離散最適化基礎論　第 3 回
充足可能性問題とその変換

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2019 年 10 月 29 日

最終更新：2019 年 10 月 31 日　00:52

スケジュール 前半 (予定)

- ⋆ 休み (大学院入学式) (10/1)
- 1 アルゴリズム的問題解決と計算複雑性 (10/8)
- 2 速習 P vs NP 問題 (10/15)
- ⋆ 休み (祝日) (10/22)
- 3 充足可能性問題とその変換 (10/29)
- 4 グラフに関する問題 (1)：部分集合の選択 (11/5)
- 5 グラフに関する問題 (2)：経路の選択 (11/12)
- 6 集合族に関する問題 (1)：グラフとの関連 (11/19)
- 7 集合族に関する問題 (2)：発展 (11/26)

注意：予定の変更もありうる

スケジュール 後半 (予定)

- 8 数値が関わる問題 (1)：2 分割問題 (12/3)
- ⋆ 出張のため休講？ (12/10)
- 9 数値が関わる問題 (2)：3 分割問題 (12/17)
- ⋆ 冬期休業 (12/24, 31)
- 10 平面性が関わる問題 (1/7)
- 11 計算幾何学に関する問題 (1/14)
- 12 文字列に関する問題 (1/21)
- 13 アルゴリズム的問題解決：再考 (1/28)
- 14 予備 (2/4)
- ⋆ 祝日のため休み (2/11)

注意：予定の変更もありうる

今日の目標

今日の目標

- 次の基本概念を理解する
  - 充足可能性問題 (SAT)，k 充足可能性問題 (k-SAT)
  - 連言標準形，選言標準形
- 多項式時間多対一帰着を用いて，
  次の問題が NP 完全であることを証明する
  - 3 充足可能性問題 (3-SAT)

目次

(2) 計画をたてること

アルゴリズム的問題解決における「計画をたてること」

1. アルゴリズム設計の方針を立てること
2. アルゴリズムを設計すること

アルゴリズム設計の方針 とは？

- 我々が目標とする「実行時間」と「メモリ消費量」の特定
- 我々が達成できる「実行時間」と「メモリ消費量」の特定
  ← この講義の内容

特に「多項式時間で解けること」が達成できるかどうか？

クラス P

クラス P とは？

クラス P とは，「多項式時間アルゴリズムで解ける判定問題」全体の集合

注意

- 「P」は「Polynomial」の頭文字
- クラス P というときは，判定問題しか対象にしていない

クラス P に所属する判定問題の例

- 入力：有理数 $a\ (\neq 0), b, c$
- 出力：Yes か No
- 条件：二次方程式 $ax^2 + bx + c = 0$ が実数解を持てば Yes，
  そうでなければ No

多項式時間アルゴリズム：

- $b^2 - 4ac \geq 0$ ならば Yes を出力，
  そうでなければ No を出力

クラス NP

クラス NP とは？

クラス NP とは，次の性質を満たす判定問題全体の集合

入力に対する正しい出力が Yes であるとき，
それを多項式時間で検証するための証拠が存在する

注意

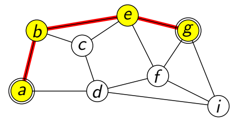
- 「NP」は「Non-deterministic Polynomial」の頭文字
- クラス NP というときは，判定問題しか対象にしていない
- クラス NP では，「入力に対する正しい出力が No であるとき」を考えない
- 「正しい出力が Yes である入力」のことを短く
  「Yes 入力」，「Yes 問題例」(Yes-instance) と呼ぶことがある

証拠 = certificate, witness

クラス NP に所属する判定問題の例 (1)

s, t 連結性判定問題

- 入力：無向グラフ $G = (V, E)$，2 頂点 $s, t \in V$
- 出力：Yes か No
- 条件：s から t へ至る経路が G にあれば，Yes，
  s から t へ至る経路が G になければ，No



