4.3. 階層定理

定理4.4: 任意の制限時間 t_1, t_2 に対し,

$$\forall c>0, \overset{\infty}{\forall} n\,[ct_1(n)^2 \le t_2(n)] \rightarrow \mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2).$$

制限時間 t が大きな関数であればあるほど, t 時間以内で計算可能な問題は増える.

4.3.1. IsProgram, evalの時間計算量

プログラムのコード化法を正確に定義することが重要.

（約束事） Σ^*型の変数名: $v1, v2, \ldots, pc$. v変数と呼ぶ.

Σ型の変数名: $u1, u2, \ldots$ u変数と呼ぶ.

入力変数としては常に $v1$ を用いる.

出力変数としても $v1$ を用いる.

4.3. Hierarchy Theorem

> **Theorem 4.4** For any time limits t_1 and t_2, we have
>
> $\forall c>0, \overset{\infty}{\forall} n\, [c t_1(n)^2 \le t_2(n)\,] \rightarrow \mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2).$

The larger the time limit t is, the more problems computable in time t there are.

4.3.1. Time complexities of IsProgram and eval

Important to define program encoding scheme precisely (convention)

variable name of type Σ^*: $v1, v2, \ldots, pc$. (called v variables)

variable name of type Σ: $u1, u2, \ldots$ (called u variables)

input variable is fixed to $v1$.

output variable is also fixed to $v1$.

標準形プログラムAのコード化

$\langle \overline{K}, \overline{L}, \overline{M}, \langle$文1のコード，文2のコード，...,文$k$のコード$\rangle\rangle$

K: case文の分岐の数

L, M: u変数，v変数の数.

文k: case文中のk番目の分岐に書かれている文.

各文kを5つ組$\langle op, a1, a2, a3, a4\rangle$でコード化.

（例）　　タイプ　　　　　　　　文の形

(3) ui:=head(vj); pc:=k;　➔ $\langle \overline{3}, \overline{i}, \overline{j}, \overline{k}, \overline{\varepsilon} \rangle$

(12) if vi=s then pc:=k1　➔ $\langle \overline{12}, \overline{i}, s, \overline{k}1, \overline{k}2\rangle$

else pc:=k2 end-if;

4.2.1.で定義した12個のタイプ

Encoding a program A in the standard form

<$\overline{K}$, $\overline{L}$, $\overline{M}$, <code of st. 1, code of st. 2, ..., code of st. *k*>>

K: number of statements in the case structure

L, M: number of *u*- and *v*-variables.

st.*k*: *k*-th statement in the case structure

Each st. is encoded by 5 tuple<*op*, *a*1, *a*2, *a*3, *a*4>.

(Example) type form of a statement

(3) u*i*:=head(v*j*); pc:=k; ➔ <$\overline{3}$, $\overline{i}$, $\overline{j}$, $\overline{k}$, ε >

(12) if v*i*=s then pc:=k1 ➔ <$\overline{12}$, $\overline{i}$, *s*, $\overline{k}$1, $\overline{k}$2>
else pc:=k2 end-if;

12 types defined in Section 4.2.1.

例4.10.

```
prog A(input v1: Σ*): Σ*;
var pc, v2: Σ*; u1: Σ;
begin
  pc:=1;
  while文
  halt(v1)
end.
```

while文の中身

```
while pc≠0 do
  case pc of
  1: u1:=head(v1); pc:=2;        タイプ3
  2: if u1=0 then pc:=3 else pc:=1 end-if    タイプ11
  3: v1:=0; pc:=0;      タイプ5
  end-case
end-while;
```

このプログラムのコードは

$$<\overline{3}, \overline{1}, \overline{2}, <<\overline{3}, \overline{1}, \overline{1}, \overline{2}, \varepsilon>, <\overline{11}, \overline{1}, \overline{0}, \overline{3}, \overline{1}>, <\overline{5}, \overline{1}, \overline{0}, \overline{0}, \varepsilon>>>$$

Ex.4.10.

```
prog A(input v1: Σ*): Σ*;
var pc, v2: Σ*; u1: Σ;
begin
  pc:=1;
  while Statement
  halt(v1)
end.
```

Content of the while statement

```
while pc≠0 do
  case pc of
  1: u1:=head(v1); pc:=2;        type 3
  2: if u1=0 then pc:=3 else pc:=1 end-if    type 11
  3: v1:=0; pc:=0;     type 5
  end-case
end-while;
```

This program is encoded as

$$<\overline{3}, \overline{1}, \overline{2}, <<\overline{3}, \overline{1}, \overline{1}, \overline{2}, \varepsilon>, <\overline{11}, \overline{1}, \overline{0}, \overline{3}, \overline{1}>, <\overline{5}, \overline{1}, \overline{0}, \overline{0}, \varepsilon>>>$$

補題4.5: 適当な定数c_{isp}, d_{isp}に対して次の時間で*IsProgram*を計算するプログラムIsProgramが構成できる.

$$\forall a \in \Sigma^* \; [\mathrm{time\_IsProgram}(a) \leq c_{\mathrm{isp}}|a| + d_{\mathrm{isp}}\,]$$

略証:

与えられたコード a に対してIsProgramが調べること.

(a) a が$<\overline{K},\overline{L},\overline{M},<\overline{s_1}, \overline{s_2}, \ldots , \overline{s_N}>>$という形で，しかも $K = N$ か？

(b) 各s_kは正しい文のコードになっているか？

(c) 使われている変数番号，case分岐番号が範囲内に収まっているか？

上記の仕事が入力の長さの線形時間でできることは明らか.

