数理解析研究所講究録
第 494 巻 1983 年 148-159

Gradually Intractable Problems

電気通信大学　笠井琢美 (Takumi Kasai)
東　海　大　学　岩田茂樹 (Shigeki Iwata)

1. 序論

近年数多くの NP 完全, PSPACE 完全問題が発見されている。NP 完全問題は intractable であろうと考えられている。P = NP は非常に重要な未解決問題である。実際 P に属する問題の中にも tractable ではないものは数多く存在する。本論文では gradually intractable problem という概念を導入し、gradually intractable problems の間の関係をしらべる。

直観的にいうと gradually intractable problem とは、問題の列 $L_0, L_1, \cdots$ のことで, 各 L_k は P に属するが, $\bigcup_{k=0}^{\infty} L_k$ は intractable と考えられる問題である。C を NP $\subseteq$ C の問題のクラスとする。gradually C complete problem とは問題の列 $L_0, L_1, \cdots$ で, 各 L_k は P に属し, $\bigcup_{k=0}^{\infty} L_k$ は C complete で満たす問題である。

例1　k - クリーク問題は gradually NP complete である。

ここで k－クリーク問題は与えられた有向グラフが k－クリークを持つかを判定する問題である。

例2　k－ナップザック問題は gradually NP complete である。k－ナップザック問題とは与えられた正の整数列 $x_1, x_2, \cdots, x_n, y$ に対し，$x_1, x_2, \cdots, x_n$ から k 個選んでその和を y とできるかを判定する問題である。

例3　k 匹のねずみとねこ問題は gradually EXPTIME complete である。この問題は $\Omega(n^{(k-1)/4})$ 時間必要であることが知られている。

$L_0, L_1, \cdots$ が gradually intractable problem であるとする。もし $P \neq NP$ ならば $\bigcup_{k=0}^{\infty} L_k \neq P$ であるから k が大きいとき L_k は intractable と予想される。よって、ある gradually intractable problem に対しては，どの $t > 0$ に対しても，ある k が存在して，L_k は $\Omega(n^t)$ 時間では解けないと予想される、ではどの程度の大きさの k に対し L_k は intractable となるであろうか、言い換えれば L_k はどのくらいの次数の多項式時間を必要とするだろうか。この問題は興味ある問題ではあるが，P＝NP 問題と同程度か、それ以上に難かしい問題であると考えられる。

gradually intractable problem 間の reducibility の議論に、s－reducibility を導入する。s は $N = \{0, 1, 2, \cdots\}$ から N へ

の関数である。

Σをアルファベット，$L_0, L_1, \cdots$ と $L_0', L_1', \cdots$ を問題の例，$L_k \subseteq \Sigma^*$，$L_k' \subseteq \Sigma^*$，$k=0,1,\cdots$，また g を N から N への関数とする。L_k が $L'_{g(k)}$ に <u>s-reducible</u> であるとは，多項式 $p(n)$ が存在し，任意の k に対し，次の (i) と (ii) を満たす $p(n)$ 時間計算可能な関数 $f: \Sigma^* \to \Sigma^*$ を構成できるときをいう。

(i) 任意の $x \in \Sigma^*$ に対し，$x \in L_k$ iff $f(x) \in L'_{g(k)}$，

(ii) 任意の $x \in \Sigma^*$ に対し，$|f(x)| \leq s(|x|)$.

条件 (i) は L_k が $L'_{g(k)}$ に reducible であること，(ii) は長さ n の語が長さが高々 $s(n)$ の語に変換されることを示している。

