トランジスタがONで出力がLのとき　$V_{OL} = 0.2\,V$
R_3にかかる電圧は　4.8 V　、電流は　$\frac{4.8}{R_3}$
これが消費電流 I_{CC} も超える事はないから，　$4.8/R_3 < 3\times10^{-3}$
$\therefore\ R_3 > 1.6\times10^{3}$
以上より　$1.6\times10^{3} < R_3 \leqq 2.5\times10^{3}$　だから　$R_3 = 2.2\times10^{3}$

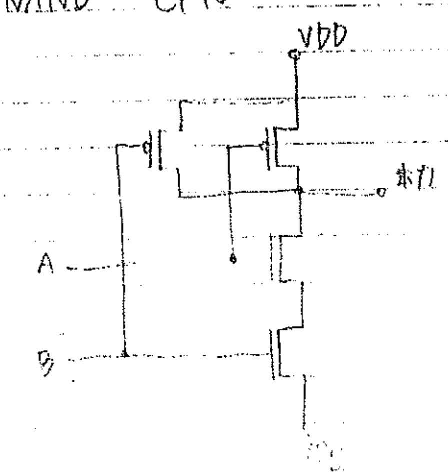
入力がLのとき　$I_{IL}(max) = 0.3\ mA$
抵抗 R_1 に流れる電流は　0.3 mA 以下．　電圧は $5-0.7 = 4.3\,V$
$\therefore\ \frac{4.3}{R_1} \leqq 0.3\times10^{-3}$　$\therefore\ R_1 \geqq 14\times10^{3}$

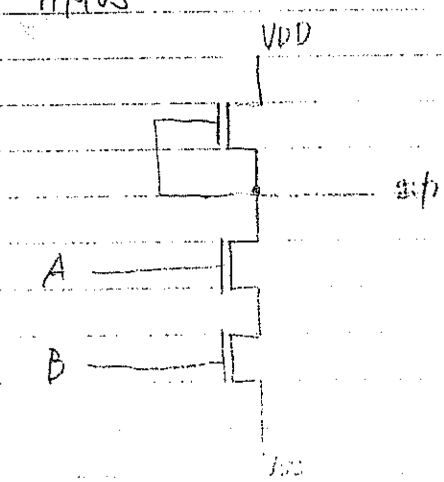
入力がHでトランジスタがONの場合、$I_{OL}(min) = 20\,mA$ より
$I_C = 20\times10^{-3} + 4.8/2.2\times10^{3} = 22.2\times10^{-3}$
$I_B\cdot\beta = I_C$ より　$I_B = 22.2\times10^{-3}/150 = 0.15\times10^{-3}$ 以上必要
$R_2 = 100k$ で十分大きいので（0.7Vで流れる電流は 7μ）無視して考えると
$(5-0.7\times3)/R_1 \geqq 0.15\times10^{-3}$　より　$R_1 \geqq 19\times10^{3}$
$1.4\times10^{4} \leqq R_1 \leqq 1.9\times10^{4}$　より　$R_1 = 1.5\times10^{4}$

(2-1)

A	ア	B	ク	C	キ
D	サ	E	シ	F	Id(VDD－VSS)
G	ウ	H	エ	I	コ
J	ケ	K	カ	L	オ

(2-2) NAND　CMOS　nMOS

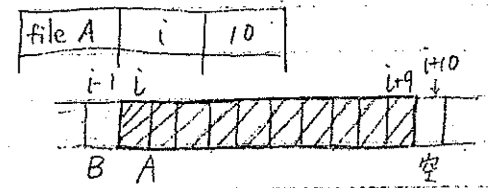




H9 [I]

オペレーティングシステム

[1] (1) 連続割当て



(a) read i+9
write i+10
read i+8
write i+9
⋮
read i
write i+1
write i

よって　$2\times10+1 = 21$　　21回

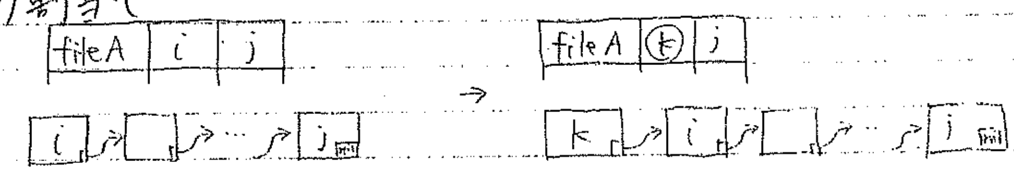
(b) read i+9
write i+10
⋮
read i+5
write i+6
write i+5

よって　$2\times5+1 = 11$　　11回

(c) write i+10　　1回

・リンク割当て

(a)



write k　（次のリンク先は i）　　1回

(b) read i　（次のブロックが i_2 と分かる）
read i_2　（次のブロックが i_3 と分かる）
⋮
read i_5　（次のブロックが i_6 と分かる）
write i_5　（次のリンク先を k とする）
write k　（kブロック追加し、次のリンク先は i_6）

