3.4. 還元可能性と完全性

- 問題の還元可能性
  …問題の相対的な難しさを測る方法
- 問題のあるクラスに関する完全性
  …そのクラス内で最も難しいことを示す方法

クラスREに属している集合の“難しさ”の比較

Aは帰納的だがBは帰納的でないとき，
BはAより難しいと言える．

では，AとBが共に帰納的でない場合は？
← 帰納的還元性による比較

A, B : 集合
AをBへ還元する ⇐ Aの認識問題をBの認識問題に言い換えること．
（AはBへ還元可能）

3.4. Reducibility and Completeness

Comparison of sets in the class RE by their “hardness”

If A is recursive but B is not recursive, then we can say that B is *harder* than A.

Then, what about if neither A or B is recursive?

$\leftarrow$ comparison based on reducibility

A, B : sets

Reduce A to B $\leftarrow$ Replace the recognition problem of A with the recognition problem of B.

(A is reducible to B)

定義3.4:

A, B :任意の集合

(1) 次の条件を満たす関数 h をAからBへの帰納的還元という.

(a) h は Σ^* からΣ^* への関数(全域的)

(b) $\forall x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B]$

(c) h は計算可能

(2) AからBへの帰納的還元が存在するとき,

AはBへ帰納的に還元可能という.

なお, AがBへ帰納的還元可能であることを$A \leq_m B$と記述する.

(m は, recursive *m*any-one reduction の m)

Definition 3.4:

A, B : arbitrary sets

(1) A function h is recursive reduction from A to B if

(a) h is a total function from Σ^* to Σ^*

(b) $\forall x \in \Sigma^* [x \in A \leftrightarrow h(x) \in B]$

(c) h is computable.

(2) If there is a recursive reduction from A to B,

we say that A is recursively reducible to B.

By $A \leq_m B$ we express that A is recursively reducible to B.

(the *m* in the suffix indicates recursive *m*any-one reduction)

例3.10

EVEN＝{ $\lceil n \rceil$:nは偶数}， ODD＝{ $\lceil n \rceil$:nは奇数}

$\lceil n \rceil$ はnの２進表記（n: 自然数）

$$h_1(x) \equiv \begin{cases} \lceil n+1 \rceil & x = \lceil n \rceil \text{となっているとき} \\ x, & \text{その他のとき} \end{cases}$$

この h_1 は明らかに全域的かつ計算可能．また，

$$\forall x \in \Sigma^*[x \in \mathrm{EVEN} \leftrightarrow h_1(x) \in \mathrm{ODD}]$$

よって，h_1 はEVENからODDへの帰納的還元

$$\therefore \mathrm{EVEN} \ \leq_m \ \mathrm{ODD}$$

同じ h_1 がODDからEVENへの帰納的還元にもなっている．

$$\forall x \in \Sigma^*[x \in \mathrm{ODD} \rightarrow h_1(x) \in \mathrm{EVEN}]$$

$$\forall x \in \Sigma^*[h_1(x) \in \mathrm{EVEN} \rightarrow \exists n \geq 0[h_1(x) = \lceil n+1 \rceil \in \mathrm{EVEN}]]$$

$$\rightarrow \exists n \geq 1[h_1(x) = \lceil n+1 \rceil \in \mathrm{EVEN}]]$$

$$\rightarrow \exists n \geq 1[x = \lceil n \rceil \in \mathrm{ODD}]] \rightarrow [x \in \mathrm{ODD}]$$

$$\therefore \mathrm{ODD} \leq_m \mathrm{EVEN}$$

Ex.3.10

EVEN$=\{\lceil n \rceil : n$ is even$\}$, ODD$=\{\lceil n \rceil : n$ is odd$\}$

$\lceil n \rceil$ is binary representation of n (n: natural number)

$$h_1(x) \equiv \begin{cases} \lceil n+1 \rceil & \text{if } x = \lceil n \rceil \\ x, & \textit{otherwise} \end{cases}$$

This h_1 is obviously total and computable. Also,

$$\forall x \in \Sigma^*[x \in \text{EVEN} \leftrightarrow h_1(x) \in \text{ODD}]$$

Therefore, h_1 is a recursive reduction from EVEN to ODD.

