1. アルゴリズムとプログラミング

(1)

（ア）$\lfloor \log_2 n \rfloor$

（イ）2^d

（ウ）1

（エ）2^d

（オ）$2^{d+1} - 1$

(2)

(2-1)

data[1] = 8

data[2] = 5

data[3] = 7

data[4] = 3

data[5] = 4

data[6] = 6

data[7] = 1

data[8] = 2

(2-2)

接点$\frac{n}{2}$から接点1まで降順に、自身を根とする部分木の最も深い葉のいずれかに辿りつくまで値の入れ替えを繰り返して移動する。

(2-3)

$n = 2^m - 1$とする．このとき，深さ$d = \log_2(n+1)$となる．深さjの接点は葉にたどり着くまでd−j回移動する．また，深さjには2^{j-1}個の接点がある．よって深さjの接点の総移動回数は$2^{j-1} * (d-j)$となる．

よって木全体の移動回数は

$$T(n) = \sum_{k=1}^{d-1} (d-k) * 2^{k-1}$$

となる．

これを計算すると$(d-1) * 2^d - (d-1)$となり，これに$d = \log_2 n + 1$を代入すると，$(\log_2(n+1) - 1) * 2^{\log_2(n+1)} - (\log_2(n+1) - 1) = (n+1) * (\log_2(n+1) - 1) - (\log_2(n+1) - 1)$となるのでオーダー表記で$O(n * \log n)$と表せる．

2. 論理回路

(1)

(1-1)

$g=(x_0+\overline{x_1})\oplus(\overline{x_2}+x_3)$

$\quad=\overline{(x_0+\overline{x_1})}\cdot(\overline{x_2}+x_3)+(x_0+\overline{x_1})\cdot\overline{(\overline{x_2}+x_3)}$

$\quad=\overline{x_0}\cdot x_1\cdot(\overline{x_2}+x_3)+x_2\cdot\overline{x_3}\cdot(x_0+\overline{x_1})$

$\quad=\overline{x_0}\cdot x_1\cdot\overline{x_2}+x_0\cdot x_1\cdot x_3+x_0\cdot x_2\cdot\overline{x_3}+\overline{x_1}\cdot x_2\cdot\overline{x_3}$

(1-2)

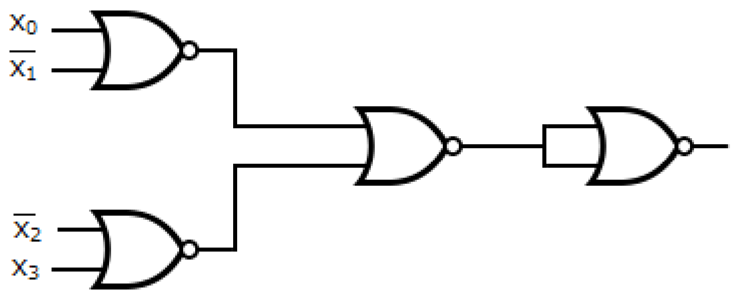
$f=x_1\cdot x_2\cdot(\overline{x_0}+\overline{x_3})+\overline{x_0}\cdot\left(\overline{x_1}\cdot x_2+x_1\cdot\overline{x_2}\right)\cdot\overline{x_3}+\overline{x_0}\cdot x_1\cdot\overline{x_2}+\overline{x_1}\cdot x_2\cdot\overline{x_3}$

$\quad=x_1\cdot x_2\cdot\overline{x_0}+x_1\cdot x_2\cdot\overline{x_3}+\overline{x_0}\cdot\overline{x_1}\cdot x_2\cdot\overline{x_3}+\overline{x_0}\cdot x_1\cdot\overline{x_2}\cdot\overline{x_3}+\overline{x_0}\cdot x_1\cdot\overline{x_2}+\overline{x_1}\cdot x_2\cdot\overline{x_3}$

$\quad=\overline{x_0}\cdot x_1+x_2\cdot\overline{x_3}$

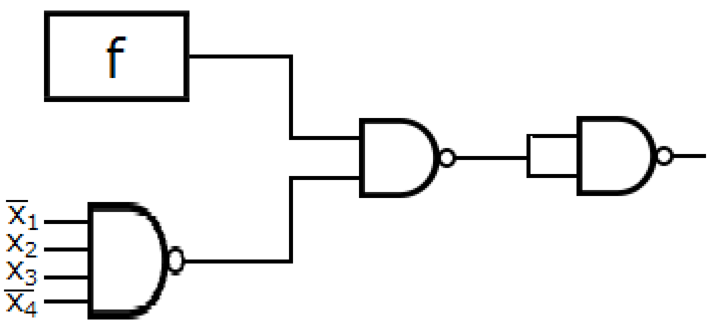
$\quad=\overline{(x_0+\overline{x_1})\cdot(\overline{x_2}+x_3)}$

