アルゴリズム入門(7)
(問題の複雑さ)

宮崎修一
京都大学　学術情報メディアセンター

問題の難しさ

ある問題が、「**これだけの計算量で解ける**」というのを示すのは、比較的簡単。（実際にアルゴリズムを設計すれば良い。）

ある問題が、「**これだけの計算量では解けない**」ことを示すのは、難しい。（「どんなアルゴリズムでも駄目だ」ということを証明する必要がある。）

今の所、問題の難しさを示す研究は、成功していない。

↓

問題の難しさを示すために、「ある問題が難しいとすると、この問題も難しい」という論法が使われている。

問題の難しさを示す手法（還元、リダクション）

A：既に難しいことが分かっている問題。
B：難しいことを示したい問題。

問題Aを問題Bに変換している（ただし、答が保存されるように）
つまり、Bを解くことにより、Aを解くことが出来る。

具体例

A：論理式の充足可能性問題（SAT）　難しいことが分かっている。

B：最小頂点被覆問題（VC）　　難しいことを示したい。

SAT

入力：和積形論理式

$$f = (x_1 + \overline{x_2} + x_3)(x_1 + x_8) \ \ldots \ (x_3 + \overline{x_6} + \overline{x_7} + x_{12})$$

出力：$x_1, x_2, \ldots, x_n$ に0/1 を割り当てて、f =1と出来るか？

YES/NO?

$\overline{x_i}$ ： x_i の否定　　x_i =1ならば $\overline{x_i}$ =0

x_i =0ならば $\overline{x_i}$ =1

$(x_1 + \overline{x_2} + x_3)$ =1　　x_1 か$\overline{x_2}$ かx_3 の少なくとも1つが1

=0　　すべて0

SATの例

$f = (x_1 + \overline{x_2})\,(x_1 + x_3)\,(x_1 + x_2)\,(\overline{x_1} + \overline{x_3})$

だと$f=1$にできる。

（ $x_1=1$　$x_2=1$　$x_3=0$ ）

$f = (x_1 + \overline{x_2})\,(\overline{x_1} + x_2)\,(x_1 + x_2)\,(\overline{x_1} + \overline{x_2})$