よって　$5+2 = 7$　　7回

No.　　Date

(c)　write j　（次のリンク先をkとする）
　　write k　さらにディレクトリも変更　　∴ 2回

(2) 連続割当て

最後に追加　アクセス回数　1 回
最後からk番目に追加　2k+1 回　（k個のブロックをreadとwriteでずらし新しいブロックを追加）
先頭に追加　2n+1 回

$$\therefore \left\{\sum_{k=0}^{n}(2k+1)\right\}/(n+1) = \{n(n+1)+(n+1)\}/(n+1)$$
$$= (n+1)^2/(n+1)$$
$$= n+1 \qquad \therefore n+1$$

リンク割当て

先頭に追加　1 回
先頭からk番目の後ろに追加　k+2 回　（k回readしてブロックをたどり、k番目のリンク先を変えて、ブロックを追加する）
最後に追加　2 回

$$\therefore \left\{1+\sum_{k=1}^{n-1}(k+2)+2\right\}/(n+1) = \left\{3+\frac{1}{2}(n-1)\cdot n+2(n-1)\right\}/(n+1)$$
$$= \cancel{\frac{1}{2}(n+2)(n+1)/(n+1)}$$
$$= \frac{n+2}{2} \qquad \therefore \frac{n+2}{2}$$

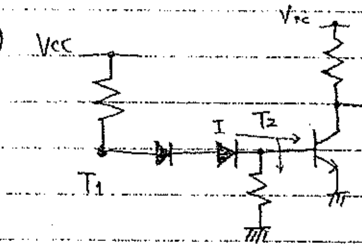
(3) 連続割当ては、ファイル作成時の割当てブロック数を必要なだけ、あらかじめ静的に指定しておかなければならない。また、ブロックが隣接しているので分散している場合に比べてreadアクセス効率はよい。
未使用ブロックの合計は必要なだけあるにもかかわらず、ブロックが連続していないためにファイルを作成できないという、フラグメンテーション（断片化）の問題が生じるので、空きブロックを1つの連続した領域にまとめるコンパクションが必要となる。
リンク割当ては、ファイル作成時の割当てブロック数をあらかじめ決める必要はない。またファイルの読出しは、ポインタをたどりながら行うのでreadアクセス効率は悪い。フラグメンテーションは生じないのでコンパクションは必要ない。

H9. II

Date　　No.

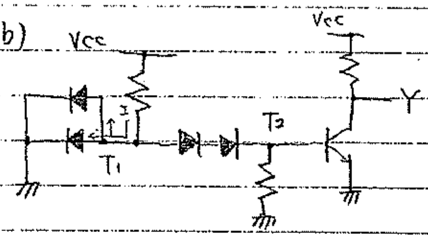
ディジタル回路

4 (1-1) (a)



∵ トランジスタには、ベース電流が流れるので、飽和状態。よって $V_{BE}=0.7V$
よって $T_2=0.7V$
∵ ダイオードの電圧降下は0.7Vより
$0.7+0.7+0.7=2.1$ で $T_1=2.1V$
∵ $V_{CE}=0.2$ より　Yは Low

(b)



∵ ダイオードの電圧降下は0.7Vより
$T_1=0.7V$
∵ 又T_2には電流が流れない
$T_2=0V$
よって Yは High

(c) A, Bどちらかが GNDなら、T_2は0VになりYはHighとなる。
∴ NAND

(d) トランジスタにベース電流が流れると、YはLow、流れないとYはHighとなる。
→ T_2の電圧が0.7Vより大きいときYはLow、0.7V以下のときYはHigh
→ T_1の電圧が2.1Vより大きいときYはLow、2.1V以下のときYはHigh（ダイオード2つつながっているから）
→ 入力の電圧が1.4Vより大きいときYはLow、1.4V以下のときYはHigh
∴ 閾値電圧は1.4V

(e) ダイオードがひとつだと入力をGNDにつないだときに、トランジスタのベースに、少し電流が流れる場合があるから。

(1-2) トランジスタがOFFで出力がHの場合、$V_{OH}(min)=3.0V$ より
R3にかかる電圧は、0以上2.0V以下、電流は0以上 $\frac{2.0}{R3}$ 以下
$I_{OH}(min)=0.8mA$ より

$$\frac{2.0}{R3} \geqq 0.8\times10^{-3}$$

よって $R3 \leqq 2.5\times10^{3}$

よって $\mathcal{X}(g(u,w))$ によって出力されるのは
$(30 - 20 - 10 - 9 - 8)$ ↓

(3-2) 関数 g において malloc が呼び出されるのは $p(t, x \to number)$
だけ。つまり return($p(t, x \to number)$) を実行するとき。
$\therefore$ m 回

