I238 計算の理論

上原 隆平

2018年I-1期（4-5月）

I238 Computation Theory

by

Prof. Ryuhei Uehara

Term I-1, April-May, 2018

計算(量)の理論

- ゴール1:
  - “計算可能な*関数/問題/言語/集合*”
    - 関数には2種類存在する;
      1. 計算不能(!)な関数
      2. 計算可能な関数
- ゴール2:
  - 「問題の困難さ」を示す方法を学ぶ
    - 計算可能な問題であっても、*手におえない* 場合がある！
      - 計算に必要な資源(時間・領域)が多すぎるとき

Computation Theory/ Computational Complexity

- Goal 1:
  - “*Computable Function/Problem/Language/Set*”
    - We have two functions;
      1. Functions that are not computable!
      2. Functions that are computable.
- Goal 2:
  - How can you show “*Difficulty of Problem*”
    - There are *intractable* problems even if they are computable!
      - because they require too many resources (time/space)!

4. 計算不能性と対角線論法

4. 計算不能な問題

以下の問題を解くチューリングマシンは存在しない：

停止性判定問題HALT (停止するかどうかを決定する問題)

入力：チューリングマシン*T*と

それへの入力*x*を符号化した文字列<*T*,*x*>

出力: *T* に入力 *x* を与えると，停止するか？

Yes: *T*(*x*) は(有限時間内に)停止する

No: 停止しない(無限ループ)

正確に言えば，停止性判定問題を解くチューリングマシン *U'* は存在しない．

…証明は「対角線論法」を用いて行う

4. Undecidability and Diagonalization

4. Undecidable problem

The following problem cannot be solved by any Turing machine:

The problem HALT (Problem of deciding halting)

input : a code <*T*,*x*> of Turing machine *T* and an input *x*

output: *T* will terminates for the input *x*?

Yes: if *T*(*x*) terminates

No: otherwise.

Precisely, we can show that there is no Turing machine *U'* that computes the halting problem.

…Proof is done by "diagonalization" essentially…

Yes/Noタイプのプログラム

5.2. 枚挙可能集合

帰納的でない集合を認識するプログラムは存在しない.
しかし弱い意味での“認識”を考えると話は別

プログラムAが集合*L*を半認識する

すべての $x \in \Sigma^*$で

$x \in L \quad \leftrightarrow \quad$ A(*x*)=accept

$x \notin L \quad \leftrightarrow \quad$ A(*x*)= $\perp$ (A(*x*)が停止しない）

集合*L*は半帰納的 ↔ 集合*L*を半認識するプログラムが存在

帰納的集合 $\subsetneq$ 半帰納的集合

i.e., 認識可能な集合 $\subsetneq$ 半認識可能な集合

5.2 Enumerable set

There is no program for recognizing a non-recursive set,
but we have a different story if we consider weak "recognition"

Program A ***semi-recognizes*** a set *L*

for every $x \in \Sigma^*$

$x \in L \quad \leftrightarrow \quad$ A(*x*)=accept

$x \notin L \quad \leftrightarrow \quad$ A(*x*)= $\perp$ (A(*x*) does not stop)

A set *L* is semi-recursive $\leftrightarrow$ semi-recognizing program of a set *L*

Recursive sets $\subsetneqq$ semi-recursive sets

i.e., recognizable sets $\subsetneqq$ semi-recognizable sets

定理5.3. 任意の無限集合Lに対し，次の2条件は同値.
(a) Lは半帰納的
(b) $L=RANGE(e)$となるような計算可能で1対1の関数 e が存在する.

定理5.3の証明は省略

⇕ cf

定理5.2. 集合Lを空でない任意の集合とする．このとき，次の2条件は同値.
(a) Lは半帰納的
(b) $L=RANGE(g)$となるような計算可能関数 g が存在する.

Theorem 5.3. For any infinite set L, the following two conditions are equivalent:
(a) L is semi-recursive.
(b) There is a computable one-to-one function e such that $L=RANGE(e)$.