（詳細はテキストp.117 – 118を参照.）

Lemma 4.5: With some constants c_{isp} and d_{isp}, we can write a program IsProgram to compute the predicate *IsProgram* in time such that

$$\forall a \in \Sigma^* \ [\mathrm{time\_IsProgram}(a) \leq c_{\mathrm{isp}}|a| + d_{\mathrm{isp}} \ ]$$

Rough Sketch of the Proof :

Given a code a, we must check the followings in the program IsProgram:

(a) Is a of the form $<\overline{K},\overline{L},\overline{M},<\overline{s}_1, \overline{s}_2, \ldots , \overline{s}_N>>$ and $K = N$?

(b) Is each s_k is a code of a grammatically correct statement?

(c) Are all variable numbers and branching numbers in the case within the limit?

The above checks are obviously done in time linear in the length of an input text.

（see pages 117 – 118 in the textbook for detail）

補題4.6: 適当な定数c_{ev}, d_{ev}に対して次の時間で*eval* を計算するプログラムevalが構成できる.

$$\forall a,x \in \Sigma^* \ [\mathrm{time\_eval}(a, x) \le c_{ev}|a|\mathrm{time\_a}(x) \times \max\{|a|, |x|, \mathrm{time\_a}(x)\} + d_{ev}\ ]$$

（補注）

a が表すプログラムに x を入力したときの実行時間 time_A(x)は，ふつうは$|a|$および$|x|$より大きい.

このとき，$\max\{|a|, |x|, \mathrm{time\_a}(x)\} = \mathrm{time\_a}(x)$

よって，$\mathrm{time\_eval}(a, x) \leqq c_{ev}|a|\mathrm{time\_a}(x)^2$

a に対応するプログラムの実行時間 t に対して，そのシミュレーションには$|a|^2t^2$ぐらいの時間がかかる。

(あるいはそれくらい時間をかければ十分).

Lemma 4.6:
With some constants c_{ev} and d_{ev}, we can write a program eval to compute the predicate *eval* in time such that

$$\forall a,x \in \Sigma^* \ [\text{time\_eval}(a, x) \leq c_{ev}|a|\text{time\_a}(x) \times \max\{|a|, |x|, \text{time\_a}(x)\} + d_{isp}\ ]$$

（Remark）
The time time_a(x), which is the time for the program A with an input x, is generally larger than $|a|$ and $|x|$.
Then, we have $\max\{|a|, |x|, \text{time\_a}(x)\} = \text{time\_a}(x)$
Thus, $\text{time\_eval}(a, x) \leqq c_{ev}|a|\text{time\_a}(x)^2$

When the program corresponding to a runs in time t,
its simulation takes time about $|a|^2t^2$.

補題4.6の略称

$(\ \forall\ a,x \in \Sigma^*\ [\mathrm{time\_eval}(a, x) \leq c_{ev}|a|\mathrm{time\_a}(x) \times \max\{|a|, |x|, \mathrm{time\_a}(x)\} + d_{ev}\ ]\ )$

変数v1はAの入力＆出力

・evalの定義

$$\mathrm{eval}(a, x) = \begin{cases} f\_a(x) & \text{IsProgram}(a)\text{のとき,} \\ ? & \text{その他のとき.} \end{cases}$$

$\lfloor a \rfloor$ =Aは
Σ 変数u1,…,uL
Σ *変数v1,…,vM
K 個のcase分岐

・eval(a, x)の計算に必要なこと

(1)IsProgramのチェック

(2) $\lfloor a \rfloor$ の入力 x に対する実行をシミュレートし, $f_a(x)$を求める.

シミュレーションのために変数*u*, *v*を用意.

変数u: プログラムA中の変数u1, ... , uLの値の管理: <u1, ... , uL>

変数v: プログラムA中の変数v1, ... , vMの値の管理: <v1, ... , vM>

uの初期値 = <0, ... , 0>

vの初期値 = <x, ε, ... , ε >　　最初のxはv1=xの値.

シミュレーション終了時の出力変数v1の値はvの第一引数の値.

Rough proof of Lemma 4.6

$(\forall a,x \in \Sigma^* [\text{time\_eval}(a, x) \le c_{ev}|a|\text{time\_a}(x) \times \max\{|a|, |x|, \text{time\_a}(x)\} + d_{ev}])$

v1 is input and output of A

$\lfloor a \rfloor$ =A is
- u1,…,uL: Σ
- v1,…,vM: Σ*
- K case statements

▪ definition of eval

$$\text{eval}(a, x) = \begin{cases} f\_a(x) & \text{if IsProgram}(a), \\ ? & \text{otherwise.} \end{cases}$$

▪ What is required to compute eval(a, x)

(1)Check of IsProgram(a)

(2)Simulate the execution of $\lfloor a \rfloor$ to compute $f\_a(x)$.

Prepare variables u and v for the simulation

u: maintain variables u1, ... , uL in the program: <u1, ... , uL>

v: maintaining variables v1, ... , vM in the program: <v1, ... , vM>

initial value of u = <0, ... , 0>

initial value of v = <x, ε, ... , ε > x is initially set to v1=x.

The value of the output variable v1 is the first argument of v when the simulation is terminated.