(3-3) 関数 g は、第1引数のリストの後ろに第2引数のリストをつなげて返す。

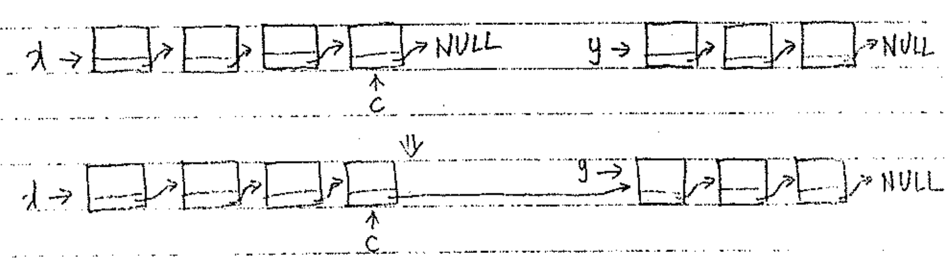
・第1引数が NULL なら、第2引数をそのまま返す。 → return(y)

・第1引数のリストの最後を指す c を求める。
→ for(c=x; c->next != NULL; c=c->next)

・c の next を y とする。 → c->next = y

・x を返す。 → return(x)

$\therefore$ ① B ② G ③ E ④ D ⑤ A



H9 [I] (3-2)

情報論

[2] (1) 区切られたある語 w を考える。
$w = vz \ (v \in \Sigma^*, z \in \Sigma)$ とする。(B) より v は辞書に登録されていて
その登録番号 $i(v)$ の2進表現 $b(v)$ と「0*」又は「1*」を符号語する。
よって w の符号語の長さは $(b(v)$ の桁数$) + 2$ である。

ここで v は w より前に登録されているので、$i(v)$ は高々 $i(w) - 1$

又、$i(v)$ の2進表現 $b(v)$ の桁数は $\lfloor \log i(v) + 1 \rfloor$

w の符号語の長さは高々 $\lfloor \log(i(w)-1) + 1 \rfloor + 2$ より
高々 $\log i(w) + 3$

今、x は $S(x)$ 個に区切られているので、符号化列 $C(x)$ の長さは
高々
$$\sum_{k=1}^{S(x)} \{\log k + 3\} + 1 = \log(S(x)!) + 3S(x) + 1$$

$$\begin{aligned} \therefore L(x) &= O(\log(S(x)!) + 3S(x) + 1) \\ &= O(\log(S(x)!)) \\ &= O(\log S(x)^{S(x)}) \\ &= O(S(x) \cdot \log S(x)) \end{aligned}$$
$\therefore$ ②

(2) $x = aaa \cdots a = a | aa | aaa | \cdots | aa \cdots a$

$i(\varepsilon) = 1,\ i(a) = 2,\ i(aa) = 3,\ i(aaa) = 4, \cdots$

$C(x) = 10*100*110*1000* \cdots$
i 番目の区切られた語 $w(=vz)$ の v は $i-1$ 番目の語で $z = a$ だから
$i-1$ を2進表現にしたものに続けて 0* としたものが符号語。

自然数を順に2進表現で表し *をつけたものとなる。

n個の a を、1個、2個、3個、…、と区切っていくので、区切られた個数 $S(x)$ は

$$1+2+3+\cdots+S(x) \geqq n$$

$$\frac{S(x)(S(x)+1)}{2} \geqq n$$ を満たす最小の整数

$$S(x)^2+S(x)-2n \geqq 0$$

$$\therefore S(x) = \left\lceil \frac{-1+\sqrt{1+8n}}{2} \right\rceil$$

・$C(x)$の長さ $L(x)$ は (1) より $L(x) = O(S(x) \log S(x))$

$$\therefore L(x) = O\left(\frac{-1+\sqrt{1+8n}}{2} \log \frac{-1+\sqrt{1+8n}}{2}\right) = O(\sqrt{n} \log n)$$

x の長さは $O(n)$ で

$n \to \infty$ のとき $O(\sqrt{n}\cdot\log n) < O(\sqrt{n}\cdot\sqrt{n}) = O(n)$

$\because O(\sqrt{n}\cdot\log n) < O(n)$ より 短縮可能

(3-1) 送信側と受信側で同期がとれていれば、$f(a)$ を受信し終った時点で、記号 a が送信されたと分かるので復号が簡単

(3-2)

H9, $\boxed{\text{II}}$

アルゴリズムとデータ構造

$\boxed{3}$ (1) $p(NULL, 1)$ によって $\rightarrow \boxed{1} \rightarrow$ NULL というリスト A が返る。

$p(A, 2)$ によって $\rightarrow \boxed{2} \rightarrow \boxed{1} \rightarrow$ NULL というリスト B が返る。

$p(B, 3)$ によって $\rightarrow \boxed{3} \rightarrow \boxed{2} \rightarrow \boxed{1} \rightarrow$ NULL というリストが返る。

