平成 26 年度 博士前期課程 入試問題 (解答)

1 アルゴリズムとプログラミング

(1)

【出力結果 (43 行目)】
```
5 6 10
```

【出力結果 (46 行目)】
```
2 5 5 7 13 18 20
```

(2)

変数 front の値は delete 関数が呼ばれた回数と等しく，変数 rear の値は insert 関数が呼ばれた回数に等しい．よって，front $= 3$, rear $= 5$

(3)

配列 A が整列されていることを利用して，二分探索を用いて挿入箇所を調べている．結果，配列の要素数を n とすると，先頭から 1 つずつ調べると $O(n)$ のコストが必要だったが，$O(\log_2 n)$ のコストですむようになった．

(4)

insert 関数

挿入箇所を調べるのに $O(\log_2 n)$，新しいデータを挿入するのに $O(n)$ の時間計算量が必要なので，$O(n+\log_2 n) \rightarrow O(n)$ の時間計算量が必要となる．

delete 関数

for 文がなく，変数 front をインクリメントするだけなので，時間計算量は $O(1)$ となる．

(5)

データの挿入を 20 回行うと，それ以降の挿入は失敗してしまう．その原因は，データの挿入が 20 回行われると，変数 rear の値が 20 となり，データ列のデータ数に関わらず 9 行目の (rear $>$ SIZE - 1) の条件式が true となってしまい挿入に失敗してしまうからである．

(6)

データ列のデータ構造をリングバッファとして持てば時間計算量はほとんど増加せずに解決ができる．具体的に説明すると，変数 rear が変数 SIZE 以上になれば，変数 rear を 0 にする．また，挿入位置を探索するときは，(rear $\geq$ front) であれば従来の方法で行い，(rear $<$ front) の場合は，front〜(rear + SIZE - 1) の範囲で二分探索を行えば良い．その時，データ列にアクセスする際の添字は変数 SIZE の剰余を取れば良い．

2 計算機システムとシステムプログラム

(1-1)

i: $90 = (01011010)_2$
値：90**，ビット列：**01011010

j: $-40 = (11011000)_2$
値：-40**，ビット列：**11011000

k: $(01011010)_2 + (11011000)_2 = (00110010)_2$
値：50**，ビット列：**00110010

m: $(01011010)_2 - (11011000)_2 = (10000010)_2$
値：-126**，ビット列：**10000010

n: $(11011000)_2 - (01011010)_2 = (01111110)_2$
値：126**，ビット列：**01111110

d: $0.4 = (-1)^0 \times (1.1001)_2 \times 2^{-2}$ より，
$s = 0, m = (1001)_2, e = 1 = (001)_2$
値：0.390625**，ビット列：**00011001

e: $0.8 = (-1)^0 \times (1.1001)_2 \times 2^{-1}$ より，

$$\begin{aligned} 0.4 + 0.8 &= (-1)^0 \times (1.1001)_2 \times 2^{-2} + (-1)^0 \times (1.1001)_2 \times 2^{-1} \\ &= (-1)^0 \times (0.11001)_2 \times 2^{-1} + (-1)^0 \times (1.1001)_2 \times 2^{-1} \\ &= (-1)^0 \times (10.010)_2 \times 2^{-1} \\ &= (-1)^0 \times (1.0010)_2 \times 2^0 \end{aligned}$$

$s = 0, m = (0010)_2, e = 3 = (011)_2$
値：1.125**，ビット列：**00110010

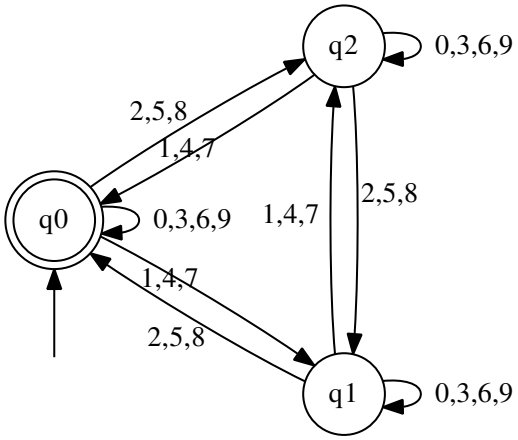
(1-2)

