第8章 問題の還元

ある問題を解くことを別の問題を解くことに帰着させることを問題の還元という．還元という手法を使うと，計算不能であることがすでにわかっている問題から別の計算不能な問題を導くこともできるし，計算可能で計算量の上界や下界がすでにわかっている問題から別の問題の計算量の上界や下界を示すこともできる．また，還元可能かどうかに従って問題の間に順序を導入することができ，この順序の下で“もっとも難しい問題”（これを“完全問題”という）というものを考えることができる．これらの概念は，7.3 節で述べたように，実は帰納的関数の理論で使われていたものを借用し計算量理論向けにアレンジしたものであるが，それが帰納的関数の理論に豊かな成果をもたらしたのと同様に（いや，それ以上に），計算量理論の発展を促す大車輪となった．

8.1 還元とは

与えられた問題を解くためにその問題を変形して，すでに解く方法が分かっている別の問題に帰着させるという方法は数学においてよく用いられる常套手段の一つである．“還元 (reduction)”と呼ばれる同様の手法は計算量の理論でも用いられ，この分野に実に豊かな成果をもたらした[†1]．与えられた問題 A がどのくらい難しいかを判定するために，その問題のサイズを大きく変えないような変形を施して，解き方（や難しさの程度）がすでに分かっている問題 B へと変換する．このように A を B へ変換できる場合，A を解く代わりに B を解けばよいから，A の計算量は（B の計算量）＋（変換にかかる計算量）である．もし，変換にかかる計算量が B の計算量に吸収されてしまうくらい小さければ，A の計算量は B の計算量以下である（A は B に帰着させる以外の方法により，もっと少ない計算量で解けるかもしれない）．よって，A を B へ変換できることを

$$\mathrm{A} \leq \mathrm{B}$$

と書くことにする．

逆に，A の計算量がすでに分かっているときに A を B へ変換することにより，B の計算量の下界を知ることができる．例えば，A が計算不能なら B も計算不能であることが導かれる．以下では，このような変換方法のいくつかについて考察する．

問 8.1. 上記の $\leq$ はこのままでは順序にならないが，$\leq$ が推移的（すなわち，任意の A, B, C について，$A \leq B$ かつ $B \leq C \Longrightarrow A \leq C$）であるならば，

$$A \equiv B \overset{\text{def}}{\Longleftrightarrow} A \leq B \text{ かつ } B \leq A$$

と定義すれば（すなわち，計算量が同じものを同一視すれば）$\equiv$ は同値関係になり（問 1），$\equiv$ の同値類の上で

$$[A]_{\equiv} \leq_{\equiv} [B]_{\equiv} \overset{\text{def}}{\Longleftrightarrow} A \leq B$$

[†1] reduce/reduction の意味は「1. 減らす，縮小する．2.（ある状態に）する，変える」の 2 つあるが，「還元」は後者の意味で使われている．「帰着する」という言い方もあろうが，現在では「還元」という用語が定着している．

と定義すれば $\leq_{\equiv}$ の定義は代表元の選び方によらず（問 2），$\leq_{\equiv}$ は順序（半順序）となる（問 3）．ここで，$[X]_{\equiv}$ は X を代表元とする $\equiv$ の同値類（すなわち，計算量が同じ問題からなるクラス）を表す．
（問 1）～（問 3）を証明せよ．

すでに何度も登場している概念であるが，**変換機** (transducer) とは出力テープを持つオフライン DTM M のことであり，入力 x に対する計算が停止したとき，出力テープ上には変換結果として文字列 $M(x)$ が書き出される．M の入力アルファベットを Σ，出力アルファベットを Δ とするとき，M は Σ^* から Δ^* への計算可能な関数（すなわち，$x \in \Sigma^*$ を $M(x) \in \Delta^*$ へ変換するアルゴリズム）である．

ある多項式 $p(n)$ が存在して M が $O(p(n))$ 時間限定であるとき M を**多項式時間限定変換機** (polynomial-time bounded transducer) といい，M が $O(\log n)$ 領域限定であるとき**対数領域限定変換機** (logarithmic-space bounded transducer) という．