$t = p(p(p(NULL,1),2),3)$ より $t \rightarrow \boxed{3} \rightarrow \boxed{2} \rightarrow \boxed{1} \rightarrow$ NULL

よって $g(t)$ によって出力されるのは (3_2_1)↓

(2) ・ポインタ a の number が削除する値のとき、

(i) a の next の指すリストを b とする → $b = a \rightarrow next$

(ii) a を削除 → free(a)

(iii) b を返す → return(b)

・ポインタ a の number が削除する値でないとき、

(i) 次のポインタをたどりながら探し、削除したリストを a の next とする。 → $a \rightarrow next = d(a \rightarrow next, n)$

(ii) a を返す → return(a)

① D ② E ③ B ④ C ⑤ A

(3-1) $u \rightarrow \boxed{30} \rightarrow \boxed{20} \rightarrow \boxed{10} \rightarrow$ NULL

$w \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow$ NULL

∴ $g(u, w)$

$t = g(u \rightarrow next, w)$

$t = g(u \rightarrow next \rightarrow next, w)$

$t = g(u \rightarrow next \rightarrow next \rightarrow next, w)$

return w

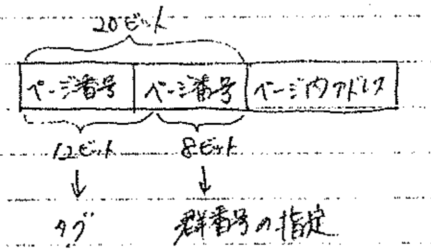
$t \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow$ NULL

return $\rightarrow \boxed{10} \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow$ NULL

return $\rightarrow \boxed{20} \rightarrow \boxed{10} \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow$ NULL

return $\rightarrow \boxed{30} \rightarrow \boxed{20} \rightarrow \boxed{10} \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow$ NULL

(4)



256 (2^8) 列なので、群番号(列番号)は8ビットだから、ディレクトリは12ビットで検索する。

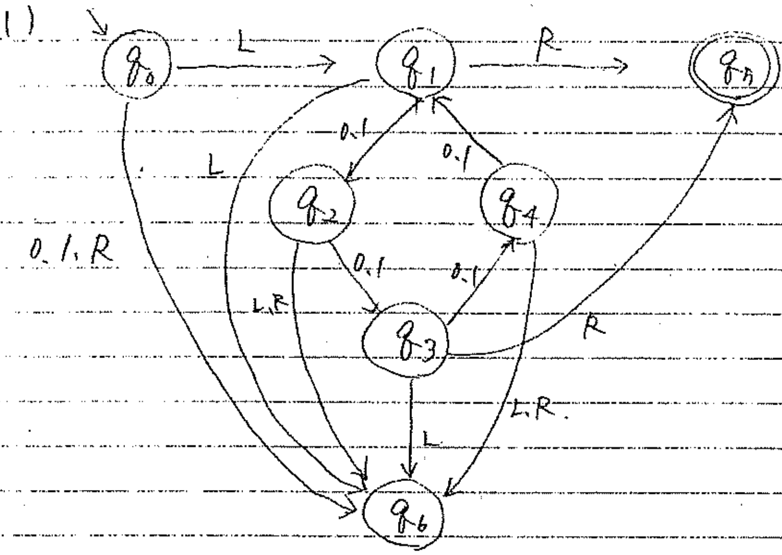
各エントリはタグ12ビットと、バリディティ・ビット1ビットで13ビット。
よって $13 \times 256 \times 8 = 13 \times 2^8 \times 2^3 = 26 \times 2^{10} = 26K$

∴ 26Kビット

H9 [I]

計算論A

[3] (1)



言語T(M1)は、Lで始まり、0か1が偶数個続いてRで終わる言語

正規表現で表すと $L((0+1)(0+1))^*R$

(2) 非到達状態はない。分割を行うと下のようになる。

0						1
q_0	q_1	q_2	q_3	q_4	q_6	q_5
0000	0001	0000	0001	0000	0000	0000

0				1		2
q_0	q_2	q_4	q_6	q_1	q_3	q_5
1000	0110	0110	0000	0002	0002	0000

0	1	2		3		4
q_0	q_6	q_2	q_4	q_1	q_3	q_5
3111	1111	1331	1331	1224	1224	1111

	L	0	1	R
a	d	b	b	b
b	b	b	b	b
c	b	d	d	b
d	b	c	c	e
(e)	b	b	b	b

(3)