$\quad=\overline{(x_0+\overline{x_1})}+\overline{(\overline{x_2}+x_3)}$



(1-3)

$g=f\cdot\overline{\overline{x_0}\cdot x_1\cdot x_2\cdot\overline{x_3}}$



(2-1)

(0, 0, 0)→(0, 0, 1)→(0, 1, 1)→(0, 1, 0)→(1, 1, 0)→(1, 0, 0)→(0, 0, 0)

(2-2)

状態遷移図

O_2	O_1	O_0	I_2	I_1	I_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	1	1	1
1	1	1	1	1	0
1	1	0	1	0	0
1	0	0	0	0	0

$I_2 = O_1$

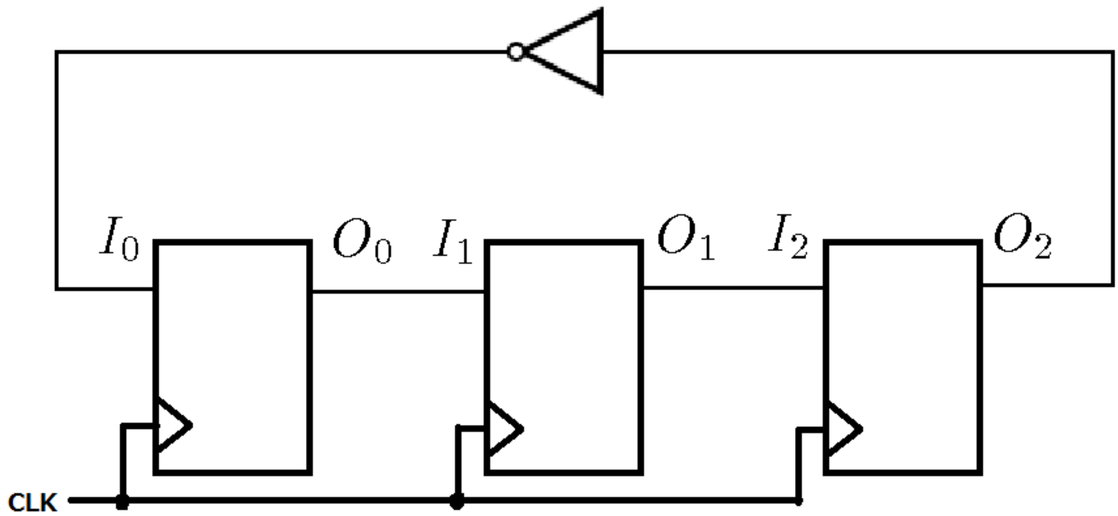
O_2O_1 \ O_0	0	1
00	0	0
01	d	1
11	1	1
10	0	d

$I_1 = O_0$

O_2O_1 \ O_0	0	1
00	0	1
01	d	1
11	0	1
10	0	d

$I_0 = \overline{O_2}$

O_2O_1 \ O_0	0	1
00	1	1
01	d	1
11	0	0
10	0	d



(2-3)

O_2	O_1	O_0	Q_2	Q_1	Q_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	1	0	1	1
1	1	1	0	1	0
1	1	0	1	1	0
1	0	0	1	0	0

$$Q_2 = O_2 \cdot \overline{O_0}$$

O_2O_1 \ O_0	0	1
00	0	0
01	d	0
11	1	0
10	1	d

$$Q_1 = O_1$$

O_2O_1 \ O_0	0	1
00	0	0
01	d	1
11	1	1
10	0	d

$$Q_0 = \overline{O_2} \cdot O_0$$

O_2O_1 \ O_0	0	1
00	0	1
01	d	1
11	0	0
10	0	d

追加するゲートは NOT2 個と AND2 個の 4 つ

3.計算機システムとシステムプログラム

(1)

(1-1)

(a)4　(b)10　(c)2　(d)7　(e)6

(1-2)

(1-2-1)

τは，各機能ブロックの遅延時間のうち，最も大きいもの以上にする必要がある.