この定義よりただちに次の補題を得る。

<u>補題 1.1</u>　T を N×N から N への関数で $n, m, k \in$ N に対し，$n \leq m$ ならば $T(n,k) \leq T(m,k)$ を満足するものとする。L_k は $L'_{g(k)}$ に s-reducible とする。もし L_k' が $T(n,k)$ 時間で解けるならば，L_k は

$$T(s(n), g(k)) + p(n)$$

時間で解ける。ただし n は入力長で $p(n)$ は n の多項式で k には関係しない。

<u>証明</u>　L_k への入力 x，$|x| = n$，は $f(x)$ に変換されるので $L'_{g(k)}$ は

$$T(|f(n)|, g(k)) + p(n)$$

時間で解くことができる。ただし $p(n)$ は reduction に要する時間で k には無関係である。$|f(n)| < s(n)$ であるので,

$$T(|f(n)|, g(k)) + p(n) < T(s(n), g(k)) + p(n).$$

よって L_k は $T(s(n), g(k)) + p(n)$ 時間で解ける。

注. $p(n)$ は本論文では高々 n^3 程度であるので、$p(n)$ の項を無視して議論することができる。

任意の k に対し, L_k は $\Omega(n^k)$ 時間必要であると仮定する。さらに、L_k が $L'_{g(k)}$ に s-reducible であると仮定する。たとえば,

$$s(n) = O(n \log n), \quad p(n) = O(n^2),$$
$$g(k) = k+2$$

とする。$p(n)$ は reduction に要する時間をあらわす。すると任意の ε ($\varepsilon > 0$) に対し L'_{k+2} は $\Omega(n^{k-\varepsilon})$ 時間では解けないことが示される。したがって、もし L_k が $\Omega(n^k)$ 時間必要であると強く予想できるような問題であれば L'_k も $\Omega(n^{k-2-\varepsilon})$ 時間必要であると強く予想される。それゆえ、もしこのような問題の列 $L_0, L_1, \cdots$ が存在すれば, この問題を基に, 次々と色々な問題が多項式時間の下界をもつという強い予想を得ることができる。

本論文では、問題のクラス NL_k に注目する。NL_k は 2^k 個の文字を使用する $\log n$ space bounded nondeterministic

Turing machine により受理される問題のクラスである。
以下では、一般性を失うことなく，Turing machine が使用する文字は2文字で 0と1 であると仮定する。すると，

$NL_k = \{ L \subseteq \{0,1\}^* \mid L$ は $k \log n$ space bounded の 2文字上の nondeterministic Turing machine により受理される $\}$

NL_k は興味あるクラスで、色々研究されている。

<u>定義</u> T を N から N への関数とする。DTIME(T(n)) で T(n) 時間計算可能な問題全体からなるクラスを表わす。a を正の実数，L を問題とする。L が <u>n^a 時間必要</u>であるとは、任意の実数 ε ($\varepsilon > 0$) に対し

$$L \notin DTIME(n^{a-\varepsilon})$$

のとき、すなわち L は $n^{a-\varepsilon}$ 時間では受理できないときをいう。

次に述べる conjecture は成立するであろうと考えられる。

<u>Conjecture A</u> 任意の $\varepsilon > 0$ に対し

$$NL_k \notin DTIME(n^{k-\varepsilon}).$$

すなわち，Conjecture A は任意の k に対し，NL_k は n^k 時間必要とするような問題を含んでいることを意味する。

2. k-pebble game

このセクションでは NL_k の元が $(k+2)$-pebble game に reduceできることを示す。

定義 pebble game は $G=(X, R, S, t)$ のことであり，

(1) X は有限集合で Xの元を頂点とよぶ。

(2) $R \subseteq \{(x, y, z) \mid x, y, z \in X,\ x \neq y,\ y \neq z,\ z \neq x\}$

Rの元を規則とよぶ。

(3) $S \subseteq X$. Sの元の個数を Gの階数とよぶ。

(4) $t \in X$. tを最終頂点とよぶ。

Gの階数が kのとき Gを k-pebble game とよぶ。

pebble game のはじめに，pebble は Sのすべての頂点に置かれる。$(x, y, z) \in R$ で pebble が x, y に置かれ，zに置かれていなければ，xにある pebble を zに移すことができる。pebble game problem とは与えられた pebble game の規則に従って pebbleを動かしていき，最終頂点に pebbleを置くことができるかを決定する問題である。与えられた pebble game の階数が kのとき，その問題を k-pebble game problem という.