だと、どうやっても$f=1$にできない。

最小頂点被覆問題（判定問題版）

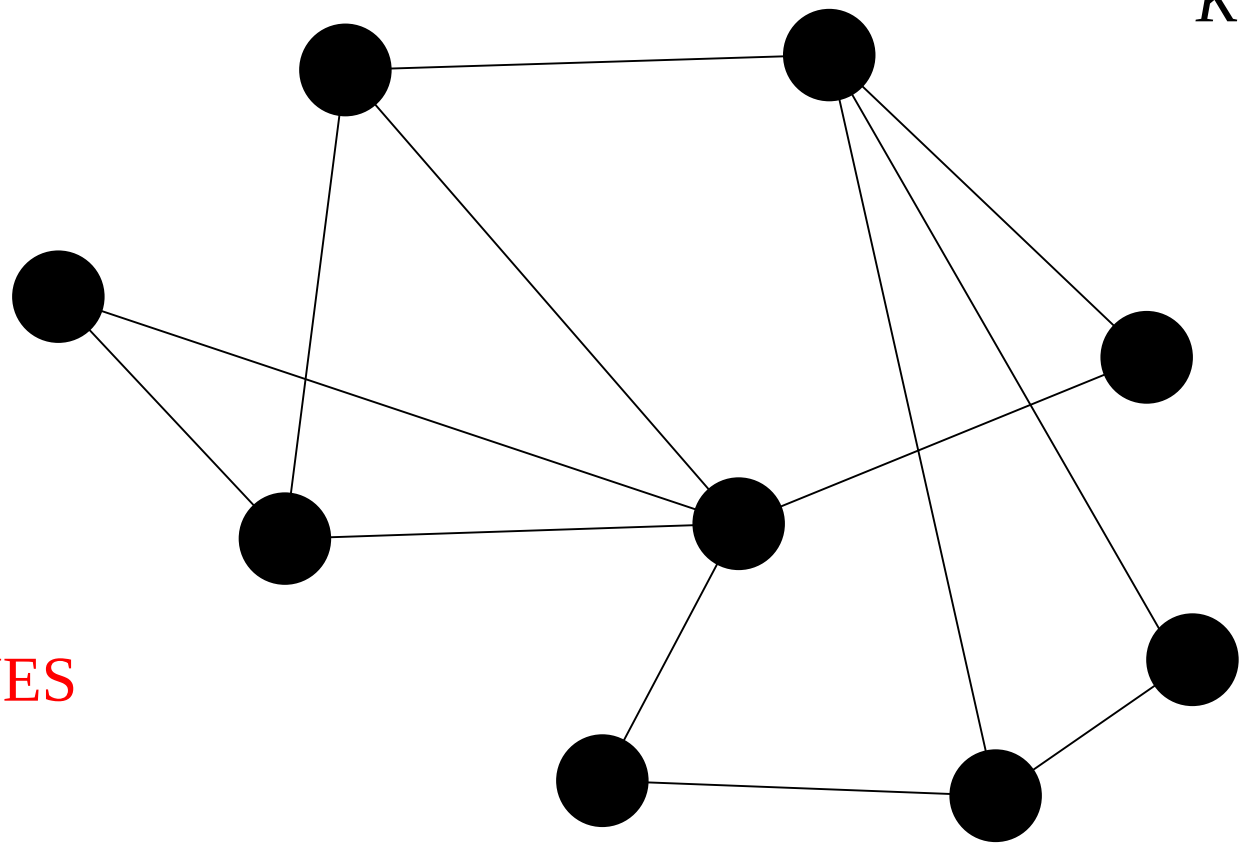
頂点の集合Cで、どの枝も端点のどちらかがCに属する。

グラフGと整数Kが与えられて、

「Gは頂点数K以下の頂点被覆を持つか？　YES／NO？」

を問う。

$K=4$



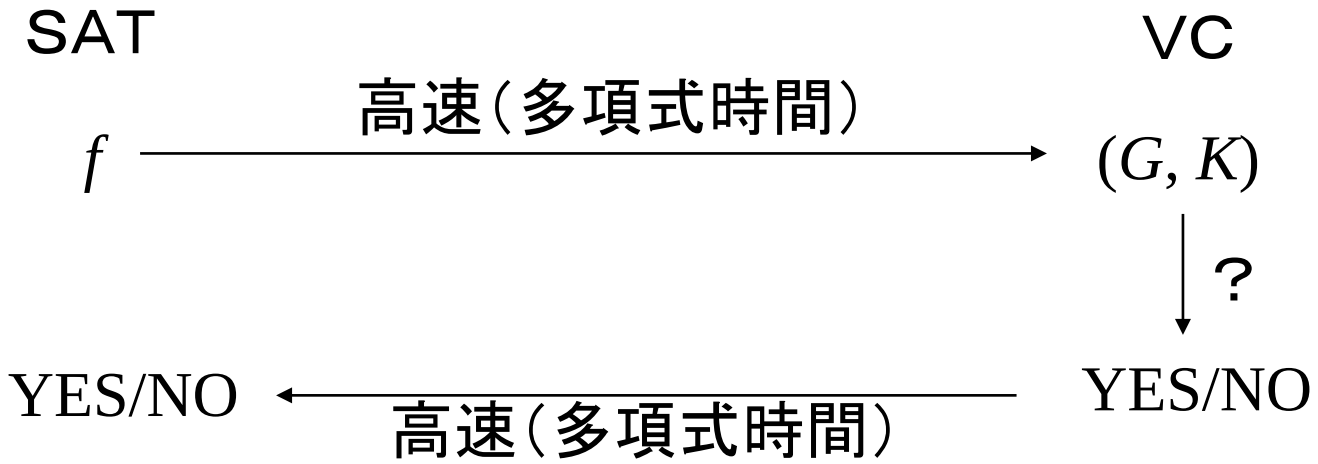
YES

リダクションの概要

元の問題で$f=1$とできる　　YES　　NO

⇕　　⇕　　⇕

K個以下の頂点で被覆できる　　YES　　NO



？のところが高速ならば、全体としてSATに対する高速な
アルゴリズムが作れたことになる。

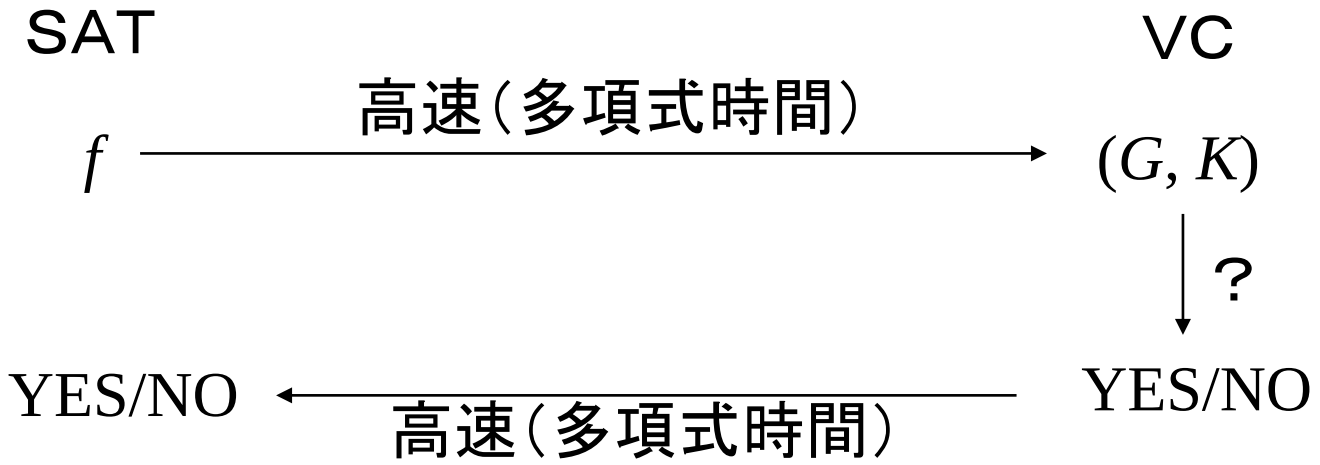
つまり、VCに対する高速なアルゴリズムが存在すれば、SATに対する高速なアルゴリズムも存在することになる。
逆に言えば、SATを解く高速なアルゴリズムが存在しないならば、
VCを解く高速なアルゴリズムも存在しない。

リダクションの概要

元の問題でf =1とできる　　YES　　NO

⇕　　⇕　　⇕

K個以下の頂点で被覆できる　　YES　　NO

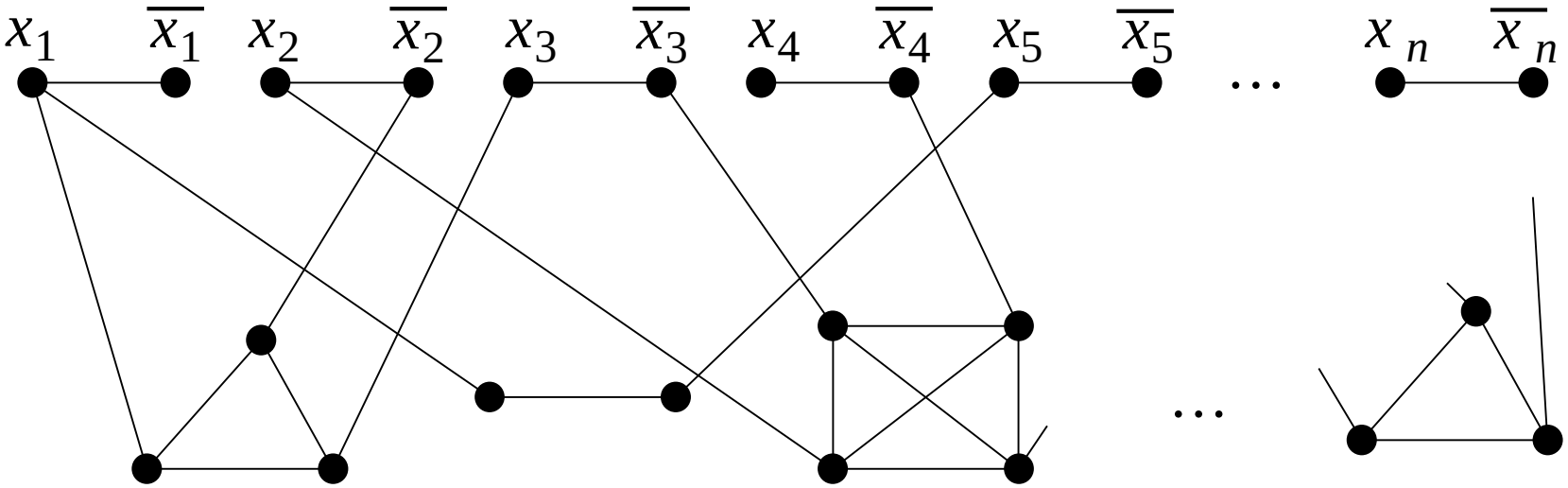


ポイント: fを解いて、YESかNOか分かった後で (G, K) を作るのは
簡単にできる。
しかし、SATは難しい問題なので、fを解くだけで指数時間かかる。
fを解かずに、答えが分からないまま「表面的な」変換だけで、
YES/NOを保存させることが出来るのがミソ。

リダクションの方法

$$f=(x_1+\overline{x}_2+x_3)(x_1+x_5)(x_2+\overline{x}_3+\overline{x}_4+x_6)\ldots(x_8+x_9+\overline{x}_{12})$$

c_1個　c_2個　c_3個　c_m個



$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$

元の問題で$f=1$とできる

⇕

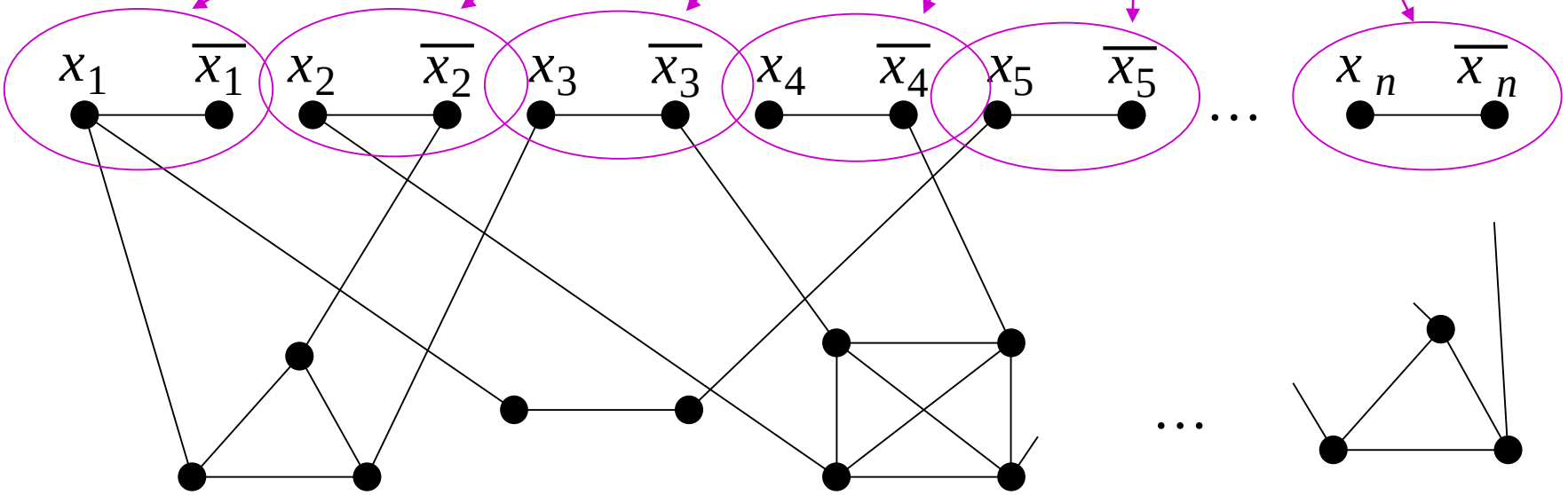
K個以下の頂点で被覆できる

全部でn個

どちらか1個選ばなければいけない。

$$f=(x_1+\overline{x}_2+x_3)(x_1+x_5)(x_2+\overline{x}_3+\overline{x}_4+x_6)\cdots(x_8+x_9+\overline{x}_{12})$$