	L	0	1	R	
q_0t_0	q_1t_1 (b)	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	a
q_1t_1	q_6t_5 (c)	q_2t_2 (d)	q_2t_2 (d)	q_5t_4 (e)	b
q_6t_5	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	c
q_2t_2	q_6t_5 (c)	q_3t_3 (f)	q_3t_3 (f)	q_6t_5 (c)	d
q_5t_4	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	e
q_3t_3	q_6t_5 (c)	q_4t_1 (g)	q_4t_1 (g)	q_5t_5 (h)	f
q_4t_1	q_6t_5 (c)	q_1t_2 (i)	q_1t_2 (i)	q_6t_4 (j)	g
q_5t_5	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	h
q_1t_2	q_6t_5 (c)	q_2t_3 (k)	q_2t_3 (k)	q_5t_5 (h)	i
q_6t_4	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	q_6t_5 (c)	j
q_2t_3	q_6t_5 (c)	q_3t_1 (l)	q_3t_1 (l)	q_6t_5 (c)	k
q_3t_1	q_6t_5 (c)	q_4t_2 (m)	q_4t_2 (m)	q_5t_4 (e)	l
q_4t_2	q_6t_5 (c)	q_1t_3 (n)	q_1t_3 (n)	q_6t_5 (c)	m
q_1t_3	q_6t_5 (c)	q_2t_1 (o)	q_2t_1 (o)	q_5t_5 (h)	n
q_2t_1	q_6t_5 (c)	q_3t_2 (p)	q_3t_2 (p)	q_6t_4 (j)	o
q_3t_2	q_6t_5 (c)	q_4t_3 (q)	q_4t_3 (q)	q_5t_5 (h)	p
q_4t_3	q_6t_5 (c)	q_1t_1 (b)	q_1t_1 (b)	q_6t_5 (c)	q

$T(M_1) \cup T(M_2)$ を受理する決定性有限オートマトンの状態遷移表は上のようになる。
これの最簡形を求めると、

H9. Ⅱ

計算機アーキテクチャ

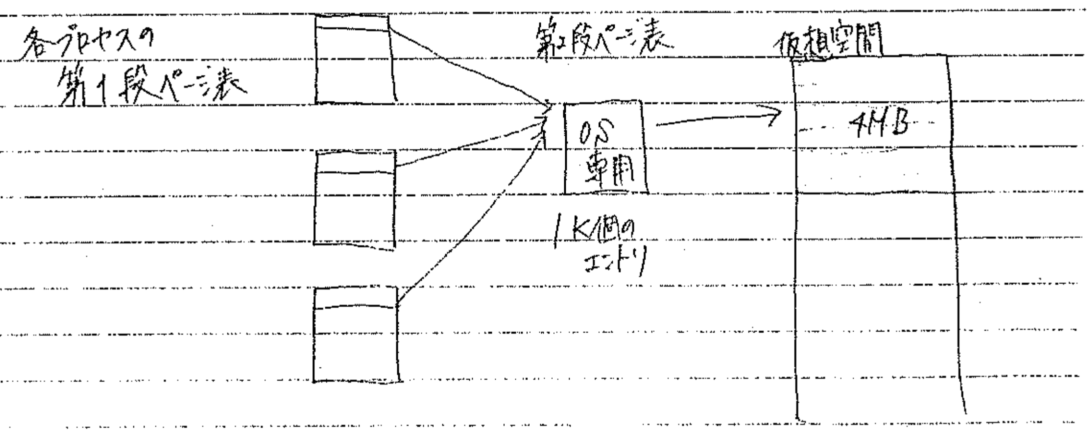
2 (1) 仮想アドレスが32ビットなので、仮想空間は $2^{32} = 4GB$

仮想アドレス32ビットの内、10ビットが第1段目のページ番号で、10ビットが第2段目のページ番号。よって残りの12ビットがページ内アドレス。
ということは、1つのページの大きさは、$2^{12} = 4K$ より 4KB

ページ番号が10ビットなので、ページ表のエントリ数は $2^{10} = 1K$
1つのエントリは4バイトだから、ページ表の大きさは $1K \times 4B = 4KB$

(2) ・プロセスを切り替える際に、ページ表を切り替えるために、オペレーティングシステムはページ表ベースレジスタを書き換える。

(3) 第2段のページ表1つをオペレーティングシステム用として、0番地から4MBを格納し、各プロセスの第1段ページ表の先頭のエントリで、そのオペレーティングシステム用の第2段ページ表のアドレスを表す。