Proof of Theorem 5.3 is omitted.

⇕ cf

Theorem 5.2 Let L be an arbitrary non-empty set. Then, the following two conditions are equivalent:
(a) L is semi-recursive.
(b) There is a computable function g such that $L=RANGE(g)$.

定理5.3

➔ 半帰納的集合 L には

$L=\{e(\varepsilon), e(0), e(1), e(00), e(01), e(10), e(11), e(000), \ldots\}$

となるような1対1の計算可能関数 e が存在する.

関数 e は L を枚挙 (enumerate)する

定義5.2. 集合 L は次のいずれかが成り立つとき，(帰納的に) 枚挙可能であるという(recursively enumerable).

(a) L は有限集合

(b) L を枚挙する関数で計算可能なものが存在.

注: 有限集合 L に対しては $L = RANGE(e)$となるような 1対1の全域関数 e などあり得ないので，例外的に扱っている.

定理5.4 すべての集合 L に対し，

L が半帰納的 $\Longleftrightarrow$ Lが枚挙可能

Theorem 5.3

➔ for a semi-recursive set *L* there exists a computable one-to-one function such that
$L=\{e(\varepsilon), e(0), e(1), e(00), e(01), e(10), e(11), e(000), \ldots\}$
We say the function *e* ***enumerates*** *L*.

Def. 5.2 A set *L* is **(recursively) enumerable** if
(a) *L* is a finite set, or
(b) there is a computable function that enumerates *L*.

Remark : Finite sets are exceptional, since for any finite set *L* there is no total on-to-one function e such that *L* = RANGE(*e*).

Theorem 5.4 For any set *L* we have
L is semi-recursive $\Longleftrightarrow$ *L* is enumerable

枚挙可能性と帰納性の比較

A: 帰納的集合

- ✓ Aの特徴述語 $R_A(x)$ が計算可能.
- ✓ $x \in \Sigma^*$に対し、$x \in A$かどうか判定可能
- ✓どんな入力 $x \in \Sigma^*$に対しても、
  いつも停止してYes/Noを答えてくれるプログラムが存在

B: 枚挙可能集合

- ✓ B を枚挙する関数が計算可能
- ✓すべてのBの要素を順番に出力するプログラムが存在

Comparison between enumerability and recursiveness

A: recursive set $\Longrightarrow$ the characteristic predicate $R_A(x)$ is computable

That is, for $x \in \Sigma^*$ it is computable whether $x \in A$

B: enumerable set $\Longrightarrow$ a function that enumerates *B* is computable

that is, we can enumerate all the elements of *B*

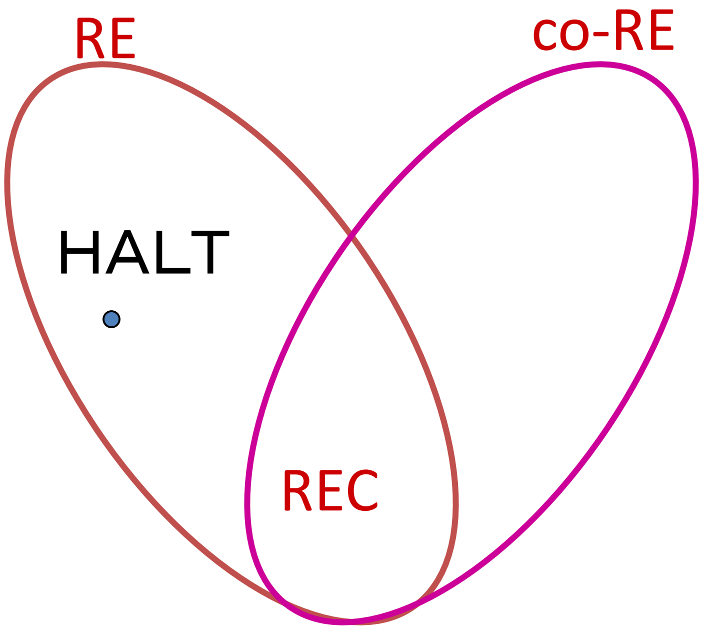
定理5.9. RE $\neq$ co-RE

証明:

RE=co-REと仮定すると，RE=RE ∩ co-RE

定理5.8より，REC＝REとなり，定理5.7に矛盾.