(1-2-2)

5τ

(1-2-3)

(m+4)τ

(1-3)

		クロックサイクル															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
命令	1	IF	ID	OF	EX	MW											
	2		IF	ID	OF	EX	MW										
	3			(s)	(s)	IF	ID	OF	EX	MW							
	4						IF	ID	OF	EX	MW						
	5							(s)	IF	ID	(d)	OF	EX	MW			
	6									IF	ID	(s)	(d)	(d)	OF	EX	MW

(2)

(2-1)

(a)5　(b)1　(c)4　(d)3

(2-2)

(2-2-1)

・ファイルシステムA: 160MB/8KB = 20Kブロック

・ファイルシステムB: 160MB/40MB = 4Kブロック

組み合わせ	1 ファイル毎のブロック数	最大ファイル数	データサイズ	使用効率
A-a	1	20000	40.0MB	25.0
A-b	63	317	158.5MB	99.1
B-a	1	4000	8MB	5.0
B-b	13	307	153.5MB	95.9

(2-2-2)

平均アクセス時間 = 平均待ち時間 + データ転送時間

= 平均シーク時間 + 平均回転遅延時間 + データ転送時間

平均シーク時間 = 5.0msec

平均回転遅延時間 = (1/2)*(1 回転にかかる時間)

= (1/2)*(60*1000/6000)

= 5.0msec

ファイルシステム A: 10KB/8KB = 1.25 ブロック

データ転送時間は,

データ転送時間 = (8KB/80KB)*(1 回転にかかる時間)

= (1/10)*(60*1000/6000)

= 1.0msec

よって, 1 ブロックへのアクセスに 11msec かかる.

全体では 11msec*1.25K = 1375msec

ファイルシステム B: 10KB/40KB = 0.25 ブロック

データ転送時間は,

データ転送時間 = (40KB/80KB)*(1 回転にかかる時間)

= (1/2)*(60*1000/6000)

= 5.0msec

よって, 1 ブロックへのアクセスに 15msec かかる.

全体では 15msec*0.25K = 375msec

(2-2-3)

・ファイルサイズとブロックサイズの関係

ファイルサイズに対してブロックサイズが大きい場合, ディスクの中に使用されない領域が増え, 使用効率は悪くなる.

・ブロックサイズとシーク回数

ブロックサイズが大きいほど同じサイズのファイルにアクセスする場合のシーク回数が減少するため, 平均転送時間は短くなる.

8. 情報論理学

(1)

(1-1) x_{117}: 偽

x_{221}: 偽

x_{135}: 真

(1-2) n^4

(1-3) $\mathrm{A(i,j)} = \bigvee_{1\le k\le n^2} x_{ijk}$

(1-4) $\mathrm{A} = \bigwedge_{1\le i\le n} \bigwedge_{1\le j\le n} A(i,j)$

(1-5) $\mathrm{B} = \bigwedge_{1\le i\le n} \bigwedge_{1\le j\le n} \bigwedge_{1\le k<h\le n^2} (\overline{x_{ijk}} \vee \overline{x_{ijh}})$

(1-6) $\mathrm{C} = \bigvee_{1\le i\le n} \bigvee_{1\le j\le n} x_{ij1}$

(1-7)

(1-7-1) $\mathrm{NP}(2,3,5) = x_{135} \vee x_{335} \vee x_{225} \vee x_{245}$

(1-7-2) $D_{ij}^{k} = \overline{x_{ijk}} \vee NP(i,j,k+1)$

(1-7-3) $\mathrm{D} = \bigwedge_{1\le k<n^2} \bigwedge_{1\le i\le n} \bigwedge_{1\le j\le n} D_{ij}^{k}$

(1-8)

(1-8-1) 充足不能

(1-8-2)

x_{111}と D よりx_{122}が真である．また，x_{122}と D よりx_{223}が真である．

ここで，x_{123}，D より，隣に 4 があるはずであるが，x_{111}，x_{223}より偽．

(2)

(2-1)○　(2-2)×　(2-3)○　(2-4)×　(2-5)×　(2-6)○

9. 計算理論

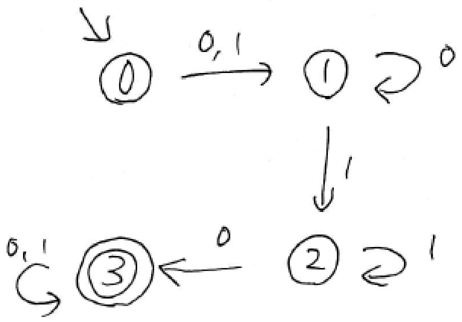
(1)