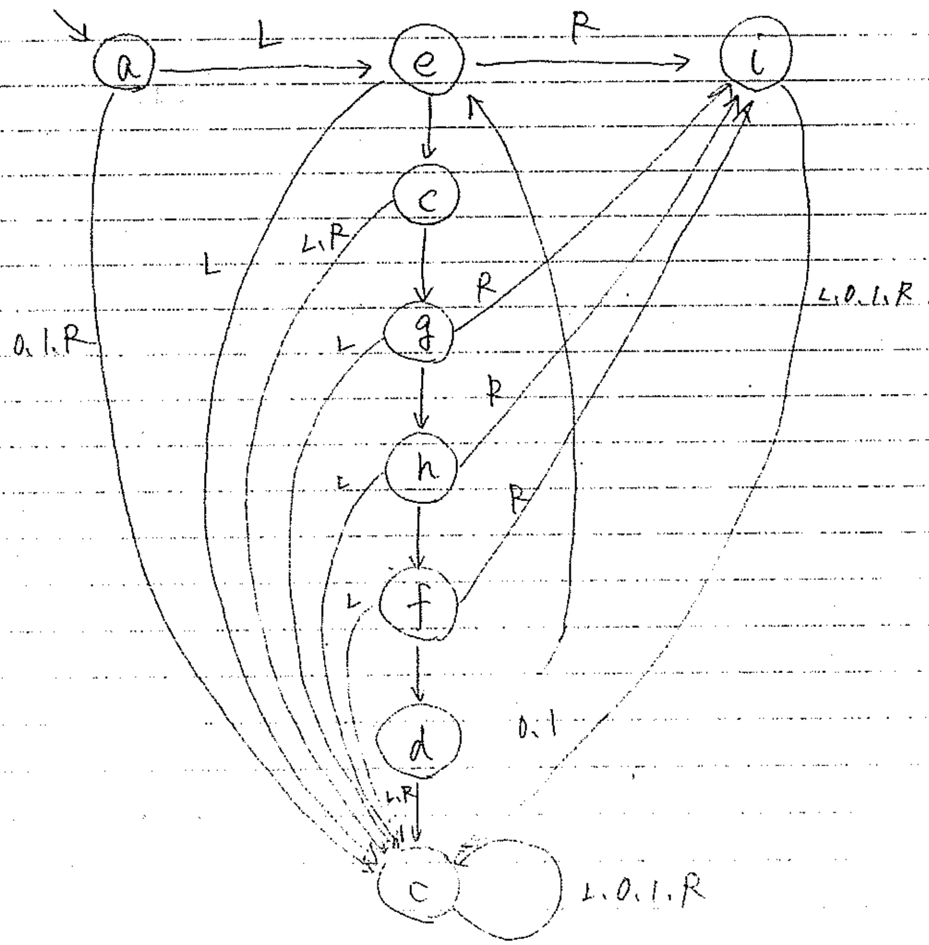


1Kページ × 4KB = 4MB

$\tilde{0}$		LOIR	$\tilde{1}$		LOIR	$\tilde{2}$		LOIR	$\tilde{3}$		LOIR
a	0	0000	a	0	1000	a	0	3111	a	0	4111
b		0001	c		0000	c	1	1111	c	1	1111
c		0000	d		0110	d	2	1441	d	2	1551
d		0000	k		0110	k		1331	m		1551
f		0001	m		0110	m		1441	k	3	1441
g		0001	q		0110	q		1331	q		1441
i		0001	b	1	0002	b	3	1225	b	4	1227
k		0000	f		0112	i		1225	i		1337
l		0001	g		0112	l		1225	l		1227
m		0000	i		0002	p		1225	p		1337
n		0001	l		0002	f	4	1445	f	5	1667
o		0001	n		0112	g		1335	n		1667
p		0001	o		0112	n		1445	g	6	1447
q		0000	p		0002	o		1335	o		1447
e	1	0000	e	2	0000	e	5	1111	e	7	1111
h		0000	h		0000	h		1111	h		1111
j		0000	j		0000	j		1111	j		1111

$\tilde{4}$		LOIR
a	0	4111
c	1	1111
d, m	2	1661
k, q	3	1441
b, l	4	1228
i, p	5	1338
f, n	6	1778
g, o	7	1558
e, h, j	8	1111

	L	0	1	R
a	e	b	b	b
b	b	b	b	b
c	b	g	g	b
d	b	e	e	b
e	b	c	c	i
f	b	d	d	i
g	b	h	h	i
h	b	f	f	i
i	b	b	b	b



H9. Ⅱ

情報解析B

Ⅰ (B1) 隣接する2つのトロッコ t_i, t_{i+1} の重さの和 w_i+w_{i+1} が最小となる i を見つけ結合する。そのコストは w_i+w_{i+1}。結合したトロッコを1つの重さ w_i+w_{i+1} のトロッコとみなして、新しく $t_1, t_2, t_3, \cdots t_{n-1}$ の $n-1$ 台のトロッコを考え、各トロッコの重さは、$w_1(t_1)$, $w_2(t_2)$, $\cdots$, $w_i+w_{i+1}(t_i)$, $w_{i+2}(t_{i+1})$, $\cdots$, $w_n(t_{n-1})$ とする。この問題について同じ事を繰り返す。