Yes 入力を検証する多項式時間アルゴリズム

- Yes 入力の証拠：s から t へ至る経路 P

1. P が s から t へ至る経路であれば，Yes 入力として検証終了

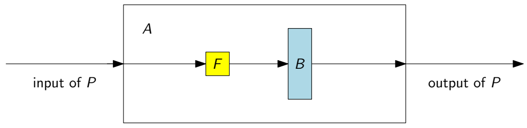
多項式時間多対一帰着

多項式時間多対一帰着 とは？

判定問題 P が判定問題 Q に多項式時間多対一帰着可能であるとは
次を満たす多項式時間アルゴリズム F が存在すること

- F は，「P の入力」を入力として，「Q の入力」を出力する
- P の入力 I が P の Yes 入力 $\Rightarrow$ $F(I)$ は Q の Yes 入力
- P の入力 I が P の No 入力 $\Rightarrow$ $F(I)$ は Q の No 入力

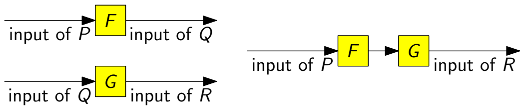
このような F を多項式時間多対一帰着と呼ぶことがある



帰着の性質

性質 (推移性)

- 判定問題 P が判定問題 Q に多項式時間多対一帰着可能，かつ，
  判定問題 Q が判定問題 R に多項式時間多対一帰着可能 $\Rightarrow$
  判定問題 P は判定問題 R に多項式時間多対一帰着可能



帰着は何に使えるか？

問題 P が問題 Q に帰着できるとき…

帰着の使い方 (1)

問題 Q を解くアルゴリズムがあれば，問題 P を解ける

つまり，アルゴリズム設計のためのツールとして使える

帰着の使い方 (2)

問題 P が解けないと分かっていたら，問題 Q が解けないことも分かる

つまり，問題の複雑性を分析するためのツールとして使える

直感的な言い方 (不正確)

問題 P が問題 Q に帰着可能 $\Rightarrow$ P は Q より難しくない
(P の方が Q より簡単)

NP 完全問題

NP 完全問題 とは？

NP に所属する判定問題 Q が NP 完全 (NP-complete) であるとは，
クラス NP に所属する任意の判定問題 P が
Q に多項式時間多対一帰着可能であること

注意

- 別の帰着を用いる場合もあり，そのときは，
  「〇〇帰着に関して NP 完全である」と定義する
- 英単語「NP-complete」は形容詞 (問題クラスを指す用語ではない)
  - おそらく，日本語としての単語「NP 完全だ」は形容動詞

重要な事実 (Cook '71, Levin '73)

NP 完全問題は存在する (実際に構成できる)

今回：この事実についての説明 $\rightsquigarrow$ 充足可能性問題

P vs NP 問題と NP 完全問題

性質

ある NP 完全問題が多項式時間で解ける $\Rightarrow$ P = NP

直感

NP 完全問題はクラス NP の中で最も難しい問題

未解決問題

NP 完全問題は多項式時間で解けるか？

性質の対偶

P $\neq$ NP $\Rightarrow$ どの NP 完全問題も多項式時間で解けない

アルゴリズム的問題解決との関係

(2) 計画をたてること

アルゴリズム的問題解決における「計画をたてること」

1. アルゴリズム設計の方針を立てること
2. アルゴリズムを設計すること

アルゴリズム設計の方針 とは？ (再)

- 我々が目標とする「実行時間」と「メモリ消費量」の特定
- 我々が達成できる「実行時間」と「メモリ消費量」の特定
  $\leftarrow$ この講義の内容

NP 完全性の使い方

対象とする問題が NP 完全問題であると分かれば
P $\neq$ NP という仮定の下で， (←落とせない仮定)
「多項式時間で解くこと」が達成できないと分かる

目次

論理関数

充足可能性問題は，論理関数を入力とする問題

定義：論理関数とは？

f が論理関数 (boolean function) であるとは，
$\{0,1\}^n$ から $\{0,1\}$ への関数のこと　　(ただし，$n \geq 1$ は整数)