シミュレーションプログラム

prog eval(input a, x);
% K, L, M, pcの値は1進表記の自然数
begin
  if ¬IsProgram(a) then halt(?);
  K:=get(a, $\overline{1}$);
    入力文字列からK,L,Mの値を取得. ssはcase文の集合
  L:=get(a, $\overline{2}$); M=get(a, $\overline{3}$); ss:=get(a, $\overline{4}$);
  u:=<0, ... , 0>; v:=<x, ε, ... , ε >;
    L個　　　　　　M-1個
  pc:=$\overline{1}$;
  while pc ≠ 0 do
    s:=get(ss, pc);   % case文の系列からpc番目の文を取得
    sに対応する文を実行(pcの更新も含む)
  end-while;
  halt(get(v, 1))
end.

変数v1の値を出力して終了

Simulation Program

prog eval(input a, x);
% K, L, M, pc are integers of unary representation
begin
if ¬IsProgram(a) then halt(?);
K:=get(a, $\overline{1}$); // get a values of K, L, and M from input string.
// ss is a set of statements in the case structure
L:=get(a, $\overline{2}$); M=get(a, $\overline{3}$); ss:=get(a, $\overline{4}$);
u:=$\langle 0, \ldots , 0 \rangle$; v:=$\langle x, \varepsilon, \ldots , \varepsilon \rangle$;
$\overline{}$ L0's M-1
pc:=$\overline{1}$;
while pc $\neq$ 0 do
s:=get(ss, pc); % get the pc-th statement from ss
execute the statement corresponding to s (and update pc)
end-while;
halt(get(v, 1))
end.

halt with the variable of v1.

プログラムevalの時間計算量…シミュレーションの時間計算量

l_a: 入力aの長さ
l_x: 入力 x の長さ
K, L, M: aが表すプログラムA中の
case文の数, u変数の数, v変数の数
l_{maxv}: シミュレーション中のevalの内部変数vの長さの最大値

項目	計算時間	理由
4つのget	$4(c_{\mathrm{get}} + l_{\mathrm{a}} + d_{\mathrm{get}})$	例4.3
IsProgram	$c_{\mathrm{isp}} l_{\mathrm{a}} + d_{\mathrm{isp}}$	補題4.5
u,vの初期化	$c_{\mathrm{cr}}(L+M) + 2d_{\mathrm{cr}} + c_{\mathrm{put}}(\lvert <, \ldots, > \rvert + l_{\mathrm{x}}) + d_{\mathrm{put}}$	
pcの初期化	1	
while文全体	$\mathrm{time\_A}(x)\{c_1(l_a + l_{\mathrm{maxv}}) + d_1\}$	
halt文のget	$c_{\mathrm{get}} l_{\mathrm{maxv}} + d_{\mathrm{get}}$	例4.3

Evaluation of time complexity of program eval (or the simulation)

l_a: length of input a
l_x: length of input x
K, L, M: the number of statements in the case structure of the program A, the number of u and v variables.
l_{maxv}: maximum length of v during the simulation of eval

Item	Computation time	reason
four get	$4(c_{\mathrm{get}} l_{\mathrm{a}} + d_{\mathrm{get}})$	Ex.4.3
IsProgram	$c_{\mathrm{isp}} l_{\mathrm{a}} + d_{\mathrm{isp}}$	Lemma 4.5
initialization of u,v	$c_{\mathrm{cr}}(L+M) + 2d_{\mathrm{cr}} + c_{\mathrm{put}}(\lvert<, \ldots ,>\rvert + l_{\mathrm{x}}) + d_{\mathrm{put}}$	
initialization of pc	1	
entire while loop	$\mathrm{time\_A}(x)\{c_1(l_a + l_{\mathrm{maxv}}) + d_1\}$	
get in the halt	$c_{\mathrm{get}} l_{\mathrm{maxv}} + d_{\mathrm{get}}$	Ex.4.3

プログラムevalの時間計算量…シミュレーションの時間計算量

l_a: 入力aの長さ
l_x: 入力 x の長さ
K, L, M: aが表すプログラムA中の
case文の数，u変数の数，v変数の数
l_{maxv}: シミュレーション中のevalの内部変数vの長さの最大値

項目	計算時間	理由
while文全体	$\mathrm{time\_A}(x)\{c_1(l_a + l_{\mathrm{maxv}})+d_1\}$	

⇒ whileループ1回分の計算： $c_1(l_a + l_{\mathrm{maxv}})+d_1$ 時間以内
ループを回る回数：$\mathrm{time\_A}(x)$
∴ while文全体では， $\mathrm{time\_A}(x)\{c_1(l_a + l_{\mathrm{maxv}})+d_1\}$

よって，全体では，ある定数c_2, d_2に対し
$\mathrm{time\_eval}(a, x) \leqq c_2(l_a + l_{\mathrm{maxv}})\ \mathrm{time\_A}(x) + d_2$

Evaluation of time complexity of program eval (or the simulation)

l_a: length of input a
l_x: length of input x
K, L, M: the number of statements in the case structure of the program A, the number of u and v variables.
l_{maxv}: maximum length of v during the simulation of eval

Item	Computation time	reason
entire while loop	time_A(x)$\{c_1(l_a + l_{\mathrm{maxv}})+d_1\}$	

computation of while loop: $c_1(l_a + l_{\mathrm{maxv}})+d_1$ time
\# of loop iteration: time_A(x)
→ in the whole of while loop: time_A(x)$\{c_1(l_a + l_{\mathrm{maxv}})+d_1\}$