A, B を言語（＝問題）とする．条件

$$\forall x\,[\,x \in A \iff M(x) \in B\,]$$

を満たす停止性変換機（すなわち，アルゴリズム）M が存在するとき A は B に **m-還元可能** (many-one reducible) であるといい，

$$A \leq_m B$$

と表す．特に，M が多項式時間限定変換機であるとき A は B に**多項式時間還元可能** (polynomial-time reducible) であるといい，

$$A \leq_m^{poly} B$$

と表す．$\leq_m^{poly}$ を**カープ還元** (Karp reduction) ともいう．以後，$\leq_m^{poly}$ を単に $\leq_{poly}$ で表す．また，M が対数領域限定変換機であるとき A は B に**対数領域還元可能** (log-space reducible) であるといい，

$$A \leq_{log} B$$

と表す．特に，M を明示したいときは

$$A \leq_{poly} B \text{ via } M \text{ あるいは } A \leq_{log} B \text{ via } M$$

などと書くこともある．

以下で，還元可能性に関する 3 つの重要な補題を示す．最初の 1 つは次の補題である：

補題 8.1. (1) $A \leq_{log} B \implies A \leq_{poly} B \implies A \leq_m B$.
(2) $\leq$ は $\leq_m$, $\leq_{poly}$, $\leq_{log}$ のどれかとする．
(a) $A \leq B \iff A^c \leq B^c$.
(b) $\leq$ は反射律，推移律を満たす．

証明 (1) $A \leq_{log} B \implies A \leq_{poly} B$ は定理 5.5 (3) による．$A \leq_{poly} B \implies A \leq_m B$ は自明．
(2)(a)
$$\begin{aligned} A \leq B \text{ via } M &\iff [\,x \in A \iff M(x) \in B\,] \\ &\iff [\,x \notin A \iff M(x) \notin B\,] \text{ すなわち } [\,x \in A^c \iff M(x) \in B^c\,] \\ &\iff A^c \leq B^c \text{ via } M. \end{aligned}$$

(2)(b) 反射律は自明なので，推移律についてだけ示す．はじめに，$\leq_{poly}$ について考える．

$$A \leq_{poly} B \text{ via } M_1 \text{ かつ } B \leq_{poly} C \text{ via } M_2 \tag{8.1.1}$$

とし，M_1, M_2 はそれぞれ $p_1(n)$, $p_2(n)$ 時間限定であるとする（$p_1(n)$, $p_2(n)$ は多項式）．DTM M_3 を次のように構成する．

1. x が入力のとき，M_3 はまず x を入力として M_1 を模倣する．その結果，$M_1(x)$ が出力される．
2. 次いで，M_3 は $M_1(x)$ を入力として M_2 を模倣する．その結果，$M_2(M_1(x))$ が生成されるので，これを M_3 の出力とする．

M_1 は $p_1(n)$ 時間限定であるから，入力 x に対する M_1 の出力 $M_1(x)$ は

$$|M_1(x)| \leqq p_1(|x|) \tag{8.1.2}$$

を満たす．また，M_2 は $p_2(n)$ 時間限定であるから，入力 $M_1(x)$ に対する計算の時間量は，(8.1.2) より，たかだか

$$p_2(|M_1(x)|) \leqq p_2(p_1(|x|)) \tag{8.1.3}$$

である（計算量の性質から，$p_1(n), p_2(n)$ は単調増加関数であることに注意）．M_1 による x の変換にかかる時間 $p_1(|x|)$ を考慮すると，入力 x に対して M_3 が使う時間量は，(8.1.3) と合算してたかだか

$$p_1(|x|) + p_2(p_1(|x|))$$

であり，これは $|x|$ の多項式である．すなわち，M_3 は多項式時間限定である．一方，(8.1.1) を考慮すると，

$$x \in A \iff M_1(x) \in B, \text{ かつ } M_1(x) \in B \iff M_2(M_1(x)) \in C$$

が成り立つ．$M_3(x) = M_2(M_1(x))$ であるから，これは $A \leq_{poly} C$ via M_3 であることを意味する．

次に，$\leq_{log}$ について考える．$\leq_{poly}$ の証明と同様に合成変換機 M_3 を作る方法は，M_1 が計算した $M_1(x)$ を保存しておくための領域が必要になり，うまくいかない（$M_1(x)$ の長さは $|x|$ の多項式にもなりうる）．そこで，M_3 は M_1 の計算終了を待たずに，M_1 が 1 文字出力するたびにそれを直接使って $M_2(M_1(x))$ の模倣を行なう．M_2 がヘッドを左に動かす必要がない限りこれでうまくいく．M_2 がヘッドを左に動かしたときは，M_1 の計算を最初からやり直して（入力 x は入力テープ上に残っている）ヘッド位置に来るべき文字を求める．このためには，ヘッド位置を保存しておくためのカウンタが必要になる．