つまり，$f\colon \{0,1\}^n \to \{0,1\}$

例：$n = 3$，$f\colon \{0,1\}^3 \to \{0,1\}$

$$f(0,0,0) = 1,\quad f(0,0,1) = 0,\quad f(0,1,0) = 0,\quad f(0,1,1) = 1,$$
$$f(1,0,0) = 0,\quad f(1,0,1) = 1,\quad f(1,1,0) = 1,\quad f(1,1,1) = 0$$

論理関数

充足可能性問題は，論理関数を入力とする問題

定義：論理関数とは？

f が論理関数 (boolean function) であるとは，
$\{0,1\}^n$ から $\{0,1\}$ への関数のこと　　(ただし，$n \geq 1$ は整数)

つまり，$f\colon \{0,1\}^n \to \{0,1\}$

例：

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

充足可能性：用語

論理関数 $f\colon \{0,1\}^n \to \{0,1\}$

用語の定義

- 割当：　$a \in \{0,1\}^n$
- 充足割当：　$f(a) = 1$ となるような割当 $a \in \{0,1\}^n$
- 充足可能関数：　$f(a) = 1$ となるような a がある f
- 充足不可能関数：$f(a) = 1$ となるような a がない f

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- $a\colon x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0$ は f の充足割当
- $\therefore$ 論理関数 f は充足可能

充足可能性問題

定義：充足可能性問題 (satisfiability problem, SAT) とは？

- 入力：整数 $n \geq 1$，論理関数 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$\rightsquigarrow$ Yes

重要な注意

論理関数 f をどのように入力するのか？
ちゃんと決めないといけない

充足可能性問題：入力を表として与えたら？

定義：充足可能性問題 (satisfiability problem, SAT) とは？

- 入力：整数 $n \geq 1$，論理関数 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

論理関数を左のような「表」で入力すると

- 表の右端の列を見ると，
  正しい出力が分かる
  (←アルゴリズム)
- $\rightsquigarrow$ 多項式時間アルゴリズム
  (注：入力のサイズ $= (n+1)2^n$)

充足可能性問題：よくある入力

定義：充足可能性問題 (satisfiability problem, SAT) とは？

- 入力：整数 $n \geq 1$，論理関数 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

論理関数の入力法としてよくある形式

- 論理回路 (boolean circuit)
- 論理式 (boolean formula)
  - 連言標準形 (conjunctive normal form) の論理式
  - 選言標準形 (disjunctive normal form) の論理式

ここでは，論理式だけを扱う．

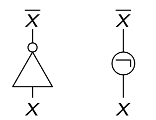
- 論理式では論理演算を用いる (3 種類の演算：否定，連言，選言)

論理演算：否定

論理変数の否定 (negation, NOT)

論理変数 x の否定 $\overline{x}$ は次で定義される

x	$\overline{x}$
0	1
1	0

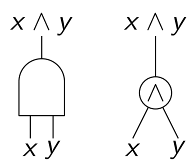


論理演算：連言

論理変数の連言 (論理積，conjunction, AND)

2 つの論理変数 x, y の連言 $x \wedge y$ は次で定義される

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

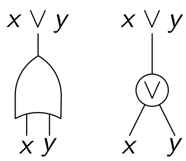


論理演算：選言

論理変数の選言 (論理和，disjunction, OR)

2 つの論理変数 x, y の選言 $x \vee y$ は次で定義される

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



論理式

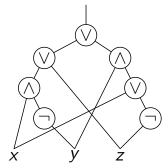
論理式は，論理変数と論理演算を組み合わせてできるもの

論理式とは？

- 論理変数 x は論理式
- 論理式 f, g に対して，$\overline{f}, f \wedge g, f \vee g$ も論理式
- 上の 2 つの規則で作られるもの以外は論理式ではない

否定，連言，選言以外の論理演算を考える (許す) 場合もある

例：$((x \wedge \overline{y}) \vee z) \vee ((x \vee \overline{z}) \wedge y)$

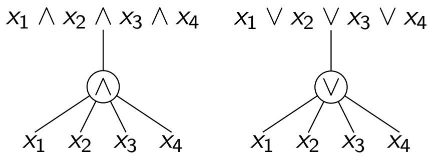


3 つ以上の変数の連言と選言

表記法のショートカット (略記) として，次を使うことが多い

3 つ以上の連言と選言

$$x_1 \wedge x_2 \wedge x_3 \wedge x_4 = (((x_1 \wedge x_2) \wedge x_3) \wedge x_4)$$
$$x_1 \vee x_2 \vee x_3 \vee x_4 = (((x_1 \vee x_2) \vee x_3) \vee x_4) \qquad \text{など}$$



