

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

- **Reducibility** of a problem
 - ...Measure of relative hardness of the problem
- **Completeness** of a problem in a class
 - ...Most difficult problem in the class

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 nor B is recursive?

← comparison based on reducibility

A, B : sets

Reduce A to B ← 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,

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

よって, h_1 は EVEN から ODD への帰納的還元

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

同じ h_1 が ODD から EVEN への帰納的還元にもなっている.

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

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

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

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

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

Ex.3.10

EVEN = $\{ \lceil n \rceil : n \text{ is even} \}$, **ODD** = $\{ \lceil n \rceil : n \text{ is odd} \}$
 $\lceil n \rceil$ is binary representation of n (n : natural number)

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

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.

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

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

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

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

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

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

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

EVENからODDへのもっと単純な還元

$$h_2(x) \equiv \begin{cases} 1 & x \in \text{EVEN} \text{ のとき} \\ 10 & \text{その他のとき} \end{cases}$$

自然数の偶奇が判定可能なので, h_2 は計算可能

$1 \in \text{ODD}$, $10 \notin \text{ODD}$ だから

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

$$x \notin \text{EVEN} \rightarrow h_2(x) = 10 \notin \text{ODD}$$

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

Simpler reduction from EVEN to ODD

$$h_2(x) \equiv \begin{cases} 1 & x \in \text{EVEN} \\ 10 & \text{otherwise} \end{cases}$$

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

Since $1 \in \text{ODD}$, $10 \notin \text{ODD}$

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

$$x \notin \text{EVEN} \rightarrow h_2(x) = 10 \notin \text{ODD}$$

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

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 $\rightarrow A$ も帰納的.

証明:

$A \leq_m B \rightarrow A$ 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 $\rightarrow h(x) \in B$?
 つまり, 次のプログラムは A を認識する.

```

prog A(input x);
begin
  if  $h(x) \in B$  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.
 $\rightarrow h(x) \in B$ 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 $\rightarrow A$ is also recursive.

Proof:

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

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

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 .

\rightarrow 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:

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

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

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

まとめると

関係

したがって,

$\overline{\text{HALT}} \leq_m \text{ZERO}$ $\text{ZERO} \notin \text{REC}$ ($\overline{\text{HALT}} \notin \text{REC}$ より)

$\overline{\text{HALT}} \leq_m \text{ZEROFT}$ $\text{ZEROFT} \notin \text{REC}$ ($\overline{\text{HALT}} \notin \text{REC}$ より)

$\text{ZERO} \leq_m \text{TOTAL}$ $\text{TOTAL} \notin \text{REC}$ ($\text{ZERO} \notin \text{REC}$ 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:

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

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

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

Summarizing,

relation

what follows

$\overline{\text{HALT}} \leq_m \text{ZERO} \quad \text{ZERO} \notin \text{REC} \quad (\text{by } \overline{\text{HALT}} \notin \text{REC})$

$\overline{\text{HALT}} \leq_m \text{ZEROFT} \quad \text{ZEROFT} \notin \text{REC} \quad (\text{by } \overline{\text{HALT}} \notin \text{REC})$

$\text{ZERO} \leq_m \text{TOTAL} \quad \text{TOTAL} \notin \text{REC} \quad (\text{by } \text{ZERO} \notin \text{REC})$

定理3.13. $A \leq_m B$ という関係にある任意の集合 A, B を考える。
このとき、次のことが成り立つ。

(1) $B \in \text{RE} \rightarrow A \in \text{RE}$ (B が枚挙可能 $\rightarrow A$ も枚挙可能)

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

(補注) 対偶を考えると、

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

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

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

性質	理由
ZERO \notin RE	$\overline{\text{HALT}} \notin \text{RE}$ 、 $\overline{\text{HALT}} \leq_m \text{ZERO}$
ZERO \notin co-RE	$\text{HALT} \notin \text{co-RE}$ 、 $\text{HALT} \leq_m \text{ZERO}$
TOTAL \notin RE	ZERO \notin RE、ZERO \leq_m TOTAL
TOTAL \notin co-RE	ZERO \notin co-RE、ZERO \leq_m TOTAL

Theorem 3.13. Consider any sets A and B such that $A \leq_m B$.

Then, we have:

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

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

(Remark) Their contrapositions:

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

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

Ex.3.11, Theorem 3.13 \rightarrow Neither **ZERO** or **TOTAL** belongs to **RE** or **co-RE**.

property	reason
$\text{ZERO} \notin \text{RE}$	$\overline{\text{HALT}} \notin \text{RE}, \overline{\text{HALT}} \leq_m \text{ZERO}$
$\text{ZERO} \notin \text{co-RE}$	$\text{HALT} \notin \text{co-RE}, \text{HALT} \leq_m \text{ZERO}$
$\text{TOTAL} \notin \text{RE}$	$\text{ZERO} \notin \text{RE}, \text{ZERO} \leq_m \text{TOTAL}$
$\text{TOTAL} \notin \text{co-RE}$	$\text{ZERO} \notin \text{co-RE}, \text{ZERO} \leq_m \text{TOTAL}$

還元可能性 : 難しさを比較する手段

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



A の難しさ \leq 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) $A \leq_m A$

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

$$A \equiv_m B \stackrel{\text{def}}{\iff} A \leq_m B \text{ かつ } B \leq_m A$$

\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

$A \leq_m B \rightarrow$ 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) $A \leq_m A$

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

$$A \equiv_m B \stackrel{\text{def}}{\iff} A \leq_m B \text{ and } B \leq_m A$$

\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.

$\text{ZERO} \notin \text{RE} \quad \therefore \text{ZERO} \not\leq_m \text{HALT}$

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

一方、 $\text{HALT} \leq_m \text{ZERO}$

$\therefore \text{ZERO}$ 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.

$\text{ZERO} \notin \text{RE} \quad \therefore \text{ZERO} \not\leq_m \text{HALT}$

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

On the other hand, $\text{HALT} \leq_m \text{ZERO}$

$\therefore \text{ZERO}$ 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

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

(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 (\leq_m) のもとで
RE-完全 (RE-complete) という。

$$(a) \forall L \in RE [L \leq_m A]$$

(A より真に難しいものはREには存在しない)

$$(b) A \in RE$$

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 \leq_m) if the following conditions hold

$$(a) \forall L \in \text{RE} [L \leq_m A]$$

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

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

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 \in REなので、条件(b)はOK。

L : 任意のRE集合とする。

$\rightarrow L$ 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、

$$x \in L \iff \text{Halt}(\langle L \rangle, x) \iff \langle \langle L \rangle, x \rangle \in \text{HALT}$$

よって、 $h(x) \stackrel{\text{def}}{=} \langle \langle L \rangle, x \rangle$ は L から HALT への帰納的還元。

(証明終)

Theorem 3.15 HALT is RE-complete.

(Proof)

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

L : any RE set.

→ a program L that semi-recognizes L .

for any $x \in \Sigma^*$

$$x \in L \iff \text{Halt}(\lceil L \rceil, x) \iff \langle \lceil L \rceil, x \rangle \in \text{HALT}$$

Thus, $h(x) \stackrel{\text{def}}{=} \langle \lceil L \rceil, x \rangle$ 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) $[A \text{ が RE-困難}] \text{ かつ } [A \leq_m B]$ ならば B は RE-困難
 (2) $A \text{ が RE-困難} \leftrightarrow A \text{ が co-RE-困難}$

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
<u>HALT</u>	RE-完全	定理3.15
<u>HALT</u>	co-RE完全	<u>HALT</u> がRE-困難、 <u>HALT</u> \in co-RE
<u>ZEROFT</u>	co-RE完全	<u>HALT</u> がco-RE困難、 <u>HALT</u> \leq_m <u>ZEROFT</u>
<u>ZEROFT</u>	RE完全	<u>ZEROFT</u> がco-RE困難、 <u>ZEROFT</u> \in RE
ZERO	RE-困難、co-RE困難	<u>HALT</u> \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) $[A \text{ is RE-hard and } A \leq_m B]$ implies B is RE-hard.
 (2) $A \text{ is RE-hard} \leftrightarrow \overline{A} \text{ is co-RE-hard.}$

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

Sets	hardness	reasons
<u>HALT</u>	RE-complete	Theorem 3.15
HALT	co-REcomplete	<u>HALT</u> is RE-hard, <u>HALT</u> \in co-RE
<u>ZEROFT</u>	co-REcomplete	<u>HALT</u> is co-REhard, <u>HALT</u> \leq_m <u>ZEROFT</u>
ZEROFT	REcomplete	<u>ZEROFT</u> is co-REhard, <u>ZEROFT</u> \in RE
ZERO	RE-hard, co-REhard	<u>HALT</u> \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.

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

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

$$(1) REC \subsetneq RE$$

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

Under the reduction \leq_m

Theorem 3.17.

$$(1) \text{REC} \cap H = \emptyset$$

$$(2) \text{RE} - (\text{REC} \cup H) \neq \emptyset$$

$$(1) \text{REC} \subsetneq \text{RE}$$

REC is closed under the equivalence relation \equiv_m .

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

Information

- 10月27日(金曜日)は中間テスト
 - 時間は11:00～12:30 (30分以上遅刻したら入室禁止)
 - 範囲は10月25日の授業分まで(テキスト3章まで)
 - テキスト、資料は持ち込み禁止
- Mid-term exam will be on October 27th, Fri.
 - Time: 11:00-12:30 (You cannot take it after 11:30)
 - About: up to Today (Chapter 3)
 - Texts and other materials are not allowed to bring

Memo:

Report (3); Today

Ans. for Report (2); Today's Office Hour