(B2) 隣接する2つのトロッコ t_i, t_{i+1} もしくは t_n, t_1 の重さの和 w_i+w_{i+1} もしくは w_n+w_1 が最小となる i を見つけ結合する。そのコストは w_i+w_{i+1} もしくは w_n+w_1。結合したトロッコを1つのトロッコとみなして、新しく $t_1, t_2, \cdots t_{n-1}$ の $n-1$ 台のトロッコを考え、各トロッコの重さは $w_1(t_1)$, $w_2(t_2)$, $\cdots$, $w_i+w_{i+1}(t_i)$, $w_{i+2}(t_{i+1})$, $\cdots$, $w_n(t_{n-1})$ もしくは $w_2(t_1)$, $w_3(t_2)$, $\cdots$, $w_n+w_1(t_{n-1})$ とする。この問題について同じ事を繰り返す。

(別解: B1) トロッコ $t_1, t_2, \cdots, t_n$ を最小コストで連結する問題を P_{1n}。
その部分問題として、$t_i, t_{i+1}, \cdots, t_j$ を連結する問題を P_{ij} とする。
P_{ij} の最適値を H_{ij} とすると

$$H_{ij} = \mathrm{Min}\,(H_{ik} + H_{k+1,j} + w(t_i) + w(t_{i+1}) + \cdots + w(t_j))$$

∴ 小さい問題から H_{ij} をもとめていくと、n の2乗個の H_{ij} を求めれば H_{1n} が求まる。各 H_{ij} を求める手間は n。
よって n の3乗時間でもとまる。

(A3) xにスコーレム関数 a を代入すると

$$
\begin{aligned}
\forall y\ (\ & (\neg A(y) \vee \neg D(y)) \\
\wedge\ & (\neg B(y) \vee \neg D(y)) \\
\wedge\ & (B(y) \vee D(y)) \\
\wedge\ & (\neg A(m(y)) \vee \neg C(y) \vee \neg B(m(y))) \\
\wedge\ & (\neg D(m(y)) \vee \neg C(y) \vee \neg B(m(y))) \\
\wedge\ & C(a) \\
\wedge\ & A(m(a))\ )
\end{aligned}
$$

母式の節集合は

$\neg A(y) \vee \neg D(y)$ (1)

$\neg B(y) \vee \neg D(y)$ (2)

$B(y) \vee D(y)$ (3)

$\neg A(m(y)) \vee \neg C(y) \vee \neg B(m(y))$ (4)

$\neg D(m(y)) \vee \neg C(y) \vee \neg B(m(y))$ (5)

$C(a)$ (6)

$A(m(a))$ (7)

(4)で $y=a$ とおいた式と(6)との導出節は $\neg A(m(a)) \vee \neg B(m(a))$ (8)

(8)と(7)の導出節は $\neg B(m(a))$ (9)

(3)で $y=m(a)$ とおいた式と(9)との導出節は $D(m(a))$ (10)

(1)で $y=m(a)$ とおいた式と(10)との導出節は $\neg A(m(a))$ (11)

(7)と(11)の導出節は 空節 □ である。

∴ 空節が導びけたので (A2) の論理式 $\neg R$ は充足不能である。

H9 [I]

論理設計

[4] (1)

$$
\begin{aligned}
y =\ & \overline{c2}\cdot\overline{c1}\cdot\overline{c0}\cdot x0 \vee \overline{c2}\cdot\overline{c1}\cdot c0\cdot x1 \vee \overline{c2}\cdot c1\cdot\overline{c0}\cdot x2 \vee \overline{c2}\cdot c1\cdot c0\cdot x3 \\
&\vee c2\cdot\overline{c1}\cdot\overline{c0}\cdot x4 \vee c2\cdot\overline{c1}\cdot c0\cdot x5 \vee c2\cdot c1\cdot\overline{c0}\cdot x6 \vee c2\cdot c1\cdot c0\cdot x7
\end{aligned}
$$

(2-1) 状態遷移表

現在の状態	次の状態		出力(y)	
/入力(x)	0	1	0	1
S0	S3	S1	0	1
S1	S0	S2	1	1
S2	S1	S2	1	1
S3	S4	S0	0	0
S4	S4	S3	0	0

(2-2) Dフリップフロップ (DFF-0, DFF-1, DFF-2) を用いると状態遷移表の次のようになる

現在の状態	次の状態		出力(y)	
/入力(x)	0	1	0	1
000	001	100	0	1
100	000	110	1	1
110	100	110	1	1
001	011	000	0	0
011	011	001	0	0

状態は (Q_2, Q_1, Q_0)

Dフリップフロップの励起表は次のようになるので、

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

D0、D1、D2 は下のようになる。

(Q_2, Q_1, Q_0)	(D2, D1, D0)		出力(y)	
/入力(x)	0	1	0	1
000	001	100	0	1
100	000	110	1	1
110	100	110	1	1
001	011	000	0	0
011	011	001	0	0

D_2

Q_2Q_1 \ Q_0x	00	01	11	10
00	0	1	0	0
01	d	d	0	0
11	1	1	d	d
10	0	1	d	d

$D_2 = \bar{Q}_0 x \vee Q_1 \bar{Q}_0$

(Q_2Q_1)