$$\therefore \text{EVEN} \le_m \text{ODD}$$

The same h_1 is also a recursive reduction from ODD to EVEN.

$$\forall x \in \Sigma^*[x \in \text{ODD} \to h_1(x) \in \text{EVEN}]$$

$$\begin{aligned} \forall x \in \Sigma^*[h_1(x) \in \text{EVEN} &\to \exists n \ge 0[h_1(x) = \lceil n+1 \rceil \in \text{EVEN}]] \\ &\to \exists n \ge 1[h_1(x) = \lceil n+1 \rceil \in \text{EVEN}]] \\ &\to \exists n \ge 1[x = \lceil n \rceil \in \text{ODD}]] \to [x \in \text{ODD}] \end{aligned}$$

$$\therefore \text{ODD} \le_m \text{EVEN}$$

EVENからODDへのもっと単純な還元

$$h_2(x) \equiv \begin{cases} 1 & x \in \mathrm{EVEN}\text{のとき} \\ 2 & \text{その他のとき} \end{cases}$$

自然数の偶奇が判定可能なので，h_2 は計算可能

$1 \in \mathrm{ODD},\ 2 \notin \mathrm{EVEN}$だから

$$x \in \mathrm{EVEN} \rightarrow h_2(x) = 1 \in \mathrm{ODD}$$

$$x \notin \mathrm{EVEN} \rightarrow h_2(x) = 2 \notin \mathrm{ODD}$$

$$\therefore x \in \mathrm{EVEN} \leftrightarrow h_2(x) \in \mathrm{ODD}$$

Simpler reduction from EVEN to ODD

$$h_2(x) \equiv \begin{cases} 1 & x \in \text{EVEN} \\ 2 & \text{otherwise} \end{cases}$$

Since odd-evenness of a natural number is computable, so is h_2.

Since $1 \in \text{ODD}$, $2 \notin \text{EVEN}$

$$x \in \text{EVEN} \rightarrow h_2(x) = 1 \in \text{ODD}$$

$$x \notin \text{EVEN} \rightarrow h_2(x) = 2 \notin \text{ODD}$$

$$\therefore x \in \text{EVEN} \leftrightarrow h_2(x) \in \text{ODD}$$

定理3.12: $A \leq_m B$という関係にある任意の集合A,Bを考える.
このとき, Bが帰納的→Aも帰納的.

証明:

$A \leq_m B$➔ AからBへの帰納的還元 h が存在する.

よって, $x \in A$という判定問題 ➔ $h(x) \in B$？

つまり, 次のプログラムはAを認識する.

```
prog A(input x);
begin
   if h(x)∈B then accept else reject end-if
end.
```

Bが帰納的なら, Bを認識するプログラムが存在する.

→$h(x) \in B$を判定するプログラム

これで上記のプログラムAが完成.

よって, Aは帰納的.　　　　証明終

Theorem 3.12: Consider any sets A and B such that $A \leq_m B$. Then, B is recursive → A is also recursive.

Proof:

$A \leq_m B$ ➔ there is a recursive reduction h from A to B.

So, the decision problem of $x \in A$ ➔ $h(x) \in B$?

That is, the following program recognizes A.

```
prog A(input x);
begin
   if h(x) ∈ B then accept else reject end-if
end.
```

If B is recursive, there is a program that recognizes B.

→a program that determines $h(x) \in B$

Now, we have a complete program A.

Thus, A is recursive. Q.E.D.

与えられた集合が“手に負えない”ことを示すための方法を示唆

(i) $A \leq_m B$ かつ
(ii) Aは帰納的でない.

このような集合Aを示せれば，Bは帰納的でない

例3.11:

ZERO $\equiv$ {a: IsProgram(a) $\wedge$ $\forall$x[f_a(x) = 0]}

ZEROFT $\equiv$ {a: IsForTimes(a) $\wedge$ $\forall$x[f_a(x) = 0]}

TOTAL $\equiv$ {a: IsProgram(a) $\wedge$ $\forall$x[f_a(x) $\neq \perp$]}

まとめると

関係	したがって，
$\overline{\text{HALT}} \leq_m$ ZERO	ZERO $\notin$ REC （$\overline{\text{HALT}} \notin$ RECより）
$\overline{\text{HALT}} \leq_m$ ZEROFT	ZEROFT $\notin$ REC （$\overline{\text{HALT}} \notin$ RECより）
ZERO $\leq_m$ TOTAL	TOTAL $\notin$ REC （ZERO $\notin$ RECより）

It suggests a method to show that a given set is "intractable"

(i) $A \leq_m B$ and

(ii) A is not recursive.