Thus, as a whole, for some constants c_2 and d_2
time_eval(a, x) $\leqq c_2(l_a + l_{\mathrm{maxv}})$ time_A(x) $+d_2$

l_{maxv}の評価（内部変数vの長さの最大値）

t ステップ目でA中のv変数に保持している文字列の内，
最も長い長さを l_t とすると，

12個の基本命令には
[文字列＋文字列]はない

$$l_0 = l_x,$$

$$l_t \le \max\{|s|, l_{t\text{-}1}+1\} \le \max\{l_a, l_{t\text{-}1}+1\}$$

（ただし，s は a 中の最も長い文字列定数）

したがって，

$$l_{\max} = \max\{l_t : 0 \le t \le \mathrm{time\_A}(x)\}$$

とおくと，

$$l_{\max} \le \max\{l_a, l_x\} + \mathrm{time\_A}(x)$$

v の値は<v1, ... , vM>なので

$$l_{\mathrm{maxv}} \le M\, l_{\max} \le M(\max\{l_a, l_x\} + \mathrm{time\_A}(x))$$

$$\le l_a \;\; 2 \max\{l_a, l_x, \mathrm{time\_A}(x)\}$$

Evaluation of l_{maxv}（Maximum length of internal variable v）

Let l_t be the maximum length of string stored in the v variable in the program A at the t-th step, then we have

$l_0 = l_x$,

$l_t \le \max\{|s|, l_{t-1}+1\} \le \max\{l_a, l_{t-1}+1\}$

Twelve basic statements do not contain “string+string”

（where, s is the longest string constant in a）

Hence, if we define

$l_{\mathrm{max}} = \max\{l_t : 0 \le t \le \mathrm{time\_A}(x)\}$

then we have

$l_{\mathrm{max}} \le \max\{l_a, l_x\} + \mathrm{time\_A}(x)$

Since the value of v is <v1, ... , vM>, we have

$l_{\mathrm{maxv}} \le M\, l_{\mathrm{max}} \le M(\max\{l_a, l_x\} + \mathrm{time\_A}(x))$

$\le l_a\ 2 \max\{l_a, l_x, \mathrm{time\_A}(x)\}$

プログラムevalの時間計算量: ある定数 c_2, d_2 に対し

$$\text{time\_eval}(a, x) \leqq c_2(l_a + l_{\text{maxv}})\ \text{time\_A}(x) + d_2$$

l_{maxv}の評価（内部変数vの長さの最大値）:

$$l_{\text{maxv}} \leqq l_a\ 2\max\{l_a, l_x, \text{time\_A}(x)\}$$

$$\begin{aligned}
&\text{time_eval}(a, x)\\
&\quad \leqq c_2(l_a + 2\, l_a \max\{l_a, l_x, \text{time_A}(x)\}) \times \text{time_A}(x) + d_2\\
&\quad = c_2 l_a\, \text{time_A}(x) \times (1+2\max\{l_a, l_x, \text{time_A}(x)\}) + d_2\\
&\quad \leqq 3\, \mathrm{c}_2 l_a\, \text{time_A}(x) \times 2\max\{l_a, l_x, \text{time_A}(x)\} + d_2\\
&\quad = c_{\text{ev}}|a|\, \text{time_A}(x) \times 2\max\{|a|, |x|, \text{time_A}(x)\} + d_{\text{ev}}
\end{aligned}$$

証明終

Evaluation of time complexity of program eval:

$$\text{time\_eval}(a, x) \leqq c_2(l_a + l_{\text{maxv}})\ \text{time\_A}(x) + d_2$$

for some constants c_2 and d_2

Evaluation of l_{maxv}（Maximum length of internal variable v）

$$l_{\text{maxv}} \leqq l_a\ \ 2 \max\{l_a, l_x, \text{time\_A}(x)\}$$

$$\begin{aligned}
&\text{time_eval}(a, x) \\
&\leqq c_2(l_a\ + 2\ l_a \max\{l_a, l_x, \text{time_A}(x)\}) \times \text{time_A}(x) + d_2 \\
&= c_2 l_a\ \text{time_A}(x) \times (1+2\max\{l_a, l_x, \text{time_A}(x)\}) + d_2 \\
&\leqq 3\ \mathrm{c}_2 l_a\ \text{time_A}(x) \times 2\max\{l_a, l_x, \text{time_A}(x)\} + d_2 \\
&= c_{\text{ev}}|a|\ \text{time_A}(x) \times 2\max\{|a|, |x|, \text{time_A}(x)\} + d_{\text{ev}}
\end{aligned}$$

End of Proof

例4.11: 文“ui:=tail(vj); pc:=k;”をシミュレートするのに要する時間

行うこと	ステップ数の概算
s:=get(ss, pc);	$c_{get}\|ss\|+d_{get}=O(l_a)$
文のタイプの判定	O(1)
i:=get(s, 2)	$c_{get}\ \|s\|+d_{get}=O(l_a)$
j:=get(s, 3)	$c_{get}\ \|s\|+d_{get}=O(l_a)$
k:=get(s, 4)	$c_{get}\ \|s\|+d_{get}=O(l_a)$
tmp:=get(v, j)	$c_{get}\ l_v+d_{get}=O(l_v)$
tmp:=tail(tmp)	O(1)
put(u, i, tmp)	$c_{put}(l_a+\|\mathrm{tmp}\|)+d_{put}=O(l_a)$
pc:=k	O(1)