c_1個　c_2個　c_3個　c_m個



$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$

元の問題で$f=1$とできる

⇕

K個以下の頂点で被覆できる

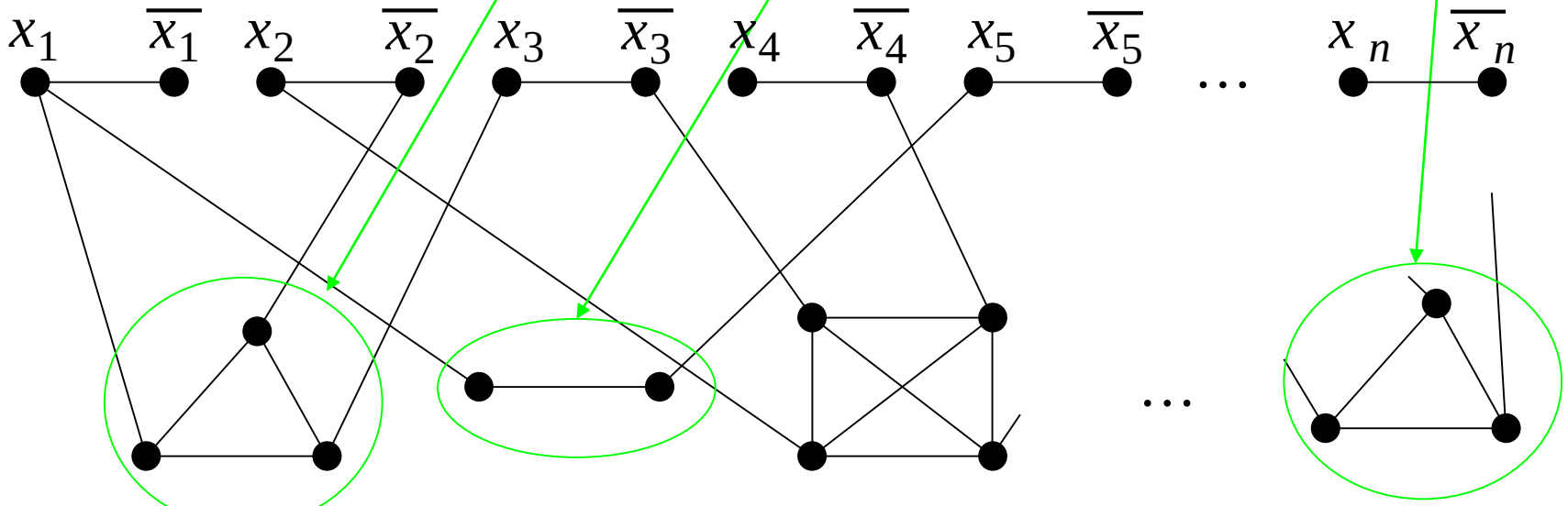
c_1−1個選ばなければならない

c_2−1個選ばなければならない

c_m−1個選ばなければならない

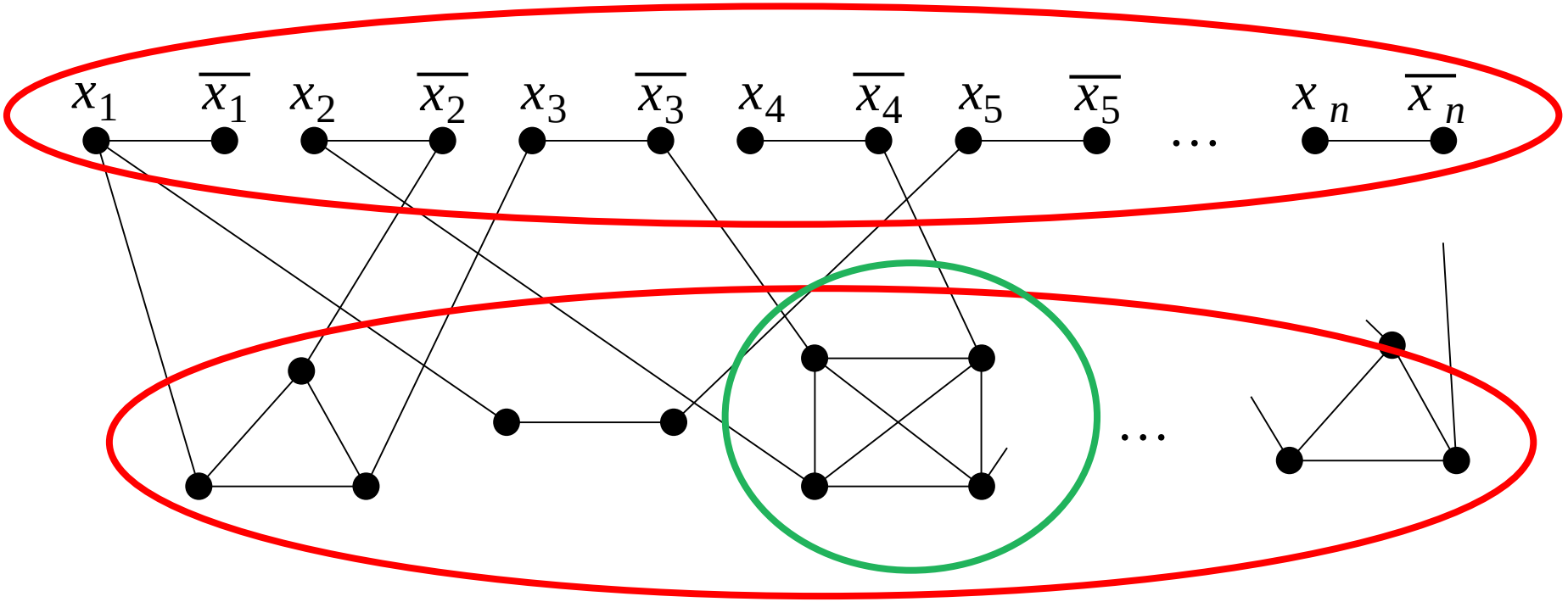
$$f=(x_1+\overline{x}_2+x_3)(x_1+x_5)(x_2+\overline{x}_3+\overline{x}_4+x_6)\ldots(x_8+x_9+\overline{x}_{12})$$

