数理解析研究所 院試過去問解答
専門科目 (理論計算機)

nabla *

2024 年 11 月 22 日

目 次

*Twitter:@nabla_delta

はじめに

数理研の院試問題の解答です．一部の問題には図がありましたが，入れるのがめんどくさいので省略してあります．解答が正しいという保証はありません．また，一部の解答は math.stackexchange.com で見つけたものを参考にしています．別解がある（かもしれない）場合でも解答は一つだけしか書いてありませんし，ここの解答より簡単な解答もあるかもしれません．この文書を使用して何らかの不利益が発生しても，私は責任を負いません．転載は禁止です．

平成 25 年度 (2012 年 8 月実施)

問 10

J を有限個の仕事からなる集合とする．各仕事 $j \in J$ の開始日 s_j, 終了日 f_j が定められている．また，各仕事 $j \in J$ には報酬 w_j が支払われる．ただし，s_j, f_j, w_j は非負整数とする．同じ日には 1 つの仕事しか実行できないとする．このとき，受け取る報酬の合計が最大となるように，実行する仕事を定める効率的なアルゴリズムを記述し，その正当性と計算量を示せ．

解答. $|J| = n$ とおく．ソートして $f_1 \leq f_2 \leq \cdots \leq f_n$ としておく．$p(j) = \max\{i\,;\, f_i \leq s_j\}$ と定める．ただし $f_i \leq s_j$ となる i が無い時は $p(j) = 0$ とする．仕事 $\{1, 2, \ldots, j\}$ から実行する仕事を選ぶ時の報酬の合計の最大値を $v(j)$, その時実行する仕事の集合を $\mathcal{O}(j)$ とおく．この時

$$v(j) = \begin{cases} \max\{v(j-1), v(p(j)) + w_j\} & (j > 0) \\ 0 & (j = 0) \end{cases}$$

となることを帰納法で示す．$j = 0$ の時は自明．$j - 1$ の時正しいとする．$j \notin \mathcal{O}(j)$ なら $\{1, \ldots, j-1\}$ から仕事を選べば良いから $v(j) = v(j-1)$. $j \in \mathcal{O}(j)$ なら，日 s_j から日 f_j までは仕事 $1, \ldots, j-1$ は実行できない．今 f_i たちは昇順にソートされているから，仕事 $1, \ldots, j-1$ のうち実行出来るのは $f_i \leq s_j$ を満たす i, すなわち $1, \ldots, p(j)$ のみ．逆に $1, \ldots, p(j)$ からは，制約を満たす限り任意に選べるから，$v(p(j))$ を達成するように選ぶのが最善．よって $v(j) = v(p(j)) + w_j$. 従って j の時も正しい．これで示せた．この証明から $j \in \mathcal{O}(j)$ と $v(p(j)) + w_j \geq v(j-1)$ は同値だから，

$$\mathcal{O}(j) = \begin{cases} \mathcal{O}(p(j)) \cup \{j\} & (v(p(j)) + w_j \geq v(j-1)) \\ \mathcal{O}(j-1) & (v(p(j)) + w_j < v(j-1)) \end{cases}$$

となる．（ただし $\mathcal{O}(0) = \emptyset$ とする．）従って $S = \mathcal{O}(n)$ を求めるアルゴリズムは以下のようになる．

```
1:  n = |J|
2:  SORT(J)
3:  for j = 1, ..., n
4:      p(j) = BINARYSEARCH(j)
5:  v(0) = 0
6:  for j = 1, ..., n
7:      v(j) = max{v(p(j)) + w_j, v(j − 1)}
8:  S = FIND(n)
```

```
9:  function FIND(j)
10:     if j = 0 then
11:         return
12:     else
13:         if v(p(j)) + w_j ≥ v(j − 1) then
14:             output {j}∪ FIND(p(j))
15:         else
16:             output FIND(j − 1)
17:         return
```

ただし，$\textsc{Sort}(J)$ は J を f_j について昇順にソートする関数，$\textsc{BinarySearch}(j)$ は $f_i \leq s_j$ となる i を二部探索で求める関数であり，output は出力を表す．この時計算量は，2 行目で $O(n \log n)$. 4 行目で $O(\log n)$ だから 3, 4 行目の for ループで $O(n \log n)$. 6, 7 行目で $O(n)$. $\textsc{Find}(j)$ で呼び出されるのは $\textsc{Find}(p(j))$ か $\textsc{Find}(j-1)$ のどちらか一つであることと $j > p(j)$ より，$\textsc{Find}(n)$ が自分自身を呼び出すのは高々 n 回．よって 8 行目の計算量は $O(n)$. 以上から全体の計算量は $O(n \log n)$. □

問 11

型のないラムダ計算のラムダ項全体の集合を Λ とする．ただし，α 同値なラムダ項は同一視するものとする．写像 $[\![-]\!] : \Lambda \to \Lambda$ を

$$\begin{array}{lcl} [\![x]\!] & = & \lambda k.kx \\ [\![\lambda x.M]\!] & = & \lambda k.k(\lambda x.[\![M]\!]) \\ [\![MN]\!] & = & \lambda k.[\![M]\!](\lambda m.[\![N]\!](\lambda n.mnk)) \end{array}$$

により定める．以下では，β 簡約関係の反射的推移的閉包を $\to_\beta^*$ で表す．このとき，次の (i), (ii) に解答せよ．

(i) N が変数か関数抽象（ラムダ抽象）であるとき，$[\![(\lambda x.M)N]\!] \to_\beta^* [\![M[x := N]]\!]$ が成り立つことを証明せよ．

(ii) $[\![(\lambda x.M)N]\!] \to_\beta^* [\![M[x := N]]\!]$ が成り立たない M と N の例を挙げよ．

解答． $\to_\beta^*$ を $\to$ と略記する．

(i) $M \in \Lambda$ の自由変数のなす集合を $\mathrm{FV}(M)$ とおく．定義から

$$\mathrm{FV}([\![x]\!]) = \{x\}, \quad \mathrm{FV}([\![\lambda x.M]\!]) = \mathrm{FV}([\![M]\!]) \setminus \{x\}, \quad \mathrm{FV}([\![MN]\!]) = \mathrm{FV}([\![M]\!]) \cup \mathrm{FV}([\![N]\!])$$

だから $\mathrm{FV}([\![M]\!]) = \mathrm{FV}(M)$ である．従って

$$\begin{aligned} [\![(\lambda x.M)N]\!] &= \lambda k.[\![\lambda x.M]\!](\lambda m.[\![N]\!](\lambda n.mnk)) = \lambda k.(\lambda k_1.k_1(\lambda x.[\![M]\!]))(\lambda m.[\![N]\!](\lambda n.mnk)) \\ &\to \lambda k.(\lambda m.[\![N]\!](\lambda n.mnk))(\lambda x.[\![M]\!]) \to \lambda k.[\![N]\!](\lambda n.(\lambda x.[\![M]\!])nk) \quad \cdots (*) \end{aligned}$$

- $N = y$ の時，

$$(*) = \lambda k.(\lambda k_1.k_1 y)(\lambda n.(\lambda x.[\![M]\!])nk) \to \lambda k.(\lambda n.(\lambda x.[\![M]\!])nk)y \to \lambda k.(\lambda x.[\![M]\!])yk$$

ここで $[\![-]\!]$ の定義から $\lambda k.[\![M]\!]k \to [\![M]\!]$ である．従って $(\lambda x.[\![M]\!])y \to [\![M[x := y]]\!]$ を示せば良い．$x \notin \mathrm{FV}(M)$ の時は明らかなので $x \in \mathrm{FV}(M)$ とする．M の構造に関する帰納法で示す．$M = x$ の時は

$$(\lambda x.[\![M]\!])y = (\lambda x.(\lambda k_1.k_1 x))y \to \lambda k_1.k_1 y = [\![y]\!] = [\![x[x := y]]\!]$$

だから良い．$M = \lambda z.M_1$ の時は $z \not\equiv x, x \in \mathrm{FV}(M_1)$ だから

$$\begin{aligned} (\lambda x.[\![M]\!])y &= (\lambda x.(\lambda k_1.k_1(\lambda z.[\![M_1]\!])))y \to \lambda k_1.k_1(\lambda z.([\![M_1]\!][x := y])) = \lambda k_1.k_1(\lambda z.((\lambda x.[\![M_1]\!])y)) \\ &\to \lambda k_1.k_1(\lambda z.[\![M_1[x := y]]\!]) = [\![\lambda z.(M_1[x := y])]\!] = [\![(\lambda z.M_1)[x := y]]\!] \end{aligned}$$

となり良い．$M = M_1 M_2$ の時は

$$\begin{aligned} (\lambda x.[\![M]\!])y &= (\lambda x.(\lambda k_1.[\![M_1]\!](\lambda m_1.[\![M_2]\!](\lambda n_1.m_1 n_1 k_1))))y \\ &\to \lambda k_1.([\![M_1]\!][x := y])(\lambda m_1.([\![M_2]\!][x := y])(\lambda n_1.m_1 n_1 k_1)) \\ &\to \lambda k_1.((\lambda x.[\![M_1]\!])y)(\lambda m_1.((\lambda x.[\![M_2]\!])y)(\lambda n_1.m_1 n_1 k_1)) \\ &\to \lambda k.[\![M_1[x := y]]\!](\lambda m_1.[\![M_2[x := y]]\!](\lambda n_1.m_1 n_1 k)) \\ &= [\![(M_1[x := y])(M_2[x := y])]\!] = [\![(M_1 M_2)[x := y]]\!] \end{aligned}$$

だから良い．これで示された．

- $N = \lambda y.N_1$ の時，

$$\begin{aligned} (*) &= \lambda k.(\lambda k_1.k_1(\lambda y.[\![N_1]\!]))(\lambda n.(\lambda x.[\![M]\!])nk) \\ &\to \lambda k.(\lambda n.(\lambda x.[\![M]\!])nk)(\lambda y.[\![N_1]\!]) \to \lambda k.(\lambda x.[\![M]\!])(\lambda y.[\![N_1]\!])k \end{aligned}$$

よって $N = y$ の時と同様，$x \in \mathrm{FV}(M)$ の場合に $(\lambda x.[\![M]\!])(\lambda y.[\![N_1]\!]) \to [\![M[x := N]]\!]$ となることを示せば良い．M の構造に関する帰納法で示す．$M = x$ の時は

$$(\lambda x.[\![x]\!])(\lambda y.[\![N_1]\!]) = (\lambda x.(\lambda k_1.k_1 x))(\lambda y.[\![N_1]\!]) \to \lambda k_1.k_1(\lambda y.[\![N_1]\!]) = [\![\lambda y.N_1]\!] = [\![x[x := N]]\!]$$

だから良い．$M = \lambda z.M_1$ の時は $z \not\equiv x, x \in \mathrm{FV}(M_1)$ だから

$$\begin{aligned}(\lambda x.[\![M]\!])(\lambda y.[\![N_1]\!]) &= (\lambda x.(\lambda k_1.k_1(\lambda z.[\![M_1]\!])))(\lambda y.[\![N_1]\!]) \to \lambda k_1.k_1(\lambda z.[\![M_1]\!][x := \lambda y.[\![N_1]\!]]) \\ &= \lambda k_1.k_1(\lambda z.(\lambda x.[\![M_1]\!])(\lambda y.[\![N_1]\!])) \to \lambda k_1.k_1(\lambda z.[\![M_1[x := N]]\!]) \\ &= [\![\lambda z.(M_1[x := N])]\!] = [\![(\lambda z.M_1)[x := N]]\!]\end{aligned}$$

となり良い．$M = M_1 M_2$ の時は

$$\begin{aligned}(\lambda x.[\![M]\!])(\lambda y.[\![N_1]\!]) &= (\lambda x.(\lambda k.[\![M_1]\!](\lambda m.[\![M_2]\!](\lambda n.mnk))))(\lambda y.[\![N_1]\!]) \\ &\to \lambda k.([\![M_1]\!][x := \lambda y.[\![N_1]\!]])(\lambda m.([\![M_2]\!][x := \lambda y.[\![N_1]\!]])(\lambda n.mnk)) \\ &\to \lambda k.((\lambda x.[\![M_1]\!])(\lambda y.[\![N_1]\!]))(\lambda m.((\lambda x.[\![M_2]\!])(\lambda y.[\![N_1]\!]))(\lambda n.mnk)) \\ &\to \lambda k.[\![M_1[x := N]]\!](\lambda m.[\![M_2[x := N]]\!](\lambda n.mnk)) \\ &= [\![(M_1[x := N])(M_2[x := N])]\!] = [\![(M_1 M_2)[x := N]]\!]\end{aligned}$$

だから良い．これで示された．

(ii) $M = x, N = yy$ とすると $M[x := N] = yy$. この時

$$\begin{aligned}[\![yy]\!] &= \lambda k.[\![y]\!](\lambda m.[\![y]\!](\lambda n.mnk)) = \lambda k.(\lambda k_1.k_1 y)(\lambda m.[\![y]\!](\lambda n.mnk)) \\ &\to \lambda k.(\lambda m.[\![y]\!](\lambda n.mnk))y = \lambda k.(\lambda m.(\lambda k_1.k_1 y)(\lambda n.mnk))y \\ &\to \lambda k.(\lambda k_1.k_1 y)(\lambda n.ynk) \to \lambda k.(\lambda n.ynk)y \to \lambda k.yyk\end{aligned}$$

であるが，$(*)$ より

$$\begin{aligned}[\![(\lambda x.x)(yy)]\!] &\to \lambda k.[\![yy]\!](\lambda n.(\lambda x.[\![x]\!])nk) \to \lambda k.(\lambda k_1.yyk_1)(\lambda n.(\lambda x.[\![x]\!])nk) \\ &\to \lambda k.yy(\lambda n.(\lambda x.[\![x]\!])nk) = \lambda k.yy(\lambda n.(\lambda x.(\lambda k_1.k_1 x))nk) \\ &\to \lambda k.yy(\lambda n.(\lambda k_1.k_1 n)k) \to \lambda k.yy(\lambda n.kn)\end{aligned}$$

だから $[\![(\lambda x.M)N]\!] \to [\![M[x := N]]\!]$ は成立しない． $\square$

平成 24 年度 (2011 年 8 月実施)

問 11

命題変数の集合 $\mathcal{V}$ が与えられているものとし，

$$x_1 \wedge x_2 \wedge \cdots \wedge x_m \qquad (m \geq 1, x_i \in \mathcal{V})$$

および

$$x_1 \wedge x_2 \wedge \cdots \wedge x_m \wedge \neg x_{m+1} \qquad (m \geq 0, x_i \in \mathcal{V})$$

の形をした論理式全体の集合を $\mathcal{A}$ とする．論理式

$$A = A_1 \vee A_2 \vee \cdots \vee A_n \qquad (n \geq 1, A_i \in \mathcal{A})$$

がトートロジー（恒真式）であるかどうかを $O(|A|^2)$ の計算量（ステップ数）で判定するアルゴリズムを与えよ．ただし，論理式 A に対し，$|A|$ は A の中の命題変数の出現回数，すなわち $|x| = 1\,(x \in \mathcal{V}), |\neg A'| = |A'|, |A' \wedge A''| = |A' \vee A''| = |A'| + |A''|$ で定まるものとする．アルゴリズムの正しさと，計算量の条件が満たされる理由も説明せよ．

解答． Horn-SAT という問題らしい.[1] □

[1] `https://people.eecs.berkeley.edu/~satishr/cs270.06/lecture1.pdf` や `https://www.sciencedirect.com/science/article/pii/0743106684900141` の Algorithm 1 などが参考になるかも．

平成 23 年度 (2010 年 8 月実施)

問 11

与えられた文字の集合 Σ に対し，有限長の文字列（空列 ε も含む）の集合 Σ^* 上の二項関係 R を以下のように定める．

- 文字列 v に 1 文字を挿入して文字列 w が得られるとき，vRw.
- 文字列 v の中の 1 文字を削除して文字列 w が得られるとき，vRw.