よって，time$\leq c(l_a+l_v)+d$,　c,dはl_a, l_vと独立

この c, d は文のタイプごとに違う

そこで，最大のものをc_1, d_1とすると，文のタイプに関係なく

　１ステップあたりの時間　$\leq c_1(l_a+l_v)+d_1$

$$\leq c_1(l_a+l_{maxv})+d_1$$

Ex.4.11 time to simulate the statement "ui:=tail(vj); pc:=k;"

what to do	approximate number of steps
s:=get(ss, pc);	$c_{\mathrm{get}}\|ss\|+d_{\mathrm{get}}=\mathrm{O}(l_a)$
type check of statement	O(1)
i:=get(s, 2)	$c_{\mathrm{get}}\ \|s\|+d_{\mathrm{get}}=\mathrm{O}(l_a)$
j:=get(s, 3)	$c_{\mathrm{get}}\ \|s\|+d_{\mathrm{get}}=\mathrm{O}(l_a)$
k:=get(s, 4)	$c_{\mathrm{get}}\ \|s\|+d_{\mathrm{get}}=\mathrm{O}(l_a)$
tmp:=get(v, j)	$c_{\mathrm{get}}\ l_v+d_{\mathrm{get}}=\mathrm{O}(l_v)$
tmp:=tail(tmp)	O(1)
put(u, i, tmp)	$c_{\mathrm{put}}(l_a+\|\mathrm{tmp}\|)+d_{\mathrm{put}}=\mathrm{O}(l_a)$
pc:=k	O(1)

Thus, time $\leq c(l_a + l_v) + d$, c,d are independent of l_a , l_v

These c and d are different for each type.

So, let the largest ones be c_1 , d_1 , then for any type we have

time per 1 step $\leq c_1\ (l_a + l_v) + d_1$

$\leq c_1\ (l_a + l_{\mathrm{maxv}}) + d_1$

今までの議論の応用として，次の関数の計算時間を評価

各$a, x \in \Sigma^*, t \in \mathrm{N}$に対して

$$eval\_in\_time(a, x, \overline{t}) = \begin{cases} f\_a(x), & IsProgram(a) \text{ かつ} \lfloor a \rfloor (x) \text{が} \\ & t \text{ 時間以内に停止するとき} \\ ? & \text{その他のとき} \end{cases}$$

補題4.7: 適当な定数 c_{evt}, d_{evt}に対して，次の計算量で *eval_in_time*を計算するプログラムeval-in-timeが構成できる．

$$\forall a, x \in \Sigma^*, t \in \mathrm{N}\ [\mathrm{time\_eval\text{-}in\text{-}time}(a, x, \overline{t}) \leq c_{evt}|a|^2 t \max\{|x|, t\} + d_{evt}]$$

（略証）

プログラム$\lfloor a \rfloor (x)$の計算時間が t を超えると強制終了するためのカウンタを用いてシミュレーションを実行．

それ以外はevalの評価と同様．

証明終

As an application of the discussions so far, evaluate the computation time of the following function:

for each $a, x \in \Sigma^*, t \in \mathrm{N}$

$$\text{eval\_in\_time}(a, x, \overline{t}) = \begin{cases} f\_a(x), & \text{IsProgram}(a) \text{ and } \lfloor a \rfloor(x) \\ & \quad \text{stops before time } t, \\ ? & \text{otherwise.} \end{cases}$$

Lemma 4.7: For some constants c_{evt} and d_{evt}, we can write a program eval-in-time to compute eval-in-time in time such that

$$\forall a, x \in \Sigma^*, t \in \mathrm{N}\ [\text{time\_eval-in-time}(a, x, \overline{t}) \le c_{\text{evt}} |a|^2 t \max\{|x|, t\} + d_{\text{evt}}]$$

(Proof)

Execute the simulation using a counter to force halting when computation time of a program $\lfloor a \rfloor(x)$ exceeds t.

The remaining analysis is just the same as in eval.

Q.E.D.

4.3.2. 階層定理の証明

定理4.4: 任意の制限時間 t_1, t_2 に対し,
$\forall c>0, \overset{\infty}{\forall} n\ [c t_1(n)^2 \leq t_2(n)] \rightarrow \mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2)$.

仮定より, 明らかに$t_1=O(t_2)$であるから, $\mathrm{TIME}(t_1) \subseteq \mathrm{TIME}(t_2)$.
よって, $\mathrm{TIME}(t_2) - \mathrm{TIME}(t_1) \neq \phi$であることを示せばよい.
すなわち, $O(t_2)$時間では認識できるが, $O(t_1)$時間では認識できない集合の存在を示せばよい.

プログラムA=$\lfloor a \rfloor$ に
$x=\langle a,w \rangle$を入力すると、
$|x| < \sqrt{t_2(|x|) / |a|}$
$t = \sqrt{t_2(|x|) / |a|}$ 以内にacceptしない

DIAG =
$\{\langle a,w \rangle$: 次の3条件を満たす.

(a) $IsProgram(a)$
(b) $l < t$
(c) eval-in-time$(a, \langle a,w \rangle, \overline{t}) \neq$ accept $\}$

ただし, $x=\langle a,w \rangle$, $l=|x|$, $t = [\sqrt{t_2(|x|) / |a|}]$
([ ]は切り捨て)

4.3.2. Proof of the Hierarchy Theorem

Theorem 4.4. For any time limits t_1 and t_2, we have
$$\forall c>0, \overset{\infty}{\forall} n\, [c t_1(n)^2 \le t_2(n)\,] \rightarrow \mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2).$$

By assumption we have $t_1=O(t_2)$ and thus $\mathrm{TIME}(t_1) \subseteq \mathrm{TIME}(t_2)$.
Thus, it suffices to show $\mathrm{TIME}(t_2) - \mathrm{TIME}(t_1) \neq \phi$.
That is, it suffices to show that there is a set which can be recognized in time $O(t_2)$ but not recognized in time $O(t_1)$.