D_1

Q_2Q_1 \ Q_0x	00	01	11	10
00	0	0	0	1
01	d	d	0	1
11	0	1	d	d
10	0	1	d	d

$D_1 = Q_0 \bar{x} \vee Q_2 x$

D_0

Q_2Q_1 \ Q_0x	00	01	11	10
00	1	0	0	1
01	d	d	1	1
11	0	0	d	d
10	0	0	d	d

$D_0 = \bar{Q}_2 \bar{x} \vee \bar{Q}_2 Q_1$

(Q_1Q_0)

y

Q_2Q_1 \ Q_0x	00	01	11	10
00	0	1	0	0
01	d	d	0	0
11	1	1	d	d
10	1	1	d	d

$y = \bar{Q}_0 x \vee Q_2$

(2-3).

(C_2, C_1, C_0)	Q_2	Q_1	Q_0	D_2	D_1	D_0	y	出力
0 0 0	0	0	0	x	0	$\bar{x}$	x	x_0
0 0 1	0	0	1	0	$\bar{x}$	$\bar{x}$	0	x_1
0 1 0	0	1	0	1	0	1	x	x_2
0 1 1	0	1	1	0	$\bar{x}$	1	0	x_3
1 0 0	1	0	0	x	x	0	1	x_4
1 0 1	1	0	1	0	1	0	1	x_5
1 1 0	1	1	0	1	x	0	1	x_6
1 1 1	1	1	1	0	1	0	1	x_7

∴ マルチプレクサの入力信号は、

入力端子	MUX8-0	MUX8-1	MUX8-2	MUX8-3
x_0	$\bar{x}$	0	x	x
x_1	$\bar{x}$	$\bar{x}$	0	0
x_2	1	0	1	x
x_3	1	$\bar{x}$	0	0
x_4	0	x	x	1
x_5	0	1	0	1
x_6	0	x	1	1
x_7	0	1	0	1

H9. [II]

情報論理学

[I] (A1)

$S_1 : \forall x((A(x) \vee B(x)) \to \neg D(x))$

$S_2 : \forall x(\neg B(x) \to D(x))$

$S_3 : \forall x((A(m(x)) \vee D(m(x))) \to \neg(C(x) \wedge B(m(x))))$

$\alpha : \forall x(C(x) \to \neg A(m(x)))$

∴

$$\forall x((A(x) \vee B(x)) \to \neg D(x))$$
$$\wedge\ \forall x(\neg B(x) \to D(x))$$
$$\wedge\ \forall x((A(m(x)) \vee D(m(x))) \to \neg(C(x) \wedge B(m(x))))$$
$$\to \forall x(C(x) \to \neg A(m(x)))$$

(A2) $\neg R$

$$= \forall x((A(x) \vee B(x)) \to \neg D(x)) \wedge \forall x(\neg B(x) \to D(x))$$
$$\wedge\ \forall x((A(m(x)) \vee D(m(x))) \to \neg(C(x) \wedge B(m(x))))$$
$$\wedge\ \neg\forall x(C(x) \to \neg A(m(x)))$$
$$= \forall x(\neg(A(x) \vee B(x)) \vee \neg D(x)) \wedge \forall x(B(x) \vee D(x))$$
$$\wedge\ \forall x(\neg(A(m(x)) \vee D(m(x))) \vee \neg(C(x) \wedge B(m(x))))$$
$$\wedge\ \neg\forall x(\neg C(x) \vee \neg A(m(x)))$$
$$= \forall x((\neg A(x) \wedge \neg B(x)) \vee \neg D(x)) \wedge \forall x(B(x) \vee D(x))$$
$$\wedge\ \forall x((\neg A(m(x)) \wedge \neg D(m(x))) \vee \neg C(x) \vee \neg B(m(x)))$$
$$\wedge\ \exists x \neg(\neg C(x) \vee \neg A(m(x)))$$
$$= \forall x((\neg A(x) \vee \neg D(x)) \wedge (\neg B(x) \vee \neg D(x)))$$
$$\wedge\ \forall x(B(x) \vee D(x))$$
$$\wedge\ \forall x((\neg A(m(x)) \vee \neg C(x) \vee \neg B(m(x))) \wedge (\neg D(m(x)) \vee \neg C(x) \vee \neg B(m($$
$$\wedge\ \exists x(C(x) \wedge A(m(x)))$$
$$= \exists x \forall y(\ (\neg A(y) \vee \neg D(y))$$
$$\wedge\ (\neg B(y) \vee \neg D(y))$$
$$\wedge\ (B(y) \vee D(y))$$
$$\wedge\ (\neg A(m(y)) \vee \neg C(y) \vee \neg B(m(y)))$$
$$\wedge\ (\neg D(m(y)) \vee \neg C(y) \vee \neg B(m(y)))$$
$$\wedge\ C(x)$$
$$\wedge\ A(m(x)))$$