大阪大学大学院情報科学研究科 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)

43 行目: 5 6 10
46 行目: 2 5 5 7 13 18 20

(2)

front: 3
rear : 5

(3)

既にソートされているデータ列中の適切な挿入位置を 2 分探索により決定している.

(4)

insert 関数:

挿入位置を求めるための時間計算量のオーダーは $O(\log n)$ であるが, 挿入位置及びそれより後ろの要素を全て後ろにずらす処理がある（24 行目）ため, `insert` 関数全体では $O(n)$ となる.

delete 関数:

関数中にループが無く, 全ての処理がデータ列のデータ数に依存しない. 従って, 定数時間 $O(1)$ で実行可能.

(5)

`insert` 関数により増加したグローバル変数 `rear` の値を減少させる処理が無いため, 配列の大きさ（図 1 のプログラム中では 20 回）よりも多く `insert` 関数を実行しようとした場合に配列外への参照を引き起こし, データの挿入に失敗する.（9 行目）

(6)

配列の末端まで使用された場合, その次の要素を配列の先頭から順に格納するようにする. そのため, 関数 `delete` に配列の大きさを示す引数 `SIZE` を追加し, `insert` 関数, `delete` 関数において, 配列 `A` の要素へアクセスする場合, インデックスの値の剰余を取るようにする（例えば, 31 行目の文 `x = A[front];` であれば, `x = A[front % SIZE];` のようにする.）また, 9 行目の if 文中の条件 `rear > SIZE-1` を `rear-front > SIZE-1` に変更する. ただし, この方法は配列の大きさより多くの要素を保持することはできない.

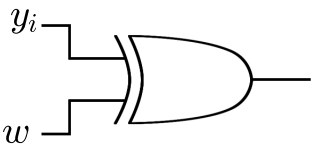
2. 計算機システムとシステムプログラム

(1)
(1-1)

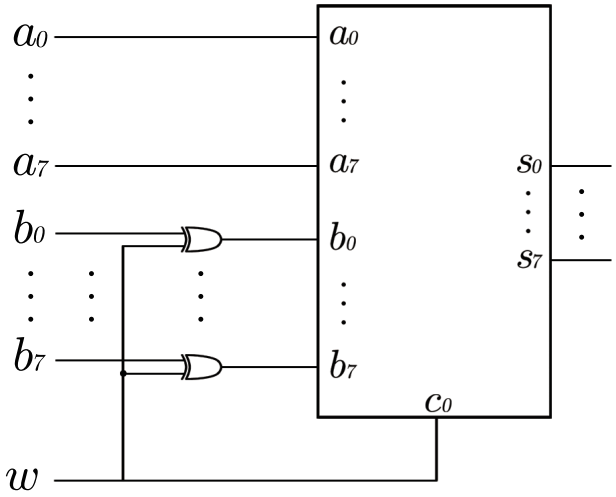
変数	10 進数	ビット列	
i	90	0101	1010
j	-40	1101	1000
k	50	0011	0010
m	-126	1000	0010
n	126	0111	1110
d	0.390625	0001	1001
e	1.125	0011	0010

(1-2)
加算と同じ仕組みで減算ができるため, 減算用の回路が不要になり, ハードウェアの構造を簡単にできる.

(1-3)



(1-4)



(2)

(2-1)

a: 空き（unlock）　　**b:** 使用中（lock）　　**c:** 競合

(2-2)

まずプロセス X がテストを実行する. 続いて, プロセス X がセットを実行する前に, プロセス Y がテストを実行したとする. すると, X と Y が同時に共有資源を使用できる状態になり, 排他制御に失敗する.

3. 離散構造

(1)

(1-1)

R_2

証明:

$R_2 = \{(x, y) \,|x$ と y を共に含む有向閉路が存在する$\} \cup S$

反射律: R_2 の定義より明らかに満たす.

対称律: x と y を共に含む有向閉路と, y と x を共に含む有向閉路は同じ意味. したがって, $^{\forall}x, y \in V; (x, y) \in R_2 \Rightarrow (y, x) \in R_2$ となり満たす.

推移律: x と y を共に含む有向閉路と, y と z を共に含む有向閉路があるとき, それらを繋ぎ合わせると x（と y）と z を含む有向閉路になるため, $^{\forall}x, y, z \in V; (x, y) \in R_2 \wedge (y, z) \in R_2 \Rightarrow (x, z) \in R_2$ となり満たす.

R_2 は反射律, 対称律, 推移律を満たすため, 同値関係である.

(1-2)

R_3

証明:

$R_3 = \{(x, y) \,|s$ から y へのすべての有向経路が x を含む$\}$

反射律: s から x への全ての有向経路は x を含むため, 満たす.

反対称律: s から x への全ての有向経路が y を含むとき, s から x でない y への全ての有向経路が x を含むならば, s から x へ y を含まない含まない有向経路が存在することになり, 矛盾する.