R を用いて，Σ^* 上の距離 d を

$$d(v,w) = \min\{k \mid \exists v_0, \dots, v_k \in \Sigma^*, v_0 = v, v_0 R v_1, \dots, v_{k-1} R v_k, v_k = w\}$$

により定める．また，$v \in \Sigma^*$ について，v の長さを $|v|$ で表す．任意の二つの文字列 $v, w \in \Sigma^*$ に対し $d(v,w)$ を $O(|v| \cdot |w|)$ の計算量（ステップ数）で求めるアルゴリズムを与えよ．アルゴリズムの正しさと計算量の条件が満たされる理由も説明せよ．

解答． vRw の時,「v に 1 回操作をすれば w が得られる」と呼ぶことにする．$v \in \Sigma^*$ の 1 文字目から i 文字からなる部分文字列を $v(i)$ と書く．また，$v \neq \varepsilon$ に対し，その i 文字目を $v[i]$ と書く．v の末尾の文字を削除することを $|v|$ 回行えば ε が得られるから $d(v,\varepsilon) \leq |v|$ である．一方 vRw なら $|v| - |w| = \pm 1$ となることから，$d(v,\varepsilon) \geq |v|$ である．従って $d(v,\varepsilon) = |v|$. 同様に $d(\varepsilon, w) = |w|$. 今 $v, w \in \Sigma^* \setminus \{\varepsilon\}$ とする．$v(i)$ に操作を行い $w(j)$ を得る時の末尾への操作は，(i) 末尾に 1 文字追加する，(ii) 末尾を削除する，(iii) 何もしないの 3 種類がある．これらの操作は $v(i)$ への操作の最後に行うとして良い．
(i) の場合，$v(i)R \cdots Rw(j-1)Rw(j-1)w[j] = w(j)$ だから操作回数は $d(v(i), w(j-1)) + 1$.
(ii) の場合，$v(i) = v(i-1)v[i]R \cdots Rw(j)v[i]Rw(j)$ だから操作回数は $d(v(i-1), w(j)) + 1$.
(iii) の場合，$v(i) = v(i-1)v[i]R \cdots Rw(j-1)v[i]$ だから $v[i] = w[j]$ となることが必要で，その時操作回数は $d(v(i-1), w(j-1))$.
$d(v(i), w(j))$ はこれらの最小値だから，アルゴリズムは以下の通り．

function EditDistance(string v, string w)
　$N = |v|, M = |w|$
　for $i = 0, \dots, N$
　　$d(i,0) = i$
　for $j = 0, \dots, M$
　　$d(0,j) = j$
　for $i = 0, \dots, N$
　　for $j = 0, \dots, M$
　　　$d(i,j) = \min(d(i,j-1), d(i-1,j)) + 1$
　　　if $v[i] = w[j]$ **then**
　　　　$d(i,j) = \min(d(i,j), d(i-1,j-1))$
　return $d(N,M)$

計算量は最初の for ループで $O(|v|)$, 次の for ループで $O(|w|)$, 最後の二重ループで $O(|v||w|)$ だから，全体で $O(|v||w|)$ となる． □

問 12

$0,1$ の有限列全体の集合を $\mathbf{B}$ で表す．$w \in \mathbf{B}$ に対してラムダ項 $\overline{w}$ を次のように定める．

$$\begin{aligned} \overline{w} &= \lambda f_0 f_1 x.M_w \\ M_\varepsilon &= x \\ M_{iv} &= f_i(M_v) \end{aligned}$$

ただし ε は空列を表し，iv は $v \in \mathbf{B}$ に $i \in \{0,1\}$ を前置して得られる列を表す．以下では，β 簡約関係 $\longrightarrow_\beta$ の反射的推移的閉包を $\longrightarrow_\beta^*$ で表す．

(i) 次の条件を満たすラムダ項 F を一つ与えよ．

$$F\overline{i_1 i_2 \cdots i_n} \longrightarrow_\beta^* \overline{(1-i_1)(1-i_2)\cdots(1-i_n)}$$

(ii) 次の条件を満たすラムダ項 R を一つ与えよ．

$$R\overline{i_1 i_2 \cdots i_n} \longrightarrow_\beta^* \overline{i_n \cdots i_2 i_1}$$

解答． $\longrightarrow_\beta^*$ を $\to$ と略記する．

(i) $F = \lambda x g_0 g_1.x g_1 g_0$ とすれば

$$\begin{aligned} F\overline{i_1 i_2 \cdots i_n} &= (\lambda x g_0 g_1.x g_1 g_0)\overline{i_1 i_2 \cdots i_n} = \lambda g_0 g_1.(\overline{i_1 i_2 \cdots i_n} g_1 g_0) \\ &= \lambda g_0 g_1.(\lambda f_0 f_1 x.f_{i_1}(f_{i_2}(\cdots(f_{i_n}(x))))g_1 g_0) \\ &\to \lambda g_0 g_1.(\lambda x.g_{1-i_1}(g_{1-i_2}(\cdots(g_{1-i_n}(x))))) \\ &= \lambda g_0 g_1 x.g_{1-i_1}(g_{1-i_2}(\cdots(g_{1-i_n}(x)))) = \overline{(1-i_1)(1-i_2)\cdots(1-i_n)} \end{aligned}$$

となり条件を満たす．

(ii) $I = \lambda x.x, L_i = \lambda ab.a(g_i b)\,(i=1,2)$ とおく．$R = \lambda w g_0 g_1 z.w L_0 L_1 I z$ とすれば

$$\begin{aligned} R\overline{i_1 i_2 \cdots i_n} &= (\lambda w g_0 g_1 z.w L_0 L_1 I z)\overline{i_1 i_2 \cdots i_n} \to \lambda g_0 g_1 z.\overline{i_1 i_2 \cdots i_n} L_0 L_1 I z \\ &\to \lambda g_0 g_1 z.(\lambda f_0 f_1 x.f_{i_1}(f_{i_2}(\cdots(f_{i_n}(x)))))L_0 L_1 I z \\ &\to \lambda g_0 g_1 z.L_{i_1}(L_{i_2}(\cdots(L_{i_n}(I))))z. \end{aligned}$$

ここで

$$\begin{aligned} L_{i_1}(L_{i_2}(\cdots(L_{i_n}(I))))z &= (\lambda ab.a(g_{i_1}b))(L_{i_2}(\cdots(L_{i_n}(I))))z \to (L_{i_2}(\cdots(L_{i_n}(I))))(g_{i_1}z) \\ &\to \cdots \to L_{i_n}(I)(g_{i_{n-1}}(\cdots(g_{i_1}z))) = L_{i_n}I(g_{i_{n-1}}(\cdots(g_{i_1}(z)))) \\ &\to I(g_{i_n}(g_{i_{n-1}}(\cdots(g_{i_1}(z))))) \to g_{i_n}(g_{i_{n-1}}(\cdots(g_{i_1}(z)))) \end{aligned}$$

だから

$$R\overline{i_1 i_2 \cdots i_n} \to \lambda g_0 g_1 z.g_{i_n}(g_{i_{n-1}}(\cdots(g_{i_1}(z)))) = \overline{i_n \cdots i_2 i_1}$$

となり条件を満たす． □

平成 22 年度 (2009 年 8 月実施)

問 10

p と q は相異なる正整数とする．$\{p^i q^j \mid i, j = 0, 1, 2, \cdots\}$ の要素を小さいものから順に n 個列挙するプログラムで，高々 n に比例したステップ数の計算しか必要としないものを書け．またそれがステップ数についての要求を満たしていることを示せ．プログラムは C, FORTRAN, Pascal 等の手続き型（命令型）プログラミング言語か，Java 等のオブジェクト指向プログラミング言語を用いて書くこと．なお言語の詳細に拘る必要はない．

解答． $\min(p,q) = 1$ の場合，$p \neq q$ より $r := \max(p,q) \geq 2$ である．この時 $r^k\,(k = 0, 1, \ldots, n-1)$ を列挙すれば良いから，$p = r, q = r^2$ の場合に帰着される．以下 $\min(p,q) > 1$ とする．この時 $p^i q^j\,(i, j = 0, 1, 2, \ldots)$ の形の整数のうち，小さい方から n 番目のものを a_n とし，$A_n = \{a_i\,;\, i = 1, 2, \ldots, n\}$ とおく．この時

$$\begin{aligned} a_{n+1} &= \min((pA_n \cup qA_n) \setminus A_n) \\ &= \min(\min(pA_n \setminus A_n), \min(qA_n \setminus A_n)) \end{aligned}$$

である．よって

$$j_n = \min\{i\,;\, pa_i \notin A_n\}, \quad k_n = \min\{i\,;\, qa_i \notin A_n\}$$

とおけば $a_{n+1} = \min(pa_{j_n}, qa_{k_n})$ が成り立つ．また，$\min(p,q) > 1$ より a_n は狭義単調増加なので $pa_i \notin A_n\,(i \geq j_n), qa_i \notin A_n\,(i \geq k_n)$ である．従って以下の漸化式が得られる．

- $pa_{j_n} < qa_{k_n}$ の時：$a_{n+1} = pa_{j_n}, j_{n+1} = j_n + 1, k_{n+1} = k_n$.
- $pa_{j_n} > qa_{k_n}$ の時：$a_{n+1} = qa_{k_n}, j_{n+1} = j_n, k_{n+1} = k_n + 1$.
- $pa_{j_n} = qa_{k_n}$ の時：$a_{n+1} = pa_{j_n} = qa_{k_n}, j_{n+1} = j_n + 1, k_{n+1} = k_n + 1$.

$a_1 = 1, j_1 = k_1 = 1$ だから，C によるアルゴリズムは以下のようになる．

```
int a[n+1], j, k, next_p, next_q;    // a[1],...,a[n] が求めるもの
if(min(p, q) == 1){
    int r = max(p, q);
    p = r; q = r * r;
}
a[1] = 1, j = 1, k = 1;
for(int i = 2; i <= n; i++){
    next_p = p * a[j]; next_q = q * a[k];
    a[i] = min(next_p, next_q);
    if(a[i] == next_p) j++;
    if(a[i] == next_q) k++;
}
```

これは for ループで計算量が $O(n)$ かかるから，全体の計算量も $O(n)$ となる． □

問 12

X を無限集合，$X^{(1)}$ を X 上の関数 $f : X \to X$ 全体の集合，$X^{(2)}$ を $X^{(1)}$ 上の関数 $F : X^{(1)} \to X^{(1)}$ 全体の集合とする．関数列 $F_0, F_1, F_2 \cdots \in X^{(2)}$ を

$$\begin{aligned} F_0(f) &= I \qquad (I \text{ は } X \text{ 上の恒等関数 } I(x) = x) \\ F_{n+1}(f) &= f \circ F_n(f) \end{aligned}$$

により定義する．また X 上の二項関係 $R \subseteq X \times X$ が与えられたとき，$R^{(1)} \subseteq X^{(1)} \times X^{(1)}, R^{(2)} \subseteq X^{(2)} \times X^{(2)}$ をそれぞれ

$$\begin{aligned} fR^{(1)}g &\iff \forall a, b \in X\,(aRb \Longrightarrow f(a)Rg(b)) \\ FR^{(2)}G &\iff \forall f, g \in X^{(1)}\,(fR^{(1)}g \Longrightarrow F(f)R^{(1)}F(g)) \end{aligned}$$

により定義する．このとき，任意の $F \in X^{(2)}$ に対して次の条件が同値であることを示せ．

1. ある $n \geq 0$ について $F = F_n$ となる．
2. 全ての $R \subseteq X \times X$ について $FR^{(2)}F$ が成り立つ．

解答． ● 1. ⇒ 2. n についての帰納法で示す．
$n = 0$ の時，任意に R を固定する．aRb なる任意の $a, b \in X$ に対し $I(a)RI(b)$ だから $IR^{(1)}I$. よって F_0 の定義から，$fR^{(1)}g$ なる任意の $f, g \in X^{(1)}$ に対し $F_0(f)R^{(1)}F_0(g)$. 従って $F_0R^{(2)}F_0$. 今 R は任意なので正しい．
n で正しいとして，任意に R を固定する．$fR^{(1)}g$ なる任意の $f, g \in X^{(1)}$ に対し $F_n(f)R^{(1)}F_n(g)$. よって aRb なる任意の $a, b \in X$ に対し $F_n(f)(a)RF_n(g)(b)$. ここで $fR^{(1)}g$ だから $f(F_n(f)(a))Rg(F_n(g)(b))$. $a, b \in X$ は aRb を満たす任意の元だったから $f \circ F_n(f)R^{(1)}g \circ F_n(g)$, すなわち $F_{n+1}(f)R^{(1)}F_{n+1}(g)$. $f, g \in X^{(1)}$ は $fR^{(1)}g$ を満たす任意の元だったから $F_{n+1}R^{(2)}F_{n+1}$. R は任意なので $n+1$ でも正しい．これで示された．

● 2. ⇒ 1. $|X| = \infty$ だから，相異なる X の点列 $x_0, x_1, \ldots$ が存在する．$X' = \{x_i\,;\, i \geq 0\}$ とおき，$f_0 \in X^{(1)}$ を $f_0(x_i) = x_{i+1}\,(i \geq 0), f_0(x) = x\,(x \in X \setminus X')$ で定める．まず $F(f_0) = f_0^n$ なる $n \geq 0$ が存在することを示す．$R = \{(x, f_0(x))\,;\, x \in X'\}$ とすると，aRb なる $a, b \in X$ は，$i \geq 0$ が存在して $a = x_i, b = f_0(x_i) = x_{i+1}$ と書ける．この時 $f_0(a) = x_{i+1}, f_0(b) = f_0(x_{i+1})$ だから $f_0(a)Rf_0(b)$. a, b は任意だから $f_0R^{(1)}f_0$. これと仮定から $F(f_0)R^{(1)}F(f_0)$. よって任意の $i \geq 0$ に対し $n_i \geq 0$ が存在して $F(f_0)(x_i) = x_{n_i}, F(f_0)(f_0(x_i)) = f_0(x_{n_i})$ と書ける．従って

$$x_{n_{i+1}} = F(f_0)(x_{i+1}) = F(f_0)(f_0(x_i)) = f_0(x_{n_i}) = x_{n_i+1}.$$

ここで x_i たちは相異なるから $n_{i+1} = n_i + 1$. よって $n_i = n_0 + i$ なので $F(f_0)(x_i) = x_{n_0+i} = f_0^{n_0}(x_i)$. 次に任意に $x \in X \setminus X'$ を固定し，$R = \{(x, x)\}$ とすると，$f_0(x) = x$ より $f_0R^{(1)}f_0$. 従って $F(f_0)R^{(1)}F(f_0)$ なので $F(f_0)(x) = x = f_0^{n_0}(x)$. 以上から $F(f_0) = f_0^{n_0}$ となり示された．この n_0 を n と書く．任意に $x \in X, f \in X^{(1)}$ を固定し，$R = \{(f^i(x), f_0^i(x_0))\,;\, i \geq 0\}$ とする．この時 $fR^{(1)}f_0$ だから $F(f)R^{(1)}F(f_0)$. よって任意の $i \geq 0$ に対し $m_i \geq 0$ が存在して $F(f)(f^i(x)) = f^{m_i}(x), F(f_0)(f_0^i(x_0)) = f_0^{m_i}(x_0)$ と書ける．上で示したことから，第 2 式の左辺は $f_0^{n+i}(x_0) = x_{n+i}$. また右辺は x_{m_i} だから，$m_i = n + i$. よって第 1 式で $i = 0$ として $F(f)(x) = f^{m_0}(x) = f^n(x)$. x は任意だから $F(f) = f^n = F_n(f)$. f は任意だから $F = F_n$.