M_3 はカウンタ用テープを 2 本使う．カウンタ 1 には M_2 の入力ヘッド位置（すなわち，$M_1(x)$ の何番目の文字を読んでいるか）を記憶する．M_2 がヘッドを右（左）に動かすたびにカウンタ 1 に 1 を加える（1 を減じる）．カウンタ 2 は，M_2 のヘッド位置にある文字が何であるかを計算するときに作業用に使う．カウンタ 1 の内容が i のとき，M_3 は $M_1(x)$ の計算を最初からやり直し，M_1 の 1 ステップごとにカウンタ 2 に 1 を加える．カウンタ 1 と 2 の内容が一致したそのとき M_1 が出力する文字が求める文字である．M_1 は $\log n$ 領域限定であるから，定理 5.5 (3) より M_1 の時間量は $O(c^{\log|x|})$ である（c は定数）．よって，カウンタ 1, 2 の長さは $O(\log|x|)$ であり，したがって M_3 の領域量は $O(\log|x|)$ である．線形領域圧縮定理（定理 4.2）より，$O()$ をはずすことができる．以上で，$A \leq_{log} C$ via M_3 であることが示せた． □

8.2 決定不能問題の還元

はじめに，ある問題が決定不能であることを，すでに決定不能であることがわかっている問題から導くことを考えよう．

補題 8.2. $A \leq_m B$ で A が帰納的言語でないならば B も帰納的言語でない．

証明 $A \leq_m B$ via M とする．B が帰納的であるとすると B を受理する停止性 TM M_B が存在する．TM M_A を図 8.2.1 のように構成すると，

$$x \in L(M_A) \iff M(x) \in L(M_B) = B \iff x \in A$$

であるから $L(M_A) = A$ である．これは A が帰納的でないことに反す．□

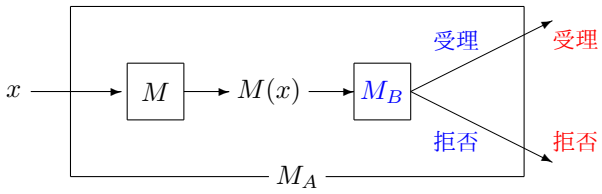

図 8.2.1: 合成 TM M_A

この補題にもとづき，以前取り上げた問題（例えば，TM の停止問題）よりも自然な決定不能問題を 1 つ示してみよう．Σ を 2 個以上の記号を含むアルファベットとする．P を $\Sigma^* \times \Sigma^*$ の有限部分集合

$$P = \{(x_1, y_1),\ \ldots,\ (x_n, y_n)\} \tag{8.2.1}$$

とする．

$$x_{i_1} x_{i_2} \cdots x_{i_k} = y_{i_1} y_{i_2} \cdots y_{i_k} \tag{8.2.2}$$

を満たす整数の組

$$(i_1,\ i_2,\ \ldots,\ i_k),\ k \geqq 1,\ 1 \leqq i_j \leqq n\ (1 \leqq j \leqq k) \tag{8.2.3}$$

が存在するとき，P は**解** $(i_1,\ i_2,\ \ldots,\ i_k)$ をもつという．任意に与えられたこのような P に対して P が解をもつかどうかを決定する問題を**ポストの対応問題** (Post's correspondence problem, **PCP**) という．

例 8.1. (1) $P_1 = \{(01, 011),\ (11, 1),\ (100, 00)\}$ とする．
$(x_1, y_1),\ (x_2, y_2),\ (x_3, y_3)$

$x_1 x_2 x_1 x_3 = 011101100 = y_1 y_2 y_1 y_3$ であるから，$(1, 2, 1, 3)$ は P_1 の解である．

(2) $P_2 = \{(a, b)\}$ は明らかに解をもたない．

(3) $P_3 = \{(x_1, y_1) = (ab, ba),\ (x_2, y_2) = (aab, aa)\}$ とする．もし，$(i_1, i_2, \ldots)$ が解だとすると $i_1 = 2$ でなければならない．すると，y 側に欠けている b を補うために，$i_2 = 1$ でなければならない．すると，再び y 側に欠けている b を補うために，$i_3 = 1$ でなければならない．以後，同じことの繰り返しで，P_3 は有限長の解をもたない．□

問 8.2. PCP P のアルファベット Σ を単元集合 (singleton: $|\Sigma| = 1$) に限定するなら，PCP は可解であることを示せ．

PCP が決定不能であることを，TM の帰属問題を PCP に還元することによって証明する：

定理 8.1. [†2] ポストの対応問題は決定不能である．

証明 補題 8.2 を使うために，TM に対する帰属性判定問題 (TM-Membership) をポストの対応問題（PCP）に還元する．すなわち，$L_U \leq_m L_{\mathrm{PCP}}$ を示す．ここで，L_{PCP} はポストの対応問題を符号化した言語であり，L_U は 7.1 節の定理 7.1 で定義した “万能言語” である．以後，このような形式的な記述はせず，アンダーラインしたようないい方をし，TM-Membership $\leq_m$ PCP のように表す．

