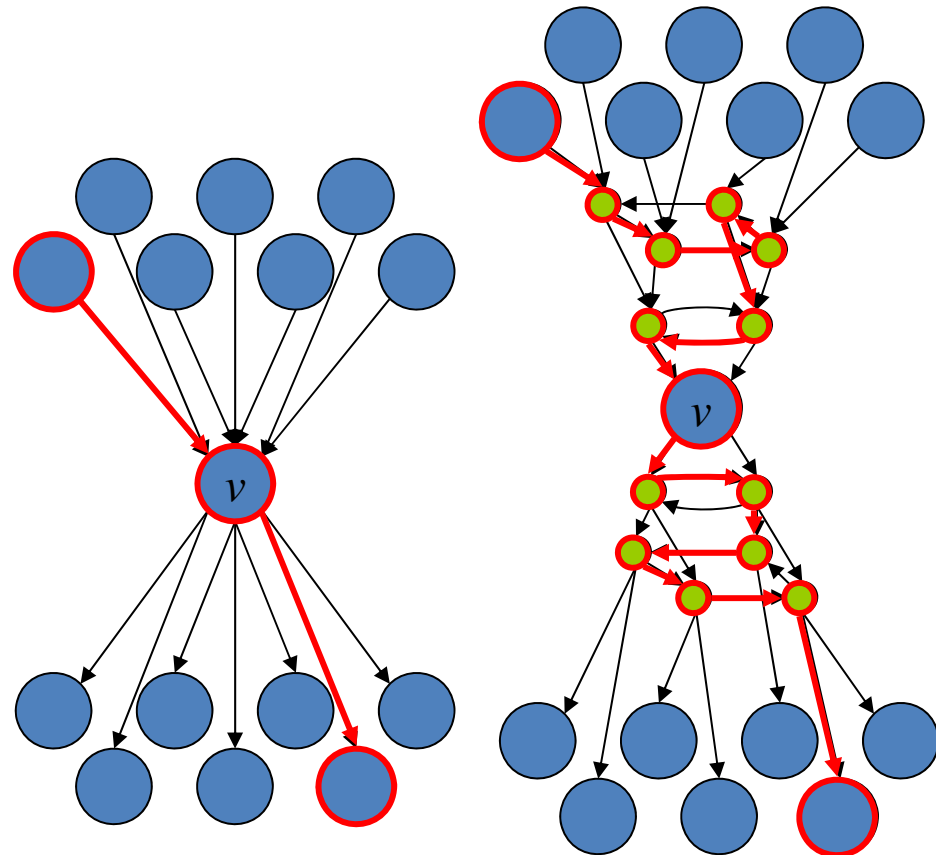
Since  $\text{DHAM} \in \text{NP}$ ,  $\text{DHAM}_{\leq 5} \in \text{NP}$ .  
We show  $\text{DHAM} \leq_m^P \text{DHAM}_{\leq 5}$ .

Idea:

Replace the set of “arcs to  $v$ ” and the set of “arcs from  $v$ ” by a right ‘gadget’.

A Hamiltonian cycle through  $v$  on the original graph corresponds to the Hamiltonian cycle through  $v$  on the resultant graph.

**degree:** the number of edges incident to a vertex



# 6. 多項式時間計算可能性の解析

## 6.2. 完全性

### 定理

DHAM はグラフの最大次数が5でも NP完全

[証明]

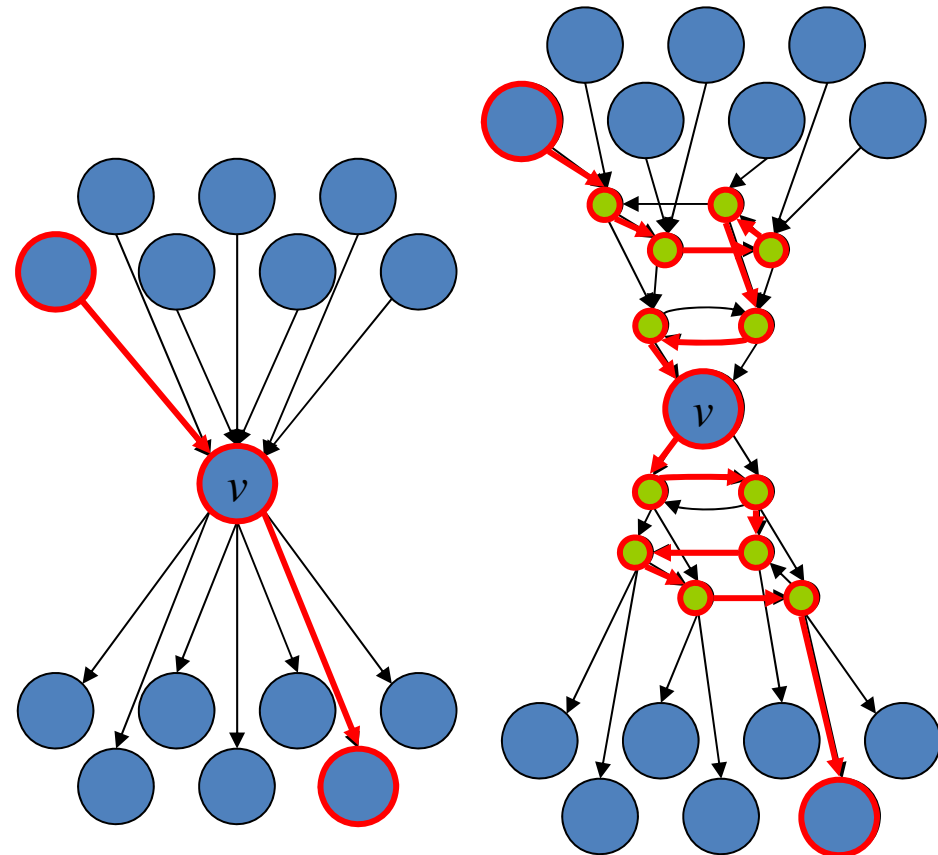
DHAM  $\in$  NPなので  $\text{DHAM}_{\leq 5} \in \text{NP}$ .  
よって  $\text{DHAM} \leq_m^P \text{DHAM}_{\leq 5}$  を示す