(補足) X が可算無限集合ならば，$X' = X$ とできるから上の議論は少し簡単になる． □

平成 21 年度 (2008 年 8 月実施)

問 11

頂点集合 V, 辺集合 $A \subset V \times V$ からなる有向グラフ $G = (V, A)$ において，頂点の番号付け（全単射）$\sigma : V \to \{1, 2, \cdots, n\}$ を考える．ただし，G は自己閉路を含まず，$n = |V|, m = |A|$ とする．始点 u, 終点 v の辺 $a = (u, v) \in A$ は，$\sigma(u) < \sigma(v)$ のとき前向き，$\sigma(u) > \sigma(v)$ のとき後向きという．有向閉路 C の中の後向きの辺の本数を C の回転数といい，$w(C)$ と表す．$w(C) = 1$ となる有向閉路 C は整合的とよばれる．すべての辺 $a \in A$ に対して，a を含む整合的な有向閉路が存在するとき，頂点の番号付け σ は整合的であるという．与えられた σ が整合的か否かを判定する高々 $O(nm)$ 時間のアルゴリズムを設計せよ．

解答. □

問 12

型のないラムダ計算において，線形なラムダ項を以下のように定義する．

1. 変数は線形なラムダ項である．
2. M が線形なラムダ項であり，x が M の自由変数に含まれるとき，$\lambda x.M$ は線形なラムダ項である．
3. M と N が線形なラムダ項であり，M と N に共通する自由変数がないとき，MN は線形なラムダ項である．

ラムダ項の上の 1 ステップの β 簡約関係を $\to_\beta$ で表すこととする．すなわち，$\to_\beta$ は

- $(\lambda x.M)N \to_\beta M[x := N]$
- $M \to_\beta M'$ ならば，$\lambda x.M \to_\beta \lambda x.M', MN \to_\beta M'N$ および $NM \to_\beta NM'$

をみたすラムダ項の上の最小の二項関係である．以下の問に答えよ．

(i) M が線形なラムダ項であり，かつ $M \to_\beta N$ であるならば，N も線形なラムダ項であることを示せ．

(ii) M が線形なラムダ項であるとき，M からはじまる無限に長い β 簡約列

$$M \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots$$

は存在しないことを示せ．

(iii) M が線形なラムダ項であり，$M \to_\beta N_1$ かつ $M \to_\beta N_2$ であるならば，ある線形なラムダ項 L が存在して，$N_1 \to_\beta L$ かつ $N_2 \to_\beta L$ となることを示せ．

解答． $\to_\beta$ を $\to$ と略記する．ラムダ項 M の自由変数の集合を $\mathrm{FV}(M)$ とおく．

(i) まず，M が線形なラムダ項で $M \to N$ ならば $\mathrm{FV}(M) = \mathrm{FV}(N)$ であることを示す．[2]

• $(\lambda x.M)N \to M[x := N]$ について：$x \in \mathrm{FV}(M)$ より

$$\mathrm{FV}((\lambda x.M)N) = \mathrm{FV}(\lambda x.M) \cup \mathrm{FV}(N) = (\mathrm{FV}(M) \setminus \{x\}) \cup \mathrm{FV}(N) = \mathrm{FV}(M[x := N])$$

だから良い．

• $M \to M'$ に対して正しい時：$\mathrm{FV}(M) = \mathrm{FV}(M')$ だから

$$\begin{aligned}
\mathrm{FV}(\lambda x.M) &= \mathrm{FV}(M) \setminus \{x\} = \mathrm{FV}(M') \setminus \{x\} = \mathrm{FV}(\lambda x.M'),\\
\mathrm{FV}(MN) &= \mathrm{FV}(M) \cup \mathrm{FV}(N) = \mathrm{FV}(M') \cup \mathrm{FV}(N) = \mathrm{FV}(M'N),\\
\mathrm{FV}(NM) &= \mathrm{FV}(N) \cup \mathrm{FV}(M) = \mathrm{FV}(N) \cup \mathrm{FV}(M') = \mathrm{FV}(NM').
\end{aligned}$$

よって示された．問題に戻る．β 簡約の形で場合分けする．

• $\lambda x.M \to \lambda x.M'$ の時：$x \in \mathrm{FV}(M) = \mathrm{FV}(M')$ だから $\lambda x.M'$ も線形なラムダ項である．

• $MN \to M'N$ の時：$\emptyset = \mathrm{FV}(M) \cap \mathrm{FV}(N) = \mathrm{FV}(M') \cap \mathrm{FV}(N)$ だから $M'N$ も線形なラムダ項である．$NM \to NM'$ の時も同様．

• $(\lambda x.M)N \to M[x := N]$ の時：M の構造に関する帰納法で示す．

(a) M が変数の時：$x \in \mathrm{FV}(M)$ より $M = x$ だから，$M[x := N] = N$ は線形なラムダ項である．

(b) $M = \lambda y.M_1$ の時：$M[x := N] = \lambda y.(M_1[x := N])$ であり，帰納法の仮定から $M_1[x := N]$ は線形なラムダ項である．また $y \in \mathrm{FV}(M_1), y \neq x$ より

$$\mathrm{FV}(M_1[x := N]) = \mathrm{FV}((\lambda x.M_1)N) = (\mathrm{FV}(M_1) \setminus \{x\}) \cup \mathrm{FV}(N) \ni y$$

だから，$M[x := N]$ は線形なラムダ項である．

(c) $M = M_1M_2$ の時：$x \in \mathrm{FV}(M_1)$ または $x \in \mathrm{FV}(M_2)$ のどちらかが成り立つ．$x \in \mathrm{FV}(M_1)$ とする．$M[x := N] = (M_1[x := N])M_2$ であり，帰納法の仮定から $M_1[x := N], M_2$ はともに線形なラムダ項である．ここで $\mathrm{FV}(\lambda x.M) = (\mathrm{FV}(M_1) \setminus \{x\}) \cup \mathrm{FV}(M_2)$ より

$$\emptyset = \mathrm{FV}(\lambda x.M) \cap \mathrm{FV}(N) = ((\mathrm{FV}(M_1) \setminus \{x\}) \cap \mathrm{FV}(N)) \cup (\mathrm{FV}(M_2) \cap \mathrm{FV}(N))$$

[2] 一般のラムダ項に対しては $(\lambda x.M)N$ のところの 2 番目の等号が $\supset$ になるため，$\mathrm{FV}(M) \supset \mathrm{FV}(N)$ しか言えない．例えば $M = (\lambda x.y)z$ とすると $\mathrm{FV}(M) = \{y, z\}, \mathrm{FV}(N) = \mathrm{FV}(y) = \{y\}$．

だから，$\mathrm{FV}(M_2)\cap\mathrm{FV}(N)=\emptyset$. よって

$$\begin{aligned}\mathrm{FV}(M_1[x:=N])\cap\mathrm{FV}(M_2)&=\mathrm{FV}((\lambda x.M_1)N)\cap\mathrm{FV}(M_2)\\&\subset(\mathrm{FV}(M_1)\cup\mathrm{FV}(N))\cap\mathrm{FV}(M_2)\\&=(\mathrm{FV}(M_1)\cap\mathrm{FV}(M_2))\cup(\mathrm{FV}(N)\cap\mathrm{FV}(M_2))\\&=\emptyset\end{aligned}$$

なので $M[x:=N]$ は線形なラムダ項である．$x\in FV(M_2)$ の時も同様.

以上で示された.

(ii)

(iii) $\square$

問 13

集合 X と，X 上の二項関係 $\xrightarrow{a}, \xrightarrow{b}$ の 3 つ組を，a と b をラベルに持つ状態遷移システム（略してシステム）といい，集合 X の元を $\mathcal{S}$ の状態とよぶ．たとえば，$X = \{x_0, x_1\}, \xrightarrow{a} = \{(x_0, x_0)\}, \xrightarrow{b} = \{(x_0, x_1)\}$ からなる 3 つ組 $\mathcal{S} = (X, \xrightarrow{a}, \xrightarrow{b})$

$$
\begin{tikzcd}
x_0 \arrow[loop above, "a"] \arrow[r, "b"] & x_1
\end{tikzcd}
$$

は，システムの例である．

システム $\mathcal{S} = (X, \xrightarrow{a}, \xrightarrow{b})$ が与えられたとする．状態集合 X の二項関係 R が次の条件 (A), (B) を同時にみたすとき，R を $\mathcal{S}$ 上の双模倣関係とよぶ．

(A) 任意の状態 $x, y \in X$ および各ラベル $l \in \{a, b\}$ について
$(x, y) \in R \wedge x \xrightarrow{l} x' \quad \Longrightarrow \quad \exists y' \in X.(y \xrightarrow{l} y' \wedge (x', y') \in R).$

(B) 任意の状態 $x, y \in X$ および各ラベル $l \in \{a, b\}$ について
$(x, y) \in R \wedge y \xrightarrow{l} y' \quad \Longrightarrow \quad \exists x' \in X.(x \xrightarrow{l} x' \wedge (x', y') \in R).$

また，システム $\mathcal{S}$ の状態 x, y について，$\mathcal{S}$ 上の双模倣関係で (x, y) を含むものが存在することを，$x \sim y$ であらわす．

次の問に答えよ．

(i) 次のシステム $\mathcal{S}_1, \mathcal{S}_2$ について，それぞれ $x_0 \sim y_0$ が成立するかどうか調べよ．

$$
\mathcal{S}_1 = \begin{pmatrix} X = \{x_0, y_0, y_1\}, \\ \xrightarrow{a} = \{(x_0, x_0), (y_0, y_1), (y_1, y_0)\}, \\ \xrightarrow{b} = \emptyset \end{pmatrix},
$$

$$
\mathcal{S}_2 = \begin{pmatrix} X = \{x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, y_4\}, \\ \xrightarrow{a} = \{(x_0, x_1), (x_1, x_3), (y_0, y_1), (y_0, y_2), (y_2, y_4)\}, \\ \xrightarrow{b} = \{(x_1, x_2), (y_1, y_3)\} \end{pmatrix}.
$$

(ii) システム $\mathcal{S}$ が与えられたとき，$\sim$ が $\mathcal{S}$ の状態集合上の同値関係となることを示せ．さらに，$\sim$ が双模倣関係であることを示せ．

解答． (i) • $\mathcal{S}_1$：

$$
\begin{tikzcd}
x_0 \arrow[loop above, "a"] & y_0 \arrow[r, shift left, "a"] & y_1 \arrow[l, shift left, "a"]
\end{tikzcd}
$$

$x_0 \sim y_0$ である．それには $R = \{(x_0, y_0), (x_0, y_1)\}$ が $\mathcal{S}_1$ 上の双模倣関係であることを示せば良い．対称性から (x_0, y_0) が (A), (B) を満たすことを示せば十分．$(x_0, y_0) \in R \wedge x_0 \xrightarrow{a} x_0$ であるが，$y_0 \xrightarrow{a} y_1 \wedge (x_0, y_1) \in R$ だから (A) は良い．$(x_0, y_0) \in R \wedge y_0 \xrightarrow{a} y_1$ であるが，$x_0 \xrightarrow{a} x_0 \wedge (x_0, y_1) \in R$ だから (B) も良い．これで示せた．

• $\mathcal{S}_2$：

$$
\begin{tikzcd}
x_0 \arrow[r, "a"] & x_1 \arrow[r, "b"] \arrow[dr, "a"] & x_2 & y_0 \arrow[r, "a"] \arrow[dr, "a"] & y_1 \arrow[r, "b"] & y_3 \\
 & & x_3 & & y_2 \arrow[r, "a"] & y_4
\end{tikzcd}
$$

$x_0 \sim y_0$ は成立しない．$x_0 \sim y_0$ であったとすると $(x_0, y_0) \in R$. $x_0 \xrightarrow{a} x_1$ だから，$(x_1, y_1) \in R$ または $(x_1, y_2) \in R$ である．$(x_1, y_1) \in R$ とすると，$x_1 \xrightarrow{a} x_3$ より $y_1 \xrightarrow{a} y' \wedge (x_3, y') \in R$ なる $y' \in X$ が存在することになり矛盾．$(x_1, y_2) \in R$ とすると，$x_1 \xrightarrow{b} x_2$ より $y_2 \xrightarrow{b} y' \wedge (x_2, y') \in R$ なる $y' \in X$ が存在することになり矛盾．

(ii) ● $\sim$ が同値関係であること：

○ 反射律：$\mathcal{S}$ 上の双模倣関係 R に対し $R' = R \cup \{(x,x)\,;\, x \in X\}$ とおく．任意に $x \in X$ を取る．$x \xrightarrow{l} x'$ となる $x' \in X, l \in \{a,b\}$ があるとすると，x' が R' に対する (A), (B) を満たす．従って R' は $\mathcal{S}$ 上の双模倣関係となるから $x \sim x$.

○ 対称律：$\mathcal{S}$ 上の双模倣関係 R に対し $R' = \{(y,x)\,;\,(x,y) \in R\}$ も $\mathcal{S}$ 上の双模倣関係となることを示す．任意に $(x,y) \in R$ を取る．$y \xrightarrow{l} y'$ となる $y' \in X, l \in \{a,b\}$ があるとする．R に対する (B) より $x \xrightarrow{l} x', (x',y') \in R$ なる $x' \in X$ が存在する．よって $x \xrightarrow{l} x', (y',x') \in R'$ だから R' は (A) を満たす．同様に (B) も満たす．これで示された．今 $x \sim y$ とすると $(x,y) \in R$ だから $(y,x) \in R'$. 従って $y \sim x$.