c_1個　c_2個　c_3個　c_m個



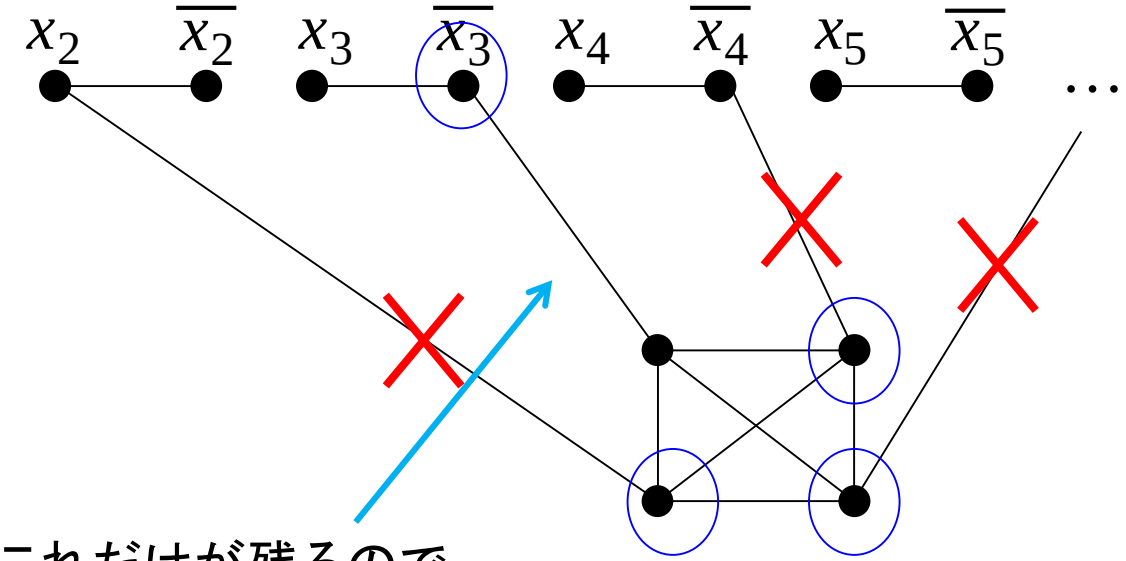
$$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$$

この内部の枝はカバーできた。
2つの間を繋ぐ枝をどうカバーするか



$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$

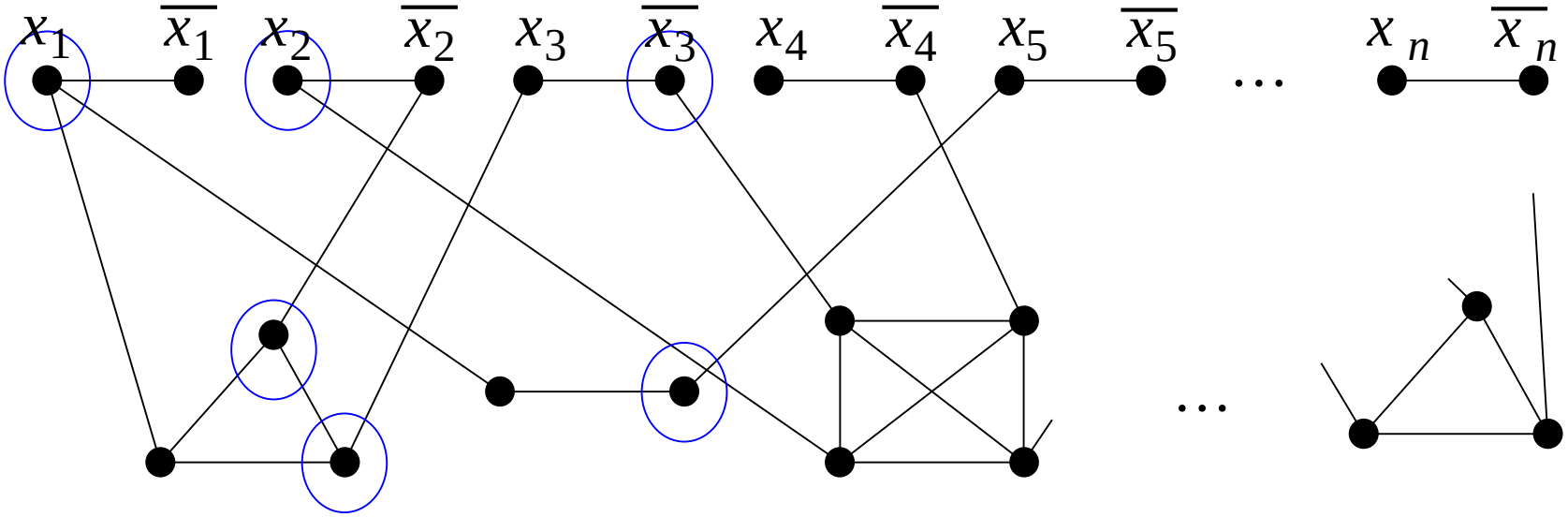
$f=(x_1+\overline{x_2}+x_3)(x_1+x_5)(x_2+\overline{x_3}+\overline{x_4}+x_6)\ldots(x_8+x_9+\overline{x_{12}})$



これだけが残るので、
上でうまくカバー
してくれると嬉しい

このうち、3頂点は選べるので、間の枝のうち、
3本は自動的にカバーできる

内部の枝はカバーできた。
2つの間を繋ぐ枝をどうカバーするか

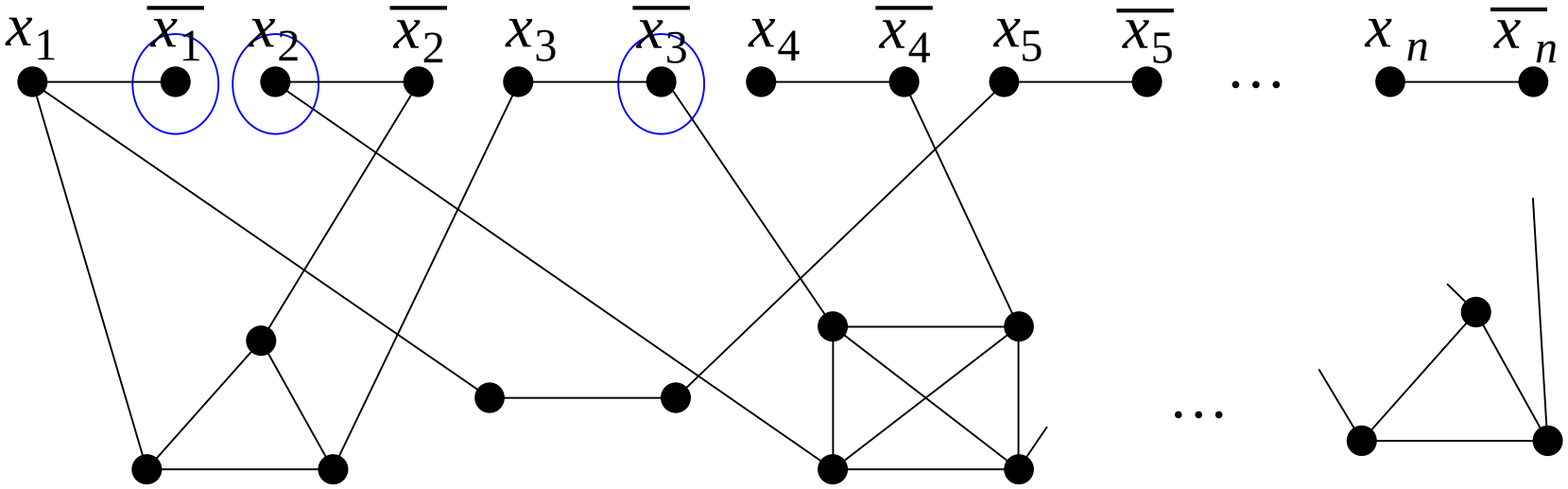


$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$

$$f=\overset{1}{(x_1+\overline{x}_2+x_3)}\overset{1}{(x_1+x_5)}(x_2+\overline{x}_3+\overline{x}_4+x_6)\ldots(x_8+x_9+\overline{x}_{12})$$

$x_1=1$ $x_2=1$ $x_3=0$

困るのは



$K=n+(c_1-1)+(c_2-1)+\ldots+(c_m-1)$

$$f=(x_1+\overline{x}_2+x_3)^{0}\,(x_1+x_5)^{0}\,(x_2+\overline{x}_3+\overline{x}_4+x_6)^{0}\ldots(x_8+x_9+\overline{x}_{12})$$