0							1	
p	r	s	t	u	w	x	q	v
00	10	00	00	10	00	00	11	11

0					1		2	
p	s	t	w	x	r	u	q	v
00	01	01	01	01	21	21	22	22

0	1				2		3	
p	s	t	w	x	r	u	q	v
11	12	12	12	12	32	32	33	33

従って,



(2)

p = 0q + 1r

q = 0p + 1q

r = 0r + 1s + ε

s = 0r + 1p

X = AX + B ならば$X = A^*B$より,

$p = (01^*0)^*1r$　・・・(1)

$q = 1^*0(01^*0)^*1r$　・・・(2)

$s = \{0 + 1(01^*0)^*1\}r$　・・・(3)

$r = \{0 + (0 + 11(01^*0)^*1\}^*$　・・・(4)

入力記号の長さを n とすると,

n=2

(2)より，　01 → ×

n=3

(2)より，101，010 → ×

n=4

0 で始まるなら，(1)より,

0101，0010 → ×

1で始まるなら，(3)より

1001，1100，1110 → ×

n=5

0で始まるなら，(1)より，

00100 → ×

00110 → ×

01101 → ×

01010 → ○

01010はすべての式を満たす．

(3)

p → q：初期状態からの遷移

q → q：入力記号2が与えられる前

q → r：入力記号2が与えられたとき

r → r：入力記号2が与えられた後

(a) (1, Z)/1　(b) (2, 0)/0　(c) (2, 1)/1　(d) (0, 0)/ε　(e) (1, 1)/ε

(4)

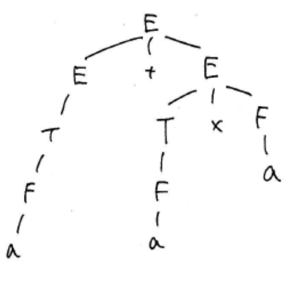
(4-1)

文法G_1：E → E + T | T，　T → T × F | F，　F → (E) | aより，

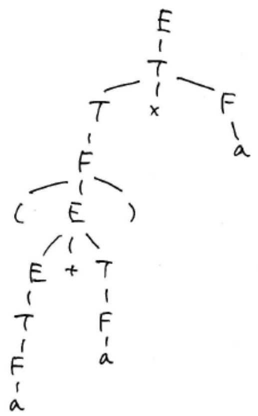
a+a，a×a，(a)

(4-2)

(a)　a + a × a　の導出木



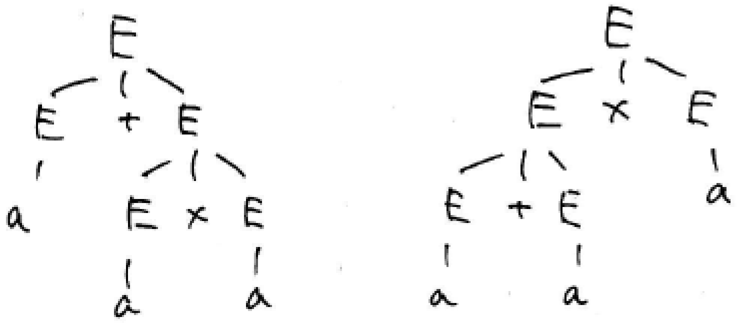
(b)　(a + a) × a　の導出木



(4-3)

文法G_2：E → E + E | E × E | (E) | a

G_2による語 a+a×a の導出木が次の 2 通り存在するため，G_2はあいまいである．



10. 情報理論

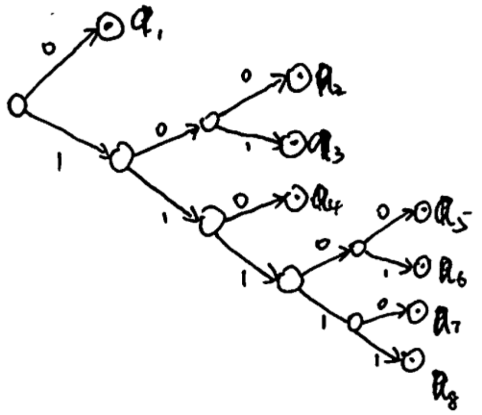
(1)

(1-1)

$$H_2(S) = -\sum_{i=1}^{8} p_i \log_2 p_i = -\left(\frac{1}{2}\log_2\frac{1}{2} + 3\left(\frac{1}{8}\log_2\frac{1}{8}\right) + 4\left(\frac{1}{32}\log_2\frac{1}{32}\right)\right) = \frac{9}{4}$$

(1-2)