DIAG = {$<a,w>$: the following three conditions are satisfied:

(a) *IsProgram*(a)

(b) $l < t$

(c) eval-in-time(a, $<a,w>$, $\overline{t}$) $\neq$ accept }

where, $x=<a,w>$, $l=|x|$, $t = [\sqrt{t_2(l)} / |a|]$ [ ] denotes round-off

If we input $x=<a,w>$ to a program a as an input, $|x| < \sqrt{t_2(|x|)} / |a|$ and it does not accept before time $t=\sqrt{t_2(|x|)} / |a|$.

補題4.8: $DIAG \notin \mathrm{TIME}(t_1)$

補題4.9: $DIAG \in \mathrm{TIME}(t_2)$

補題4.9の証明:

$x \in DIAG$ を調べるプログラムDIAGの計算時間.

$l = |x|,\ \ t = [\sqrt{t_2(l)}/|a|]$

(1) xが$<a,w>$の形をしているか？
(2) $IsProgram(a)$?

$\left.\right\}$ c_1, d_1を定数として $c_1 l + d_1$時間で判定可能

(3) $l < t$?

(4) $eval\text{-}in\text{-}time(a, <a,w>, \bar{t}) = eval\text{-}in\text{-}time(a, x, \bar{t}) \neq \mathrm{accept}$?

(3) $l < t$? $t = [\sqrt{t_2(l)}/|a|]$はO($t_2(l)$)時間で計算可能.

→演習問題4.12

(a) 2進表記の自然数nから1進表記$\bar{n}$への変換：O(n)時間
逆の変換 inv-transもO(n)でできる.

(b) n から$\sqrt{n}$ はO($|n|^2$)時間で計算可能→プログラムsqrt

(c) プログラムinv-trans, div, sqrt, transを用いて
$\sqrt{t_2(l)}/|a|$の1進表記を求める.

Lemma 4.8: $DIAG \notin \mathrm{TIME}(t_1)$

Lemma 4.9: $DIAG \in \mathrm{TIME}(t_2)$

Proof of Lemma 4.9:

computation time of a program DIAG to check $x \in DIAG$.

$l = |x|$, $t = [\sqrt{t_2(l)}/|a|]$

(1) Is x of the form $<a,w>$?

(2) $IsProgram(a)$?

can be checked in time $c_1 l + d_1$ where c_1, d_1 are constants

(3) $l < t$?

(4) eval-in-time$(a, <a,w>, \bar{t})$ = eval-in-time$(a, x, \bar{t}) \neq$ accept ?

(3) $l < t$? $t = [\sqrt{t_2(l)}/|a|]$ can be computed in time $O(t_2(l))$.

→Exercise 4.12

(a) Transformation from binary to unary representation.: $O(n)$ time

inverse transformation inv-trans is also done in time $O(n)$.

(b) $\sqrt{n}$ can be computed in time $O(|n|^2)$→program sqrt

(c) Using programs inv-trans, div, sqrt, and trans

we compute unary representation of $\sqrt{t_2(l)}/|a|$.

(4) eval-in-time(a,x,$\bar{t}$) ≠ accept?の判定

補題4.7より，$c_{\mathrm{evt}}|a|^2 t \max\{l,t\}+d_{\mathrm{evt}}$時間で判定可能.

$$c_{\mathrm{evt}}|a|^2\lceil\sqrt{t_2(l)}/|a|\rceil \max\{l, \lceil\sqrt{t_2(l)}/|a|\rceil\}+d_{\mathrm{evt}}$$
$$\le c_{\mathrm{evt}}|a| \max\{l\sqrt{t_2(l)}, t_2(l)/|a|\}+d_{\mathrm{evt}}$$
$$= c_{\mathrm{evt}} \max\{l|a|\sqrt{t_2(l)}, t_2(l)\}+d_{\mathrm{evt}} = \mathrm{O}(t_2(l)).$$

結局，(1)-(4)はO($t_2(l)$)時間で判定可能.

よって，$DIAG \in \mathrm{TIME}(t_2)$.

補題4.9の証明終

(4)Check of eval-in-time($a,x,\bar{t}$) $\neq$ accept?

From Lemma 4.7, it can be checke in time $c_{\mathrm{evt}}|a|^2 t \max\{l,t\}+d_{\mathrm{evt}}$.

$$c_{\mathrm{evt}}|a|^2[\sqrt{t_2(l)}/|a|] \max\{l, [\sqrt{t_2(l)}/|a|]\}+d_{\mathrm{evt}}$$
$$\leq c_{\mathrm{evt}}|a| \max\{l\sqrt{t_2(l)}, t_2(l)/|a|\}+d_{\mathrm{evt}}$$
$$= c_{\mathrm{evt}} \max\{l|a|\sqrt{t_2(l)}, t_2(l)\}+d_{\mathrm{evt}} = \mathrm{O}(t_2(l)).$$

(1)-(4) can be checked in time $\mathrm{O}(t_2(l))$.

Thus, we have $DIAG \in \mathrm{TIME}(t_2)$.