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ を $\Sigma = \{0,1\}$ を入力アルファベットとする任意の 1 テープ DTM とし，$x \in \{0,1\}^*$ とする．

$$x \in L(M) \iff P_{M,x} \text{ が解を持つ}$$

が成り立つように MPCP のインスタンス $P_{M,x}$ を M と x から構成するアルゴリズムを以下に示す．形式的には，M, x を符号化したコード $\overline{M}\#x$ を入力として，

$$\overline{M}\#x \in L_U \iff N(\overline{M}\#x) = \overline{P_{M,x}} \in L_{\mathrm{PCP}}$$

を満たすような $P_{M,x}$ のコード $\overline{P_{M,x}}$ を出力する変換機 N を構成しなければならないが，今後は上述のように，“符号化” は（具体的に述べようとすると大変であるが，やろうと思えばできるので）具体的には明示しない．また，停止性変換機とは “変換アルゴリズム” にほかならないことをあらためて注意する．さらに，M は受理状態に入ったら停止すると仮定してもよいことにも注意する．

$Q \cap \Gamma = \emptyset$ とし，M の ID を uqv で表すことにする（$uv \in \Gamma^*$ は M のその時点のテープ内容，q はそのときの状態で，ヘッドは v の先頭記号を指している）．$\mathbb{C}, \$, \#$ を $Q \cup \Gamma$ にない新しい記号とし，$\Delta = Q \cup \Gamma \cup \{\#\}$ とおく．さらに，Δ の各記号 X に対し新しい記号 X' を導入し，$\Delta' = \{X' \mid X \in \Delta\}$ とおく．$\xi \in \Delta^*$ に対し，ξ の中の各記号 X を X' で置き換えたものを ξ' で表すことにする．

$P_{M,x} \subseteqq (\Delta \cup \{\mathbb{C}, \$\} \cup \Delta')^* \times (\Delta \cup \{\mathbb{C}, \$\} \cup \Delta')^*$ を，図 8.2.2 のように構成する．M と x から（事実上，M だけから）この表を出力するアルゴリズムが N であるが，そのような N が存在することは明らかであろう．

$P_{M,x}$ は入力として x が与えられたときの M の動作を模倣するように作られている．$(i_1, \ldots, i_k)$ が $P_{M,x}$ の解であるとき，$(i_1, \ldots, i_j)$, $j \leqq k$, をその**途中解**と呼び，$x_{i_1} \cdots x_{i_j}$, $y_{i_1} \cdots y_{i_j}$ をそれぞれこの途中解の x **部**，y **部**と呼ぶ．

$q_0 x \vdash_M \alpha_1 q_1 \beta_1 \vdash_M \alpha_2 q_2 \beta_2 \vdash_M \alpha_3 q_3 \beta_3 \vdash_M \cdots \vdash_M \alpha_{k-1} q_{k-1} \beta_{k-1} \vdash_M \alpha_k q_k \beta_k$ （各 $q_i \in Q$, 各 $\alpha_i \beta_i \in \Gamma^*$）

とする（仮に，k は偶数とする．奇数の場合も同様）．このとき，$P_{M,x}$ は，x 部，y 部が

$$x \text{ 部：} \quad \mathbb{C} q_0 x \# \alpha_1' q_1' \beta_1' \#' \alpha_2 q_2 \beta_2 \# \alpha_3' q_3' \beta_3' \#' \cdots \# \alpha_{k-1}' q_{k-1}' \beta_{k-1}' \#'$$
$$y \text{ 部：} \quad \mathbb{C} q_0 x \# \alpha_1' q_1' \beta_1' \#' \alpha_2 q_2 \beta_2 \# \alpha_3' q_3' \beta_3' \#' \cdots \# \alpha_{k-1}' q_{k-1}' \beta_{k-1}' \#' \alpha_k q_k \beta_k \#$$

[†2] E.Post, A variant of a recursively unsolvable problem, *Bull. Amer. Math. Soc.* 52, pp.264–268, 1946.

	x_i のリスト	y_i のリスト	条件
グループ 1	$\not\vdash$	$\not\vdash q_0 x \#$	
グループ 2	X X'	X' X	$X \in Q \cup \Gamma \cup \{\#\}$
グループ 3	ZqX $Z'q'X'$	$p'Z'Y'$ pZY	$\delta(q, X) = (p, Y, -1)$
	qX $q'X'$	$p'Y'$ pY	$\delta(q, X) = (p, Y, 0)$
	qX $q'X'$	$Y'p'$ Yp	$\delta(q, X) = (p, Y, 1)$
	$Zq\#$ $Z'q'\#'$	$p'Z'Y'\#'$ $pZY\#$	$\delta(q, \square) = (p, Y, -1)$
	$q\#$ $q'\#'$	$p'Y'\#'$ $pY\#$	$\delta(q, \square) = (p, Y, 0)$
	$q\#$ $q'\#'$	$Y'p'\#'$ $Yp\#$	$\delta(q, \square) = (p, Y, 1)$
グループ 4	XqY $X'q'Y'$ Xq $X'q'$ qY $q'Y'$	q' q q' q q' q	$X, Y \in \Gamma,\ q \in F$
グループ 5	$\#q'\$$ $\#'q\$$	$\$$ $\$$	$q \in F$