○ 推移律：R を $(x,y),(y,z) \in R$ なる $\mathcal{S}$ 上の双模倣関係とする．この時 $R' = R \cup \{(x,z)\,;\,\exists y \in X, (x,y),(y,z) \in R\}$ が $\mathcal{S}$ 上の双模倣関係となることを示せば良い．$x \xrightarrow{l} x'$ となる $x' \in X, l \in \{a,b\}$ があるとすると，R に対する (A) より $y \xrightarrow{l} y', (x',y') \in R$ なる $y' \in X$ が存在する．再び R に対する (A) より $z \xrightarrow{l} z', (y',z') \in R$ なる $z' \in X$ が存在する．よって $(x,z) \in R, x \xrightarrow{l} x'$ に対し $z' \in X$ は R' に対する (A) を満たす．(B) も同様．$y \xrightarrow{l} y', z \xrightarrow{l} z'$ がある場合も同様．これで示された．

以上から $\sim$ は $\mathcal{S}$ の状態集合上の同値関係となる．

● $\sim$ が双模倣関係であること：$x \sim y \wedge x \xrightarrow{l} x'$ となる任意の $x, y \in X$ を取る．$x \sim y$ だから，双模倣関係 R であって $(x,y) \in R$ を満たすものが存在する．この時 $(x,y) \in R \wedge x \xrightarrow{l} x'$ だから (A) より $\exists y' \in X.(y \xrightarrow{l} y' \wedge (x',y') \in R)$ となる．$(x',y') \in R$ だから $x' \sim y'$. 従って $\sim$ は (A) を満たす．(B) も同様なので，$\sim$ は双模倣関係となる．□

平成 20 年度 (2007 年 8 月実施)

問 10

n を 2 以上の整数とする．n 個の要素からなる有限集合 S と関数 $d : S \times S \to \mathbb{R}$ が与えられている．各 $k = 2, \cdots, n$ に対して，S を空でない部分集合 $S_1, S_2, \cdots, S_k$ に分割し，異なる成分に属する要素の順序対に対する d の最小値

$$\min\{d(u,v) \mid u \in S_i, v \in S_j, i \neq j\}$$

をできるだけ大きくしたい．この観点から最適な分割を，すべての $k = 2, \cdots, n$ に対して見出す効率的なアルゴリズムを設計し，その正当性を示すとともに，計算量を評価せよ．ただし，d の関数値計算は定数時間で実行可能とする．

解答. $d(u,v)\,(u,v \in S)$ を昇順にソートして $d_1 \leq \cdots \leq d_{n^2}$ としておく．$u_i, v_i \in S$ を $d_i = d(u_i, v_i)$ なるものとする．また，k に対する最適な分割を $S_i^{(k)}\,(i = 1, \ldots, k)$ とおく．$S = \{x_1, \ldots, x_n\}$ とすると，$k = n$ の時は $S_i^{(n)} = \{x_i\}\,(i = 1, \ldots, n)$ とするしかない．最適な分割が $n, n-1, \ldots, k+1$ まで構成出来たとする．異なる $a, b \in \{1, \ldots, k+1\}$ が存在して $u_i \in S_a^{(k+1)}, v_i \in S_b^{(k+1)}$ となるような最小の i を t_k とおく．この時 $u_{t_k} \in S_a^{(k+1)}, v_{t_k} \in S_b^{(k+1)}$ なる a, b を取り，$S_i^{(k+1)}\,(1 \leq i \leq k+1, i \neq b), S_a^{(k+1)} \cup S_b^{(k+1)}$ を $S_i^{(k)}\,(i = 1, \ldots, k)$ とすれば良いことを示す．もし $S_i^{(k+1)}$ のうち分割するものがあれば，

$$\min\{d(u,v)\,;\, u \in S_i^{(k)}, v \in S_j^{(k)}, i \neq j\} = \min_a \min\{d(u,v)\,;\, u, v \in S_a^{(k+1)}\} < d_{t_{k+1}}$$

である．ただし $\min_a$ は，分割する $S_a^{(k+1)}$ に対する a 全体を渡る．一方分割するものがない場合，t_k の定義から，任意の $i < t_{k+1}$ に対し $1 \leq a \leq k+1, 1 \leq a' \leq k$ が存在して $u_i, v_i \in S_a^{(k+1)} \subset S_{a'}^{(k)}$ だから

$$\min\{d(u,v)\,;\, u \in S_i^{(k)}, v \in S_j^{(k)}, i \neq j\} \geq d_{t_{k+1}}.$$

従って $S_i^{(k+1)}$ は分割せずに $S_i^{(k)}$ を作るのが最良である．この時ある a, b を選んで $S_a^{(k+1)}, S_b^{(k+1)}$ を合併させるしかない．a', b' を $u_{t_k} \in S_{a'}^{(k+1)}, v_{t_k} \in S_{b'}^{(k+1)}$ なるものとする時，

$$\min\{d(u,v)\,;\, u \in S_i^{(k)}, v \in S_j^{(k)}, i \neq j\} = \begin{cases} d_{t_k+1} & (S_{a'}^{(k+1)} \text{ と } S_{b'}^{(k+1)} \text{ を合併させる時}) \\ d_{t_k} & (\text{それ以外}) \end{cases}$$

だから，$S_{a'}^{(k+1)}$ と $S_{b'}^{(k+1)}$ を合併させるのが最良．よって k でも正しい．これで示された．この証明から $1 \leq t_{n-1} < \cdots < t_2 \leq n^2$ だから，アルゴリズムは以下の通り．

1: $\textsc{Sort}(d)$
2: **for** i = 1, ..., n
3: $\quad S_i^{(1)} = \{x_i\}$
4: $t = 1$
5: **for** $k = n-1, \ldots, 2$
6: $\quad$ **while** $\textsc{Find}(d_t.first) = \textsc{Find}(d_t.second)$
7: $\quad\quad t = t + 1$
8: $\quad \textsc{Merge}(1, d_t.first, d_t.second)$
9: $\quad p_k = d_t$
10: **for** $k = n, \ldots, 2$
11: $\quad$ **for** $i = 1, \ldots, n$
12: $\quad\quad S_i^{(k)} = \{x_i\}$
13: **for** $k = n-1, \ldots, 2$
14: $\quad$ **for** $j = n-1, \ldots, k$
15: $\quad\quad \textsc{Merge}(k, p_j.first, p_j.second)$

ただし，$d_t.first, d_t.second$ は $d_t = d(u_t, v_t)$ なる u_t, v_t をそれぞれ表す．また，$\textsc{Find}(i)$ は $x_i \in S_a^{(1)}$ なる a を求める関数，$\textsc{Merge}(k, i, j)$ は $x_i \in S_a^{(k)}, x_j \in S_b^{(k)}$ の時 $S_a^{(k)}, S_b^{(k)}$ を合併させる関数である．Union-Find を用いると，$\textsc{Find}, \textsc{Merge}$ の計算時間は $O(\alpha(n))$ である．(ただし $\alpha(n)$ は Ackermann 関数の逆関数.）よって計算量は 1 行目で $O(n^2 \log(n^2)) = O(n^2 \log n)$, 2 〜 3 行目で $O(n)$, 5 〜 9 行目で

$$\sum_{k=2}^{n-1} ((t_k - t_{k+1}) + \alpha(n)) = t_2 - t_n + n\alpha(n) = O(n^2),$$

10 〜 12 行目で $O(n^2)$, 13 〜 15 行目で

$$\sum_{k=2}^{n-1}\sum_{j=k}^{n-1}\alpha(n) = \sum_{k=2}^{n-1}(n-k)\alpha(n) = \frac{1}{2}(n-2)(n-1)\alpha(n) = O(n^2\alpha(n)).$$

従って全体の計算量は $O(n^2 \log n)$ となる. □

問 11

以下の λ 項を考える．

$$\mathbf{S}_1 = \lambda xyz.x \qquad \mathbf{S}_2 = \lambda xyz.y \qquad \mathbf{S}_3 = \lambda xyz.z$$
$$\mathbf{B} = \lambda x.x\mathbf{S}_1 \qquad \mathbf{P} = \lambda sx.x\mathbf{S}_1 \qquad \mathbf{T} = \lambda sx.x(s\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3)$$

λ 項 $M_1, \cdots, M_n \in \{\mathbf{P}, \mathbf{T}\}$ を任意に取る $(n \geq 0)$. このとき，

$$\mathbf{B}M_1M_2\cdots M_n$$

の β 正規形を，$M_1, \cdots, M_n$ の取り方により分類せよ．

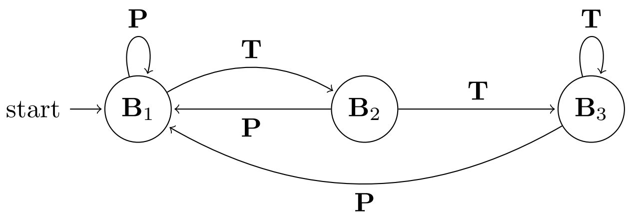
解答． $\mathbf{B}_i = \lambda x.x\mathbf{S}_i \, (i = 1, 2, 3)$ とおく．$\to$ を β 変換の反射的推移的閉包とする．この時

$$\begin{aligned}
\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 &= (\lambda xyz.x)\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 \to \mathbf{S}_2, \\
\mathbf{S}_2\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 &= (\lambda xyz.y)\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 \to \mathbf{S}_3, \\
\mathbf{S}_3\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 &= (\lambda xyz.z)\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3 \to \mathbf{S}_3
\end{aligned}$$

だから

$$\begin{aligned}
\mathbf{BP} &= (\lambda x.x\mathbf{S}_1)\mathbf{P} \to \mathbf{PS}_1 = (\lambda sx.x\mathbf{S}_1)\mathbf{S}_1 \to \lambda x.x\mathbf{S}_1 = \mathbf{B}_1, \\
\mathbf{BT} &= (\lambda x.x\mathbf{S}_1)\mathbf{T} \to \mathbf{TS}_1 = (\lambda sx.x(s\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3))\mathbf{S}_1 \to \lambda x.x(\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3) \to \lambda x.x\mathbf{S}_2 = \mathbf{B}_2, \\
\mathbf{BTP} &\to \mathbf{B}_2\mathbf{P} = (\lambda x.x\mathbf{S}_2)\mathbf{P} \to \mathbf{PS}_2 = (\lambda sx.x\mathbf{S}_1)\mathbf{S}_2 \to \lambda x.x\mathbf{S}_1 = \mathbf{B}_1, \\
\mathbf{BTT} &\to \mathbf{B}_2\mathbf{T} = (\lambda x.x\mathbf{S}_2)\mathbf{T} \to \mathbf{TS}_2 = (\lambda sx.x(s\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3))\mathbf{S}_2 \to \lambda x.x(\mathbf{S}_2\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3) \to \lambda x.x\mathbf{S}_3 = \mathbf{B}_3, \\
\mathbf{BTTP} &\to \mathbf{B}_3\mathbf{P} = (\lambda x.x\mathbf{S}_3)\mathbf{P} \to \mathbf{PS}_3 = (\lambda sx.x\mathbf{S}_1)\mathbf{S}_3 \to \lambda x.x\mathbf{S}_1 = \mathbf{B}_1, \\
\mathbf{BTTT} &\to \mathbf{B}_3\mathbf{T} = (\lambda x.x\mathbf{S}_3)\mathbf{T} \to \mathbf{TS}_3 = (\lambda sx.x(s\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3))\mathbf{S}_3 \to \lambda x.x(\mathbf{S}_3\mathbf{S}_2\mathbf{S}_3\mathbf{S}_3) \to \lambda x.x\mathbf{S}_3 = \mathbf{B}_3.
\end{aligned}$$

よって $\mathbf{B}, \mathbf{B}M_1, \mathbf{B}M_1M_2, \ldots, \mathbf{B}M_1M_2\cdots M_n$ の遷移は以下のようになる.（末尾の M_i に対応する矢印の先に移動する.）



$\mathbf{B}_i$ は β 正規形だから，$\mathbf{B}M_1M_2\cdots M_n$ の β 正規形は，

$$\begin{cases}
\mathbf{B}_1 & (n = 0 \text{ or } M_n = \mathbf{P}) \\
\mathbf{B}_2 & (n = 1, M_1 = \mathbf{T} \text{ or } n \geq 2, (M_{n-1}, M_n) = (\mathbf{P}, \mathbf{T})) \\
\mathbf{B}_3 & (n \geq 2, (M_{n-1}, M_n) = (\mathbf{T}, \mathbf{T}))
\end{cases}$$

□

平成 19 年度 (2006 年 8 月実施)

問 11

純粋な（型のない）ラムダ計算において，以下のような，有限列を表現するラムダ式を考える．

$$\begin{array}{lcl} [\,] & = & \lambda xy.x \\ [M_1, M_2, \ldots, M_n] & = & \lambda xy.yM_1(yM_2 \cdots (yM_n x)) \end{array}$$

ただし $M_1, M_2, \ldots, M_n$ は自由な x, y の出現がない任意のラムダ式とする．

(i) 以下の条件をみたす閉じたラムダ式 C をひとつ与えよ．

$$\begin{array}{lcl} CM[\,] & \to^* & [M] \\ CM[N_1, \ldots, N_n] & \to^* & [M, N_1, \ldots, N_n] \end{array}$$

ただし，$\to^*$ は β 簡約関係の推移的反射的閉包とする．

(ii) 以下の条件をみたす閉じたラムダ式 A をひとつ与えよ．

$$\begin{array}{lcl} A[\,][\,] & \to^* & [\,] \\ A[\,][N_1, \ldots, N_n] & \to^* & [N_1, \ldots, N_n] \\ A[M_1, \ldots, M_m][\,] & \to^* & [M_1, \ldots, M_m] \\ A[M_1, \ldots, M_m][N_1, \ldots, N_n] & \to^* & [M_1, \ldots, M_m, N_1, \ldots, N_n] \end{array}$$

解答． $\to^*$ を $\to$ と略記する．

(i) $C = \lambda abpq.qa(bpq)$ が条件を満たすことを示す．$X \in \{[\,], [N_1, \ldots, N_n]\}$ に対し

$$CMX = (\lambda abpq.qa(bpq))MX \to \lambda pq.qM(Xpq)$$

である．また，

$$[\,]pq = (\lambda xy.x)pq \to p, \quad [N_1, \ldots, N_n]pq = (\lambda xy.yN_1(yN_2 \cdots (yN_n x)))pq \to qN_1(qN_2 \cdots (qN_n p))$$

だから

$$\begin{aligned} CM[\,] &\to \lambda pq.qMp = [M], \\ CM[N_1, \ldots, N_n] &\to \lambda pq.qM(qN_1(qN_2 \cdots (qN_n p))) = [M, N_1, \ldots, N_n] \end{aligned}$$

となり条件を満たす．

(ii) $A = \lambda abpq.a(bpq)q$ が条件を満たすことを示す．$X, Y \in \{[\,], [M_1, \ldots, M_m], [N_1, \ldots, N_n]\}$ に対し

$$AXY = (\lambda abpq.a(bpq)q)XY \to \lambda pq.X(Ypq)q$$

である．また，(i) で示したことから

$$[\,]([\,]pq)q \to [\,]pq \to p, \quad [\,]([N_1, \ldots, N_n]qp)q \to [\,](qN_1(qN_2 \cdots (qN_n p)))q \to qN_1(qN_2 \cdots (qN_n p)),$$
$$[M_1, \ldots, M_m]([\,]pq)q \to [M_1, \ldots, M_m]pq \to qM_1(qM_2 \cdots (qM_m p)),$$
$$[M_1, \ldots, M_m]([N_1, \ldots, N_n]qp)q \to qM_1(qM_2 \cdots (qM_m(qN_1(qN_2 \cdots (qN_n p)))))$$