⇒ If we can show such a set A, then B is not recursive.

Ex.3.11:

$$\mathrm{ZERO} \equiv \{a : \mathrm{IsProgram}(a) \wedge \forall x[f\_a(x) = 0]\}$$

$$\mathrm{ZEROFT} \equiv \{a : \mathrm{IsForTimes}(a) \wedge \forall x[f\_a(x) = 0]\}$$

$$\mathrm{TOTAL} \equiv \{a : \mathrm{IsProgram}(a) \wedge \forall x[f\_a(x) \neq \perp]\}$$

Summarizing,

relation	what follows	
$\overline{\mathrm{HALT}} \leq_m \mathrm{ZERO}$	$\mathrm{ZERO} \notin \mathrm{REC}$	(by $\overline{\mathrm{HALT}} \notin \mathrm{REC}$)
$\overline{\mathrm{HALT}} \leq_m \mathrm{ZEROFT}$	$\mathrm{ZEROFT} \notin \mathrm{REC}$	(by $\overline{\mathrm{HALT}} \notin \mathrm{REC}$)
$\mathrm{ZERO} \leq_m \mathrm{TOTAL}$	$\mathrm{TOTAL} \notin \mathrm{REC}$	(by $\mathrm{ZERO} \notin \mathrm{REC}$)

定理3.13. $A \leqq_m B$という関係にある任意の集合A, Bを考える.
このとき、次のことが成り立つ。
(1) B∈RE →A∈RE （Bが枚挙可能 → Aも枚挙可能）
(2) B∈co-RE →A∈co-RE

(補注) 対偶を考えると、
(1) A∉RE → B∉RE
(2) A∉co-RE → B∉co-RE

例3.11, 定理3.13 → ZERO、TOTALは
REにもco-REにも属さない。

性質	理由
ZERO∉RE	$\overline{\text{HALT}}$∉RE、$\overline{\text{HALT}} \leqq_m$ ZERO
ZERO∉co-RE	HALT∉co-RE、HALT $\leqq_m$ ZERO
TOTAL∉RE	ZERO∉RE、ZERO $\leqq_m$ TOTAL
TOTAL∉co-RE	ZERO∉co-RE、ZERO $\leqq_m$ TOTAL

Theorem 3.13. Consider any sets A and B such that $A \leqq_m B$. Then, we have:

(1) $B \in \mathrm{RE} \rightarrow A \in \mathrm{RE}$ (B is enumerable → so is A)

(2) $B \in \text{co-RE} \rightarrow A \in \text{co-RE}$

Remark Consider their contrapositions:

(1) $A \notin \mathrm{RE} \rightarrow B \notin \mathrm{RE}$

(2) $A \notin \text{co-RE} \rightarrow B \notin \text{co-RE}$

Ex.3.11, Theorem 3.13 → Neither ZERO or TOTAL belongs to RE or co-RE.

property	reason
$\mathrm{ZERO} \notin \mathrm{RE}$	$\overline{\mathrm{HALT}} \notin \mathrm{RE}$、$\overline{\mathrm{HALT}} \leqq_m \mathrm{ZERO}$
$\mathrm{ZERO} \notin \text{co-RE}$	$\mathrm{HALT} \notin \text{co-RE}$、$\mathrm{HALT} \leqq_m \mathrm{ZERO}$
$\mathrm{TOTAL} \notin \mathrm{RE}$	$\mathrm{ZERO} \notin \mathrm{RE}$、$\mathrm{ZERO} \leqq_m \mathrm{TOTAL}$
$\mathrm{TOTAL} \notin \text{co-RE}$	$\mathrm{ZERO} \notin \text{co-RE}$、$\mathrm{ZERO} \leqq_m \mathrm{TOTAL}$

還元可能性　：　難しさを比較する手段

$A \leqq_m B \rightarrow A$の認識問題を$B$の認識問題に変換できる。

↓

Aの難しさ $\leqq$ Bの難しさ

（Bを認識するプログラムがあればAの認識に使える。）

定理3.14.