証明終

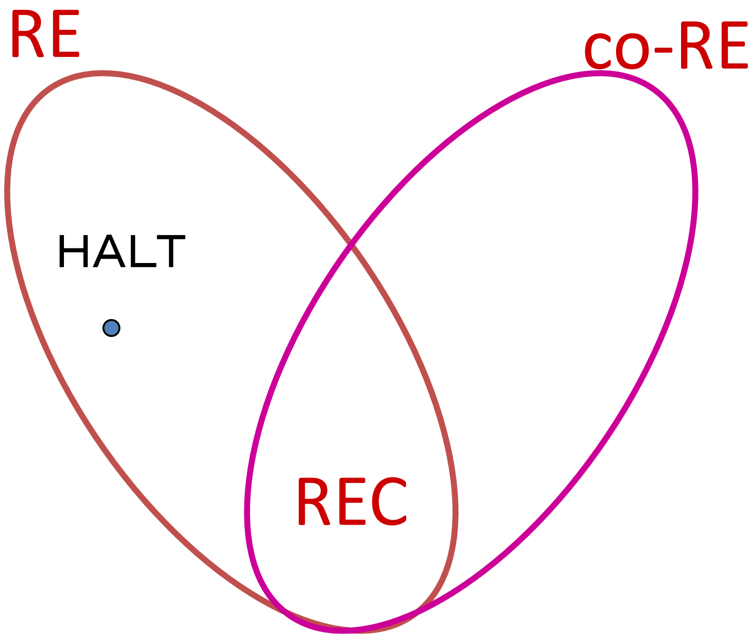


Theorem 5.9. RE $\neq$ co-RE

Proof :

If we assume RE=co-RE, we have RE=RE $\cap$ co-RE.
Hence, by Theorem 5.8 we have REC = RE, contradicts to Theorem 5.7.

Q.E.D.



短期の予定　Schedule for short term

5月7日：レポート1出題(締切：5月14日10:50)
(Report 1; deadline is 10:50 on May 14)

5月9日：午後，補講あり
(Supplemental lecture on Tutorial Hour)

5月14日：中間テスト (Mid-term Examination)

- 出題範囲(Scope)：今日の講義の範囲まで
- 持ち込んで良いもの(You can/should bring)
  - 筆記用具(Pen/Pencil)

1. スライドのコピー（書き込み可）＋手書きノート(Copies of slides (with your marks)+ hand-written notes)
2. ~~手書きノート(hand-written notes)~~
3. ~~A4用紙1枚（回収します）(A sheet of paper of A4 size, which will be submitted)~~

Information

- 5月14日（火曜日）は中間テスト
  - 10:50-12:30（30分以上遅刻したら入室禁止）
  - 範囲は5月9日の授業分まで
  - スライドのコピー（書き込み可）＋手書きノート
  - 場所はI3･4教室（隣は空席にすること）

- Mid-term exam will be on May 14th, Tue.
  - Time: 10:50-12:30 (You cannot take it after 30 minutes)
  - Scope: up to May 9th
  - Copies of slides (with your marks)+ hand-written notes
  - Place: I3･4 (by rule, your have to have no neighbor)

5.6. 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）

5.6. 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?

$\leftarrow$ 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5.3.

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 5.3.

$A,\ B$: 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5.7.

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. 5.7.

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)

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

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

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

自然数の偶奇が判定可能なので，h_2 は計算可能

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

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

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

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

Simpler reduction from EVEN to ODD

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

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5.10. $A \leq_m B$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 5.10. 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5.8.