$x_1=0$ $x_2=1$ $x_3=0$

SATもVCも、「難しい問題」として知られている。
そういう問題が「NP完全問題」。

多項式時間アルゴリズムが存在しないだろうと
思われている（が、まだ証明はされていない）。

グラフに関するNP完全問題は、VC以外にも数限りなくある。

- k-クリーク問題
- k-独立頂点集合問題
- 3彩色問題
- ハミルトン閉路問題

P：　多項式時間アルゴリズムを持つ問題の集合
（つまり、高速なアルゴリズムが存在する。）

PとNP

ここでは、答えがYESかNOを判定する判定問題だけを考える。

- SAT（入力：論理式 f）
  f の全部の項を1にする変数割り当てがある？
  YES? NO?
- 最小頂点被覆問題（入力：グラフGと整数k）
  k個の頂点を選んで、Gのすべての枝をカバーできる？
  YES? NO?
- 最短経路問題（入力：重み付グラフG、2頂点sとt、整数k）
  sからtまで、長さk以下のルートがある？
  YES? NO?
- 最小全域木問題（入力：重み付グラフG、整数k）
  Gから枝を選んで、重みの合計がk以下の木を作れる？
  YES? NO?

NP: 答が与えられれば、それが正しい答であるかを多項式時間で検証できる問題の集合。
もう少し厳密には、「答がYesである証拠」が与えられたら、確かにYesであることを多項式時間で検証できる。

- SAT
  変数への0/1割り当てが与えられたら、それが全ての項を充足するか否かを、多項式時間で検証できる。
- 最小頂点被覆問題
  頂点集合が与えられたら、それがk個以下で、かつ、正しい被覆になっているか否かを、多項式時間で検証できる。
- 最短経路問題
  sからtへのルートが与えられたら、その長さがk以下であるか否かを、多項式時間で検証できる。
- 最小全域木問題
  木が与えられたら、それが全域木で、その重みがk以下であるか否かを、多項式時間で検証できる。

つまり、これら4つの問題は、全てNPに入る。

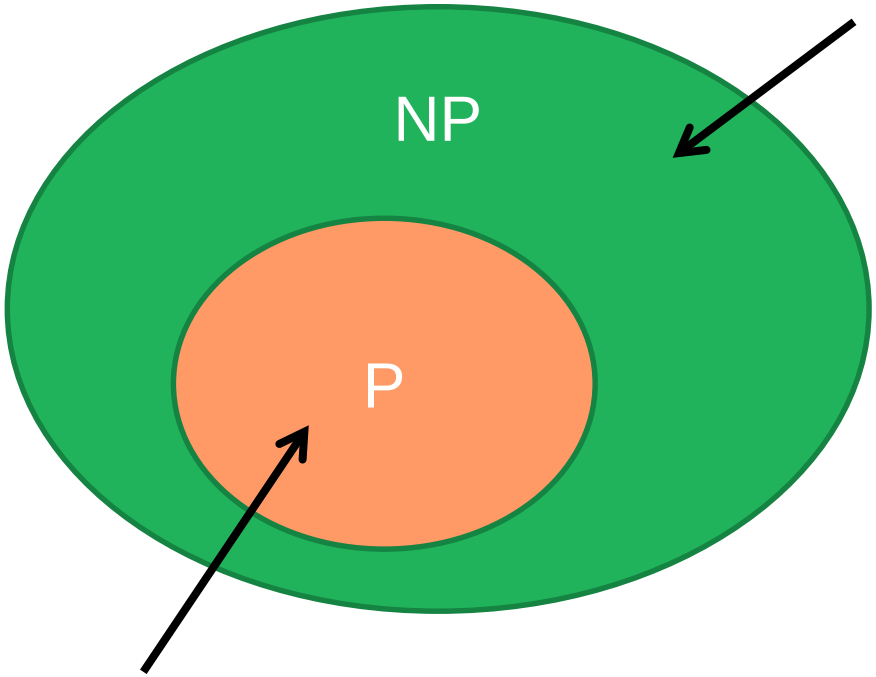
P:「答がYesである証拠」が与えられなくとも、Yesであることを（自力で）多項式時間で検証できる。

- 最短経路問題
  ダイクストラのアルゴリズムで最適解を求めて、それをkと比較すれば、多項式時間でYESかNOか分かる。
- 最小全域木問題
  クラスカルのアルゴリズムで最適解を求めて、それをkと比較すれば、多項式時間でYESかNOか分かる。