図 8.2.2: $P_{M,x}$ の構成

であるような途中解をもつ．x 部は，つねに y 部より 1 ステップ遅れながら M の動作を模倣する．M の奇数番目のステップに対応する部分語はダッシュ（′）が付いた語である．この部分語はグループ 1 ～ 3 だけを使って得られる．もし $q_k \in F$ であれば，このあとさらにグループ 4 と，最後にグループ 2 を適用してすべてのテープ記号を消去し，

$$x \text{ 部：} \quad \not\vdash q_0 x \# \alpha'_1 q'_1 \beta'_1 \#' \alpha_2 q_2 \beta_2 \# \cdots \alpha'_{k-1} q'_{k-1} \beta'_{k-1} \#' \alpha_k q_k \beta_k \# \cdots$$
$$y \text{ 部：} \quad \not\vdash q_0 x \# \alpha'_1 q'_1 \beta'_1 \#' \alpha_2 q_2 \beta_2 \# \cdots \alpha'_{k-1} q'_{k-1} \beta'_{k-1} \#' \alpha_k q_k \beta_k \# \cdots \#' q_k$$

を得て（$\alpha_k q_k \beta_k$ の内容次第で $\#' q_k$ は $\# q'_k$ になる），最後のステップとしてグループ 5 を適用して，解

$$\not\vdash q_0 x \# \alpha'_1 q'_1 \beta'_1 \#' \alpha_2 q_2 \beta_2 \# \cdots \alpha'_{k-1} q'_{k-1} \beta'_{k-1} \#' \alpha_k q_k \beta_k \# \cdots \#' q_k \$$$

が得られる．

以上述べたように，$x \in L(M)$ ならば $P_{M,x}$ は解をもつ．

逆に，$P_{M,x}$ が解をもつならば $x \in L(M)$ であることを示す．最初に，$P_{M,x}$ の解はダッシュが付いた部分語と付かない部分語が交互に現れるようなものしかないことに注意する．また，解はグループ 1 で始まりグループ 5 で終わるものしかないことが容易にわかる．あとは，解の長さに関する帰納法で $x \in L(M)$ であることを証明すればよい． □

問 8.3. 次の DTM M に対して，定理 8.1 の証明中で構成した PCP のインスタンスを示し，$x = 01$ を M に入力したときの受理計算に対応する PCP の解を求めよ．

$$M = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,\square\}, \delta, q_0, \{q_2\}),$$

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$	$\delta(q_i, \square)$
q_0	$(q_1, 1, 1)$	$(q_1, 0, -1)$	$(q_1, 1, -1)$
q_1	$(q_2, 0, -1)$	$(q_0, 0, 1)$	$(q_1, 0, 1)$
q_2	–	–	–

問 8.4. 定理 8.1 の証明ではステップの偶奇がわかるように PCP を構成したが，TM の初期ステップに PCP の第 1 ペアが必ず対応するようにすると PCP のルールを少し簡単にできる．

PCP (8.2.1) において，解 (8.2.3) が $i_1 = 1$ でなければならないと制約したものを制約 PCP (modified PCP; **MPCP**) と呼ぶことにする．次の補題をヒントに従って証明せよ．

補題 8.3. MPCP $\leq_m$ PCP. すなわち，MPCP が決定不能ならば PCP も決定不能である．

【ヒント】MPCP のインスタンス P から PCP のインスタンス P' を次のように定義する．¢ と \$ を Σ の元でない記号とし，x_i $(1 \leqq i \leqq n)$ の各文字の後ろに ¢ を挿入してできる語を x'_i とし，y_i の各文字の前に ¢ を挿入してできる語を y'_i とする．また，

$$\begin{array}{rclcrcl} x'_0 & := & ¢x_1, & & y'_0 & := & y_1 \\ x'_{n+1} & := & \$ & & y'_{n+1} & := & ¢\$ \end{array}$$