IsProgram(a): a はプログラム A のコードか？
これは計算可能な関数である.

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

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

まとめると

関係	したがって，
$\overline{\mathrm{HALT}} \leq_m \mathrm{ZERO}$	$\mathrm{ZERO} \notin \mathrm{REC}$ （$\overline{\mathrm{HALT}} \notin \mathrm{REC}$より）
$\mathrm{ZERO} \leq_m \mathrm{TOTAL}$	$\mathrm{TOTAL} \notin \mathrm{REC}$ （$\mathrm{ZERO} \notin \mathrm{REC}$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. 5.8.

IsProgram(a): Is a the program code of A?
This problem is computable.

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

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

Summarizing,

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

定理5.11. $A \leqq_m B$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）$B \in$ RE → $A \in$ RE　（Bが枚挙可能 → Aも枚挙可能）
（2）$B \in$ co-RE → $A \in$ co-RE

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

例5.8, 定理5.11 → ZERO、TOTALは
REにもco-REにも属さない。

性質	理由
ZERO $\notin$ RE	$\overline{\mathrm{HALT}} \notin$ RE、$\overline{\mathrm{HALT}} \leqq_m$ ZERO
ZERO $\notin$ co-RE	HALT $\notin$ co-RE、HALT $\leqq_m$ ZERO
TOTAL $\notin$ RE	ZERO $\notin$ RE、ZERO $\leqq_m$ TOTAL
TOTAL $\notin$ co-RE	ZERO $\notin$ co-RE、ZERO $\leqq_m$ TOTAL

Theorem 5.11. 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) 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. 5.8, Theorem 5.11 → Neither ZERO or TOTAL belongs to RE or co-RE.

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

還元可能性　：　難しさを比較する手段

$A \leqq_m B$ → Aの認識問題をBの認識問題に変換できる。

↓

Aの難しさ ≦ 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5.12.

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

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

$\downarrow$

hardness of $A \leq$ hardness of B

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

Theorem 5.12. 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5.9.

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5.10.

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

$$\mathrm{EVEN} \equiv_m \mathrm{PRIME}$$

（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. 5.9.

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. 5.10.

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5.4.

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

(A より真に難しいものはREには存在しない)

（b）$A \in \mathrm{RE}$

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. 5.4.
A set A is called RE-complete (under $\leqq_m$) if the following conditions hold

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

(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5.13. 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。

→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、

$$x \in L \Longleftrightarrow \mathrm{Halt}(\langle \mathrm{L}, x \rangle) \Longleftrightarrow \langle \mathrm{L}, x \rangle \in \mathrm{HALT}$$

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

（証明終）

Theorem 5.13. 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 $x \in \Sigma^*$

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

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

Q.E.D.

定理5.14. 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5.11. 定理5.14 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5.13
$\overline{\text{HALT}}$	co-RE完全	HALTがRE-困難、$\overline{\text{HALT}} \in$ co-RE
ZERO	RE-困難、co-RE困難	HALT $\leq_m$ZERO、
$\overline{\text{TOTAL}}$	RE-困難、co-RE困難	ZERO $\leq_m$TOTAL

Theorem 5.14. 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.5.11. Using Theorem 5.14, we can show hardness of various sets.

Sets	hardness	reasons
HALT	RE-complete	Theorem 5.13
$\overline{\mathrm{HALT}}$	co-RE complete	HALT is RE-hard, $\overline{\mathrm{HALT}} \in$ co-RE
ZERO	RE-hard, co-RE hard	HALT $\leq_m$ ZERO、
$\overline{\mathrm{TOTAL}}$	RE-hard, co-RE hard	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 $\leqq_m$ のもとで

定理5.15.

(1) $\mathrm{REC} \cap H = \phi$

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

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

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 $\leqq_m$

Theorem 5.15.

(1) $\mathrm{REC} \cap \mathrm{H} = \phi$

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

(1) $\mathrm{REC} \subsetneqq \mathrm{RE}$

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

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