つまり、これら2つの問題は、Pに入る。

SATやVCがPに入るかどうかは不明

PはNPの部分集合



答を多項式時間で検証できる問題の集合

- SAT
- 最小頂点被覆問題

など

多項式時間アルゴリズムを持つ問題の集合（易しい問題の集合）

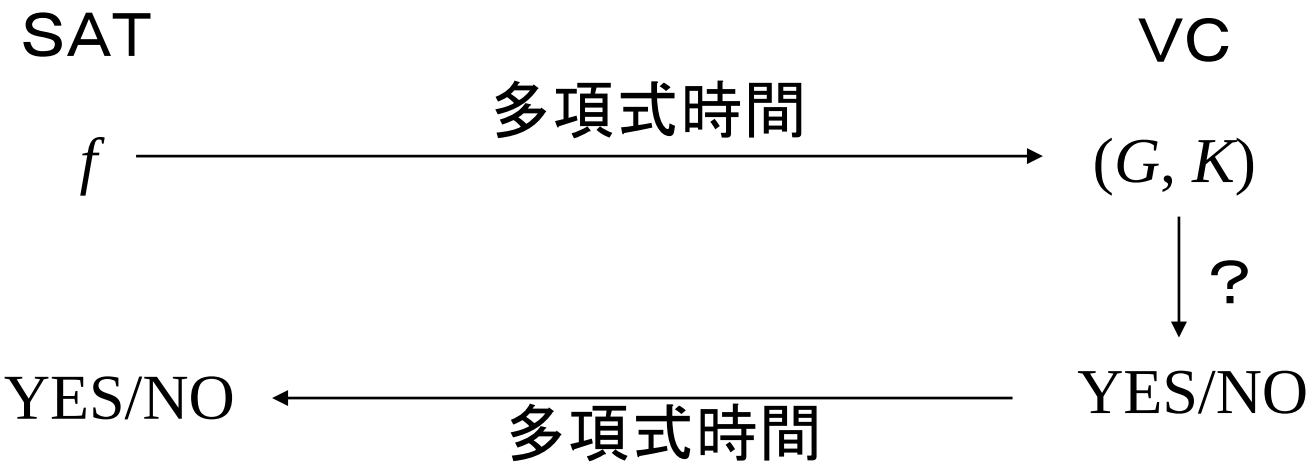
- 最短経路問題
- 最小全域木問題

など

P=NPかP≠NPかは
まだ分かっていない。
（多くの人がP≠NPだろうと
思っている。）

NP完全性は、P vs NP問題を解く有力な手掛かり

SATからVCへのリダクション（前出）

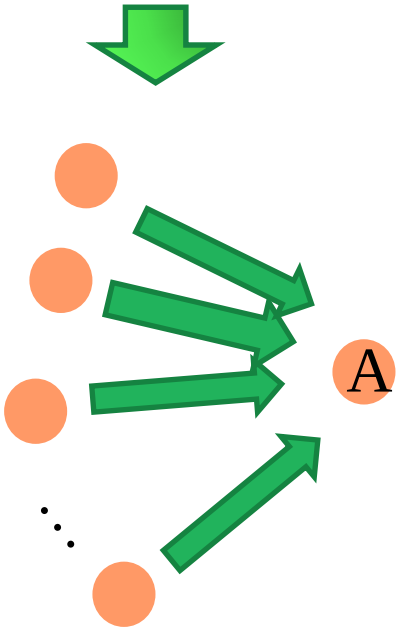


これは、VCを解くアルゴリズムをサブルーチン（部品）として使って、SATを解くアルゴリズムを構築していると解釈できる。

「？」の部分が多項式時間⇒全体が多項式時間

VCがPに入る⇒SATがPに入る

問題AがNP完全であるとは、
(1) A自身がNPに入る。
(2) NPのどの問題からもAにリダクション出来る。

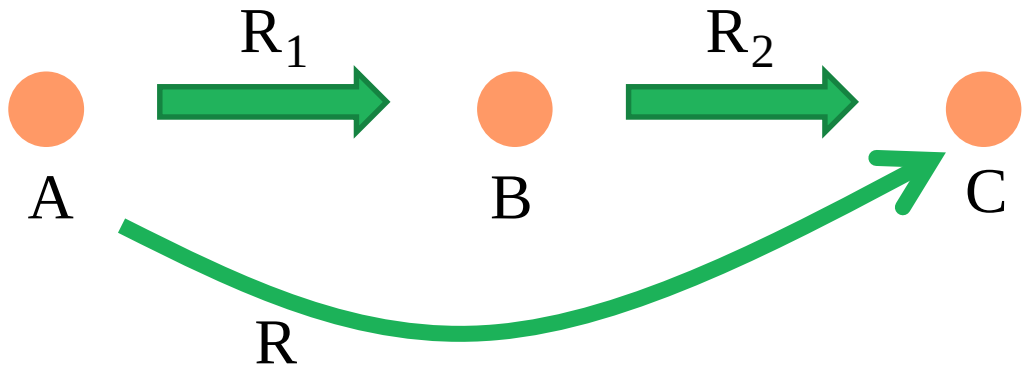


※世の中のたくさんの（何千、何万という）実用的な問題が、
NP完全問題であるということが知られている。

定義の意味は、後のスライドで。

(1)を示すのは難しくないが、(2)は難しそうに思える。
（NPの問題は無数にあるから。）

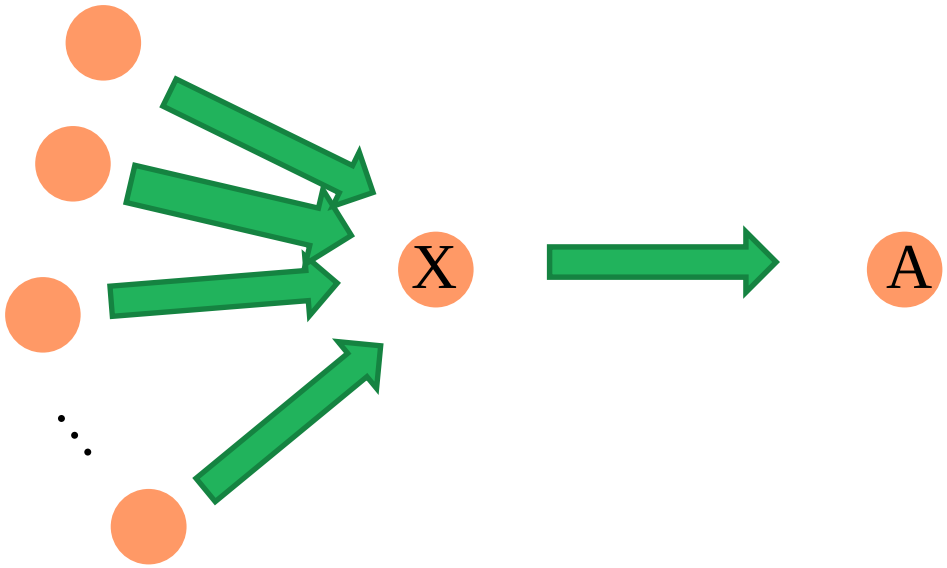
しかし、リダクションは推移的である。すなわち、
AからBにリダクション出来て、BからCにリダクション出来れば、
AからCにリダクション出来たことになる。



R_1 :Aに対する入力IからBに対する入力I'を作る
R_2 :Bに対する入力I'からCに対する入力I''を作る

R: R_1とR_2を使って、Aに対する入力IからCに対する入力I''を作る。

ある問題XがNP完全だということを示せたとしよう。



すると、問題Aに対して(2)を示すには、XからAへのリダクションを見つけさえすれば良い。
なぜなら、前ページの議論より、全ての問題から（Xを経由して）Aにリダクションできることになるから。