$B_2 = \{0, 1\}$



a_1 : 0
a_2 : 100
a_3 : 101
a_4 : 110
a_5 : 11100
a_6 : 11101
a_7 : 11110
a_8 : 11111

(1-3)

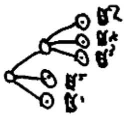
ア. 20　イ. 21　ウ. 32　エ. 33

(1-4)

オ. 6　カ. 17　キ. 24　ク. 16

(2)

(2-1)

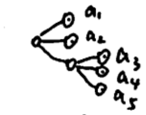


$$p_1 = \frac{1}{3}, \quad p_2 = \frac{1}{3}, \quad p_3 = \frac{1}{9}, \quad p_4 = \frac{1}{9}, \quad p_5 = \frac{1}{9}$$

$$H_2(S) = -\sum_{i=1}^{5} p_i \log_2 p_i = -\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} - 3\left(\frac{1}{9}\log_2\frac{1}{9}\right) = \frac{4}{3}\log_2 3$$

$$(\log_2 3)\,\overline{l_3} = (\log_2 3)(p_1 + p_2 + 2p_3 + 2p_4 + 2p_5) = \frac{4}{3}\log_2 3$$

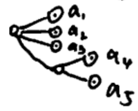
$$\therefore H_2(S) = (\log_2 3)\overline{l_3}$$



(2-2)

$\overline{l_3} = p_1 + p_2 + 2p_3 + 2p_4 + 2p_5$　･･･（1）

$\overline{l_4} = p_1 + p_2 + 2p_3 + 2p_4 + 2p_5$　･･･（2）



条件

$(\log_2 3)\overline{l_3} = (\log_2 4)\overline{l_4}$　･･･（3）

$p_1 > p_2 > p_3 > p_4 > p_5$　･･･（4）

$p_1 + p_2 + p_3 + p_4 + p_5 = 1$　･･･（5）

$p_4 + p_5 = \frac{1}{100}$　･･･（6）

（1）,（2）→（3）:

$(\log_2 3)\,(p_1 + p_2 + 2p_3 + 2p_4 + 2p_5) = (\log_2 4)(p_1 + p_2 + p_3 + 2p_4 + 2p_5)$　･･･（7）

（5）,（6）→（7）:

$\left(\frac{101}{100} + p_3\right)\log_2 3 = \frac{101}{100}(\log_2 4)$

$$p_3 = \frac{101}{50\log_2 3} - \frac{101}{100}$$

11. ネットワーク

(1)

(a) 17 ホスト (b) 11 ノード (c) 4　回線 (d) 19 ルータ　(e) 14 ブリッジ

(f) 18　リピータ (g) 9　帯域幅

(2)

(2-1)

ダイクストラのアルゴリズム

(2-2)

	N	d(B),p(B)	d(C),p(C)	d(D),p(D)	d(E),p(E)	d(F),p(F)
初期状態	{A}	3,A	5,A	2,A	∞,φ	∞,φ
1	{A,D}	3,A	4,D		6,D	∞,φ
2	{A,B,D}		4,D		6,D	∞,φ
3	{A,B,C,D}				5,C	7,C
4	{A,B,C,D,E}					7,C
5	{A,B,C,D,E,F}					

(2-3)

B:　A>B　　　　コスト 3

C:　A>D>C　　　コスト 4

D:　A>D　　　　コスト 2

E:　A>D>E　　　コスト 5

F:　A>D>C>F　　コスト 7

(2-4)

プロトコルの名称

Open Shortest Path First

各ノードの動作

制御パケットを送ることで隣接ノードのアドレスを取得し, 隣接するノードに対し制御パケットを送出し，その返事を返してもらうことでラウンドトリップ時間を計算し，2 で割ることでコストをだす.

送信元ノードアドレスや順序番号, 隣接ノードへの遅延などの情報からなるパケットを組み立て全てのノードに送る.

隣接するノードとのコストなどの情報が手に入ったのでダイクストラのアルゴリズムが実行出来る.

(2-5)

静的ルーティングが人間の手によってルーティングテーブルを設定するのに対して, 動的ルーティングはルータ同士が経路情報を自動で収集して, 経路を決定する.

動的ルーティングの利点

経路に障害が発生したときに, 自動的に別の迂回経路に移行できる.

ネットワークやトラフィックの状態によって, 常に最適なルートを選択できる.

動的ルーティングの欠点

情報交換のための処理やトラフィックが発生する.

機器障害などで, 誤った経路情報が伝達すると広範囲に渡り通信不能に陥る可能性がある.