第5章 代表的な計算量クラス

5.1. 代表的な時間計算量クラス

$$\mathrm{P} \equiv \bigcup_{p:\text{多項式}} \mathrm{TIME}(p(l))$$

$$\mathrm{E} \equiv \bigcup_{c>1} \mathrm{TIME}(2^{cl})$$

$$\mathrm{EXP} \equiv \bigcup_{p:\text{多項式}} \mathrm{TIME}(2^{p(l)})$$

C集合: 計算量クラスCに入る集合.

C問題: C集合の認識問題

ある問題がPに入っていないなら、
現実的には手に負えない…

Chapter 5
Representative Complexity Classes

5.1. Representative time complexity classes

$$\mathrm{P} \equiv \bigcup_{p:\text{polynomial}} \mathrm{TIME}(p(l))$$

$$\mathrm{E} \equiv \bigcup_{c>1} \mathrm{TIME}(2^{cl})$$

$$\mathrm{EXP} \equiv \bigcup_{p:\text{polynomial}} \mathrm{TIME}(2^{p(l)})$$

C set : set in the complexity class C.

C problem : problem of recognizing a C set.

Problems not in P are intractable
from the practical viewpoint…

例5.1: クラスP, E, EXPでは，多項式時間程度の違いは問題ではない．

P: 多項式 × 多項式→多項式

E: 2の線形乗 × 多項式 → 2の線形乗

EXP: 2の多項式乗 × 多項式 → 2の多項式乗

例5.2: PRIMEの計算量クラス

例4.7 → PRIME $\in$ TIME(2^l)

故に，PRIME $\in$ E

余談: 2002年に $O(l^6)$ のアルゴリズムが考案されたので、今ではP

定義5.1. T: 制限時間の集合

$\bigcup_{t \in T}$ TIME(t): T時間計算量クラス

→これをTIME(T)と表す．

定理5.1: (1) P = $\cup_{c>0}$TIME(l^c), (2) EXP = $\cup_{c>0}$ TIME(2^{l^c})

Ex.5.1: Polynomial makes no serious difference in the classes P, E, EXP.

P: polynomial × polynomial → polynomial

E: linear power of 2 × polynomial → linear power of 2

EXP: poly. power of 2 × poly. → poly. power of 2

Ex.5.2: Complexity class of PRIME

Ex.4.7 → PRIME $\in$ TIME(2^l)

Thus, PRIME $\in$ E

$O(l^6)$time algorithm put it in P!!

Def.5.1: T: set of time limits

$\bigcup_{t \in T}$TIME(t): T time complexity class

→It is denoted by TIME(T).

Theorem5.1 (1) P = $\bigcup_{c>0}$TIME(l^c), (2) EXP = $\bigcup_{c>0}$TIME(2^{l^c})

定理5.1: (1) P = $\cup_{c>0}\mathrm{TIME}(l^c)$,　(2) EXP = $\cup_{c>0}\mathrm{TIME}(2^{l^c})$

証明: (2)の証明は省略.
T1: n^cという形の多項式の集合.
T_2: 多項式の全体
→ $T_1 \subseteq T_2$ なので，$\mathrm{TIME}(T_1) \subseteq \mathrm{TIME}(T_2)$
p: 任意の多項式 (pはT_2の任意の要素）
多項式pの最大次数をkとすると，$p(n) = O(n^k)$
定理4.3より,
$\mathrm{TIME}(p(l)) \subseteq \mathrm{TIME}(l^k) \subseteq \mathrm{TIME}(T_1)$
したがって，$\mathrm{TIME}(T_1) = \mathrm{TIME}(T_2)$

証明終

Theorem 5.1: (1) P $= \bigcup_{c>0}$ TIME(l^c), (2) EXP $= \bigcup_{c>0}$ TIME(2^{l^c})

Proof: The proof of (2) is omitted.

T_1: set of polynomials of the form of n^c.

T_2: set of all polynomials

→ since $T_1 \subseteq T_2$, TIME(T_1) $\subseteq$ TIME(T_2)

p: arbitrary polynomial (p is any element of T_2)

if the maximum degree of a polynomial p is k, $p(n) = O(n^k)$

From Theorem 4.3,

TIME($p(l)$) $\subseteq$ TIME(l^k) $\subseteq$ TIME(T_1)

Therefore, TIME(T_1) $=$ TIME(T_2)

Q.E.D.