だから

$$\begin{aligned} A[\,][\,] &\to \lambda pq.p = [\,], \\ A[\,][N_1, \ldots, N_n] &\to \lambda pq.qN_1(qN_2 \cdots (qN_n p)) = [N_1, \ldots, N_n], \\ A[M_1, \ldots, M_m][\,] &\to \lambda pq.qM_1(qM_2 \cdots (qM_m p)) = [M_1, \ldots, M_m], \\ A[M_1, \ldots, M_m][N_1, \ldots, N_n] &\to \lambda pq.qM_1(qM_2 \cdots (qM_m(qN_1(qN_2 \cdots (qN_n p))))) \\ &= [M_1, \ldots, M_m, N_1, \ldots, N_n] \end{aligned}$$

となり条件を満たす． □

平成 18 年度 (2005 年 8 月実施)

問 11

以下のような，整数の 3 つ組を入力にとる再帰プログラム P を考える．

$$P(x,y,z) = \texttt{if}\ \ x \le y\ \ \texttt{then}\ \ y\ \ \texttt{else}\ \ P(P(x-1,y,z), P(y-1,z,x), P(z-1,x,y))$$

また，関数 $F : \mathbb{Z}^3 \to \mathbb{Z}$ を，以下のように与える．

$$F(x,y,z) = \begin{cases} y & (x \le y) \\ z & (y < x, y \le z) \\ x & (\text{その他の場合}) \end{cases}$$

(i) 任意の $x,y,z \in \mathbb{Z}$ について

$$F(x,y,z) = \begin{cases} y & (x \le y) \\ F(F(x-1,y,z), F(y-1,z,x), F(z-1,x,y)) & (y < x) \end{cases}$$

が成り立つことを示せ．

(ii) 関数 $M : \mathbb{Z}^3 \to \mathbb{N}^3$ を以下のように定める．

$$M(x,y,z) = (M_1(x,y,z), M_2(x,y,z), M_3(x,y,z))$$

ただし

$$M_1(x,y,z) = \begin{cases} 0 & (x \le y) \\ 1 & (y < x) \end{cases}$$
$$M_2(x,y,z) = \max\{x,y,z\} - \min\{x,y,z\}$$
$$M_3(x,y,z) = x - \min\{x,y,z\}$$

であり，max と min は最大値，最小値を意味する．任意の $x,y,z \in \mathbb{Z}$ について，$y < x$ ならば，$M(x-1,y,z), M(y-1,z,x), M(z-1,x,y)$ および $M(F(x-1,y,z), F(y-1,z,x), F(z-1,x,y))$ は，$\mathbb{N}^3$ 上の辞書式順序に関していずれも $M(x,y,z)$ より真に小さくなることを示せ．

(iii) 任意の $x,y,z \in \mathbb{Z}$ について，$P(x,y,z)$ の計算が停止してその結果が $F(x,y,z)$ と一致することを示せ．

解答． (i) $x > y$ の場合を示せば良い．$(n_1,n_2,n_3) = (F(x-1,y,z), F(y-1,z,x), F(z-1,x,y))$ とおく．$y \le z$ の時は

$$F(n_1,n_2,n_3) = \begin{cases} F(y,z,*) = z & (x-1 = y) \\ F(z,z,*) = z & (x-1 > y). \end{cases}$$

$y > z$ の時は

$$F(n_1,n_2,n_3) = \begin{cases} F(y,z,x) = x & (x-1 = y \wedge y-1 = z) \\ F(y,x,x) = x & (x-1 = y \wedge y-1 > z) \\ F(x-1,z,x) = x & (x-1 > y \wedge y-1 = z) \\ F(x-1,x,x) = x & (x-1 > y \wedge y-1 > z). \end{cases}$$

(ii) $\mathbb{N}^3$ の辞書式順序を $<_3$ と書く．

- $M(x-1,y,z)$: $x-1 \le y$ の時は $M_1(x-1,y,z)=0$ だから良い．$x-1>y$ の時は

$$\begin{aligned}M(x-1,y,z) &= (1,\max\{x-1,z\}-\min\{y,z\},(x-1)-\min\{y,z\}) \\ &<_3 (1,\max\{x,z\}-\min\{y,z\},x-\min\{y,z\}) = M(x,y,z).\end{aligned}$$

- $M(y-1,z,x)$: $y-1 \le z$ の時は $M_1(y-1,z,x)=0$ だから良い．$y-1>z$ の時は

$$M(y-1,z,x) = (1,x-z,(y-1)-z) <_3 (1,x-z,x-z) = M(x,y,z).$$

- $M(z-1,x,y)$: $z-1 \le x$ の時は $M_1(z-1,x,y)=0$ だから良い．$z-1>x$ の時は

$$M(z-1,x,y) = (1,z-y,(z-1)-y) <_3 (1,z-y,x-y) = M(x,y,z).$$

- $M(n_1,n_2,n_3)$: $y \le z$ の時は

$$\begin{aligned}M(n_1,n_2,n_3) &= \begin{cases} M(y,z,*) = (0,*,*) & (x-1=y) \\ M(z,z,*) = (0,*,*) & (x-1>y)\end{cases} \\ &<_3 (1,*,*) = M(x,y,z).\end{aligned}$$

$y>z$ の時は

$$\begin{aligned}M(n_1,n_2,n_3) &= \begin{cases} M(y,z,x) = (1,x-z,y-z) & (x-1=y \wedge y-1=z) \\ M(y,x,x) = (0,*,*) & (x-1=y \wedge y-1>z) \\ M(x-1,z,x) = (1,x-z,(x-1)-z) & (x-1>y \wedge y-1=z) \\ M(x-1,x,x) = (0,*,*) & (x-1>y \wedge y-1>z)\end{cases} \\ &<_3 (1,x-z,x-z) = M(x,y,z).\end{aligned}$$

(iii) $M(x,y,z) \in \mathbb{N}^3$ の辞書式順序 $<_3$ に関する帰納法で示す．$M(x,y,z)=(0,*,*)$，すなわち $x \le y$ の時は明らか．$x>y$ なる $(x,y,z) \in \mathbb{Z}^3$ を任意に取る．$M(x',y',z') <_3 M(x,y,z)$ なる任意の $(x',y',z') \in \mathbb{Z}^3$ に対して $P(x',y',z') = F(x',y',z')$ であるとする．この時 (i), (ii) と帰納法の仮定より

$$\begin{aligned}P(x,y,z) &= P(P(x-1,y,z),P(y-1,z,x),P(z-1,x,y)) \\ &= P(F(x-1,y,z),F(y-1,z,x),F(z-1,x,y)) \\ &= F(F(x-1,y,z),F(y-1,z,x),F(z-1,x,y)) \\ &= F(x,y,z)\end{aligned}$$

だから示された． □

(補足) $a \in \mathbb{Z}$ に対し $a_+ = \max\{a,0\}$ とおく時，$(x-y)_+ + (y-z)_+ + (z-x)_+$ に関する帰納法[3] や，「$\max\{x,y,z\} - \min\{x,y,z\}$」に関する帰納法[4] で (iii) を直接示すこともできるらしい．

[3] https://cybozushiki.cybozu.co.jp/articles/m000434.html の写真 3.

[4] 野崎，計算機数学（共立数学講座），共立出版，P221. 国会図書館のデジタルコレクションにあり．

平成 17 年度 (2004 年 8 月実施)

問 13

a は非負整数とし，以下のプログラムを考える．

$x := 0$;
$y := 1$;
$z := 1$;
while $z \leq a$ **do**
  $x := x + 1$;
  $y := y + 2$;
  $z := z + y$;
done
return x

このプログラムの実行は必ず停止することを示せ．また，出力される結果を与え，その理由を説明せよ．

解答． while ループを n 回回った後の x, y, z の値をそれぞれ x_n, y_n, z_n とすると

$$\begin{cases} x_{n+1} = x_n + 1 \\ y_{n+1} = y_n + 2 \\ z_{n+1} = z_n + y_{n+1} \end{cases} \qquad (x_0, y_0, z_0) = (0, 1, 1).$$

よって $x_n = n, y_n = 2n+1, z_{n+1} = z_n + 2n + 3$. $z_{n+1} - (n+2)^2 = z_n - (n+1)^2$ より $z_n = (n+1)^2$. z_n は単調増加で $n \to \infty$ の時 $z_n \to \infty$ だから，$z_n \leq a < z_{n+1}$ を満たす n が一意に存在する．従ってプログラムは必ず停止する．while ループを n 回回って停止したとすると，$z_{n-1} \leq a < z_n$ だから $n^2 \leq a < (n+1)^2$. よって $n = \lfloor \sqrt{a} \rfloor$ なので，出力結果は $x_n = \lfloor \sqrt{a} \rfloor$. □

問 14

0 と 1 とからなる文字列で，2 進数としてみたとき素数を表すもの全体の集合は正則言語でないこと，すなわちこの集合を受理する有限オートマトンが存在しないことを示せ．ただし，以下の結果 $(*)$ は証明なしに用いてもよい:

$(*)$ 2 より大きい素数 p について $2^{p-1} \equiv 1 \bmod p$.

解答. 問題の集合を L とする．L が正則言語であったとする．反復補題による定数を N とする．素数は無限に存在するから， $p \in L$ であって $|p| \geq 2N$ となるものが存在する．この時反復補題による p の分割を $p = xyz$ とすると $q := xy^p z \in L$ であり，$p \geq 2$ より $q > p$ である．$y \neq \varepsilon$ より $2^{|y|} - 1 > 0$. また，$|p| \geq 2N, |y| \leq |xy| \leq N$ より

$$p - (2^{|y|} - 1) \geq 2^{2N-1} - (2^N - 1) = 2^N(2^{N-1} - 1) + 1 > 0.$$

よって $0 < 2^{|y|} - 1 < p$ である．以下 $\bmod p$ とする．$2^{|y|} - 1 \not\equiv 0$ であることに注意すると

$$\begin{aligned}
q &= 2^{p|y|+|z|}x + 2^{(p-1)|y|+|z|}y + \cdots + 2^{|z|}y + z \\
&= 2^{p|y|+|z|}x + \frac{2^{p|y|} - 1}{2^{|y|} - 1}2^{|z|}y + z \\
&\equiv 2^{|y|+|z|}x + \frac{2^{|y|} - 1}{2^{|y|} - 1}2^{|z|}y + z \\
&= 2^{|y|+|z|}x + 2^{|z|}y + z \\
&= p \equiv 0.
\end{aligned}$$

よって q は素数 p の倍数だが，$q > p$ より q は合成数．これは $q \in L$ に矛盾． $\square$

平成 16 年度 (2003 年 8 月実施)

問 12

有限個の文字からなる集合を一つ固定し，これらの文字からなる有限の長さの文字列について考える．長さ 0 の文字列を ε であらわす．また，文字 a と長さ n の文字列 $u = b_1 b_2 \dots b_n$ に対して，これらを並べて得られる長さ $n+1$ の文字列 $ab_1 b_2 \dots b_n$ を $a \cdot u$ とあらわす（とくに，$a \cdot \varepsilon = a$ である）．このとき，以下のような文字列上の操作を考える．

$$
\begin{aligned}
\mathsf{rev\_append}(u, v) &= \begin{cases} v & u = \varepsilon \text{ のとき} \\ \mathsf{rev\_append}(u', a \cdot v) & u = a \cdot u' \text{ のとき} \end{cases} \\
\mathsf{reverse}(u) &= \mathsf{rev\_append}(u, \varepsilon) \\
\mathsf{mirror}(u) &= \mathsf{rev\_append}(u, u) \\
\mathsf{rev\_mirror}(u) &= \mathsf{reverse}(\mathsf{mirror}(u))
\end{aligned}
$$

任意の文字列 u に対して，$\mathsf{rev\_mirror}(u)$ の結果と $\mathsf{mirror}(u)$ の結果が一致することを示せ．

解答． 文字列 $u = u_1 \dots u_n, v = v_1 \dots v_m$ に対し $u_1 \dots u_n v_1 \dots v_m$ を uv と書く．$u = u_1 \dots u_n$ とした時 $\mathsf{rev\_append}(u, v) = u_n \dots u_1 v$ となることを n についての帰納法で示す．$n = 0$ の時は定義から良い．n で正しい時,

$$\mathsf{rev\_append}(u_1 \dots u_{n+1}, v) = \mathsf{rev\_append}(u_2 \dots u_{n+1}, u_1 v) = u_{n+1} \dots u_2 \cdot u_1 v$$

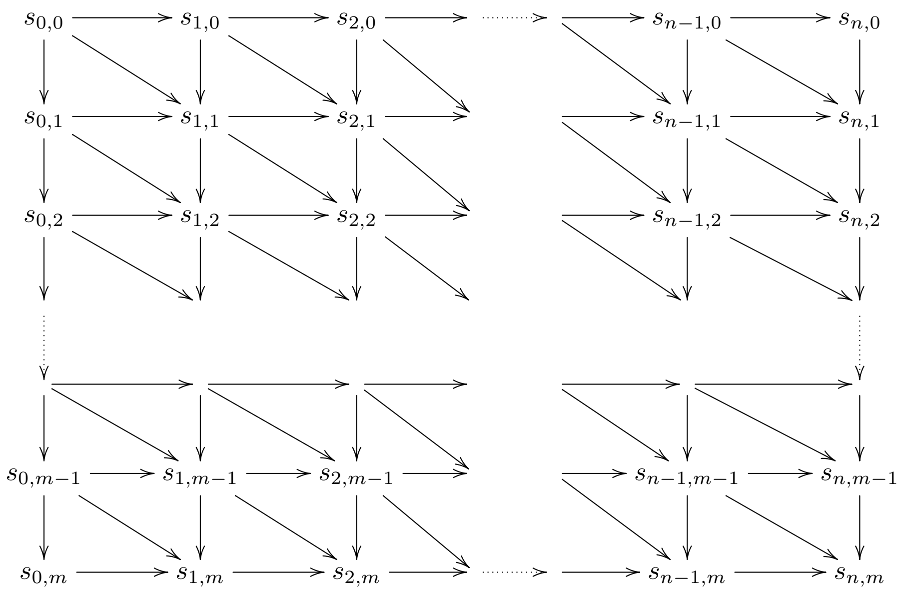
だから $n+1$ でも正しい．これで示せた．よって任意の文字列 $u = u_1 \dots u_n$ に対し

$$
\begin{aligned}
&\mathsf{reverse}(u) = u_n \dots u_1 \cdot \varepsilon = u_n \dots u_1 \\
&\mathsf{mirror}(u) = u_n \dots u_1 \cdot u_1 \dots u_n \\
&\mathsf{rev\_mirror}(u) = \mathsf{reverse}(u_n \dots u_1 \cdot u_1 \dots u_n) = u_n \dots u_1 u_1 \dots u_n
\end{aligned}
$$

だから $\mathsf{rev\_mirror}(u) = \mathsf{mirror}(u)$. □

問 13

自然数 $a_0, \ldots, a_n$ および $b_0, \ldots, b_m$ $(n, m \geq 1)$ が与えられているとする．下図のような $(n{+}1){\times}(m{+}1)$ 個の節点 $\{s_{i,j} \,|\, 0 \leq i \leq n, 0 \leq j \leq m\}$ からなる有向グラフを考える．