と定義し，P' を

$$P' := \{(x'_0, y'_0), (x'_1, y'_1), \ldots, (x'_n, y'_n)\}$$

と定義する．このとき，P が解をもつ必要十分条件は P' が解をもつことである．【ヒント終わり】

問 8.5. ポストの対応問題が決定不能であることを証明するために，下記の MPCP のインスタンスを考える．TM-Membership $\leq_m$ MPCP であることを証明せよ．

	x_i のリスト	y_i のリスト	条件
グループ 1	$\#$	$\#q_0x\#$	
グループ 2	X	X	$X \in \Gamma$
	$\#$	$\#$	
グループ 3	ZqX	pZY	$\delta(q, X) = (p, Y, -1)$
	qX	pY	$\delta(q, X) = (p, Y, 0)$
	qX	Yp	$\delta(q, X) = (p, Y, 1)$
	$Zq\#$	$pZY\#$	$\delta(q, \square) = (p, Y, -1)$
	$q\#$	$pY\#$	$\delta(q, \square) = (p, Y, 0)$
	$q\#$	$Yp\#$	$\delta(q, \square) = (p, Y, 1)$
グループ 4	XqY	q	$X, Y \in \Gamma,\ q \in F$
	Xq	q	
	qY	q	
グループ 5	$q\#\#\$$	$\#$	$q \in F$

系 8.1. PCP は決定不能である． □

問 8.6. 補題 8.2，定理 8.1 の証明からわかるように，ある問題 B が決定不能であることを証明するには，決定不能であることがすでにわかっている問題 A を B へ還元できればよい．すなわち，「B が決定可能であるとすると A も決定可能になってしまう」ような問題 A を見つければよい．次の各問題に還元できる決定不能問題を見つけよ．(a) は TM の停止問題を還元せよ．

(a) M_1, M_2 を TM とするとき，$L(M_1) \subseteqq L(M_2)$ か？
(b) M_1 と M_2 は同値か？ すなわち，$L(M_1) = L(M_2)$ か？
(c) $L(M_1) - L(M_2) = \emptyset$ か？
(d) $L(M_1) \cap L(M_2) = \emptyset$ か？

問 8.7. 補題 8.2 は決定可能でない問題を示すのには有効であるだけでなく，決定可能問題を示すのにも使える．すなわち，問題 A を決定可能であることがすでにわかっている（あるいは，決定可能であることが容易にわかる）問題 B へ還元できれば，A は決定可能である．次の各問題を決定可能な問題へ還元せよ：

(a) 与えられたグラフがオイラーグラフであるか否かを決定すること．

(b) 平面上に凸多角形 (その頂点の座標 $(x_i, y_i), 1 \leqq i \leqq n$，によって与えられるとする) と，1 点 (x, y) が与えられたとき，(x, y) がその凸多角形の内部の点であるか否かを決定すること．

(c) $ax + by = c$ $(a, b, c \in \mathbb{N})$ が整数解 (x, y) を持つか否かを判定すること．

補題 8.2（問 8.6, 8.7）は決定可能性や決定不能性を示すのに有効であるものの，変換の計算量に制約がないので決定可能問題の場合の計算量の上界を示すのには使うことはできない．そこで，変換に要する計算量を小さめに抑えた還元法，$\leq_{poly}$ や $\leq_{log}$，に限定すると，次のことが成り立つ：

補題 8.4. (1) $\mathcal{C} \in \{\mathbf{P}, \mathbf{NP}, \mathbf{PSPACE}\}$ のとき，$L \leq_{poly} L'$ かつ $L' \in \mathcal{C} \implies L \in \mathcal{C}$.

(2) k を正整数とし，$\mathcal{C} \in \{\mathbf{P}, \mathbf{NP}, \mathbf{PSPACE}, \mathbf{DSPACE}(\log^k n), \mathbf{NSPACE}(\log^k n)\}$ のとき，$L \leq_{log} L'$ かつ $L' \in \mathcal{C} \implies L \in \mathcal{C}$.

証明 (1) $\mathcal{C} = \mathbf{P}$ の場合についてだけ示す．他の場合も証明は同様である．

$L \leq_{poly} L'$ via M とし，M は $p(n)$ 時間限定であるとする．また，L' を受理する $q(n)$ 時間限定 DTM を $M_{L'}$ とする．ただし，$p(n)$, $q(n)$ は多項式である（時間量の定義から，これらは単調増加関数である）．DTM M_L を図 8.2.1 のように作ると M_L は L を受理する．M は $p(n)$ 時間限定であるから，入力 x に対する M の出力 $M(x)$ は

