4.3. 階層定理

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

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.

標準形プログラムAのコード化

$<\bar{K}, \bar{L}, \bar{M}$, <文1のコード, 文2のコード, ...,文$k$のコード>>
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$<op, a1, a2, a3, a4>$でコード化.
（例）　タイプ　文の形
(3) ui:=head(vj); pc:=k;　➔ $<\bar{3}, \bar{i}, \bar{j}, \bar{k}, \bar{\varepsilon}>$
(12) if vi=s then pc:=k1　➔ $<\bar{12}, \bar{i}, s, \bar{k}1, \bar{k}2>$
else pc:=k2 end-if;

4.2.1.で定義した12個のタイプ

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

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.
$\forall a \in \Sigma^*$ [time_IsProgram(a) $\le c_{isp}|a| + d_{isp}$]

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$<\bar{K},\bar{L},\bar{M},<\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_N>>$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.）

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.
$\forall a,x \in\Sigma^*$ [time_eval(a, x)$\le c_{ev}|a|$time_a(x)
× max{$|a|$, $|x|$, 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{$|a|$, $|x|$, time_a(x)} = time_a(x)
よって, time_eval(a, x) ≦ $c_{ev}|a|$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).

補題4.6の略称
($\forall a,x \in \Sigma^*$ [time_eval(a, x)$\le c_{ev}|a|$time_a(x)
× max{$|a|$, $|x|$, time_a(x)}+ d_{ev}])

変数v1はAの
入力＆出力

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

・evalの定義
eval(a, x)= { f_a(x)　IsProgram(a)のとき, ?　その他のとき. }

・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 = $<x, \varepsilon, \ldots, \varepsilon>$　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.