### アイデア

「 $v$ に入る辺」や「 $v$ から出る辺」を  
しかるべきガジェットで置き換える

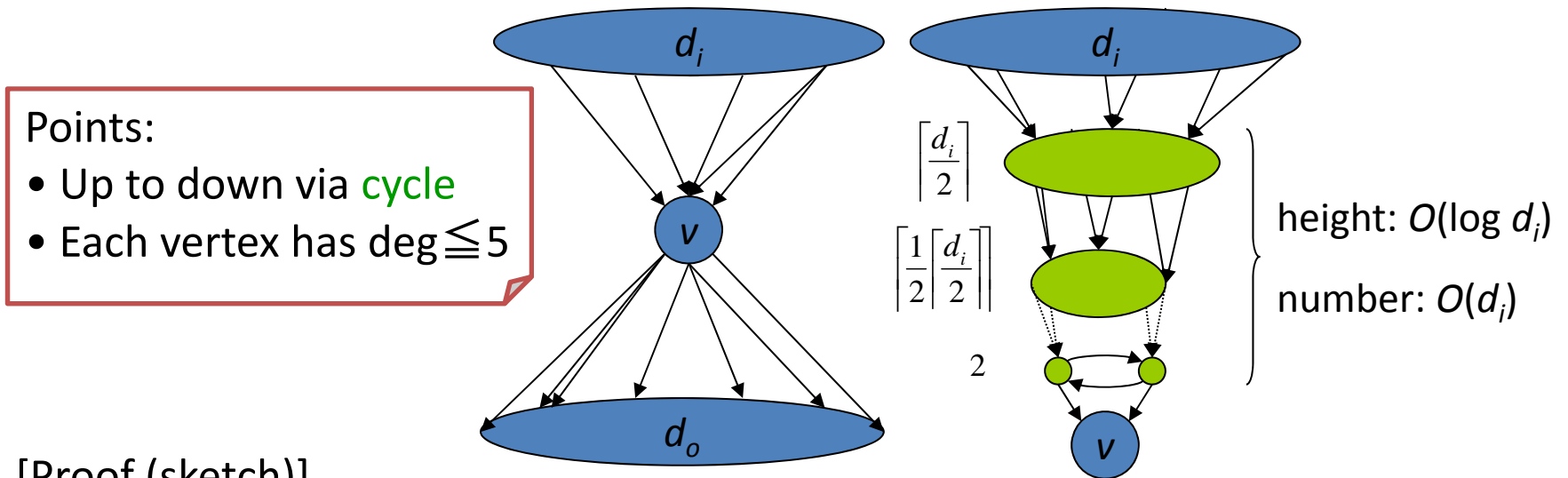
元のグラフで $v$ を通る  
ハミルトン閉路は、  
置き換えたグラフで $v$ を  
通るハミルトン閉路に  
対応づけられる。

次数: 頂点につながる  
辺の本数



## 6.2. Completeness

**Theorem** DHAM is NP-complete even if max. degree=5.



[Proof (sketch)]

For each vertex  $v$  of degree  $\geq 6$ , replace the edges around  $v$  by the gadget.

1. If the original graph  $G$  has  $n$  vertices with  $m$  edges, the resultant graph  $G'$  contains  $O(n+m)$  vertices with  $O(m)$  edges. Hence the reduction can be done in polynomial time of  $n$  &  $m$ .
2. Each vertex in  $G'$  has degree **at most 5**.
3.  $G$  has a Hamiltonian cycle  $\Leftrightarrow G'$  has a Hamiltonian cycle.