よって, $^{\forall}x, y \in V; (x, y) \in R_3 \wedge (y, x) \in R_3 \Rightarrow x = y$ となり満たす.

推移律: s から y への全ての有向経路が x を含み, s から z への全ての有向経路が y を含むとき, s から z への全ての有向経路は x を含むため, 満たす.

R_3 は反射律, 反対称律, 推移律を満たすため, 半順序関係である.

(1-3)

R_3

(2)
(2-1)

(a) $r\,(h,x,y) \to \bigvee_{z\in X} m\,(h,z,y)$

(b) $m\,(h-1,z,y) \to \bigvee_{z\in X} m\,(h,z,y)$

(c) $m\,(h,x,y) \to (m\,(h-1,x,y) \vee r\,(h,x,y))$

(2-2)

$$P \equiv (CX1 \wedge CX2 \wedge CX3 \wedge CX4 \wedge CX5 \wedge CY1 \wedge CY2 \wedge CY3 \wedge CY4 \wedge CY5 \wedge CZ1 \wedge CZ2) \to H$$

(2-3)
(2-3-1)

(d) $r\,(1,u1,v2)$

(e) $r\,(1,u2,v2)$

(f) $r\,(1,u1,v1) \to (m\,(1,u1,v1) \vee m\,(1,u2,v1))$

(g) $m\,(1,u1,v2) \to r\,(1,u1,v2)$

(h) $m\,(1,u2,v2) \to r\,(1,u2,v2)$

(2-3-2)

$$P \equiv (CX1 \wedge \ldots \wedge CX5 \wedge CY1 \wedge \ldots \wedge CY5) \to (m\,(1,u2,v1) \wedge m\,(1,u2,v2) \wedge \neg m\,(1,u1,v1) \wedge \neg m\,(1,u1,v2))$$

P の恒真性を導出原理を利用して示す.

$$\neg P \equiv CX1 \wedge \ldots \wedge CX5 \wedge CY1 \wedge \ldots \wedge CY5 \wedge (\neg m\,(1,u2,v1) \vee m\,(1,u2,v2) \vee m\,(1,u1,v1) \wedge m\,(1,u1,v2))$$

$(\neg m\,(1,u2,v1) \vee m\,(1,u2,v2) \vee m\,(1,u1,v1) \wedge m\,(1,u1,v2))$ を ① とおく.

$CX5$ で $(h,k,x,y,w) \leftarrow (1,0,u1,v2,v1)$ とすると, $r\,(1,u1,v2) \cdots$ ②
$CX5$ で $(h,k,x,y,w) \leftarrow (1,0,u2,v2,v1)$ とすると, $r\,(1,u2,v2) \cdots$ ③
$CX1$ で $(h,x,y,w) \leftarrow (1,u1,v1,v2)$ とすると, $\neg r\,(1,u1,v1) \vee \neg r\,(1,u1,v2) \cdots$ ④
$(1,u2,v1,v2)$ とすると, $\neg r\,(1,u2,v1) \vee \neg r\,(1,u2,v2) \cdots$ ⑤
②と④より導出 $\neg r\,(1,u1,v1) \cdots$ ⑥
③と⑤より導出 $\neg r\,(1,u2,v1) \cdots$ ⑦

$CY5$ で $(h,x,y) \leftarrow (1,u1,v2)$ とすると, $\neg m\,(1,u1,v2) \vee r\,(1,u1,v2) \cdots$ ⑧
$CY5$ で $(h,x,y) \leftarrow (1,u2,v2)$ とすると, $\neg m\,(1,u2,v2) \vee r\,(1,u2,v2) \cdots$ ⑨

$CY5$ で $(h,x,y) \leftarrow (1,u1,v1)$ とすると, $\neg m\,(1,u1,v1) \vee r\,(1,u1,v1) \cdots$ ⑩
$CY5$ で $(h,x,y) \leftarrow (1,u2,v1)$ とすると, $\neg m\,(1,u2,v1) \vee r\,(1,u2,v1) \cdots$ ⑪
⑥と⑩より導出　　$\neg m\,(1,u1,v1) \cdots$ ⑫
⑦と⑪より導出　　$\neg m\,(1,u2,v1) \cdots$ ⑬
$CY2$ で $(h,x,y,z) \leftarrow (1,u1,v2,u2)$ とすると,
　　$\neg r\,(1,u1,v2) \vee \neg r\,(1,u2,v2) \vee \neg m\,(1,u1,v2) \cdots$ ⑭
②と③と⑭より　　$\neg m\,(1,u1,v2) \cdots$ ⑮
$CY4$ で $(h,x,y) \leftarrow (1,u1,v1)$ とすると, $m\,(1,u1,v1) \vee m\,(1,u2,v1) \cdots$ ⑯
⑫と⑬と⑯より空節を導出. よって, $\neg P$ は恒偽. したがって, P は恒真である.