連言標準形の論理式

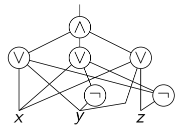
連言標準形 (乗法標準形，和積標準形) とは？

論理式が連言標準形 (conjunctive normal form, CNF) にあるとは，
それが「リテラルの選言の連言」として表されること

リテラル ＝ 「論理変数」 と 「論理変数の否定」 のこと

- 連言標準形の論理式を CNF 論理式と呼ぶことがある
- 事実：任意の論理関数は CNF 論理式として表現できる

$\boxed{例}$：$(x \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee \overline{z}) \wedge (x \vee y \vee z)$



節とは？

「リテラルの選言」のことを節 (clause) と呼ぶ

選言標準形の論理式

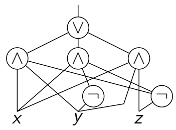
連言標準形 (加法標準形，積和標準形) とは？

論理式が選言標準形 (disjunctive normal form, DNF) にあるとは，
それが「リテラルの連言の選言」として表されること

リテラル ＝ 「論理変数」 と 「論理変数の否定」 のこと

- 選言標準形の論理式を DNF 論理式と呼ぶことがある
- 事実：任意の論理関数は DNF 論理式で表現できる

$\boxed{例}$：$(x \wedge y \wedge \overline{z}) \vee (x \wedge \overline{y} \wedge \overline{z}) \vee (x \wedge y \wedge z)$



項とは？

「リテラルの連言」のことを項 (term) と呼ぶ

CNF 充足可能性問題 と DNF 充足可能性問題

定義：CNF 充足可能性問題 (CNF-SAT) とは？

- 入力：整数 $n \geq 1$，CNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

定義：DNF 充足可能性問題 (DNF-SAT) とは？

- 入力：整数 $n \geq 1$，DNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

重要な事実

どちらかは多項式時間で解け，どちらかは NP 完全

CNF-SAT は NP 完全

定理 (Cook '71, Levin '73)

CNF 充足可能性問題は NP 完全

この講義で，証明はしない

帰結

- $P \in$ NP $\Rightarrow$ P は CNF 充足可能性問題に多項式時間で帰着される
- CNF 充足可能性問題が多項式時間で解ける $\Rightarrow$ P ＝ NP

- CNF 充足可能性問題を解くためのソルバーが多く開発されている

問題解決に対する帰結

解きたい問題が NP に所属する $\Rightarrow$
CNF 充足可能性問題のソルバーを使って解ける (かもしれない)

$\rightsquigarrow$ SAT 符号化，SAT 定式化

DNF-SAT は多項式時間で解ける

事実 (容易に分かる)

DNF 充足可能性問題は多項式時間で解ける

$\boxed{例}$：$(x \wedge y \wedge \overline{z}) \vee (x \wedge \overline{y} \wedge \overline{z}) \vee (x \wedge \overline{x} \wedge z)$

充足割当：$x \mapsto 1, y \mapsto 1, z \mapsto 0$ など

アルゴリズム

1. 各項を順に見る
2. 項の中に，ある変数とその否定が両方含まれていれば，continue
   そうでなければ，Yes を出力して終了
3. 最終的に，全ての項を見終わり，Yes を出力してなければ，
   No を出力して終了

疑問 ???