2 の補数系表現において，減算処理は減算数の正負を反転させたものを被減算数に加算処理することと等価になるので，計算機上に加算器だけ用意しておけば減算処理も実現可能である．よって，2 の補数系表現を用いるとコストや回路の簡素さの面でメリットがある．

4 計算理論

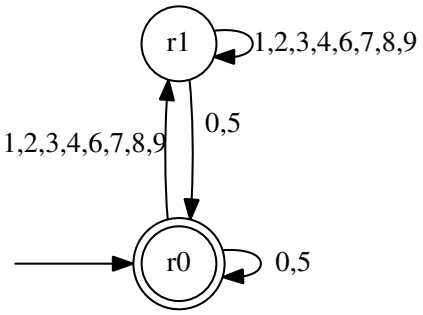
(1-1)

10 進数の非負整数で 3 で割り切れる数は各桁の和が 3 の倍数であるものである．よって，現時点での各桁の和の 3 の剰余が 0,1,2 の 3 状態を用意し，剰余が 0 の状態を受理状態にすれば良い．

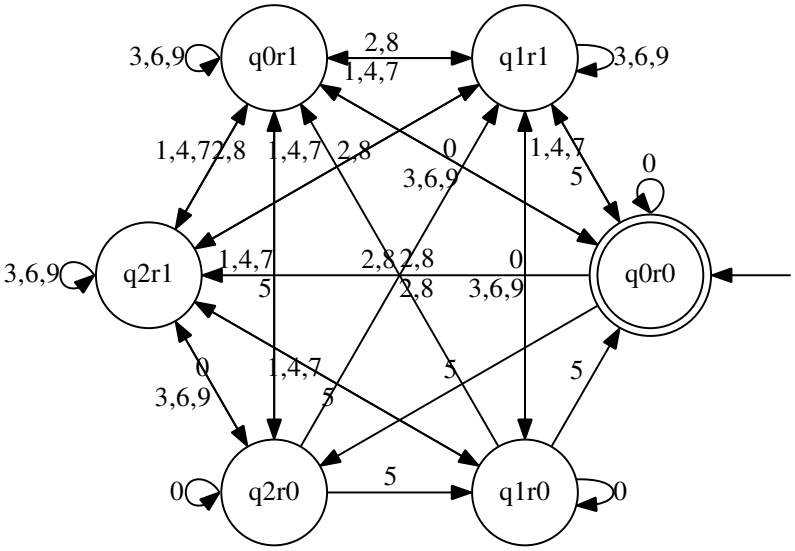


(1-2)

10 進数の非負整数で 5 で割り切れる数は 1 の位の数が 0 または 5 である数である．よって，最後の入力が 0 または 5 であれば受理とするオートマトンを作成すれば良い．有限オートマトン C の図を以下に示す．

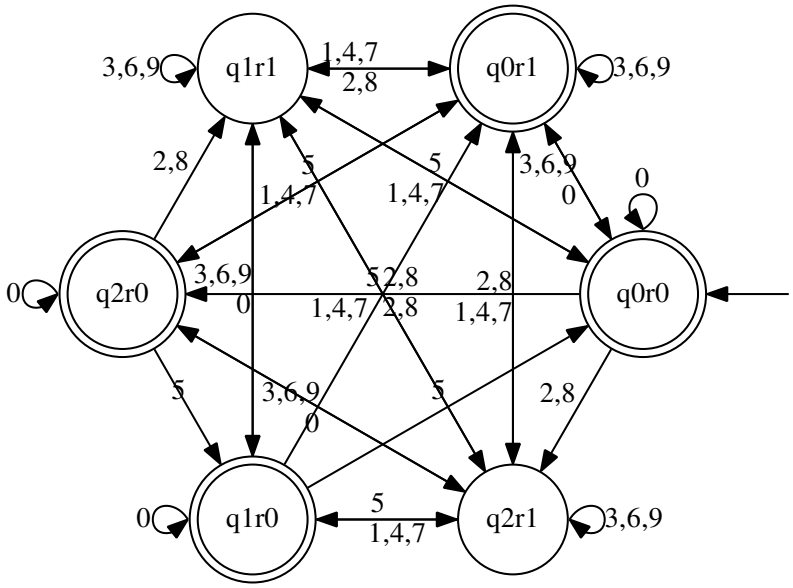


15 で割り切れる数は 3 と 5 の両方で割り切れる数である．よって，有限オートマトン B および C の両方を受理する入力は 15 の倍数であると見なせる．2 つのオートマトンの状態を合成した 6 状態 $(q_0r_0, q_1r_0, q_2r_0, q_0r_1, q_1r_1, q_2r_1)$ を持つ有限オートマトン D を以下に示す．