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, 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, 2); M=get(a, 3); ss:=get(a, 4);
  u:=<0, ... , 0>; v:=<x, ε, ... , ε >;
          L個              M-1個
  pc:=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))     変数v1の値を出力して終了
end.
```


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	$4(c_{\mathrm{get}} + l_a + d_{\mathrm{get}})$	例4.3
IsProgram	$c_{\mathrm{isp}} l_a + d_{\mathrm{isp}}$	補題4.5
u,vの初期化	$c_{\mathrm{cr}}(L+M) + 2d_{\mathrm{cr}} + c_{\mathrm{put}}(\lvert<, \dots ,>\rvert + l_x) + d_{\mathrm{put}}$	
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

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

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

$l_0 = l_x$,
$l_t \le \max\{|s|, l_{t-1}+1\} \le \max\{l_a, l_{t-1}+1\}$
（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
$l_{\mathrm{maxv}} \le M\, l_{\max} \le M(\max\{l_a, l_x\} + \mathrm{time\_A}(x))$
$\le l_a\ 2\max\{l_a, l_x, \mathrm{time\_A}(x)\}$

プログラムevalの時間計算量: ある定数 c_2, d_2 に対し

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

l_{maxv}の評価（内部変数vの長さの最大値）:

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

$\mathrm{time\_eval}(a, x)$
$\leqq c_2(l_a + 2\, l_a \max\{l_a, l_x, \mathrm{time\_A}(x)\}) \times \mathrm{time\_A}(x) + d_2$
$= c_2 l_a\ \mathrm{time\_A}(x) \times (1+2\max\{l_a, l_x, \mathrm{time\_A}(x)\}) + d_2$
$\leqq 3\, c_2 l_a\ \mathrm{time\_A}(x) \times 2\max\{l_a, l_x, \mathrm{time\_A}(x)\} + d_2$
$= c_{\mathrm{ev}}|a|\ \mathrm{time\_A}(x) \times 2\max\{|a|, |x|, \mathrm{time\_A}(x)\} + d_{\mathrm{ev}}$

証明終

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);	$c_{\mathrm{get}}\lvert ss\rvert + d_{\mathrm{get}} = O(l_a)$
文のタイプの判定	$O(1)$
i:=get(s, 2)	$c_{\mathrm{get}}\lvert s\rvert + d_{\mathrm{get}} = O(l_a)$
j:=get(s, 3)	$c_{\mathrm{get}}\lvert s\rvert + d_{\mathrm{get}} = O(l_a)$
k:=get(s, 4)	$c_{\mathrm{get}}\lvert s\rvert + d_{\mathrm{get}} = O(l_a)$
tmp:=get(v, j)	$c_{\mathrm{get}} l_v + d_{\mathrm{get}} = O(l_v)$
tmp:=tail(tmp)	$O(1)$
put(u, i, tmp)	$c_{\mathrm{put}}(l_a + \lvert\mathrm{tmp}\rvert) + d_{\mathrm{put}} = O(l_a)$
pc:=k	$O(1)$

よって，time$\le c(l_a + l_v) + d$,　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 $\le c_1(l_a + l_v) + d_1$
$\le c_1(l_a + l_{\mathrm{maxv}}) + d_1$

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$a, x \in \Sigma^*, t \in$ N に対して

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

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.

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

（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

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

仮定より，明らかにt_1=O(t_2)であるから， TIME(t_1) ⊆ TIME(t_2).
よって，TIME(t_2) – TIME(t_1) ≠ ϕ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.

DIAG =
{<*a*,*w*>: 次の3条件を満たす.
(a) *IsProgram*(*a*)
(b) $l < t$
(c) eval-in-time(a, <a,w>, $\overline{t}$) ≠ accept }
ただし，x=<a,w>, l=|x|, $t = [\sqrt{t_2(|x|)} / |a|]$
([ ]は切り捨て)

プログラムA=$\lfloor a \rfloor$ に
x=<a,w>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

補題4.8: *DIAG* ∉ TIME(t_1)

補題4.9: *DIAG* ∈ TIME(t_2)

補題4.9の証明:

x∈*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)?
(3) $l < t$?
(4) *eval-in-time*(a, <a,w>, $\overline{t}$) = *eval-in-time*(a, x, $\overline{t}$) ≠ accept ?

c_1,d_1を定数として $c_1 l+d_1$時間で判定可能

(3) $l < t$? $t = [\sqrt{t_2(l)} / |a|]$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$\overline{n}$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.

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

補題4.7より，$c_{evt}|a|^2 t \max\{l, t\} + d_{evt}$時間で判定可能.
$c_{evt}|a|^2[\sqrt{t_2(l)} / |a|] \max\{l, [\sqrt{t_2(l)} / |a|]\} + d_{evt}$
$\le c_{evt}|a| \max\{l\sqrt{t_2(l)}, t_2(l) / |a|\} + d_{evt}$
$= c_{evt} \max\{l|a|\sqrt{t_2(l)}, t_2(l)\} + d_{evt}$ = O($t_2(l)$).

結局，(1)-(4)はO($t_2(l)$)時間で判定可能.
よって，*DIAG* ∈ TIME(t_2).

補題4.9の証明終

補題4.8 (*DIAG* ∉ TIME(t_1))の証明:

DIAG ∈ 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 ≦ $t_1(l)$)

c_0は定数

定理の仮定：∀c>0, $\overset{\infty}{\forall} n\ [ct_1(n)^2 \leqq t_2(n)]$ 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， $n \le t_1(n)$
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>|)
$c_0\, t_1(l_0) \le [\sqrt{t_2(l_0)} / |a_0|]$
$t_1(l_0) \le [\sqrt{t_2(l_0)} / |a_0|] / c_0 \le [\sqrt{t_2(l_0)} / |a_0|]$
よって，
$l_0 \le [\sqrt{t_2(l_0)} / |a_0|]$ ....(2)

(1) 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> ∈ *DIAG* ↔ $l \leqq t$ ∧ eval-in-time(a_0,<a_0,w>,$\overline{t}$) ≠ 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>∈ *DIAG* ↔ eval-in-time(a_0,<a_0,w_0>,$\overline{t_0}$) ≠ accept .... (3)
(1), (2)より,
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， 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>)

$\langle a_0, w_0\rangle \in \mathit{DIAG} \leftrightarrow$ eval-in-time$(a_0, \langle a_0, w_0\rangle, \overline{t_0}) \neq$ accept .... (3)
に
eval-in-time$(a_0, \langle a_0, w_0\rangle, t_0) =$ eval$(a_0, \langle a_0, w_0\rangle)$
をあてはめると、

$\langle a_0, w_0\rangle \in \mathit{DIAG} \leftrightarrow$ eval$(a_0, \langle a_0, w_0\rangle) \neq$ accept
$\leftrightarrow$ A_0が$\langle a_0, w_0\rangle$をacceptしない.

これは,「A_0がDIAGを認識する」という仮定に矛盾.
（証明終）

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
$= \{\mathrm{A}_1, \mathrm{A}_2, \ldots\}$
それらのプログラムのコードを $a_1, a_2, \ldots$ とする.
各a_iごとに適当な定数c_iを考えると,
time_$\mathrm{A}_i(l) \leq c_i t_1(l)$
が成立. さらに, 各a_i, c_i に対し, 十分長いw_iを取ると,
$c_i t_1(l_i) \leq [\sqrt{t_2(l_i)} / |a_i|\,]$, $l_i \leq [\sqrt{t_2(l_i)} / |a_i|\,]$
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

$\mathrm{A}_i(w_i)$の値

対角線でRとAが逆

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

$w_i \in \mathit{DIAG}$?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

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$\mathrm{TIME}(t_1) \subsetneq \mathrm{TIME}(t_2)$