例5.3. 命題論理式評価問題(PROP-EVAL)

入力: $\langle F, \langle a_1, a_2, \ldots, a_n \rangle\rangle$

Fは拡張命題論理式 $\wedge \vee \neg \rightarrow \leftrightarrow$

$(a_1, a_2, \ldots, a_n)$は F に対する真理値割り当て

質問: $F(a_1, a_2, \ldots, a_n) = 1$?

(x,y)	$x \rightarrow y$ $(\neg x \vee y)$	$x \leftrightarrow y$ $((x \rightarrow y) \wedge (y \rightarrow x))$
(0,0)	1	1
(0,1)	1	0
(1,0)	0	0
(1,1)	1	1

Ex.5.3. Problem of evaluating propositional expression(PROP-EVAL)

Input: $<F, < a_1, a_2, \ldots, a_n >>$

F is an extended prop. expression

$(a_1, a_2, \ldots, a_n)$ is a truth assignment to F

Question: $F(a_1, a_2, \ldots, a_n) = 1$?

(x,y)	$x \rightarrow y$ $(\neg x \vee y)$	$x \leftrightarrow y$ $((x \rightarrow y) \wedge (y \rightarrow x))$
(0,0)	1	1
(0,1)	1	0
(1,0)	0	0
(1,1)	1	1

例5.3. 命題論理式評価問題(PROP-EVAL)

入力: $<F, < a_1, a_2, \ldots , a_n >>$

Fは拡張命題論理式 $\wedge \vee \neg \rightarrow \leftrightarrow$

$(a_1, a_2, \ldots , a_n)$はFに対する真理値割り当て

質問: $F(a_1, a_2, \ldots , a_n) = 1?$

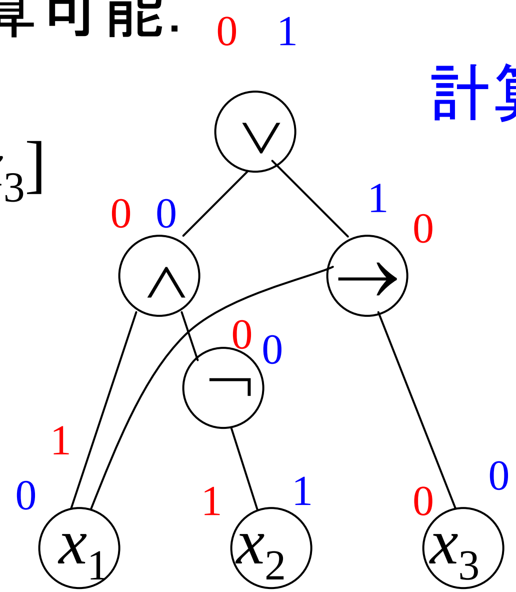
拡張命題論理式 F がコード化されたもの $\lceil F \rceil$ から計算木を作る.

計算木は$O(|\lceil F \rceil|^3)$時間で構成できる.

計算木が得られていれば, ボトムアップ式で

$F(a_1, a_2, \ldots , a_n)$ の値は容易に計算可能.

計算木



例: $F(x_1, x_2 , x_3) = [x_1 \wedge \neg x_2] \vee [x_1 \rightarrow x_3]$

$F(0,1,0)=1$

$F(1,1,0)=0$

よって PROP-EVAL ∈ P

Ex.5.3. Problem of evaluating propositional expression(PROP-EVAL)

Input : $\langle F, \langle a_1, a_2, \ldots, a_n \rangle\rangle$

F is an extended prop. expression

$(a_1, a_2, \ldots, a_n)$ is a truth assignment to F

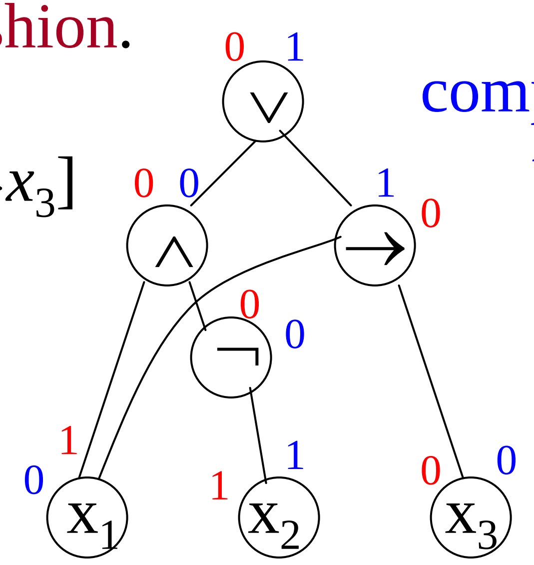
Question : $F(a_1, a_2, \ldots, a_n) = 1$?