ここまで述べてきた事実をまとめると…

- CNF 充足可能性問題は NP 完全
  - CNF 充足可能性問題が多項式時間で解ける $\Rightarrow$ P $=$ NP
- DNF 充足可能性問題は多項式時間で解ける
- 任意の論理関数は CNF 論理式として表現できる
- 任意の論理関数は DNF 論理式として表現できる

疑問？

これは「P $=$ NP」を証明してることにならないか？

CNF 充足可能性問題を解くアルゴリズム

1. 入力の CNF 論理式を，DNF 論理式として表現する
2. その DNF 論理式を入力として，DNF 充足可能性問題を解く
3. その出力を そのまま出力する

CNF 論理式を DNF 論理式として表現する

例

CNF 論理式 $(x_1 \vee y_1) \wedge (x_2 \vee y_2)$ 節の数 $= 2$

$\downarrow$

DNF 論理式 $(x_1 \wedge x_2) \vee (x_1 \wedge y_2) \vee (y_1 \wedge x_2) \vee (y_1 \wedge y_2)$ 項の数 $= 4$

この例を一般化すると，次のような CNF 論理式が構成できる

CNF 論理式 変数の数 $= 2n$ 節の数 $= n$

$\downarrow$

DNF 論理式 変数の数 $= 2n$ 項の数 $= 2^n$

結論

前のページのアルゴリズムは多項式時間アルゴリズムではない

補足：CNF 充足可能性問題が NP に所属すること

次の命題は簡単なので，証明する

命題

CNF 充足可能性問題はクラス NP に所属する

今一度，クラス NP の定義を復習

クラス NP とは？

クラス NP とは，次の性質を満たす判定問題全体の集合

入力に対する正しい出力が Yes であるとき，
それを多項式時間で検証するための証拠が存在する

つまり，多項式時間で検証できる証拠が存在することを言えばよい

補足：CNF 充足可能性問題が NP に所属すること — 証明

定義：CNF 充足可能性問題 (CNF-SAT) とは？

- 入力：整数 $n \geq 1$，CNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：f が充足可能であれば，Yes
  そうでなければ，No

Yes 入力の証拠：$f(a) = 1$ となる $a \in \{0,1\}^n$

多項式時間検証アルゴリズム

- 入力：整数 $n \geq 1$，CNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$，$a \in \{0,1\}^n$

1. $f(a) = 1$ となることを代入して確かめ，$f(a) = 1$ となれば検証成功

Yes 入力の例：$(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (x \vee y \vee z)$
$\rightsquigarrow$ 証拠 $a\colon x \mapsto 0, y \mapsto 1, z \mapsto 0$

目次

ここまでのまとめ と ここからの流れ

ここまでのまとめ

- NP 完全性の定義 と NP 完全問題の存在 (CNF-SAT)

- 疑問：NP 完全問題は稀にしか存在しないのか？
- 回答：NP 完全問題は数多に存在！ 重要な多くの問題も NP 完全！

ここからの流れ

- 問題が NP 完全であることの証明法 の習得

今日の残り：3 充足可能性問題が NP 完全であることの証明

k 連言標準形 (k-CNF)

自然数 $k \geq 1$

定義：k 連言標準形 (k 乗法標準形，k 和積標準形)

論理式が k 連言標準形 (k-CNF) にあるとは，
それが連言標準形にあり，各節が含むリテラルの数が k 以下であること

2-CNF の例：$(x \vee \bar{y}) \wedge (y \vee z) \wedge (\bar{y} \vee \bar{z})$
3-CNF の例：$(x \vee y \vee \bar{z}) \wedge (x \vee z \vee w) \wedge (\bar{x} \wedge \bar{y})$

注：「リテラルの数が k 以下」ではなく「リテラルの数がちょうど k」とする流儀もある

k 充足可能性問題 (k-CNF 充足可能性問題)

自然数 $k \geq 1$

定義：k 充足可能性問題 (k-CNF 充足可能性問題，k-SAT) とは？

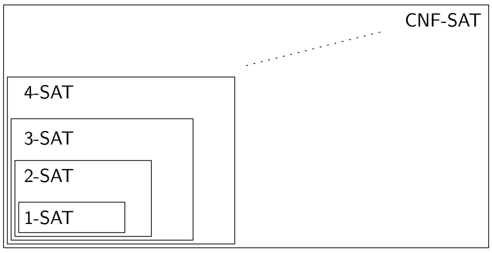
- 入力：整数 $n \geq 1$，k-CNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$
- 出力：Yes または No
- 条件：$f(a) = 1$ となる $a \in \{0,1\}^n$ が存在すれば，Yes
  そうでなければ，No