注. pebble game problem は PSPACE 完全であり，各k $(k \geq 3)$ に対して k-pebble game problem は NLOGSPACE完全であることが知られている。NLOGSPACEは $O(\log n)$ space bounded nondeterministic Turing machine で受理される問題のクラスである。したがって k-pebble game problem は P に属する. よって, k-pebble game problem は gradually PSPACE complete である。また k-pebble game problem は NL_k に属することに注意する。(長さ $\log n$ の k個のカウンタを使用すれば解ける.)

次に問題のクラスの列から問題の列への s-reduction を定義する。

<u>定義</u> $C_0, C_1, \cdots$ を問題のクラスの列, $L_0, L_1, \cdots$ を問題の列, s, g を N から N への関数とする。$\Sigma = \{0, 1\}$, 問題は Σ^* の部分集合であると仮定する。C_k が L_k に <u>s-reducible</u> であるとは, 多項式 $p(n)$ が存在し, 任意の k と任意の $L \in C_k$ に対し次の (i), (ii) を満たす $p(n)$ 時間計算可能な関数 f: $\Sigma^* \to \Sigma^*$ が存在するときをいう.

(i) 任意の $x \in \Sigma^*$ に対し, $x \in L$ iff $f(x) \in L_{g(k)}$,

(ii) 任意の $x \in \Sigma^*$ に対し, $|f(x)| \leq s(|x|)$.

<u>定理 2.1</u> NL_k は $(k+2)$-pebble game problem に $O(n \log^2 n)$-reducible である.

<u>証明</u> $L \in NL_k$ とする。すると L を受理する２文字上の $k \log n$ space bounded nondeterministic Turing machine M が存在する。M の状態集合を Q とする。Σ^* の元 x が与え

られたとき，次を満たす $(k+2)$-pebble game G_x を構成すればよい。

(1) $x \in L$ iff G_x において最終頂点に pebble を置ける.

(2) $|G_x| = O(n \log^2 n)$, $n = |x|$.

Mの状態関数は $Q \times \Sigma \times \Sigma \times Q \times \{-1,0,1\} \times \Sigma \times \{-1,0,1\}$ の元で表わされているものと仮定する。状態関数 $r = (q, \sigma, a, q^*, d_I, a^*, d_W)$ は次のことを意味する。もし M が状態 q で入力記号 σ，作業用テープ文字 a を読むと，Mは状態 q^* になり，作業用テープ文字を a から a^* に変え，入力テープと作業用テープのヘッドを d_I, d_W に動かす。一般性を失うことなく，Mが最初に適用する状態関数を r_0 とする。

与えられた入力 x, $|x| = n$, に対し，Mの動作を模倣するのに作業用テープを k 個のブロックに分割する。各ブロックの長さは $\log n$ である。（図2.1 参照）

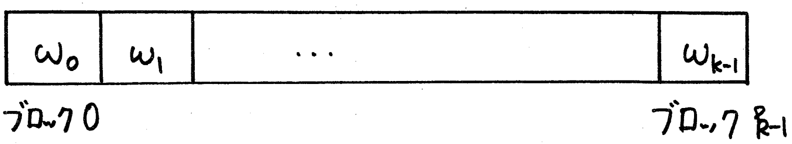

図2.1　Mの作業用テープを k 個に分割する.