任意に与えられた集合 A, B, C に対し、次の関係が成り立つ

(1) $A \leqq_m A$

(2) $A \leqq_m B$ かつ $B \leqq_m C$ ならば $A \leqq_m C$

$A \equiv_m B \overset{\text{def}}{\leftrightarrow} A \leqq_m B$ かつ $B \leqq_m A$

$\equiv_m$ は同値関係（同程度の難しさ）

$A \equiv_m B$ のとき、A と B は $\equiv_m$-同値という。

Reducibility : a means of comparing hardness

$A \leqq_m B$ → We can convert the recognition problem of A into that of B.

↓

hardness of $A \leq$ hardness of B

（A program recognizing B can be used to recognize A.）

Theorem 3.14. For any given sets A, B, C, we have

（1） $A \leqq_m A$

（2） $A \leqq_m B$ and $B \leqq_m C$ implies $A \leqq_m C$

$A \equiv_m B \overset{\text{def}}{\leftrightarrow} A \leqq_m B$ and $B \leqq_m A$

$\equiv_m$ is an equivalence relation (equal hardness)

If $A \equiv_m B$, we say that A and B are $\equiv_m$-equivalent.

例3.13.

ZERO $\notin$ RE　　$\therefore$ ZERO $\not\leq_m$ HALT

（$\because$ ZERO $\leq_m$ HALTとすると、HALT $\in$ REなので
ZERO $\in$REとなり矛盾）

一方、HALT $\leq_m$ ZERO

$\therefore$ZEROはHALTより真に難しい。

例3.14.

すべての帰納的集合は互いに帰納的に同値。
たとえば、EVEN（偶数の集合）とPRIME（素数の集合）は
帰納的に同値

$$\text{EVEN} \equiv_m \text{PRIME}$$

（両方とも帰納的という意味で同程度の難しさ）

どちらも計算できるという
意味で同程度に難しい

Ex. 3.13.

ZERO $\notin$ RE $\quad\therefore$ ZERO $\not\leq_m$ HALT

($\because$ if ZERO $\leq_m$ HALT we have HALT $\in$ RE and ZERO $\in$ RE, a contradiction)

On the other hand, HALT $\leq_m$ ZERO

$\therefore$ ZERO is strictly harder than HALT.

Ex. 3.14.

All the recursive sets are recursively equivalent to each other.

For example, EVEN (set of even numbers) and PRIME (set of primes) are recursively equivalent

EVEN $\equiv_m$ PRIME

(both of them are equally hard in the sense that they are recursive.)

both computable

“クラスREの中で最も難しい集合”の定義

(one of the most difficult sets in RE)

定義3.5.

集合 A が次の条件を満たすとき、それを($\leqq_m$ のもとで)
RE-完全(RE-complete)という。

(a) $\forall L \in \mathrm{RE}$ [$L \leqq_m \mathrm{A}$]

(A より真に難しいものはREには存在しない

(b) $\mathrm{A} \in \mathrm{RE}$

集合Aが上記の条件(a)だけを満たすとき、
RE-困難(RE-Hard)という。

(すべてのRE集合より難しい集合のこと)

Definition of "**the hardest sets in the class RE**"

Def. 3.5.
A set A is called RE-complete (under $\leqq_m$) if the following conditions hold

(a) $\forall L \in \mathrm{RE}\ [L \leqq_m A]$

(no element of RE is strictly harder than A).

(b) $A \in \mathrm{RE}$

If a set A satisfies only (a) above, it is called RE-hard.

(meaning sets harder than any RE set)

定理3.15: HALTはRE-完全

（証明）

HALT ∈ REなので、条件(b)はOK。

L: 任意のRE集合とする。

→Lを半認識するプログラム L が存在する

すべての $x \in \Sigma^*$ に対し、

$x \in L$ ⟺ Halt(⌜L⌝, x) ⟺ <⌜L⌝, x> ∈ HALT

よって、$h(x) \overset{\mathrm{def}}{=}$ <⌜L⌝, x> は L からHALTへの帰納的還元。

（証明終）

Theorem 3.15 HALT is RE-complete.