4. 計算理論

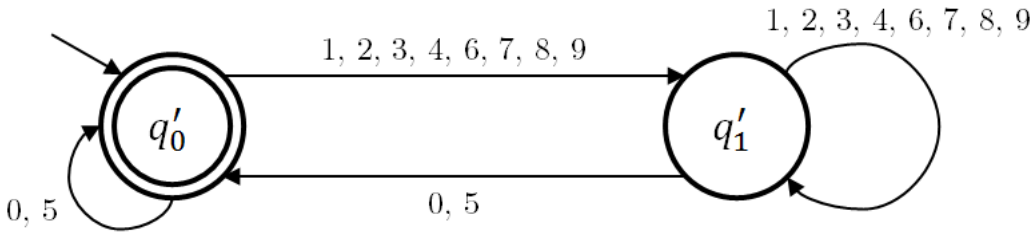
(1)

(1-1)

与えられた語のうち，それまでに読みこまれた部分を 3 で割った余りが 0 になる場合は状態 q_0，1 になる場合は状態 q_1，2 になる場合は状態 q_2 になるようにオートマトンを構成する． オートマトン $\mathcal{B}$ は，$\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, \delta_{\mathcal{B}}, q_0, \{q_0\})$ と表される． ここで，$Q_{\mathcal{B}} = \{q_0, q_1, q_2\}$ であり，遷移関数 $\delta_{\mathcal{B}}$ は，次状態が $\{(\text{現状態に対応する余りの値}) \times 10 + (\text{入力記号})\}$ を 3 で割った余りに対応する状態に遷移するように定める．

(1-2)

有限オートマトン $\mathcal{C}$ は以下のようになる．



オートマトン $\mathcal{C}$ を $\mathcal{C} = (Q_{\mathcal{C}}, \Sigma, \delta_{\mathcal{C}}, q_0', \{q_0'\})$ と表す． ただし，$Q_{\mathcal{C}} = \{q_0', q_1'\}$ であり，$\delta_{\mathcal{C}}$ は上図のオートマトンに従うものとする． オートマトン $\mathcal{D}$ は，$\mathcal{D} = (Q_{\mathcal{B}} \times Q_{\mathcal{C}}, \Sigma, \delta_{\mathcal{D}}, (q_0, q_0'), \{(q_0, q_0')\})$ と表される． ただし，全ての $((P_1, P_2), a) \in Q_{\mathcal{B}} \times Q_{\mathcal{C}} \times \Sigma$ について，$\delta_{\mathcal{D}}((P_1, P_2), a) = (\delta_{\mathcal{B}}(P_1, a), \delta_{\mathcal{C}}(P_2, a))$ とする．

(1-3)

オートマトン $\mathcal{E}$ を, 以下のように定める．

$$\mathcal{E} = \left(Q_{\mathcal{B}} \times Q_{\mathcal{C}}, \Sigma, \delta_{\mathcal{E}}, (q_0, q_0'), \left\{(q_0, q_1'), (q_1, q_0'), (q_2, q_0')\right\}\right)$$

ただし, $\delta_{\mathcal{E}} = \delta_{\mathcal{D}}$.

（※「3 または 5 で割り切れるときのみ」を 3 または 5 のどちらか一方でのみ割り切れるときと捉えた場合）．

「3 または 5 で割り切れるときのみ」を 3 で割り切れるか, または, 5 で割り切れるときと捉えた場合は,

$$\mathcal{E} = \left(Q_{\mathcal{B}} \times Q_{\mathcal{C}}, \Sigma, \delta_{\mathcal{E}}, (q_0, q_0'), \left\{(q_0, q_0'), (q_0, q_1'), (q_1, q_0'), (q_2, q_0')\right\}\right)$$

ただし, $\delta_{\mathcal{E}} = \delta_{\mathcal{D}}$.

(2)
(2-1)

$G_1 = (\{S\}, \{c\}, \{S \to cS, S \to c\}, S)$

(2-2)

L_2 を生成する文法 G_2 は,

$$G_2 = (\{S, X, Y\}, \{a, b, c\}, \{S \to XY, X \to aXb, X \to ab, Y \to cY, Y \to c\}, S)$$

である. 文法 G_2 から得られた語 w_2 について, その導出過程で規則 $X \to aXb$ が適用された回数を n' 回（$n' \geq 0$）, 規則 $Y \to cY$ が適用された回数を m' 回（$m' \geq 0$）とすると, $w_2 = a^{n'+1}b^{n'+1}c^{m'+1}$ となる. $n' \geq 0$, $m' \geq 0$ より, 文法 G_2 により生成される言語は L_2 に等しい. ゆえに, 言語 L_2 は文脈自由言語.
L_3 についても, 文法 G_3

$$G_3 = (\{S, X, Y\}, \{a, b, c\}, \{S \to XY, X \to Xa, X \to a, Y \to bYc, Y \to bc\}, S)$$

について同様に示される.

(2-3)

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

より, 言語 L_2 と言語 L_3 の積は L_4 に等しい. 補題 2 より, L_4 は文脈自由言語でないので, $L_2 \cap L_3$ も文脈自由言語でない. 従って, 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.