$$|M(x)| \leqq p(|x|) \tag{8.2.4}$$

を満たす．また，M' は $q(n)$ 時間限定であるから，入力 $M(x)$ に対する計算の時間量は，(8.2.4) より，たかだか

$$q(|M(x)|) \leqq q(p(|x|)) \tag{8.2.5}$$

である．M による x の変換にかかる時間 $p(|x|)$ を考慮すると，入力 x に対して M_L が使う時間量は，(8.2.5) と合算してたかだか

$$p(|x|) + q(p(|x|))$$

であり，これは $|x|$ の多項式である．よって，$L \in \mathbf{P}$ である．

(2) 略. □

問 8.8. 補題 8.4 の残りを証明せよ．補題 8.1 の証明を参考にするとよい．

問 8.9. 2 進数 $n_2 \in \{0,1\}^*$ に関するある決定問題を 10 進数 $n_{10} \in \{0, 1, \ldots, 9\}^*$ に関する同じ決定問題に還元するアルゴリズムを示せ．それは多項式時間還元か？ 対数領域還元か？ いずれでもないか？

問 8.10. $\leq_m$ の下では補題 8.4 の (1) が成り立たないことを示せ．また，$\mathbf{NL} \subsetneqq \mathbf{P}$ と仮定すると（$\mathbf{NL} \subsetneqq \mathbf{P}$ であるかどうかは現在のところ未解決である），$\leq_{poly}$ の下では $\mathcal{C} \in \{\mathbf{DSPACE}(\log^k n), \mathbf{NSPACE}(\log^k n)\}$ に対して (2) が成り立たないことを示せ．

問 8.11. （問 8.1 も参照せよ）

$A \leq_{poly} B$ かつ $B \leq_{poly} A$ であるとき $A \equiv_{poly} B$ と定義すると $\equiv_{poly}$ は同値関係である．$\equiv_{poly}$ の同値類を**多項式時間 m 級** (polynomial-time m-degree) という．次のことを示せ．

(1) 多項式時間 m 級の全体は順序 $\leq_{poly}$ の下で半順序集合である．

(2) $\mathbf{P}$ は $\leq_{poly}$ の下で最小元である．極大な m 級は存在しない．

8.3 完全問題

$\mathcal{C}$ を問題（＝言語）のクラスとする．言語 L_0 が多項式時間（対数領域）還元性に関して $\mathcal{C}$ **困難** ($\mathcal{C}$-hard) であるとは，$\mathcal{C}$ の任意の元 L が L_0 に多項式時間（対数領域）還元可能なことをいう．$\mathbf{NP} \subseteqq \mathcal{C}$ の場合，L_0 が 多項式時間還元性に関して $\mathcal{C}$ 困難かつ $L_0 \in \mathcal{C}$ であるとき，L_0 は $\mathcal{C}$ **完全** ($\mathcal{C}$-complete) であるという．$\mathbf{NL} \subseteqq \mathcal{C} \subseteqq \mathbf{P}$ の場合，L_0 が 対数領域還元性に関して $\mathcal{C}$ 困難かつ $L_0 \in \mathcal{C}$ であるとき，L_0 は $\mathcal{C}$ 完全であるという．

これまで見てきたように（問題 8.1，補題 8.1 参照），還元 $\leq_{poly}$ や $\leq_{log}$ は問題の複雑さを表す順序に相当していることに注意する．したがって，粗っぽくいうと，$\mathcal{C}$ 困難問題とは $\mathcal{C}$ のどの問題よりも難しい（計算量が大きい）問題のことであり（$\mathcal{C}$ に属すとは限らない），$\mathcal{C}$ 完全問題とは $\mathcal{C}$ の中で最も難しい問題のことである．例えば，**NP 完全問題**とは，

- $L_0 \in \mathbf{NP}$ かつ
- 任意の $L \in \mathbf{NP}$ に対して $L \leq_{poly} L_0$

が成り立つような L_0 のことである．

完全性の定義をする際，**NP** を含むクラスと **P** に含まれるクラスに対して還元可能性を使い分けている理由は，$\mathcal{C}$ に属する問題を分類するためには，還元に要する計算量が $\mathcal{C}$ の問題自体が持つ計算量よりも小さくなければならないからである．しかし，**NP** 以上のクラスに対して完全であることが証明されている問題の多くは対数領域還元性に関しても完全であることが知られている．

問 8.12. A が $\mathcal{C}$ 完全で $A \leq_{log} B$, $B \in \mathcal{C}$ ならば B も $\mathcal{C}$ 完全であることを示せ．

次の定理が示唆するように，$\mathbf{L} \stackrel{?}{=} \mathbf{NL}$ 問題や $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ 問題など，主要な計算量のクラスの間の未解決な包含関係は完全問題と密接な関係がある:

定理 8.2. それぞれの計算量のクラスについて，以下の関係が成り立つ：

(1) L を **NP** 完全問題または *co***NP** 完全問題とすると，$L \in \mathbf{P} \iff \mathbf{P} = \mathbf{NP}$.
(2) L を *co***NP** 完全問題とすると，$L \in \mathbf{NP} \iff \mathbf{NP} = \mathit{co}\mathbf{NP}$.
(3) L を **PSPACE** 完全問題とすると，$L \in \mathbf{P} \iff \mathbf{P} = \mathbf{PSPACE}$.
(4) L を **P** 完全問題とすると，$L \in \mathbf{NL}$ $(L \in \mathbf{L}) \iff \mathbf{NL} = \mathbf{P}$ $(\mathbf{L} = \mathbf{P})$.
(5) L を **NL** 完全問題とすると，$L \in \mathbf{L} \iff \mathbf{L} = \mathbf{NL}$.

証明. (1) についてだけ証明する．他も同様である．

L を **NP** 完全問題とすると任意の $L' \in \mathbf{NP}$ に対して $L' \leq_{poly} L$ であるから，補題 8.4 (1) より，$L \in \mathbf{P}$ ならば $L' \in \mathbf{P}$ である．よって，$\mathbf{NP} \subseteqq \mathbf{P}$.

同様に，$L \in \mathbf{P}$ が *co***NP** 完全問題であるとすると $\mathit{co}\mathbf{NP} \subseteqq \mathbf{P}$ が導かれるが，**P** が補集合で閉じているので $\mathbf{NP} \subseteqq \mathit{co}\mathbf{P} = \mathbf{P}$. よって，いずれの場合も $\mathbf{P} = \mathbf{NP}$ である．

逆に，$\mathbf{P} = \mathbf{NP}$ ならば $L \in \mathbf{P}$ であることは明らかである． □

問 8.13. 定理 8.2 の残りすべてを証明せよ．

還元可能性とか，完全問題といった概念は，もともと帰納的関数の理論において導入されて豊かな成果をもたらしたものであるが[†3]，クック (S.A.Cook) はこれを巧みに計算量理論に取り込んで **NP** 完全問題の存在を初めて示した[†4]．それにつづいて，カープ (R.M.Karp) が数多くの組合せ論的問題が **NP** 完全になることを示してその重要性を指摘した[†5]．これを契機として各種のクラスに対する完全問題の研究が活発に行なわれ，その後の計算量理論の発展の大きな原動力となった．

クックが最初に示した，クラス **NP** に対する完全問題は**ブール式の充足可能性判定問題** (SAT（サット）: satisfiability problem for Boolean expressions) である．

入力: ブール式 E
出力: E が充足可能か否か

ここで，$x_1, \ldots, x_n$ を $\{0,1\}$ 上を動く変数とし，否定（$\neg$ あるいは $\overline{\ }$），論理積（$\wedge$ あるいは $\cdot$），論理和（$\vee$ あるいは $+$）を演算子とする**論理式**（**ブール式**）(logical formula, Boolean formula)

$$f(x_1, \ldots, x_n) : \{0, 1\}^n \mapsto \{0, 1\}$$

（$0, 1$ は false, true に相当する）が**充足可能** (satisfiable) であるとは，$f(x_1, \ldots, x_n) = 1$ となる付値 (value assignment) $x_1, \ldots, x_n \in \{0,1\}$ が存在することをいう．充足可能でないとき，**充足不能** (unsatisfiable) であるという．

例えば，$f_1(x_1, x_2) = x_1 x_2$ は $f_1(1,1) = 1$ であるから充足可能であり，$f_2(x_1, x_2, x_3) = x_1(x_2 + x_3)\overline{x_1}$ はいかなる付値 $x_1, x_2, x_3 \in \{0,1\}$ に対しても $f_2(x_1, x_2, x_3) = 0$ であるから充足不能である．

定理 8.3 (クック・レヴィンの定理)．[†6] SAT は **NP** 完全である．

[†3] 例えば，H.Rogers Jr., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967 を参照せよ．

[†4] S.A.Cook, The complexity of theorem-proving procedures, *Proc. 3rd Annual ACM Symp. on the Theory of Computing*, pp.151–158, 1971. この論文ではチューリング還元（後述）が用いられた．

[†5] R.M.Karp, Reducibilities among combinatorial problems, in *Complexity of Computer Computations* edited by J.W.Thatcher and R.E.Miller, Plenum Press, pp.85–103, 1972. この論文で多項式時間還元が導入された．

[†6] この定理はレビン (L.Levin) も独立に証明したのでクック-レヴィンの定理と呼ばれている：L.Levin, Universal search problems, *Problems of Information Transmission* 9, pp.115–116, 1973（原文はロシア語）．