(Proof)

Since HALT $\in$ RE, the condition (b) is satisfied.

L : any RE set.

$\rightarrow$ a program L that semi-recognizes L.

for any $x \in \Sigma^*$

$$x \in L \Longleftrightarrow \text{Halt}(\lceil \mathrm{L} \rceil, x) \Longleftrightarrow < \lceil \mathrm{L} \rceil, x > \in \text{HALT}$$

Thus, $h(x) \stackrel{\text{def}}{=} < \lceil \mathrm{L} \rceil, x >$ is a recursive reduction from L to HALT.

Q.E.D.

定理3.16: A, B を任意の集合とする。
（１）[AがRE-困難] かつ [$A \leqq_m B$] ならば B はRE-困難
（２）A がRE-困難 $\leftrightarrow$ $\overline{A}$ がco-RE-困難

例3.15. 定理3.16 を用いて、いろいろな集合の困難性（完全性）を示す。

集合	難しさ	主な理由
HALT	RE-完全	定理3. 15
$\overline{\text{HALT}}$	co-RE完全	HALTがRE-困難、$\overline{\text{HALT}}$ ∈co-RE
ZEROFT	co-RE完全	$\overline{\text{HALT}}$がco-RE困難 、$\overline{\text{HALT}} \leq_m$ ZEROFT
$\overline{\text{ZEROFT}}$	RE完全	ZEROFTがco-RE困難 、$\overline{\text{ZEROFT}}$ ∈RE
ZERO	RE-困難、co-RE困難	HALT$\leq_m$ZERO、
TOTAL	RE-困難、co-RE困難	ZERO$\leq_m$TOTAL

Theorem 3.16: Let A and B be arbitrary sets.
(1) $[A$ is RE-hard and $A \leqq_m B]$ implies B is RE-hard.
(2) A is RE-hard $\leftrightarrow$ $\overline{A}$ is co-RE-hard.

Ex.3.15 Using Theorem 3.16, we can show hardness of various sets.

Sets	hardness	reasons
HALT	RE-complete	Theorem3.15
$\overline{\text{HALT}}$	co-REcomplete	HALT is RE-hard, $\overline{\text{HALT}} \in$ co-RE
ZEROFT	co-REcomplete	$\overline{\text{HALT}}$ is co-REhard、$\overline{\text{HALT}} \leq_m$ ZEROFT
$\overline{\text{ZEROFT}}$	REcomplete	ZEROFT is co-REhard、$\overline{\text{ZEROFT}} \in$ RE
ZERO	RE-hard、co-REhard	HALT $\leq_m$ ZERO、
TOTAL	RE-hard、co-REhard	ZERO $\leq_m$ TOTAL

H:RE-完全集合の集合

H:REの中で“最も難しい集合”

REC:REの中で“最もやさしい集合”

還元 $\leq_m$ のもとで

定理3.17.

(1)REC ∩ $H = \phi$

(2)RE−(REC ∪ H) $\neq \phi$

(1)REC $\subsetneq$ RE

REC は同値関係 $\equiv_m$ のもとで閉じている。

(2)の証明は複雑なので省略。

H: an RE-complete set

H: "hardest set" in RE

REC : "easiest set" in RE

Theorem 3.17.

(1) REC $\cap$ H $= \phi$

(2) RE $-$ (REC $\cup$ H) $\neq \phi$

(1) REC $\subsetneqq$ RE

REC is closed under the equivalence relation $\equiv_m$.

(2) The proof is complicated, and so omitted.

重要なお知らせ

- 7月6日（水）は中間テスト
  - ➢場所は大講義室
  - ➢時間は9:20～（30分以上遅刻したら入室禁止）
  - ➢範囲は7月1日の授業分まで（テキスト3章まで）
  - ➢簡単で持込なし または 難しくて持込あり

メモ:
7月6日（水）の午後と
7月13日（水）の午後は補講

重要なお知らせ

- 7月6日（水）は中間テスト
  - 場所は大講義室
  - 時間は9:20～（30分以上遅刻したら入室禁止）
  - 範囲は7月1日の授業分まで（テキスト3章まで）
  - 簡単で持込なし または 難しくて持込あり

メモ:
7月6日（水）の午後と
7月13日（水）の午後は補講