Mの configuration を表わすのに G_x では次の頂点を使用する。

$$X_1 = \{\langle q, i\rangle, \langle q, i\rangle' \mid q \in Q,\ 0 \le i \le n-1\} \cup \{t\},$$

$$X_2 = \{\langle l, w\rangle, \langle l, w\rangle' \mid 0 \le l \le k-1,\ 0 \le w \le n-1\},$$
$$X_3 = \{\langle j, r, s\rangle \mid 0 \le j \le k \log n,\ r \text{ は M の状態関数},\ 0 \le s \le 3\}$$

X_1の元$\langle q, i\rangle$にpebbleが置かれるのはMの現在の状態がqで入力テープヘッドがi番目の位置にあることを示す。次に作業用テープの内容をk個のpebbleをX_2に置くことによって表わす。$\langle l, w\rangle$にpebbleが置かれるのはl番目のブロックの内容がwであることを表わす。各ブロックの長さは$\log n$でブロックの内容はwで表わせることに注意する。X_3の元$\langle j, r, s\rangle$にpebbleがあるのは、作業用テープのヘッドの位置がj，次に実行される状態関数がrであることを示す。sはシミュレーションの制御に使用される。以上のように全体では$(k+2)$個のpebbleを使用する。$\langle q, i\rangle'$，$\langle l, w\rangle'$は作業用頂点を表わす。q_0を初期configuration，r_0をMが最初に実行する状態関数とすると，

$$S = \{\langle q_0, 0\rangle, \langle 0, 0\rangle, \langle 1, 0\rangle, \cdots, \langle k-1, 0\rangle, \langle 0, r_0, 0\rangle\}$$

とする。

G_xの規則RはMの各状態関数$r = (q, \sigma, a, q^*, d_I, a^*, d_W)$に対して図2.3(a)を含む。また各最終状態$q_f \in Q$については図2.3(b)を含む。ただしRの規則$(x, y, z)$を図2.2のように表示することとする。