グラフの各辺 (s, t) に対し，重み $w(s, t)$ を以下のように定める．

$$
\begin{aligned}
w(s_{i,j}, s_{i+1,j}) &= 1 \\
w(s_{i,j}, s_{i,j+1}) &= 1 \\
w(s_{i,j}, s_{i+1,j+1}) &= \begin{cases} 0 & (a_i = b_j) \\ 1 & (a_i \neq b_j) \end{cases}
\end{aligned}
$$

このとき，$s_{0,0}$ から $s_{n,m}$ に至る最短距離（$s_{0,0}$ から $s_{n,m}$ に至る経路における辺の重みの総和のうち最小のもの）を求めるアルゴリズムで，計算量が $O(n \times m)$ であるものを与えよ．なお，アルゴリズムを記述する言語は任意とするが，そのアルゴリズムの正しさと，計算量が $O(n \times m)$ である理由を，簡潔に説明すること．

解答． $d[i][j]$ を $s_{0,0}$ から $s_{i,j}$ への最短距離とする．明らかに $d[i][0] = i, d[0][j] = j$ である．また，$s_{i+1,j+1}$ に到達するには $s_{i+1,j}, s_{i,j+1}, s_{i,j}$ のいずれかを経由する必要がある．$s_{i+1,j}$ を経由する時は $d[i+1][j+1] = d[i+1][j] + 1$, $s_{i,j+1}$ を経由する時は $d[i+1][j+1] = d[i][j+1] + 1$, $s_{i,j}$ を経由する時は $a_i = b_j$ なら $d[i+1][j+1] = d[i][j]$, そうでないなら $d[i+1][j+1] = d[i][j] + 1$. よって

$$
d[i+1][j+1] = \begin{cases} \min\{d[i+1][j]+1, d[i][j+1]+1, d[i][j]\} & (a_i = b_j) \\ \min\{d[i+1][j]+1, d[i][j+1]+1, d[i][j]+1\} & (a_i \neq b_j) \end{cases}.
$$

従ってアルゴリズムは以下のようになる．

procedure DISTANCE
  for $i = 0, \ldots, n$
    $d[i][0] = i$
  for $j = 0, \ldots, m$
    $d[0][j] = j$
  for $i = 0, \ldots, n$
    for $j = 0, \ldots, m$

$tmp = \min\{d[i+1][j]+1, d[i][j+1]+1\}$
if $a_i = b_j$ **then**
$d[i+1][j+1] = \min\{tmp, d[i][j]\}$
else
$d[i+1][j+1] = \min\{tmp, d[i][j]+1\}$
return $d[n][m]$

計算量は，最初とその次の for ループでそれぞれ $O(n), O(m)$, 最後の二重ループで $O(nm)$ だから，全体で $O(nm)$. $\square$

平成 15 年度 (2002 年 8 月実施)

問 9

次に示すプログラムについて，以下の問に答えよ．

```
int a[1..N]    /* 整数の配列 a[1], a[2], .., a[N] */
function swap(i) =
    if a[i] ≤ a[i+1]
      then return(false)
      else tmp ← a[i]; a[i] ← a[i+1]; a[i+1] ← tmp; return(true)
    endif
procedure bubblesort() =
    begin
      changed ← true;
      while changed do
        changed ← false;
        for i=1 to N-1 do
          if swap(i) then changed ← true endif
        done
      done
    end
```

(i) 手続き *bubblesort* が配列 a を昇順に整列することを示せ．

(ii) *bubblesort* が *swap* を呼び出す回数は最悪の場合何回か．

(iii) *bubblesort* が *swap* を呼び出す最悪の回数を，*bubblesort* を改良することで半分に減らすことができる．このような改良された *bubblesort* を書け．

解答． (i) $a[1], \dots, a[N]$ を昇順にソートしたものを $b[1], \dots, b[N]$ とする．while ループを j 回回った後 $a[i] = b[i]\,(i = N+1-j, \dots, N)$ となることを帰納法で示す．$j = 1$ の時，$a[k] = b[N]$ とする．この時 $a[k] \geq a[i]\,(i = k+1, \dots, N)$ だから，for ループの $i = k+1, \dots, N$ の後 $a[N] = b[N]$ となり正しい．j で正しいとする．$a[k] = b[N-j]$ とする．帰納法の仮定から $a[i] = b[i]\,(i = N+1-j, \dots, N)$ だから，$a[k] \leq a[i]\,(i = N+1-j, \dots, N)$ である．よって for ループの $i = N-j$ の後 $a[N-j] = b[N-j]$ となり正しい．これで示された．従って while ループを N 回回ると $a[i] = b[i]\,(i = 1, \dots, N)$ となり昇順にソートされる．再び while ループに入ると swap の戻り値が false となり while ループから抜けるので，bubblesort は停止する．

(ii) while ループを 1 回回ると swap は $N-1$ 回呼び出される．(i) で示したように，while ループは高々 N 回しか回らないから，swap の呼び出し回数は高々 $N(N-1)$. ここで $a[i] = N+1-i\,(i = 1, \dots, N)$ の時，while ループを $N-1$ 回回ると $(a[1], \dots, a[N])$ は

$$
\begin{aligned}
(N, N-1, \dots, 2, 1) &\to (N-1, N-2, \dots, 1, N) \to (N-2, N-3, \dots, 1, N-1, N) \\
&\to \cdots \to (2, 1, 3, 4, \dots, N-1, N) \to (1, 2, \dots, N)
\end{aligned}
$$

と変化する．この時最後の for ループで $a[1]$ と $a[2]$ の交換が実行されたから changed は true である．従ってもう一度 while ループを回るが，既に $a[i]$ はソートされているから，for ループを抜けた後 changed は false となり while ループからも抜ける．よって while ループは N 回回るので，swap 呼び出しの最悪回数は $N(N-1)$.

(iii) (i) で示したように，while ループを j 回回ると $a[i] = b[i]\,(i = N+1-j, \dots, N)$ となるから，for ループは $i = N-1-j$ までで良い．while ループを N 回回ったあと for ループは実

行されず，while ループを抜けるから bubblesort は停止する．この時 swap 呼び出しの最悪回数は $(N-1)+(N-2)+\cdots+1=\frac{N(N-1)}{2}$. 従って改良された bubblesort は以下の通り．

```
bubblesort(){
    j = 0;  // while ループのカウンタ
    changed = true;
    while(changed){
        changed = false;
        for(i = 1, i < N-j; i++){
            if(swap(i)) changed = true;
        }
        j++;
    }
}
```

$\square$

問 10

D を ω 完備半順序集合（ω CPO）とし，$\mathrm{fix} : [D \to D] \to D$ を D 上の最小不動点演算子（すなわち，連続関数 $f : D \to D$ について $\mathrm{fix}(f) = \bigsqcup_{n\geq 0} f^n(\bot)$ ）とする．
連続関数 $F : D \times D \to D$ が与えられたとき，連続関数 $G : D \to [D \to D]$ および $H : D \to D$ を

$$\begin{array}{lcl} G(x)(y) & = & F(x,y) \\ H(x) & = & F(x,x) \end{array}$$

と定めると，

$$\mathrm{fix}(\mathrm{fix} \circ G) = \mathrm{fix}(H)$$

が成り立つことを示せ．

（注）D が ω 完備半順序集合であるとは，D が半順序集合で，かつ，以下の二つの条件を満たすことをいう．

(a) D は最小元 $\bot$ をもつ．

(b) D の可算単調列 $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$ についてその最小上界 $\bigsqcup_{n\geq 0} d_n$ が存在する．

また，ω 完備半順序集合 D, E について，関数 $f : D \to E$ が連続であるとは，f が単調（すなわち，$d \sqsubseteq d'$ ならば $f(d) \sqsubseteq f(d')$）であり，かつ，D の任意の可算単調列 $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$ について

$$f\Big(\bigsqcup_{n\geq 0} d_n\Big) = \bigsqcup_{n\geq 0} f(d_n)$$

が成り立つことをいう．

解答． $G_1 = \mathrm{fix} \circ G, \theta = \mathrm{fix}(H), \theta' = \mathrm{fix}(G_1)$ とおく．

$$\begin{aligned} G_1(x) &= \bigsqcup_{n\geq 0} G(x)^n(\bot) = \bot \sqcup \bigsqcup_{n\geq 0} G(x)^{n+1}(\bot) = \bigsqcup_{n\geq 0} F(x, G(x)^n(\bot)) \\ &= F\Big(\bigsqcup_{n\geq 0}(x, G(x)^n(\bot))\Big) = F\Big(x, \bigsqcup_{n\geq 0} G(x)^n(\bot)\Big) = F(x, G_1(x)) \end{aligned}$$

であるから $\theta' = G_1(\theta') = F(\theta', G_1(\theta')) = F(\theta', \theta') = H(\theta')$. 従って θ' は H の不動点なので $\theta \sqsubseteq \theta'$.
ここで $G(\theta)^n(\bot) \sqsubseteq \theta\,(n \geq 0)$ を帰納法で示す．$n = 0$ の時は自明．$n-1$ で正しい時 $G(\theta)^n(\bot) = F(\theta, G(\theta)^{n-1}(\bot)) \sqsubseteq F(\theta, \theta) = \theta$ だから n でも正しい．これで示せた．よって

$$G_1(\theta) = \bigsqcup_{n\geq 0} G(\theta)^n(\bot) \sqsubseteq \bigsqcup_{n\geq 0} \theta = \theta.$$

これと G_1 の連続性から $G_1(\bot) \sqsubseteq G_1(\theta) \sqsubseteq \theta$. また，$G_1^{n-1}(\bot) \sqsubseteq \theta$ が成り立つ時 $G_1^n(\bot) \sqsubseteq G_1(\theta) \sqsubseteq \theta$ だから，帰納的に $G_1^n(\bot) \sqsubseteq \theta\,(n \geq 0)$ が成り立つ．従って

$$\theta' = \bigsqcup_{n\geq 0} G_1^n(\bot) \sqsubseteq \bigsqcup_{n\geq 0} \theta = \theta.$$

以上から $\theta = \theta'$ となる． $\square$

平成 14 年度 (2001 年 8 月実施)

問 10

g を，2 つの正の整数 x, y に対し，その最大公約数 $g(x, y)$ を与える関数とする．

(i) x, y と $g(x, y)$ の関係を一階述語の論理式で表せ．ただし，関数記号としては，g に加えて，加法，乗法 $(+, *)$, また述語記号としては等号と不等号 $(=, <, \leq)$ のみを用いよ．

(ii) g を計算するような再帰的プログラムを書き，その正しさを示せ．ただし，用いてよい基本演算は，整数の間の比較 $(\leq, <, =)$, 加法，乗法および減法 $(+, *, -)$ のみとする．(特定のプログラミング言語の細かい文法にこだわる必要はない．)

解答． (i) $g = g(x, y)$ は x, y の公約数であり，かつ x, y の任意の公約数は g の倍数であるから，

$$(\exists n, m.(x = gn \wedge y = gm)) \wedge ((\exists n, m, d.(x = dn \wedge y = dm)) \to (\exists k.d = kg))$$

(ii) $x = y$ の時は $g(x, y) = x$, $x < y$ の時は $g(x, y) = g(y, x)$ である．以下 $x > y$ とする．$d = g(x, y)$ とおくと，$g(x_1, y_1) = 1, x_1 > y_1$ なる正整数 x_1, y_1 が存在して $x = dx_1, y = dy_1$ と書ける．この時 $g(x - y, y) = g(d(x_1 - y_1), dy_1) = dg(x_1 - y_1, y_1)$ である．$d_1 = g(x_1 - y_1, y_1)$ とおくと，$g(x_2, y_2) = 1$ なる正整数 x_2, y_2 が存在して $x_1 - y_1 = d_1x_2, y_1 = d_1y_2$ と書ける．よって $x_1 = y_1 + d_1x_2 = d_1(x_2 + y_2)$ だから，

$$1 = g(x_1, y_1) = g(d_1(x_2 + y_2), d_1y_2) = d_1g(x_2 + y_2, y_2) \geq d_1.$$

従って $d_1 = 1$ なので $g(x - y, y) = d$. 以上からプログラムは以下の通り．

```
g(x, y){
    if(x = y) return x;
    else if(x < y) return g(y, x);
    else return g(x - y, y);
}
```

□

問 11

型なしのラムダ式に関する以下の問に答えよ．ただし，$\to^*$ は β 変換の反射的推移的閉包を表すものとする．また，以下，$T = \lambda xy.x, F = \lambda xy.y$ とする．

(i) 次の条件を満たす閉じたラムダ式 M を与えよ：$P = \lambda v.vMT, Q = \lambda v.vMF$ としたとき，

$$\begin{array}{llll} PP \to^* P & \text{かつ} & PQ \to^* Q & \text{かつ} \\ QP \to^* Q & \text{かつ} & QQ \to^* Q & \end{array}$$

(ii) 次の条件を満たす閉じたラムダ式 L を与えよ：

$$\lambda zw.b_1 \cdots b_k \qquad (k \geq 1, \text{ 各 } b_i \text{ は } z \text{ か } w \text{ のいずれか})$$

の形をした任意のラムダ式 B について

$$LB \to^* \begin{cases} T & (b_1, \ldots, b_k \text{ が全て } z) \\ F & (\text{それ以外}) \end{cases}$$

解答. $\to^*$ を $\to$ と略記する．

(i) $M = \lambda abcd.da(bcb)$ が条件を満たすことを示す．$X_1, X_2 \in \{T, F\}$ とすると M, X_1, X_2 は自由変数を含まないから

$$\begin{aligned}(\lambda v.vMX_1)(\lambda v.vMX_2) &\to (\lambda v.vMX_2)MX_1 \to MMX_2X_1 \\ &= (\lambda abcd.da(bcb))MX_2X_1 \to \lambda d.dM(X_2X_1X_2).\end{aligned}$$

これと

$$\begin{aligned} TTT &= (\lambda xy.x)TT \to T, \qquad TFT = (\lambda xy.x)FT \to F, \\ FTF &= (\lambda xy.y)TF \to F, \qquad FFF = (\lambda xy.y)FF \to F \end{aligned}$$

より $PP \to P, PQ \to Q, QP \to Q, QQ \to Q$ が成り立つ．

(ii) $L = \lambda x.xPQF$ が条件を満たすことを示す．P, Q, F は自由変数を含まないから L も自由変数を含まない．また，P, Q からなるラムダ式 $R_1R_2 \cdots R_k\,(R_i \in \{P, Q\})$ は，(i) より $R_1, \ldots, R_k$ が全て P の時 $\to P$, それ以外の時 $\to Q$ だから

$$LB = (\lambda x.xPQF)B \to BPQF = (\lambda zw.b_1 \cdots b_k)PQF \to \begin{cases} PF & (b_1, \ldots, b_k \text{ が全て } z) \\ QF & (\text{それ以外}). \end{cases}$$

また $X \in \{T, F\}$ に対し

$$(\lambda v.vMX)F \to FMX = (\lambda xy.y)MX \to X$$

だから $PF \to T, QF \to F$ となり条件を満たす． □

平成 13 年度 (2000 年 8 月実施)

問 10

可算無限個の定数記号 $c_i\,(i \in \mathbb{N})$, 1 変数の関数記号 s および 2 変数の述語記号 P をもち，非論理公理 (nonlogical axiom) として

$$\begin{aligned}
&\neg\exists x.P(x,x)\\
&\forall x.\forall y.\forall z.P(x,y) \wedge P(y,z) \to P(x,z)\\
&\forall y.\exists x.P(x,y)\\
&\forall x.P(x,s(x))\\
&\forall x.P(x,c_i) \to P(s(x),c_i) \quad (i \in \mathbb{N})
\end{aligned}$$