最初にNP完全だということが示されたのがSAT。
それ以降は、SATから芋づる式にリダクションして得られた。

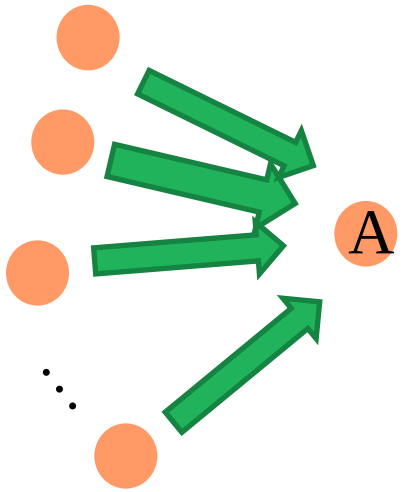
なぜNP完全性が、P vs NP問題に関係している？

好きなNP完全問題Aを選ぶ。

- AがPに入らないことを示せば、P≠NPを示せたことになる。
  AはNPに入るのにPに入らないから。

- AがPに入ることを示せば、P=NPを示したことになる。
  リダクションの性質から、NP問題全てがPに入る。

つまり、NPは無数の問題を含んでいるが、
P vs NP問題を考える上では、何でも良いから
ある1つのNP完全問題が多項式時間で
解けるか否かに集中すれば良い。



3彩色問題のNP完全性の証明

(1) NPに入ることは明らか。与えられた3彩色に対して、
どの枝を見ても、その両端の色が異なっていることを
チェックするのは、多項式時間で出来る。

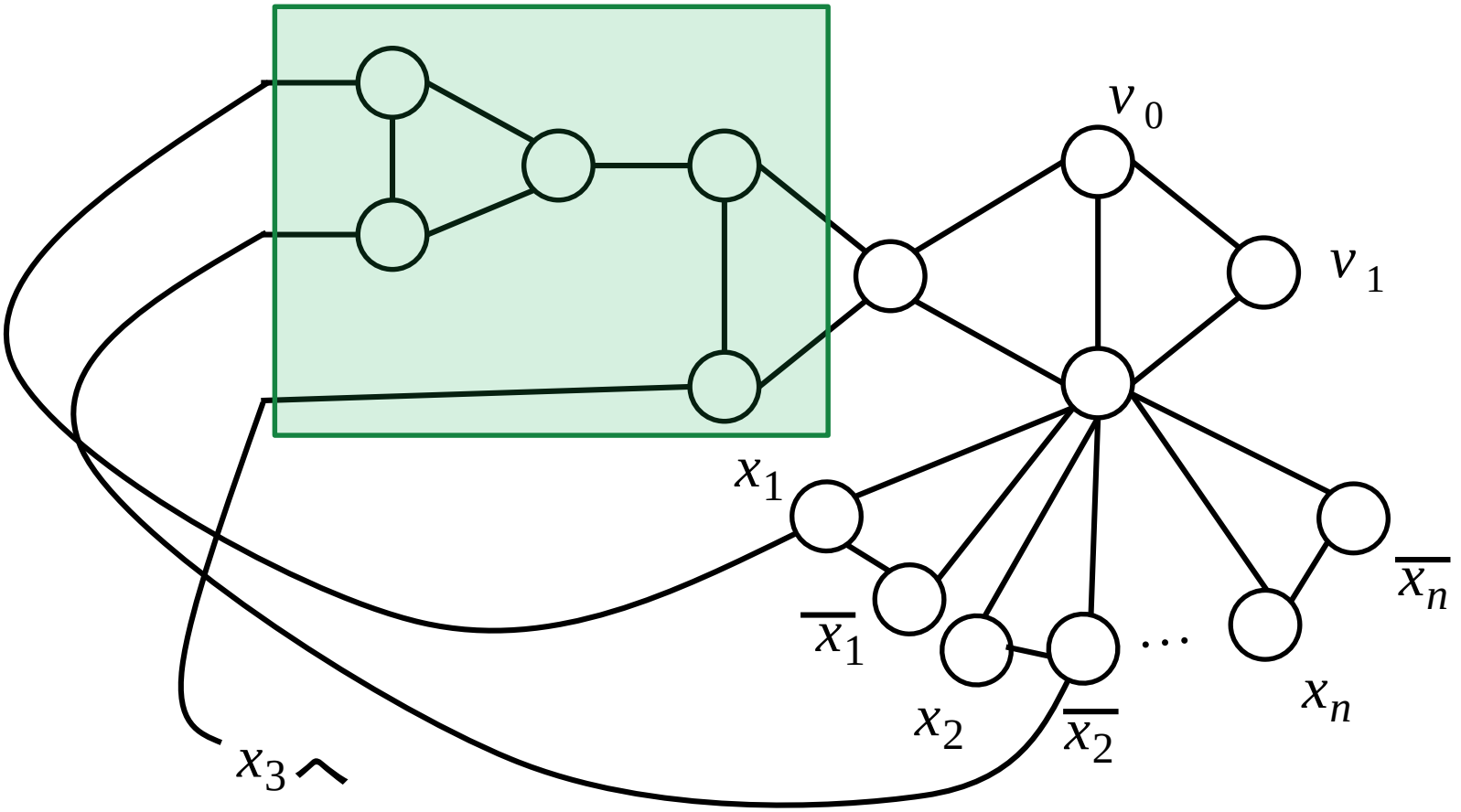
(2) 任意のNP問題からリダクション出来ることについて。
既にNP完全であることが分かっている3SATから
リダクションする。

SATの特殊なもの。各項に現れるリテラルの
数がちょうど3のもの。

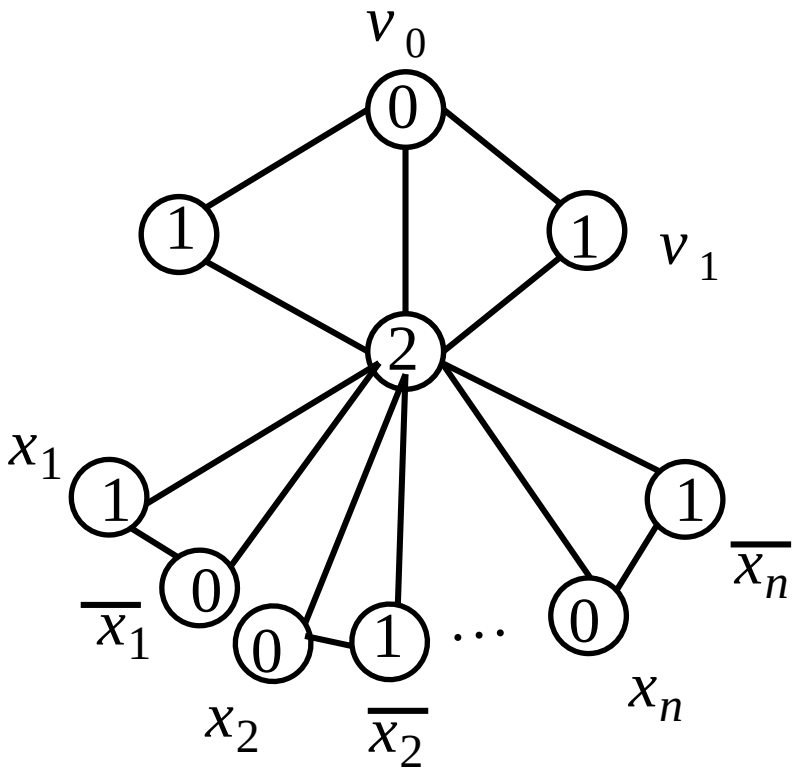
$$f = (x_1 + \overline{x_2} + x_3)(x_1 + x_5 + x_2)(\overline{x_3} + \overline{x_4} + x_6) \ldots (x_8 + x_9 + \overline{x_{12}})$$

3個　3個　3個　3個

全ての項に対して同じように作る



$$f = (x_1 + \overline{x_2} + x_3)\,(x_1 + x_5 + x_2)\,(\overline{x_3} + \overline{x_4} + x_6)\,\ldots\,(x_8 + x_9 + \overline{x_{12}})$$