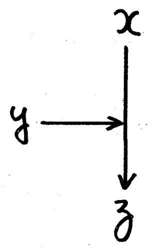

図2.2 規則 (x, y, z) の表示

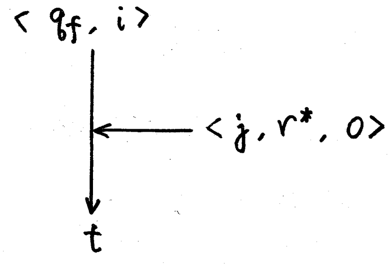

図2.3(b) 各最終状態 q_f についての規則

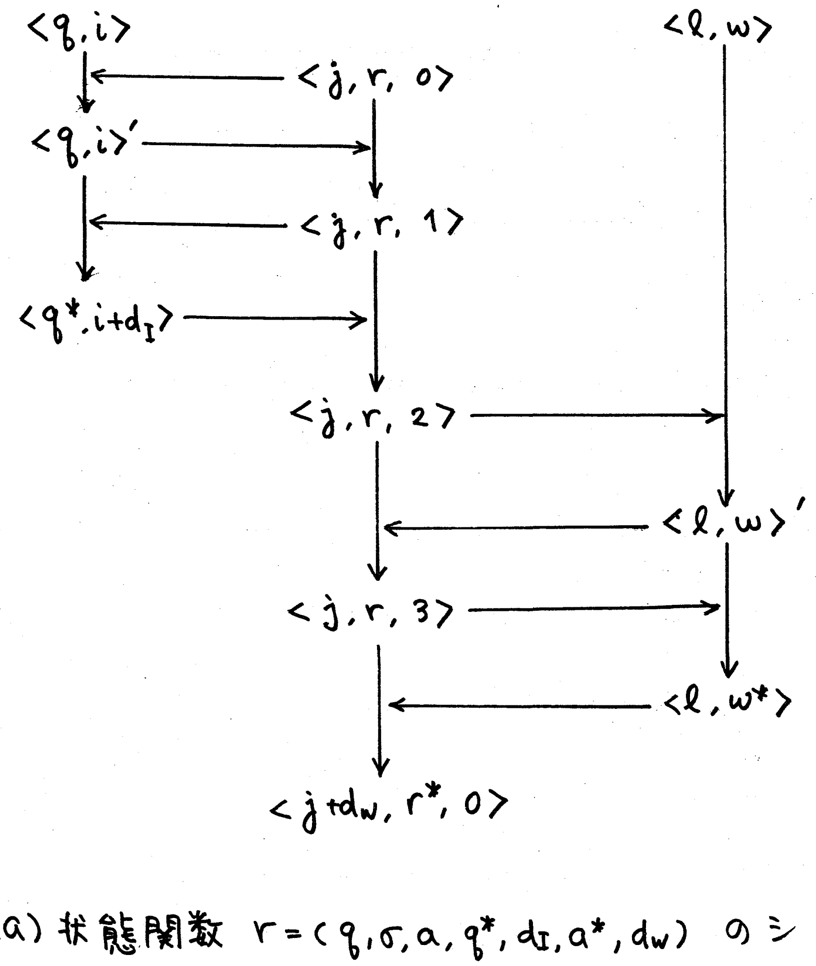

図2.3(a) 状態関数 $r=(q, \sigma, a, q^*, d_I, a^*, d_W)$ のシミレーション.

ただし $j=l\cdot\log n+t$, $0\leq t\leq\log n$, w^* は w の t 番目のビットが a から a^* に変わったもの, r^* は M の任意の状態関数。

図2.3 (a)の規則は各 i $(0\le i\le n-1)$, j $(0\le j\le k\log n)$, l $(0\le l\le k)$, w $(0\le w<n)$ について構成するので、規則の数は $O(n\log n)$ である。各規則は $O(\log n)$ の長さで表現できるから、全体として $O(n\log^2 n)$ の長さで表現できる。すなわち G_x を $O(n\log^2 n)$ で表わすことができる。G_x の構成から明らかなように、Mが x を受理するときかつそのときに限り G_x で t に pebble が置けるので、定理は証明された。

系 Conjecture A は k-pebble game problem が $\Omega(n^{k-2})$ 時間必要であることを意味する。

系 自然数 k と実数 a $(a\ge 2)$ に対し、もし

$$NL_k \not\subset DTIME(n^a)$$

ならば $(k+2)$-pebble game problem は n^a 時間必要である。

3. 応用

このセクションでは Conjecture A を仮定すると、いろいろな問題が多項式時間の下界を持つことを示す。記述を簡単にするため "Conjecture A が成立するなら" が省略されているものとする。ここでは結果だけを述べる。

定理3.1 k-FNU (Finite automaton Non Universability) の問題：

入力：$\Sigma=\{0,1\}$ 上の k 個の有限オートマトン $A_1,\cdots,A_k$

決定すること：　$L(A_1)\cup L(A_2)\cup\cdots\cup L(A_k)\ne\Sigma^*$

は $\Omega(n^{k/3})$ 時間必要である。

系 入力：$\Sigma=\{0,1\}$ 上の k 個のオートマトン $A_1, \cdots, A_k$

決定すること：$L(A_1) \cap L(A_2) \cap \cdots \cap L(A_k) \neq \emptyset$

の問題は $\Omega(n^{k/3})$ 時間必要である。

定理3.2 k 次元グラフ上の次の問題はいずれも $\Omega(n^{k/3})$ 時間必要である。(1) k-GAP（k 次元グラフ上の道発見問題）(2) k-STRONG（k 次元グラフが強連結），(3) lSCON on k-DIM（k 次元グラフが l-強連結，l は定数）

系 k 次元無向グラフのマッチングが最大でないかどうかの問題は $\Omega(n^{(k-1)/3})$ 時間必要とする。

定理3.3 n 次元 conservative vector addition system の k-到達性問題は $\Omega(n^{(k-2)/3})$ 時間必要とする。

〈参考文献〉

[1] Adachi, Iwata, Kasai, Low level complexity for combinatorial Games, Proc. ACM Symp. on Theory of Comput., 1981

[2] Kasai, Adachi, Iwata, Classes of pebble games and complete problems, SIAM J. on Comput. 8, 1979

[3] Ibarra, On two-way multihead automata, JCSS 7, 1973

[4] Monien, Transformational methods and their application to complexity problems, Acta Informatica 6, 1976

[5] Seiferas, Relating relined space complexity classes, JCSS 7, 1973.