注意：k は入力の一部ではない (k の値によって，異なる問題ができる)

- $k = 1$ のとき $\rightsquigarrow$ 1-SAT
- $k = 2$ のとき $\rightsquigarrow$ 2-SAT
- $k = 3$ のとき $\rightsquigarrow$ 3-SAT
- ……

k-SAT と CNF-SAT の関係

- 各 $k \geq 1$ に対して，k-SAT の入力は，CNF-SAT の入力
- 各 $k \geq 1$ に対して，k-SAT の入力は，$(k+1)$-SAT の入力



帰結

各 $k \geq 1$ に対して，k-SAT は NP に所属する

3-SAT は NP 完全

定理 (Karp '72)

3-SAT は NP 完全

証明の流れ

1. 3-SAT が NP に所属することを証明する (済)
2. CNF-SAT が 3-SAT に多項式時間多対一帰着可能であることを証明する

なぜ，これが NP 完全性の証明になるのか？

- NP に所属するすべての問題は，CNF-SAT に多項式時間多対一帰着可能 (CNF-SAT の NP 完全性と，NP 完全性の定義より)
- CNF-SAT は，3-SAT に多項式時間多対一帰着可能 (上の 2 より)
- 多項式時間多対一帰着可能性は推移性を持つ (前回)
- $\therefore$ NP に所属するすべての問題は，3-SAT に多項式時間多対一帰着可能

3-SAT は NP 完全：証明 (1)

CNF-SAT の入力して与えられた
整数 $n \geq 1$ と CNF 論理式 $f\colon \{0,1\}^n \to \{0,1\}$ を考える

行うこと

f から 3-CNF 論理式 f' を構成する

- f の変数は $x_1, x_2, \ldots, x_n$ であるとする
- f のすべての節に含まれるリテラルの数が 3 以下 $\Rightarrow$ f' は 3-CNF 論理式 ($\because f = f'$ として構成終了)
- そうでないとき，含むリテラルの数が 4 以上の節が f に存在
- それを C として，C に含まれるリテラルの 2 つを ℓ, ℓ' とする
- つまり，ある節 C' を用いて，$C = C' \vee \ell \vee \ell'$ と書ける

$$C = \underbrace{x_1 \vee \overline{x_2} \vee x_3}_{=C'} \vee \underbrace{x_4}_{=\ell} \vee \underbrace{\overline{x_5}}_{=\ell'}$$

3-SAT は NP 完全：証明 (2)

$C = C' \vee \ell \vee \ell'$

- C を以下の CNF 論理式に置き換える (y は新しい論理変数)

$$(C' \vee y) \wedge (\overline{y} \vee \ell \vee \ell')$$

- このとき,「C のリテラル数」 $>$ 「$C' \vee y$ のリテラル数」

この操作を繰り返すと，最終的に，3-CNF 論理式 f' が得られる

例

$$\begin{aligned}
f &= (x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee y_2) \wedge (\overline{y_2} \wedge \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee y_2) \wedge (\overline{y_2} \wedge \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee y_3)\\
&\quad \wedge (\overline{y_3} \vee x_4 \vee x_5) = f'
\end{aligned}$$

f から f' を構成するのは，多項式時間でできる (確認せよ)

3-SAT は NP 完全：証明 (3)

これが多対一帰着であることを言うには，次を証明しないといけない

証明すべきこと

1. f が CNF-SAT の Yes 入力 $\Rightarrow$ f' は 3-SAT の Yes 入力
2. f が CNF-SAT の No 入力 $\Rightarrow$ f' は 3-SAT の No 入力

上の 2 は直接証明しにくいので，代わりに次を証明する

証明すべきこと (言い換え)

1. f が CNF-SAT の Yes 入力 $\Rightarrow$ f' は 3-SAT の Yes 入力
2. f' が 3-SAT の Yes 入力 $\Rightarrow$ f は CNF-SAT の Yes 入力

つまり，

f が CNF-SAT の Yes 入力 $\Leftrightarrow$ f' は 3-SAT の Yes 入力

この言い換えは，今後も頻繁に用いる

3-SAT は NP 完全：証明 (4)

f も f' も CNF 論理式なので，次が証明できればよい

証明すべきこと (さらに言い換え)