Construct a computation tree from a code $\ulcorner F \urcorner$ of ext. prop. expression

It is built in time $O(|\ulcorner F \urcorner|^3)$.

If computation tree is available, we can easily obtain the value $F(a_1, a_2, \ldots, a_n)$ in a bottom-up fashion.

Ex. : $F(x_1, x_2, x_3) = [x_1 \wedge \neg x_2] \vee [x_1 \rightarrow x_3]$



computation tree

$F(0,1,0)=1$

$F(1,1,0)=0$

Hence PROP-EVAL ∈ P

例5.3. 命題論理式充足性問題：2和積形(2SAT)

入力：$<F, < a_1, a_2, \ldots , a_{\mathrm{n}} >>$ Fは2和積形命題論理式

質問： $F(a_1, a_2, \ldots , a_n) = 1$を満たす割り当てがあるか?

和積形:

$F = (● \vee ● \vee \ldots \vee ●) \wedge (● \vee \ldots \vee ●) \wedge \ldots \wedge (\ldots)$

- リテラルの論理<u>和</u>の論理<u>積</u>で表現されたもの

k和積形(k SAT)

- 和積形の各論理和が k 個のリテラルを含む

<u>ちょうど</u>/たかだか

- 3SAT, 4SAT も同様に定義できる。
- SAT: 各論理和のリテラルの個数に制限がないもの
- ExSAT: 入力が拡張命題論理式(→や↔も許す)

Ex. 5.3. 2-Satisfiability (2SAT)

Input : $\langle F, \langle a_1, a_2, \ldots, a_n \rangle\rangle$ F is 2-conjunctive normal form

Question : Is there any assignment such that $F(a_1, a_2, \ldots, a_n) = 1$?

Conjunctive Normal Form (CNF)

$$F = (\bullet \vee \bullet \vee \ldots \vee \bullet) \wedge (\bullet \vee \ldots \vee \bullet) \wedge \ldots \wedge (\ldots)$$

- described by $\wedge$ of $\vee$ of literals.

exactly/at most

k SAT

- Each closure contains k literals

- We can define 3SAT, 4SAT similarly.
- SAT consists of any CNF.
- ExSAT consists of any extended propositional expression.

例5.4: 到達可能性問題(ST-CON)

入力: $<G,s,t>$: 無向グラフG, $1 \leqq s,t \leqq n(=|G|)$
質問: G上で s から t への道があるか?

- **閉路**とは、始点と終点が同じである路
- **オイラー閉路**とは、すべての辺を一度づつ通る閉路
- **ハミルトン閉路**とは、すべての頂点を一度づつ通る閉路

例5.4: 一筆書き閉路問題(DEULER)

入力: $<G>$: 有向グラフG
質問: Gはオイラー閉路をもつか?

例5.5: ハミルトン閉路問題(DHAM)

入力: $<G>$: 有向グラフG
質問: Gはハミルトン閉路をもつか?

Ex. 5.4: Graph reachability problem (ST-CON)

Input : <G,s,t> : an undirectd graph G, $1 \leqq s,t \leqq n(=|G|)$

Question : Does G have a path from s to t?

- Cycle is a path that shares two endpoints.
- Euler cycle is a cycle that visits all edges once.
- Hamiltonian cycle is a cycle that visits all vertices once.

Ex. 5.4: Euler cycle problem (DEULER)

Input : <G>: a directed graph G

Question : Does G have an Euler cycle?

Ex. 5.5 Hamiltonian cycle problem (DHAM)

Input : <G>: a directed graph G

Question : Does G have a Hamiltonian cycle?

知られていること:

- 以下の問題はP:
  - ✓ PROP-EVAL, 2SAT, ST-CON, DEULER
- 以下の問題はEであることはわかっているが…
  - ✓ 3SAT, DHAM

PとEの間にある(?)クラスNP

It is known that:

- The following problems are in P:
  - ✓ PROP-EVAL, 2SAT, ST-CON, DEULER

- The following problems are in E, but…
  - ✓ 3SAT, DHAM

The class NP between P and E?