をもつ一階述語論理の理論を T とする．このとき，T は無矛盾 (consistent) であることを証明せよ．

解答. T がモデルを持つことを示せば良い．T の公理の論理式を上から順に $A_1, A_2, A_3, A_4, A_{5,i}$ とする．$L = \{c_0, c_1, \dots, s, P\}$ を言語とする．構造の領域を $\mathbb{Q}_{>0}$ とする L 構造を，解釈

$$P(x,y) = (x < y), \qquad c_i = i+1\,(i=0,1,\dots), \qquad s\left(n+\frac{p}{q}\right) = n + \frac{p+1}{q+1}$$

により定める．ここで n, p, q は $0 \le p < q, (p,q) = 1$ を満たす非負整数である.（ただし $x \in \mathbb{N}$ の時は $p = 0, q = 1$ とみなす.）これが T のモデルとなることを示す．A_1, A_2 は自明．任意の $y \in \mathbb{Q}_{>0}$ に対し $x = \frac{y}{2} \in \mathbb{Q}_{>0}$ は $x < y$ を満たすから A_3 も良い．任意の $x = n + \frac{p}{q}$ に対し

$$s(x) - x = \frac{p+1}{q+1} - \frac{p}{q} = \frac{q-p}{q(q+1)} > 0$$

だから A_4 も良い．さらにこの x が $x < c_i = i+1$ を満たす時，$n \le i$ だから

$$c_i - s(x) = i + 1 - \left(n + \frac{p+1}{q+1}\right) = i - n + \frac{q-p}{q+1} > 0$$

となり $A_{5,i}$ も良い．これで示された． □

問 11

基底型 b を持つ型付きラムダ計算の型と項を次のとおり定義する．

$$\begin{array}{lccl} \text{型} & \tau & ::= & b \mid \tau \to \tau \\ \text{項} & M & ::= & x \mid \lambda x:\tau.M \mid (MM) \end{array}$$

型環境 Γ は変数の有限集合から型の集合への関数とし，型判定を $\Gamma \rhd M : \tau$ と書く．その型判定の導出規則は下記の 3 つである．

$$\begin{array}{ll} \text{(var)} & \Gamma \rhd x : \tau \qquad \text{if } \Gamma(x) = \tau \\ \text{(abs)} & \dfrac{\Gamma\{x:\tau_1\} \rhd M : \tau_2}{\Gamma \rhd \lambda x:\tau_1.M : \tau_1 \to \tau_2} \\ \text{(app)} & \dfrac{\Gamma \rhd M_1 : \tau_2 \to \tau_1 \qquad \Gamma \rhd M_2 : \tau_2}{\Gamma \rhd (M_1M_2) : \tau_1} \end{array}$$

ただし，記法 $\Gamma\{x:\tau\}$ は，$\Gamma\{x:\tau\}(x) = \tau$ かつ，$x \neq y$ なる y に対して $\Gamma\{x:\tau\}(y) = \Gamma(y)$ となる型環境を表す．

(i) 任意の項 M に対して，空の型環境 $\emptyset$ のもとで型判定 $\emptyset \rhd M : \tau$ が導出できるとき，M は自由変数を含まないことを示せ．

(ii) 任意の項 M に対して，型判定 $\emptyset \rhd M : \tau$ が導出できるとき，τ は唯一に定まることを示せ．すなわち $\emptyset \rhd M : \tau$ と $\emptyset \rhd M : \tau'$ が導出できれば，必ず $\tau = \tau'$．

解答. (i) 任意の型環境 Γ に対し，$\Gamma \rhd M : \tau$ が導出できるなら $\mathrm{FV}(M) \subset \Gamma$ となることを，導出に関する帰納法で示す．

• $\Gamma \rhd x : \tau$ の時，(var) より $x : \tau \in \Gamma$ だから $\mathrm{FV}(x) = x \in \Gamma$ となり正しい．

• $\Gamma \rhd \lambda x : \tau_1.M : \tau_1 \to \tau_2$ の時，導出

$$\frac{\Gamma\{x:\tau_1\} \rhd M : \tau_2}{\Gamma \rhd \lambda x:\tau_1.M : \tau_1 \to \tau_2}$$

が存在する．帰納法の仮定から $\mathrm{FV}(M) \subset \Gamma \cup \{x:\tau_1\}$ だから $\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) \setminus \{x\} \subset \Gamma$ となり正しい．

• $\Gamma \rhd (M_1M_2) : \tau_1$ の時，導出

$$\frac{\Gamma \rhd M_1 : \tau_2 \to \tau_1 \qquad \Gamma \rhd M_2 : \tau_2}{\Gamma \rhd (M_1M_2) : \tau_1}$$

が存在する．帰納法の仮定から $\mathrm{FV}(M_1), \mathrm{FV}(M_2) \subset \Gamma$ だから $\mathrm{FV}(M_1M_2) = \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) \subset \Gamma$ となり正しい．

これで示された．特に $\emptyset \rhd M : \tau$ が導出できる時，$\mathrm{FV}(M) \subset \emptyset$ だから M は自由変数を含まない．

(ii) 任意の型環境 Γ に対し，$\Gamma \rhd M : \tau$ が導出できるなら τ は一意に定まることを示せば良い．これを導出に関する帰納法で示す．

• $\Gamma \rhd x : \tau_i \, (i = 1, 2)$ の時，$\tau_1 = \Gamma(x) = \tau_2$ だから良い．

• $\Gamma \rhd \lambda x : \sigma.M : \sigma \to \tau_i \, (i = 1, 2)$ の時，導出

$$\frac{\Gamma\{x:\sigma\} \rhd M : \tau_i}{\Gamma \rhd \lambda x:\sigma.M : \sigma \to \tau_i}$$

が存在する．帰納法の仮定から M の型は一意なので，$\tau_1 = \tau_2$. よって $\sigma \to \tau_1 = \sigma \to \tau_2$ となり正しい．

• $\Gamma \rhd (M_1M_2) : \tau_i \, (i = 1, 2)$ の時，型 $\sigma_i \, (i = 1, 2)$ と導出

$$\frac{\Gamma \rhd M_1 : \sigma_i \to \tau_i \qquad \Gamma \rhd M_2 : \sigma_i}{\Gamma \rhd (M_1M_2) : \tau_i}$$

が存在する．帰納法の仮定から M_1, M_2 の型は一意なので，$\sigma_1 \to \tau_1 = \sigma_2 \to \tau_2, \sigma_1 = \sigma_2$. よって $\tau_1 = \tau_2$ となり正しい．

これで示された． □

平成 12 年度 (1999 年 8 月実施)

問 10

二分木とは，以下の (a), (b) により構成されるものをいう．

(a) 空の木 ϕ.

(b) 二分木 T_1, T_2 を合わせて得られる木．

空でない二分木に対して，T_1, T_2 をそれぞれ T の左部分木，右部分木と呼び，$T_1 = L(T), L_2 = R(T)$ と書く．

二分木の部分木は，それ自身，左部分木，または右部分木の部分木として帰納的に定義される．

また，二分木 T について，その大きさ $|T|$ および高さ $h(T)$ を以下のように定める．

$$|T| = \begin{cases} 0 & T \text{ が空のとき}, \\ 1 + |L(T)| + |R(T)| & T \text{ が空でないとき}, \end{cases}$$

$$h(T) = \begin{cases} 0 & T \text{ が空のとき}, \\ 1 + \max(h(L(T)), h(R(T))) & T \text{ が空でないとき}. \end{cases}$$

二分木は，任意の空でない部分木 T について

$$0 \leq |L(T)| - |R(T)| \leq 1$$

が成り立つとき，Braun 木と呼ばれる．

例えば，T は Braun 木であり，$|T| = 4, h(T) = 3$ である．

(i) 空でない Braun 木 T に対して $h(T) = \lfloor \log_2 |T| \rfloor + 1$ であることを示せ．ただし，$\lfloor \ \rfloor$ は整数部分を表すものとする．

(ii) Braun 木に対して次のように関数 Δ を定める：

$$\Delta(T) = \begin{cases} 0 & T \text{ が空のとき}, \\ |L(T)| - |R(T)| & T \text{ が空でないとき}. \end{cases}$$

このとき，$|T| \geq 2$ を満たす任意の Braun 木 T に対して

$$\Delta(T) = \begin{cases} \Delta(L(T)) & |R(T)| \text{ が奇数のとき}, \\ 1 - \Delta(L(T)) & |R(T)| \text{ が偶数のとき}, \end{cases}$$

が成り立つことを示せ．

(iii) Braun 木に関する次のような再帰的アルゴリズムを考える：

$$\begin{array}{rl} \texttt{size}(T) = & T \text{ が空のとき} \quad 0 \\ & T \text{ が空でないとき} \quad m = \texttt{size}(R(T)) \text{ として} \\ & \qquad\qquad 1 + 2m + \texttt{diff}(L(T), m) \end{array}$$

$$\begin{array}{rl} \texttt{diff}(T, m) = & T \text{ が空のとき} \quad 0 \\ & T \text{ が空でないとき} \\ & \qquad m = 0 \text{ のとき} \quad 1 \\ & \qquad m = 2k + 1\,(k \geq 0) \text{ のとき} \quad \texttt{diff}(L(T), k) \\ & \qquad m = 2k + 2\,(k \geq 0) \text{ のとき} \quad \texttt{diff}(R(T), k) \end{array}$$

このとき，$\texttt{size}(T)$ が $|T|$ を計算することを証明せよ．また，その計算量が高々 $O((\log_2 |T|)^2)$ であることを示せ．

解答．(i) $|T| < 2^k$ なる任意の Braun 木について正しいことを k に関する帰納法で示す．$k = 1$ の時は $|T| = 1$ だから $h(T) = 1$ となり正しい．k で正しいとして，$2^k \le |T| < 2^{k+1}$ なる Braun 木を任意に取る．

• $|L(T)| - |R(T)| = 0$ の時：$2^k \le 1+2|L(T)| < 2^{k+1}$ より $2^{k-1} \le |L(T)| < 2^k$. よって帰納法の仮定より $h(L(T)) = \lfloor \log_2 |L(T)| \rfloor + 1 = (k-1)+1 = k$. 同様に $h(R(T)) = k$ なので $h(T) = 1+\max(k, k) = k+1$ となり良い．

• $|L(T)| - |R(T)| = 1$ の時：$2^k \le 1 + |L(T)| + (|L(T)| - 1) < 2^{k+1}$ より $2^{k-1} \le |L(T)| < 2^k$. よって帰納法の仮定から $h(L(T)) = \lfloor \log_2 |L(T)| \rfloor + 1 = (k-1) + 1 = k$. また $2^{k-1} - 1 \le |R(T)| < 2^k - 1$ だから，$h(R(T)) = \lfloor \log_2 |R(T)| \rfloor + 1 \le (k-1) + 1 = k$. （$|R(T)| = 1$ の時も $h(R(T)) = 0$ だから正しい．）従って $h(T) = 1 + \max(k, h(R(T))) = k+1$ となり良い．

これで示された．

(ii) Braun 木 T が $|T| \ge 2$ を満たすとする．もし $L(T)$ が空なら $|R(T)| \ge 1$ となり $|L(T)| - |R(T)| \in \{0, 1\}$ に反する．よって $L(T)$ は空でない．従って mod2 において

$$\begin{aligned}\Delta(T) &= |L(T)| - |R(T)| = 1 + |L(L(T))| + |R(L(T))| - |R(T)| \\ &\equiv 1 + |L(L(T))| - |R(L(T))| - |R(T)| = 1 + \Delta(L(T)) - |R(T)| \\ &\equiv \begin{cases} \Delta(L(T)) & (|R(T)| \text{ が奇数}) \\ 1 - \Delta(L(T)) & (|R(T)| \text{ が偶数}) \end{cases}\end{aligned}$$

である．ところが，この左辺と右辺は 0 または 1 なので，（$\equiv$ ではなく $=$ の）等号が成立する．

(iii) $h(T)$ に関する帰納法で $\texttt{size}(T) = |T|$ を示す．$h(T) = 0$ の時，T は空だから良い．$n \ge 1$ として，$h(T) < n$ なる任意の Braun 木で正しいとする．$h(T) = n$ なる任意の Braun 木 T を取る．$T \neq \emptyset$ である．また $h(R(T)) < h(T) = n$ だから，帰納法の仮定より

$$\texttt{size}(T) = 1 + 2\texttt{size}(R(T)) + \texttt{diff}(L(T), \texttt{size}(R(T))) = 1 + 2|R(T)| + \texttt{diff}(L(T), |R(T)|).$$

よって $\Delta(T) = \texttt{diff}(L(T), |R(T)|)$ を示せば良い．帰納法の仮定から，$h(T) < n$ なる T に対してはこれは成立している．すなわち，$h(T') < n-1$ なる任意の Braun 木 T' と任意の $m \ge 0$ に対し，$|T'| = m$ or $m+1$ なら $\texttt{diff}(T', m) = |T'| - m$ である．

• $|R(T)| = 0$ の時：$|L(T)| = 0$ なら $\texttt{diff}(L(T), |R(T)|) = 0 = \Delta(T)$ だから良い．$|L(T)| = 1$ なら $\texttt{diff}(L(T), |R(T)|) = 1 = \Delta(T)$ だから良い．

• $|R(T)| = 2k+1\,(k \ge 0)$ の時：$\Delta(T) = 0$ なら $|L(T)| = 2k+1$. また (ii) より $\Delta(L(T)) = 0$ だから $|R(L(T))| = k$. $\Delta(T) = 1$ なら $|L(T)| = 2k+2$. また (ii) より $\Delta(L(T)) = 1$ だから $|R(L(T))| = k$. いずれにしても $|R(L(T))| = k$ である．これと $h(L(T)) < h(T) = n$ より

$$\Delta(T) = \Delta(L(T)) = \texttt{diff}(L(L(T)), |R(L(T))|) = \texttt{diff}(L(L(T)), k) = \texttt{diff}(L(T), |R(T)|).$$

• $|R(T)| = 2k+2\,(k \ge 0)$ の時：$\texttt{diff}(L(T), |R(T)|) = \texttt{diff}(R(L(T)), k)$ である．$\Delta(T) = 0$ の場合 $|L(T)| = 2k+2$. また (ii) より $\Delta(L(T)) = 1$ だから $|R(L(T))| = k$. よって $R(L(T))$ に帰納法の仮定が使えて $\texttt{diff}(R(L(T)), k) = 0$. $\Delta(T) = 1$ の場合 $|L(T)| = 2k+3$. また (ii) より $\Delta(L(T)) = 0$ だから $|R(L(T))| = k+1$. よって $R(L(T))$ に帰納法の仮定が使えて $\texttt{diff}(R(L(T)), k) = 0$.