End of proof of Lemma 4.9

補題4.8 ($DIAG \notin$ TIME(t_1))の証明:

$DIAG \in$ TIME(t_1)として矛盾を導く.

・$DIAG$ をO(t_1)時間で認識するプログラムをA_0, コードを a_0とする.

・time_ $A_0(l) \leqq c\, t_1(l) + d$ を満たす定数c, d が存在する.

$c_0=c+d$とおく. ただし, $c_0>1$.

→ time_ $A_0(l) \leqq c_0\, t_1(l)$ .... (1) (if $1 \leqq t_1(l)$)

c_0は定数

定理の仮定: $\forall c>0, \exists n\ [c t_1(n)^2 \leqq t_2(n)]$ より、

n を十分大きく取ると,

$$(|a_0|^2 c_0^2)\, t_1(n)^2 \leqq t_2(n) \rightarrow c_0\, t_1(n) \leqq \sqrt{t_2(n)}\ /|a_0|$$

また, 自然な制限時間の条件より, $n \leq t_1(n)$

そこで, 十分長い文字列 w_0 を考えると (ただし, $l_0=|<a_0,w_0>|$)

$$c_0\, t_1(l_0) \leq [\sqrt{t_2(l_0)}\ /|a_0|\ ]$$

$$t_1(l_0) \leq [\sqrt{t_2(l_0)}\ /|a_0|\ ] / c_0 \leq [\sqrt{t_2(l_0)}\ /|a_0|\ ]$$

よって,

$$l_0 \leq [\sqrt{t_2(l_0)}\ /|a_0|\ ] \text{ ....(2)}$$

Proof of Lemma 4.8 ($DIAG \notin TIME(t_1)$):

To derive a contradiction we assume $DIAG \in \mathrm{TIME}(t_1)$.

- Let A_0 be a program to recognize $DIAG$ in time $O(t_1)$, and let its code be a_0.
- $\mathrm{time\_}A_0(l) \leq c\, t_1(l) + d$ for some constants c and d.
  we set $c_0=c+d$, where $c_0>1$.
  $\rightarrow \mathrm{time\_}A_0(l) \leq c_0\, t_1(l)$ .... (1) (if $1 \leq t_1(l)$)

c_0 ;constant

in the theorem: $\forall c>0, \overset{\infty}{\forall} n\ [c t_1(n)^2 \leq t_2(n)]$

when n is sufficiently large,

$$(|a_0|^2 c_0^2)\, t_1(n)^2 \leq t_2(n) \rightarrow c_0\, t_1(n) \leq \sqrt{t_2(n)}\, /|a_0|$$

Also, by the condition of natural time limit, $n \leq t_1(n)$

So, considering sufficiently long string w_0 (where, $l_0=|\langle a_0,w_0\rangle|$)

$$c_0\, t_1(l_0) \leq [\sqrt{t_2(l_0)}\, /|a_0|\,]$$

$$t_1(l_0) \leq [\sqrt{t_2(l_0)}\, /|a_0|\,] / c_0 \leq [\sqrt{t_2(l_0)}\, /|a_0|\,]$$

Thus,

$$l_0 \leq [\sqrt{t_2(l_0)}\, /|a_0|\,] \text{ ....(2)}$$

$(1)\ \mathrm{time\_A}_0(l) \le c_0 t_1(l)$ ——— 一般の l について

$(2)\ l_0 \le [\sqrt{t_2(l_0)} / |a_0|]$ ——— 十分長い $l_0=<a_0,w_0>$

$$<a_0,w> \in DIAG \leftrightarrow l \leqq t \wedge \text{eval-in-time}(a_0,<a_0,w>,\bar{t}) \neq \text{accept}$$
$$(w\in \Sigma^*,\ l=|<a_0,w>|,\ t=[\sqrt{t_2(l)}/|a_0|\ ])$$

w_0に対しては第1の条件 $l_0 \le t_0=[\sqrt{t_2(l_0)}/|a_0|\ ]$が満たされているので，

$<a_0,w_0>\in DIAG \leftrightarrow \text{eval-in-time}(a_0,<a_0,w_0>,\bar{t}_0) \neq \text{accept}$ .... (3)

(1), (2)より，

$\mathrm{time\_A}_0(<a_0,w_0>) \le c_0 t_1(l_0) \le t_0 = [\sqrt{t_2(l_0)}/|a_0|\ ]$

すなわち，　$\mathrm{time\_A}_0(<a_0,w_0>) \le t_0$

これは$<a_0,w_0>$をプログラムA_0に入力したとき，

計算は必ず t_0時間以内に終ることを意味しているから，

eval-in-timeでの制限時間 t_0 は本質的ではない.

つまり，eval-in-time$(a_0, <a_0,w_0>, t_0)$ = eval$(a_0, <a_0,w_0>)$

For general l

(1) $\text{time\_A}_0(l) \le c_0 t_1(l)$

For sufficiently large $l_0 = \langle a_0, w_0 \rangle$

(2) $l_0 \le [\sqrt{t_2(l_0)} / |a_0|]$

$\langle a_0,w\rangle \in DIAG \leftrightarrow l \le t \wedge$ eval-in-time$(a_0, \langle a_0,w\rangle, \bar{t}) \ne$ accept

$(w \in \Sigma^*,\ l = |\langle a_0,w\rangle|,\ t = [\sqrt{t_2(l)} / |a_0|\,])$

Since the condition $l_0 \le t_0 = [\sqrt{t_2(l_0)} / |a_0|\,]$ is satisfied for w_0,

$\langle a_0,w_0\rangle \in DIAG \leftrightarrow$ eval-in-time$(a_0, \langle a_0,w_0\rangle, t_0) \ne$ accept .... (3)

From (1), (2),

$\text{time\_A}_0(\langle a_0,w_0\rangle) \le c_0 t_1(l_0) \le t_0 = [\sqrt{t_2(l_0)} / |a_0|\,]$

Therefore, $\text{time\_A}_0(\langle a_0,w_0\rangle) \le t_0$

This implies that program A_0 terminates within time t_0 for the input $\langle a_0,w_0\rangle$, and so

the time limit t_0 in eval-in-time is not essential.