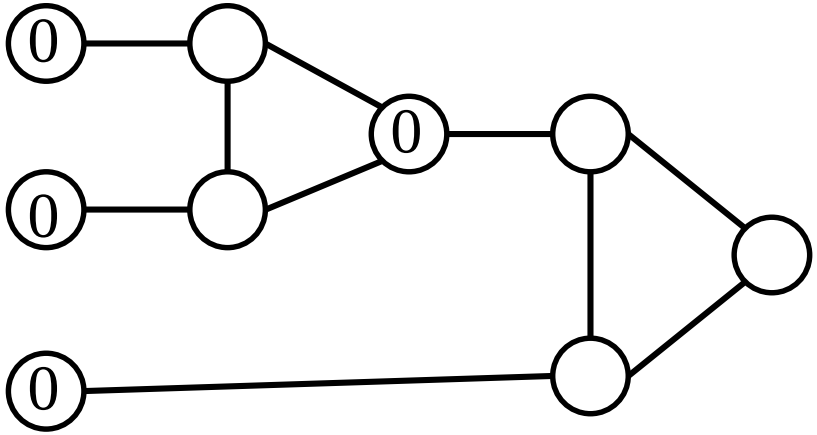
x_i や $\overline{x_i}$ は0か1でしか塗れない

x_i を0、$\overline{x_i}$ を1か、またはその逆しかない。

前者を x_i =0、後者を x_i =1と解釈する。

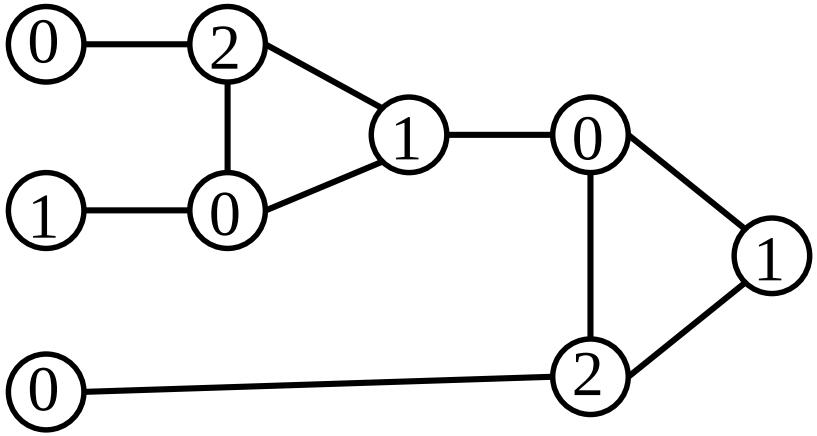
項に対応する部分

変数部分が全部0ならば



ここには0を使わざるを
得ない

変数部分に1つでも1があれば



ここを1で塗れる
（他の6つの場合も
正しいことを確かめよ）

3SATの入力 f がYES。

→ 全部の項を1とする変数割り当てに従って、変数頂点を0と1で塗る。

→ それぞれの項に対応する部分グラフを0, 1, 2で塗る。
（どの項も、1となる変数が少なくとも1つあるので、
正しく塗れる。）

→ 3彩色問題の入力 G がYES。

3彩色問題の入力 G がYES。

→ 全ての頂点を0, 1, 2で正しく塗れる。

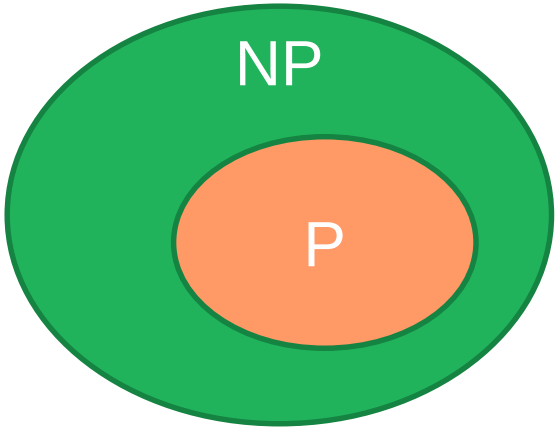
→ v_0 が0、v_1 が1になるように、色を入れ替える。

→ 各項に対応する部分グラフの一番左にある3頂点のうち、
少なくとも1つは色1で塗られている。

→ 変数頂点の色（0または1）に従って、f の変数の0/1割り当てを決めると、各項少なくとも1つは1となる変数を含む。

→ 3SATの入力 f がYES。

有名な未解決問題
（21世紀の7大未解決問題の1つ）



P≠NP予想

- 「NPとPは異なる、つまり、NPに入るがPに入らない問題が存在する」という予想。
- 「NP完全問題に多項式時間アルゴリズムが存在しない」と言っているのとも等価。
- 「与えられた答が正しい答であるかどうかの検証は簡単だが、自分で答を見つけるのは難しい問題は存在する。」という意味も持つ。

ウィキペディア
フリー百科事典

メインページ
コミュニティ・ポータル

ミレニアム懸賞問題

ミレニアム懸賞問題(ミレニアムけんしょうもんだい、英: **millennium prize problems**)とは、アメリカのクレイ数学研究所によって2000年に発表された100万ドルの懸賞金がかけられている7つの数学上の未解決問題のことである。**ミレニアム賞問題**、**ミレニアム問題**とも呼ばれる。

他言語版
العربية
Azərbaycanca
Català
Čeština
Deutsch
English
Español
فارسی
Français
עברית
Magyar
Bahasa Indonesia
Italiano
한국어
ລາວ
മലയാളം
Монгол
Nederlands
Norsk bokmål
Polski
Português

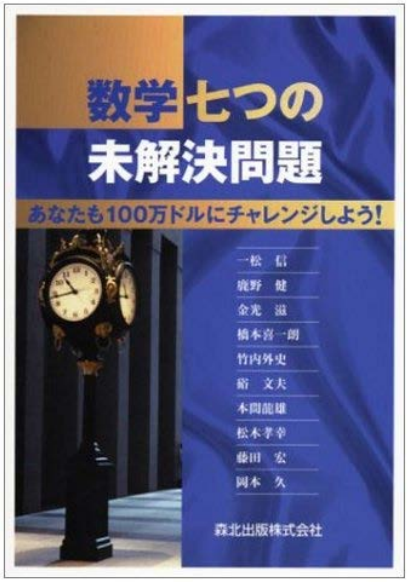
と認められ、2010年3月18日にクレイ数学研究所はペレルマンの受賞を発表した[4]。ただし、ペレルマンはこの受賞を拒否しており、彼に与えられる賞金100万ドルは数学界へ貢献するかたちで使われることになると発表している。

一覧 [編集]

1. ヤン-ミルズ方程式と質量ギャップ問題 (Yang–Mills and Mass Gap)
2. リーマン予想 (Riemann Hypothesis)
3. P≠NP予想 (P vs NP Problem)
4. ナビエ-ストークス方程式の解の存在と滑らかさ (Navier–Stokes Equation)
5. ホッジ予想 (Hodge Conjecture)
6. **ポアンカレ予想** (Poincaré Conjecture) – グリゴリー・ペレルマンにより解決済。
7. バーチ・スウィンナートン＝ダイアー予想(BSD予想) (Birch and Swinnerton-Dyer Conjecture)

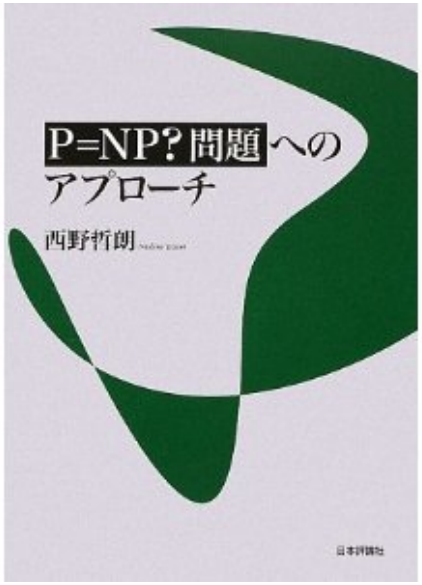
脚注 [編集]

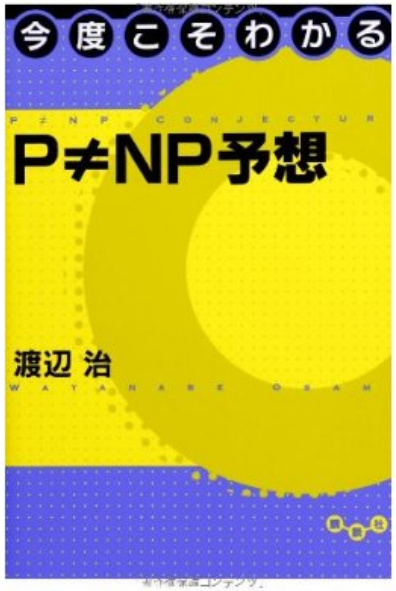
1. ^ *a* *b* Jaffe 2006
2. ^ Jaffe & Wiles (2006) の Rules for the Millennium Prizes の章を参照
3. ^ CMI 2013
4. ^ Carlson 2010

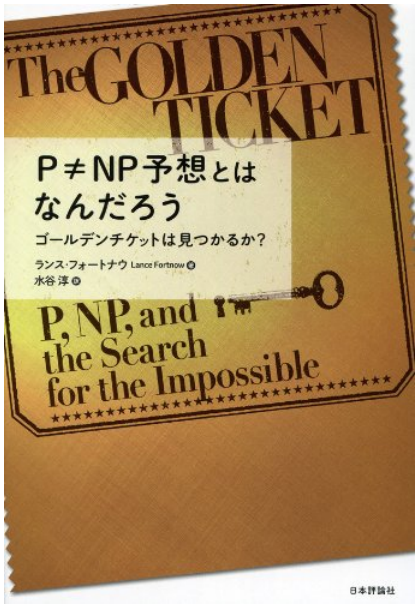

数学七つの
未解決問題
あなたも100万ドルにチャレンジしよう!
一松 信
鹿野 健
金光 滋
橋本喜一朗
竹内外史
前 文夫
本間龍雄
松本幸幸
藤田 宏
岡本 久
森北出版株式会社


BLUE BACKS
数学21世紀の
7大難問
数学の未来をのぞいてみよう
中村 亨


ミレニアム賞問題
7つの未解決問題はどうなったか?
ミレニアム賞問題10周年
ミレニアム賞問題
ポアンカレ予想完全解決
ミレニアム賞問題の現在


P=NP? 問題への
アプローチ
西野哲朗
日本評論社


今度こそわかる
P≠NP予想
渡辺 治


The GOLDEN TICKET
P≠NP予想とは
なんだろう
ゴールデンチケットは見つかるか?
ランス・フォートナウ Lance Fortnow
水谷 淳
P, NP, and the Search for the Impossible
日本評論社