以上から $h(T) = n$ の時も正しいので示された．

m が与えられた時，$\texttt{diff}(T, m)$ は，自分自身を呼び出す毎に $h(T)$ が 1 減るから，計算量は $O(h(T))$. よって $\texttt{size}(R(T)) = |R(T)|$ が計算出来た時，$\texttt{diff}(L(T), |R(T)|)$ は計算量 $O(h(L(T))) = O(h(T))$ で計算できる．従って $h(T) = n$ なる任意の Braun 木に対する $\texttt{size}(T)$ の最悪計算量を $f(n)$ とすると，$f(n) = f(n-1) + O(n)$ であるから $f(n) = O(n^2)$. ゆえに $\texttt{size}(T)$ の計算量は $O(h(T)^2) = O((\log_2 |T|)^2)$. □

問 11

定数を含まない純粋なラムダ式の集合について以下の問に答えよ．

(i) β 正規形のラムダ式は以下の形に書けることを示せ．

$$\lambda x_1 \ldots \lambda x_n . x M_1 \cdots M_m$$

ここで $x, x_1, \ldots, x_n$ は変数，$M_1, \ldots, M_m$ は β 正規形のラムダ式，$n \geq 0, m \geq 0$ とする．

(ii) ラムダ式に対して，ただ一つの原子型 b と関数型構成子 $\rightarrow$ からなる単純な型システムを考える．空の型環境 ϕ のもとで

$$\phi \rhd M : (b \rightarrow b) \rightarrow (b \rightarrow b)$$

なる型をもつ β 正規形のラムダ式 M の可能な形を決定せよ．

参考：単純な型システムとは以下の型付け規則で与えられる証明システムである．

$$\Gamma\{x : \tau\} \rhd x : \tau$$

$$\frac{\Gamma\{x : \tau_1\} \rhd M : \tau_2}{\Gamma \rhd \lambda x . M : \tau_1 \rightarrow \tau_2}$$

$$\frac{\Gamma \rhd M_1 : \tau_1 \rightarrow \tau_2 \qquad \Gamma \rhd M_2 : \tau_1}{\Gamma \rhd M_1 M_2 : \tau_2}$$

ここで，τ, τ_1, τ_2 は型，Γ は型環境（変数の有限集合から型の集合への関数）を表す．また，記法 $\Gamma\{x : \tau\}$ は，$\Gamma\{x : \tau\}(x) = \tau$ かつ，$x \neq y$ なる y に対して $\Gamma\{x : \tau\}(y) = \Gamma(y)$ となる型環境を表す．

解答． β 正規形のラムダ式の集合を βNF とおく．

(i) ラムダ式の構造に関する帰納法で示す．

• 変数 x の時は β 簡約出来ないから $n = m = 0$ とすれば良い．

• $\lambda x . M \,(M \in \beta\mathrm{NF})$ と書ける時，帰納法の仮定から $M = \lambda x_1 \ldots x_n . y M_1 \ldots M_m \,(M_1, \ldots, M_m \in \beta\mathrm{NF})$ と書ける．よって $\lambda x . M = \lambda x x_1 \ldots x_n . y M_1 \ldots M_m$ だから良い．

• $MN \,(M, N \in \beta\mathrm{NF})$ と書ける時，帰納法の仮定から $M = \lambda x_1 \ldots x_n . y M_1 \ldots M_m \,(M_1, \ldots, M_m \in \beta\mathrm{NF})$ と書ける．ところが MN は β 正規形だからこれ以上 β 簡約出来ない．よって $n = 0$ なので $MN = (y M_1 \ldots M_m) N = y M_1 \ldots M_m N$ となり良い．

これで示された．

(ii) 導出規則を上から順に (var), (abs), (app) とおく．M の導出の最後に用いられるのは (abs), (app) のどちらかである．

(1) (app) の場合：$M = M_1 M_2$ とおけて，型 σ が存在して導出

$$\frac{\emptyset \rhd M_1 : \sigma \rightarrow ((b \rightarrow b) \rightarrow (b \rightarrow b)) \qquad \emptyset \rhd M_2 : \sigma}{\emptyset \rhd M_1 M_2 : (b \rightarrow b) \rightarrow (b \rightarrow b)}$$

が存在する．$M_1 M_2 \in \beta\mathrm{NF}$ だから，(i) の証明と同様に $M_1 = x M_{11} \ldots M_{1m}$ とおける．ただし x は変数，$m \geq 0, M_{11}, \ldots, M_{1m} \in \beta\mathrm{NF}$ である．よって $M = x M_1 \cdots M_m \,(m \geq 1, M_1, \ldots, M_m \in \beta\mathrm{NF})$.

(2) (abs) の場合：$M = \lambda x . M_1$ とおけて，導出

$$\frac{\{x : b \rightarrow b\} \rhd M_1 : b \rightarrow b}{\emptyset \rhd \lambda x . M_1 : (b \rightarrow b) \rightarrow (b \rightarrow b)}$$

が存在する．M_1 の β 正規形を求める．M_1 の導出の最後に用いられるのが

(2-a) (var) の時：$\{x : b \rightarrow b\} \rhd x : b \rightarrow b$ だから $M_1 = x$.

(2-b) (app) の時：$M_1 = M_{11} M_{12}$ とおけて，型 σ が存在して導出

$$\frac{\{x : b \rightarrow b\} \rhd M_{11} : \sigma \rightarrow (b \rightarrow b) \qquad \{x : b \rightarrow b\} \rhd M_{12} : \sigma}{\{x : b \rightarrow b\} \rhd M_{11} M_{12} : b \rightarrow b}$$

が存在する．よって (1) と同様に $M_1 = yM_{11}\cdots M_{1m}\,(m \geq 1, M_{11}, \ldots, M_{1m} \in \beta\mathrm{NF})$.
(2-c) (abs) の時：$M_1 = \lambda y.M_2$ とおけて，導出

$$\frac{\{x : b \to b, y : b\} \rhd M_2 : b}{\{x : b \to b\} \rhd \lambda y.M_2 : b \to b}$$

が存在する．M_2 の β 正規形を求める．M_2 の導出の最後に用いられるのは (var) または (app) である．(var) の時は $M_2 = y$ だから $M_1 = \lambda y.y$ である．(app) の時は (1), (2-b) と同様に $M_2 = zM_{21}\cdots M_{2m}\,(m \geq 1)$ だから $M_1 = \lambda y.zM_{21}\cdots M_{2m}$.
よって $M = \lambda x.x, \lambda x.yM_1\cdots M_m, \lambda x.\lambda y.y, \lambda x.\lambda y.zM_1\cdots M_m\,(m \geq 1)$.
　以上から M の β 正規形は

$$xM_1\cdots M_m, \quad \lambda x.x, \quad \lambda x.yM_1\cdots M_m, \quad \lambda xy.y, \quad \lambda xy.zM_1\cdots M_m$$

である．ただし x, y, z は変数，$m \geq 1, M_1, \ldots, M_m \in \beta\mathrm{NF}$. □

平成 11 年度 (1998 年 8 月実施)

問 10

高々有限個の 1 を含む 0, 1 の無限列 $a = \{a_i\}_{i=0}^{\infty}$ $(a_i \in \{0,1\})$ に対して次のアルゴリズムを考える：

$i \leftarrow 0$
while $a_i = 1$ **do**
  $a_i \leftarrow 0$
  $i \leftarrow i+1$
endwhile
$a_i \leftarrow 1$

ここで，$\displaystyle\sum_{i=0}^{\infty} a_i = n$ なる数列 a に対して，このアルゴリズムを m 回続けて実行したとき，数列要素への代入回数は $O(n+m)$ であることを示せ．

解答． $\sum_{i\geq 0} a_i = n$ の時に m 回続けて実行した場合の a への代入回数の最大値を $f(n,m)$ とおく．明らかに $f(n,0) = 0$ である．$m \geq 1$ とする．$0 \leq j \leq n$ なる j を固定する．$a_0 = a_1 = \cdots = a_{j-1} = 1, a_j = 0$ の状態から 1 回実行すると $a_0 = a_1 = \cdots = a_{j-1} = 0, a_j = 1$ となり，代入回数は $j+1$ で，$\sum a_i$ は $j-1$ だけ減る．ここからさらに $m-1$ 回続けて実行する時の代入回数は最大でも $f(n-(j-1), m-1)$ だから，m 回実行した時の代入回数は最大でも $j+1+f(n-(j-1), m-1)$ となる．よって

$$f(n,m) \begin{cases} = 0 & (m=0) \\ \leq \displaystyle\max_{0\leq j\leq n} \{j+1+f(n-(j-1), m-1)\} & (m \geq 1) \end{cases}$$

が成り立つ．ここで $m \geq 1$ の時 $f(n,m) \leq n+2m-1$ となることを m に関する帰納法で示す．$m=1$ の時は

$$f(n,1) \leq \max_{0\leq j\leq n} \{j+1+f(n-(j-1),0)\} = \max_{0\leq j\leq n} \{j+1\} = n+1$$

だから良い．$m \geq 1$ で正しい時，

$$\begin{aligned} f(n,m+1) &\leq \max_{0\leq j\leq n} \{j+1+f(n-(j-1),m)\} \\ &\leq \max_{0\leq j\leq n} \{j+1+n-(j-1)+2m-1\} \\ &= \max_{0\leq j\leq n} \{n+2m+1\} = n+2m+1 \end{aligned}$$

だから $m+1$ でも正しい．よって示された．従って代入回数は最悪でも $n+2m-1 = O(n+m)$． □

問 11

文脈自由文法 L_1, L_2 を以下のように定義する．

$$
\begin{array}{llcll}
L_1: & E \to E+T & \qquad & L_2: & E \to TF \\
 & E \to T & & & E \to (E)F \\
 & E \to (E) & & & F \to +TF \\
 & T \to 0 & & & F \to \varepsilon \\
 & T \to 1 & & & T \to 0 \\
 & & & & T \to 1
\end{array}
$$

ただし，E, T, F は非終端記号，$+, (,), 0, 1$ は終端記号であり，ε は空列を表す．

このとき，L_1 において E が生成する言語（すなわち，$\{w | E \overset{*}{\underset{L_1}{\Longrightarrow}} w, w$ は終端記号列$\}$ ）と，L_2 において E が生成する言語とは同一であることを示せ．

解答． 簡単のため $\overset{*}{\underset{L_i}{\Longrightarrow}}$ を $\to_i$ と書く．$L_i\,(i=1,2)$ において E が生成する言語を $L_i(E)$ とする．さらに，$L_i(E)$ の元のうち導出木の高さが n のものの集合を $L_i(E;n)$ とする．同様に $L_2(F;n)$ も定める．

● $L_1(E;n) \subset L_2(E)$ を n についての帰納法で示す．L_1 における導出木の高さの最小値は 2 である．$n=2$ の時，導出木の高さが 2 となる導出は $E \to_1 T \to_1 0, E \to_1 T \to_1 1$ の 2 つである．L_2 において $E \to_2 TF \to_2 0\varepsilon = 0, E \to_2 TF \to_2 1\varepsilon = 1$ だから $n=2$ の時は良い．$n-1$ で正しいとする．任意に $w \in L_1(E;n)$ を取る．$E \to_1 w$ を導出する時に最初に用いる生成規則で場合分けする．

○ $E \to T$ の時，これは $n=2$ の場合だから $w \in L_2(E)$ となり良い．

○ $E \to (E)$ の時，$E \to_1 (E) \to_1 w$ だから，$w' \in L_1(E;n-1)$ が存在して $w = (w')$ と書ける．よって帰納法の仮定から $E \to_2 w'$ だから，$E \to_2 (E)F \to_2 (w')\varepsilon = (w') = w$ となり $w \in L_2(E)$.

○ $E \to E+T$ の時，$E \to_1 E+T \to_1 w$ だから，$w' \in L_1(E;n-1), a \in \{0,1\}$ が存在して $w = w'+a$ と書ける．帰納法の仮定から $E \to_2 w'$ である．ところが，L_2 においては E からの導出は必ず末尾に F がつくから，この導出は $E \to_2 w'F \to_2 w'\varepsilon = w'$ の形をしている．特に $E \to_2 w'F$ だから，$E \to_2 w'F \to_2 w'+TF \to_2 w'+a\varepsilon = w'+a = w$ となり $w \in L_2(E)$.

以上から n の時も正しいので示された．これより $L_1(E) = \bigcup_{n \geq 2} L_1(E;n) \subset L_2(E)$.

● $L_2(E) \subset L_1(E)$ を示す．まず $L_2(F;1) = \{\varepsilon\}, L_2(F;n) = \{+a_1 + \cdots + a_{n-1}\,;\, a_i \in \{0,1\}\}\,(n \geq 2)$ を n についての帰納法で示す．$n=1$ の時は自明．n で正しいとする．この時 $w \in L_2(F;n+1)$ の導出は，$w' \in L_2(F;n), a_1 \in \{0,1\}$ が存在して $E \to_2 +TF \to_2 +a_1w' = w$ となる．$n=1$ の時は $w' = \varepsilon$ だから $w = +a_1$. $n \geq 2$ の時は $a_2, \ldots, a_n \in \{0,1\}$ が存在して $w' = +a_2 + \cdots + a_n$ と書けるから $w = +a_1 + a_2 + \cdots + a_n$. 従って $n+1$ でも正しい．これで示せた．これを用いて，n についての帰納法で $L_2(E;n) \subset L_1(E)$ を示す．L_2 における導出木の高さの最小値は 2 である．$n=2$ の時，導出木の高さが 2 となる導出は $E \to_2 TF \to_2 0\varepsilon = 0, E \to_2 TF \to_2 1\varepsilon = 1$ の 2 つである．L_1 において $E \to_1 0, E \to_1 1$ となることは上で既に見たから，$n=2$ の時は良い．$n-1$ で正しいとする．任意に $w \in L_2(E;n)$ を取る．$E \to_2 w$ の導出時に最初に用いる導出規則で場合分けする．

○ $E \to TF$ の時，$w' \in L_2(F;n-1), a_0 \in \{0,1\}$ が存在して $E \to_2 TF \to_2 a_0w' = w$ となる．$n-1 \geq 2$ より $n \geq 3$ だから，$a_1, \ldots, a_{n-2} \in \{0,1\}$ を用いて $w' = +a_1 + \cdots + a_{n-2}$ と書ける．従って $E \to_1 E+T \to_1 E+T \cdots + T \to_1 T + \cdots + T \to_1 a_0 + a_1 + \cdots + a_{n-2} = w$ となり $w \in L_1(E)$.

○ $E \to (E)F$ の時，$w' \in L_2(E;n_1), w'' \in L_2(F;n_2)\,(\max\{n_1,n_2\} = n-1)$ が存在して $E \to_2 (E)F \to_2 (w')w'' = w$ となる．帰納法の仮定から $E \to_1 w'$ である．$n_2 = 1$ の場合は $w'' = \varepsilon$ だから $E \to_1 (E) \to_1 (w') = w$ となり $w \in L_1(E)$. $n_2 \geq 2$ の場合は $a_1, \ldots, a_{n_2-1} \in \{0,1\}$ を用いて $w'' = +a_1 + \cdots + a_{n_2-1}$ と書けるから，$E \to_1 E+T \to_1 E+T+\cdots+T \to_1 (E)+T+\cdots+T \to_1 (w')+a_1+\cdots+a_{n_2-1} = (w')w'' = w$ となり $w \in L_1(E)$.

以上から n の時も正しいので示された．これより $L_2(E) = \bigcup_{n \geq 2} L_2(E;n) \subset L_1(E)$. □