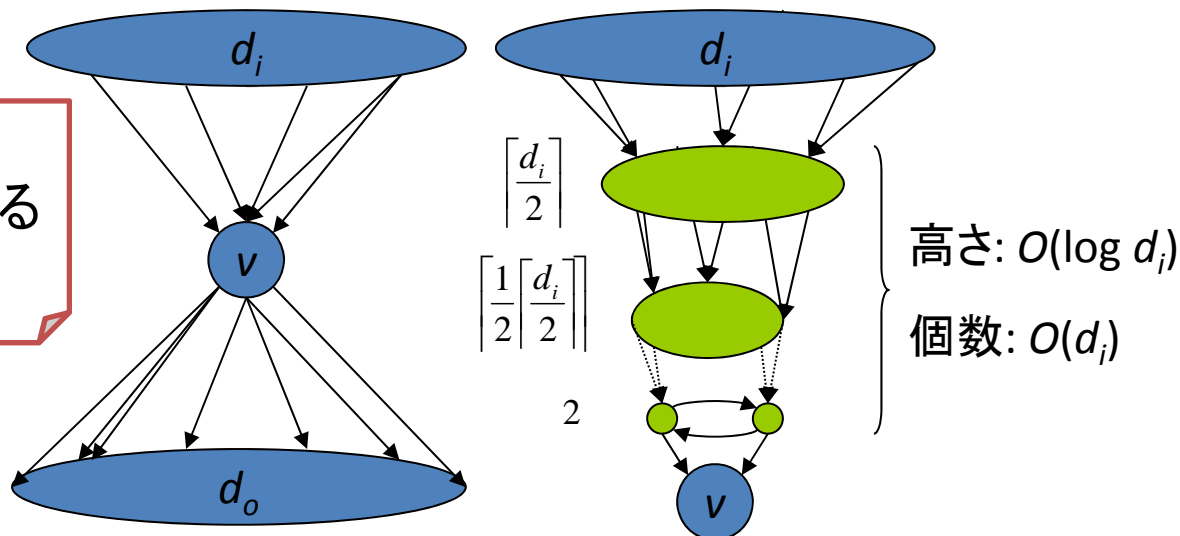
QED.

## 6.2.完全性

定理 DHAM はグラフの最大次数が5でも NP完全

ポイント:

- 上から下に閉路を通る
- 各頂点の次数  $\leq 5$



[証明 (概要)]

次数  $\geq 6$  の各頂点  $v$  に対して,  $v$  の頂点の周りの辺をガジェットで置き換える

1. もとのグラフ  $G$  の頂点数を  $n$ , 辺数を  $m$  とすると, 還元後に得られるグラフ  $G'$  の頂点数は  $O(n+m)$  で辺数は  $O(m)$  となる. よってこの還元は  $n$  と  $m$  の多項式時間で実行できる.
2.  $G'$  の各頂点の次数は たかだか 5.
3.  $G$  がハミルトン閉路をもつ  $\Leftrightarrow G'$  がハミルトン閉路をもつ

QED.

## Addition (おまけ)

- R. Uehara, S. Iwata:

Generalized Hi-Q is NP-complete,  
*The Transactions of the IEICE*, E73, p.270-273, 1990.

- P. Zhang, H. Sheng, R. Uehara:

A Double Classification Tree Search Algorithm for  
Index SNP Selection, *BMC Bioinformatics*, 5:89, 2004.

- R. Uehara, S. Teramoto:

Computational Complexity of a **Pop-up Book**  
*4<sup>th</sup> International Conference on Origami in Science, Mathematics, and  
Education*, 2006.

- E. Demaine, M. Demaine, R. Uehara, T. **UNO**, Y. **UNO**:

**UNO** is hard, even for a single player,  
*Theoretical Computer Science*, Vol. 521, pp.51-61, 2014.

- E. D. Demaine, Y. Okamoto, R. Uehara, and Y. Uno:

Computational complexity and an integer programming model of Shakashaka,  
*IEICE Trans.* Vol. E97-A, No. 6, pp. 1213-1219, 2014.

Many natural hard problems are either


- Poly-time solvable, or
- NP-hard



Copyright ©2003 by Robert Sabuda



# Schedule(残りの予定)

- 10/30(Thu): Last class (前半最後の講義)
  - Submission of the report (2) (レポート(2)提出)
  - Course Evaluation Questionnaire (授業アンケート)
  - Office Hour: Comments/Answers on report (2)
- 11/ 6(Thu): mid-term exam (中間試験)
  - 40 points  Notes, Textbook, Copy, Printout,...
  - Only pens and pencils (持ち込み不可)
  - Lesson 3~Lesson 6 (講義3~講義6),  
which means that “no diagonalization”
  - Office Hour: Next lesson by Prof. Miyaji