That is, eval-in-time$(a_0, \langle a_0,w_0\rangle, \bar{t_0})$ = eval$(a_0, \langle a_0,w_0\rangle)$

$<a_0,w_0> \in DIAG \leftrightarrow$ eval-in-time$(a_0,<a_0,w_0>, t_0) \neq$ accept .... (3)

に

eval-in-time$(a_0, <a_0,w_0>, t_0) =$ eval$(a_0, <a_0,w_0>)$

をあてはめると、

$$
\begin{aligned}
<a_0,w_0> \in DIAG &\leftrightarrow \mathrm{eval}(a_0, <a_0,w_0>) \neq \mathrm{accept} \\
&\leftrightarrow \mathrm{A}_0\text{が}<a_0,w_0>\text{をaccept}\text{しない}.
\end{aligned}
$$

これは,「 A_0が$DIAG$を認識する」という仮定に矛盾.

（証明終）

We apply

$$\text{eval-in-time}(a_0, <a_0,w_0>, t_0) = \text{eval}(a_0, <a_0,w_0>)$$

to

$$<a_0,w_0> \in DIAG \leftrightarrow \text{eval-in-time}(a_0,<a_0,w_0>, \bar{t}_0) \neq \text{accept} \text{ .... (3),}$$

we have

$$<a_0,w_0> \in DIAG \leftrightarrow \text{eval}(a_0, <a_0,w_0>) \neq \text{accept}$$
$$\leftrightarrow A_0 \text{ does not accept } <a_0,w_0>.$$

This contradicts to the assumption that A_0 recognizes *DIAG*.

（Q.E.D.）

対角線論法に基づく解釈

F_1 = O(t_1)の時間計算量をもつ認識プログラムすべての集合
= {A_1, A_2, ...}
それらのプログラムのコードを a_1, a_2, ... とする.
各a_iごとに適当な定数c_iを考えると,
$\text{time\_A}_i(l) \leq c_i t_1(l)$
が成立. さらに, 各a_i, c_i に対し, 十分長いw_iを取ると,
$c_i t_1(l_i) \leq [\sqrt{t_2(l_i)} / |a_i|\,],\quad l_i \leq [\sqrt{t_2(l_i)} / |a_i|\,]$
とできる. 各プログラムの入力w_iに対する出力の表を作ると,

	w_1	w_2	w_3	....	w_k
A_1	A	R	A	....	A
A_2	R	R	R	....	A
A_3	A	A	A	....	R
....	....	....	....	....	....
A_k	R	R	A	....	A

$A_i(w_i)$の値

対角線で
RとAが
逆

w_1	w_2	w_3	....	w_k
R				
	A			
		R		
			....	
				R

$w_i \in DIAG$?の答

*DIAG*を認識するプログラムはこの表に登場できない...矛盾

Interpretation based on Diagonalization

F_1 = a set of all recognizing programs of time complexity $O(t_1)$
= $\{A_1, A_2, \ldots\}$
Let their program codes be $a_1, a_2, \ldots$.

Considering an appropriate constant c_i for each a_i, we have
$\text{time\_A}_i(l) \leq c_i t_1(l)$

Moreover, we can take sufficiently long w_i for each a_i and c_i s.t.
$c_i t_1(l_i) \leq [\sqrt{t_2(l_i)} / |a_i|\,]$, $l_i \leq [\sqrt{t_2(l_i)} / |a_i|\,]$

Putting the outputs w_i for input in the table:

	w_1	w_2	w_3	....	w_k
A_1	A	R	A	....	A
A_2	R	R	R	....	A
A_3	A	A	A	....	R
......	......	......	......	......	......
A_k	R	R	A	....	A

values of $A_i(w_i)$

compare diagonals

	w_1	w_2	w_3	....	w_k
	R				
		A			
			R		
				....	
					R

answer to $w_i \in DIAG$?

This table can't include a program recognizing *DIAG*...contradiction.

例4.12: $\mathrm{TIME}(n^2) \subsetneq \mathrm{TIME}(n^5)$

$\Leftarrow\ \forall c>0, \forall n \geq c\ [\ c(n^2)^2 \leq n^5\ ]$

要するに,

$$\liminf_{n\to\infty} \frac{t_2(n)}{t_1(n)^2} = \infty$$

となれば，階層定理より$\mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2)$

Ex.4.12: $\mathrm{TIME}(n^2) \subsetneq \mathrm{TIME}(n^5)$

$\Leftarrow\ \forall c>0, \forall n \geq c\ [\ c(n^2)^2 \leq n^5\ ]$

If we have

$$\liminf_{n \to \infty} \frac{t_2(n)}{t_1(n)^2} = \infty$$

then the hierarchy theorem tells us that $\mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2)$