先の帰着で，$C = C' \vee \ell_1 \vee \ell_2$ を $(C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ に置き換えたとき

- C が充足可能 $\Leftrightarrow$ $(C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ が充足可能

$$\begin{aligned}
f &= (x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee y_2) \wedge (\overline{y_2} \wedge \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)\\
&\rightsquigarrow (x_1 \vee x_2 \vee y_2) \wedge (\overline{y_2} \wedge \overline{x_3} \vee y_1) \wedge (\overline{y_1} \vee x_4 \vee \overline{x_5}) \wedge (x_2 \vee x_3 \vee y_3)\\
&\quad \wedge (\overline{y_3} \vee x_4 \vee x_5) = f'
\end{aligned}$$

3-SAT は NP 完全：証明 (5)

$\boxed{\Rightarrow \text{の証明}}$：$C = C' \vee \ell_1 \vee \ell_2$ が充足可能であると仮定する

- C の充足割当 a が存在する

1. a が C' を充足する場合
   - a は $C' \vee y$ も充足する
   - $y \mapsto 0$ とすれば，$\overline{y} \vee \ell_1 \vee \ell_2$ も充足する
   - $\therefore (C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ も充足可能
2. a が $\ell_1 \vee \ell_2$ を充足する場合
   - a は $\overline{y} \vee \ell_1 \vee \ell_2$ も充足する
   - $y \mapsto 1$ とすれば，$C' \vee y$ も充足する
   - $\therefore (C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ も充足可能 $\square$

3-SAT は NP 完全：証明 (6)

$\boxed{\Leftarrow \text{の証明}}$：$(C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ が充足可能であると仮定する

- $(C' \vee y) \wedge (\overline{y} \vee \ell_1 \vee \ell_2)$ の充足割当 a' が存在する

1. $a'\colon y \mapsto 0$ のとき
   - a' は $C' \vee y$ を充足するので，C' も充足する
   - $\therefore a'$ は $C = C' \vee \ell_1 \vee \ell_2$ も充足する
2. $a'\colon y \mapsto 1$ のとき
   - a' は $\overline{y} \vee \ell_1 \vee \ell_2$ を充足するので，$\ell_1 \vee \ell_2$ も充足する
   - $\therefore a'$ は $C = C' \vee \ell_1 \vee \ell_2$ も充足する $\square$

これで，考えている帰着が多対一帰着であることが分かった $\square$

3-SAT は NP 完全：証明のまとめ

3-SAT は NP 完全：証明全体の構造

- 3-SAT が NP に所属することの証明
- CNF-SAT を 3-SAT に多項式時間多対一帰着可能であることの証明
  - CNF-SAT の入力 f から 3-SAT の入力 f' の構成
  - 構成が多項式時間でできることの確認
  - 「f が Yes 入力 $\Rightarrow$ f' が Yes 入力」の証明
  - 「f' が Yes 入力 $\Rightarrow$ f が Yes 入力」の証明

この証明構造は，他の NP 完全性証明でも見られる (ことになる)

目次

今日のまとめ と 次回の予告

今日の目標

- 次の基本概念を理解する
  - 充足可能性問題 (SAT)，k 充足可能性問題 (k-SAT)
  - 連言標準形，選言標準形
- 多項式時間多対一帰着を用いて，
  次の問題が NP 完全であることを証明する
  - 3 充足可能性問題 (3-SAT)

次回の予告

グラフに関する様々な問題が NP 完全であることを証明する

文献案内

- S. Cook, The complexity of theorem proving procedures. Proceedings of the Third Annual ACM Symposium on Theory of Computing (1971) 151–158.
- L. A. Levin, Universal sequential search problems. Probl. Peredachi Inf. 9 (1973) 115–116.
- R. M. Karp, Reducibility among combinatorial problems. In: R. E. Miller, J. W. Thatcher, J.D. Bohlinger (eds.), Complexity of Computer Computations. New York, Plenum (1972) pp. 85–103.