(1-3)

有限オートマトン D 上で，3 で割り切れる状態 (q_0) または 5 で割り切れる状態 (r_0) であるときに受理状態とすれば良い．ようするに，$q_0r_0, q_0r_1, q_1r_0, q_2r_0$ の 4 状態を受理状態としたものを有限オートマトン E とする．
有限オートマトン E を以下に示す．



(2-1)

次のように文脈自由文法 G_1 の 4 項組 (N_1, T_1, P_1, S_1) を与える．

$$N_1 = \{S\},\ T_1 = \{c\},\ P_1 = \{S \to c,\ S \to cS\},\ S_1 = S$$

(2-2)

次のように文脈自由文法 G_5 の 4 項組 (N_5, T_5, P_5, S_5) を与える．

$$N_5 = \{S\},\ T_5 = \{a, b\},\ P_5 = \{S \to ab,\ S \to aSb\},\ S_5 = S$$

文脈自由文法 G_5 から生成される文は，$S \to ab$，$S \to aSb \to aabb$，$S \to aSb \to aaSbb \to aaabbb...$ となるので，$L_5 = L(G_5) = \{a^n b^n \mid n \geq 1\}$ となる．また，L_5 は文脈自由言語である．
ここで，文脈自由言語 L_5, L_1 の連接演算を行う．$L_5 \cdot L_1 = \{a^n b^n c^m \mid n \geq 1, m \geq 1\} = L_2$ となる．よって，文脈自由言語 L_1, L_5 および補題 1 より，L_2 は文脈自由言語である．

同様に，次のように文脈自由文法 G_6 の 4 項組 (N_6, T_6, P_6, S_6)，文脈自由文法 G_7 の 4 項組 (N_7, T_7, P_7, S_7) を与える．

$$N_6 = \{S\},\ T_6 = \{a\},\ P_6 = \{S \to a,\ S \to aS\},\ S_6 = S$$
$$N_7 = \{S\},\ T_7 = \{b, c\},\ P_7 = \{S \to bc,\ S \to bSc\},\ S_7 = S$$

先程と同様に考えると，$L_6 = L(G_6) = \{a^m \mid m \geq 1\}$，$L_7 = L(G_7) = \{b^n c^n \mid n \geq 1\}$ となる．この L_6, L_7 は文脈自由言語である．ここで，文脈自由言語 L_6, L_7 の連接演算を行う．$L_6 \cdot L_7 = \{a^m b^n c^n \mid n \geq 1, m \geq 1\} = L_3$ となる．よって，文脈自由言語 L_6, L_7 および補題 1 より，L_3 も文脈自由言語である．

(2-3)

文脈自由言語全体の集合が積演算について閉じていると仮定する．(背理法)
いま，文脈自由言語 L_2, L_3 の積演算を行うと，

$$L_2 \cap L_3 = \{a^n b^n c^n \mid n \geq 1\} = L_4$$

となる．仮定より文脈自由言語全体の集合が積演算について閉じているので言語 L_4 は文脈自由言語である．しかし，補題 2 より，L_4 は文脈自由言語ではないので矛盾が発生し，仮定が間違っていることになる．よって，背理法により，文脈自由言語全体の集合は積演算について閉じていないことが証明された．

6 電子回路と論理設計

(1-1)

select	a	b	out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

(1-2)

select \ ab	00	01	11	10
0	0	1	1	0
1	0	0	1	1

(1-3)

$$\mathrm{out} = \mathrm{select} \cdot \mathrm{a} + \overline{\mathrm{select}} \cdot \mathrm{b}$$

(1-4)

$$\mathrm{out} = \overline{\overline{\mathrm{a} \cdot \mathrm{select}} \cdot \overline{\mathrm{b} \cdot \overline{\mathrm{select}}}}$$

より，論理回路図は以下のようになる．
(図省略)

(2-1)
遅延時間: 3T

(2-2)
(select, a, b) が (0,0,1) から (1,0,1) に変わるとき，マルチプレクサの遅延時間が 5T となり最大となる．