1. アルゴリズムとプログラミング （作成者：山口）

(1)

（ア）i-- （イ）data[MAX-i]

(2)

(2-1)

{1, 'b'}, {2, 'j'}, {3, 'c'}, {4, 'h'}, {5, 'a'}, {6, 'e'}, {7, 'g'}, {8, 'f'}

(2-2)

（a）6 回 （b）13 回 （c）7 回 （d）16 回
　∗ （d）について，関数 sort は，data[j-1].key = data[j].key のときも入れ替え操作が起きることに注意する．

(2-3)

（エ） $a_i \geq a_j$ （オ）k

(3)

　図 1 のプログラムの 15 行目において，data[j-1].key >= data[j].key と書かれている．これは，key の値が同じ要素であっても入れ替え操作が行われることを意味するため，安定ではない．

2. 計算機システムとシステムプログラム （作成者：柏原・秦野）

(1)

(1-1)

群番号 セット数 2（$=2^1$）だから 1 ビット

ブロック内アドレス ブロックサイズは 4 バイトでアドレスはバイト単位なので，$4=2^2$ で 2 ビット

群内ブロック番号 アドレス長 8 ビットなので 8-1-2=5 ビット

アドレス（8 ビット） 群内ブロック番号（5 ビット） + 群番号（1 ビット） + ブロック内アドレス（2 ビット）

実行順	アドレス			群内ブロック番号	群番号
①	01011	0	11	11	0
②	01001	0	10	9	0
③	01001	0	01	9	0
④	00010	1	01	2	1
⑤	01101	0	01	13	0
⑥	00010	1	00	2	1
⑦	01101	0	01	13	0
⑧	01001	0	10	9	0
⑨	01011	0	10	11	0
⑩	00010	1	01	2	1

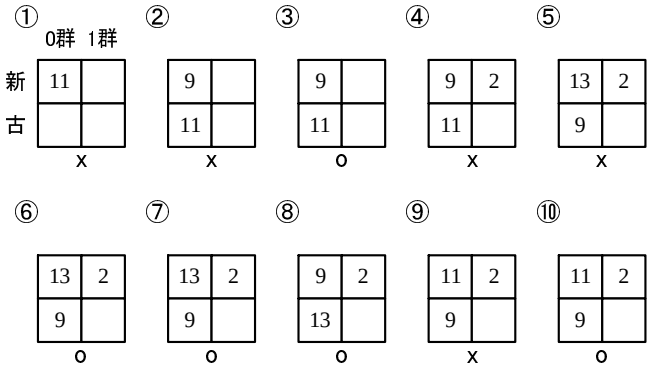


図 1: キャッシュメモリの内容

表 1: f がキャッシュミス

実行順	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
初期参照	f	f		f	f					
ブロック置き換え					f			f		

(1-1-1)

4 回

(1-1-2)

2 回

(1-1-3)

5/10=0.5 より 50 %

(1-2)

空間的局所性と時間的局所性

空間的局所性により，あるデータとその近くに位置するデータが 1 つのブロック内に存在することで，同一ブロック内のアクセスが増え，キャッシュのヒット率が上昇するためメモリアクセスの時間が小さくなる．また，時間的局所性により，最近参照したデータを含むブロックがキャッシュに存在する確率が高くなり，キャッシュのヒット率が上昇するためメモリアクセスの時間が小さくなる．

(1-3)

ブロックサイズが小さいと，ブロック内に参照したいデータが存在しない確率が高くなり，主記憶の参照が増加するため平均メモリアクセス時間が大きくなる．ブロックサイズが大きいと，キャッシュに保存できるブロック数が少なくなり，ブロック置き換えの回数が増加するため平均メモリアクセス時間が大きくなる．

(2)

(2-1)

(a)（サ）　(b)（ク）　(c)（オ）　(d)（キ）
(e)（コ）　(f)（イ）　(g)（ス）

(2-2)

(2-2-1)

到着順 $(20 + 52 + 52 + 76) / 4 = 50$
処理時間順 $(20 + 92 + 12 + 36) / 4 = 40$

(2-2-2)

(a) $(40 + 90 + 36 + 76) / 4 = 60.5$
(b) $(52 + 86 + 44 + 76) / 4 = 64.5$

(2-2-3)

タイムスライスの値が小さいとプロセス切り替えが頻繁に発生し，その処理にかかるオーバーヘッドが大きくなるため，平均ターンアラウンド時間が大きくなる．また，タイムスライスの値が大きいと処理時間の短いプロセスが長く待たされることになり，平均ターンアラウンド時間が大きくなる．

3. 離散構造 （作成者：オドフー）

(1)

（a）恒真
（b）恒真ではないが充足可能

真にする解釈　$p(a) = true,\ p(b) = true$
偽にする解釈　$p(a) = true,\ p(b) = false$

（c）充足不能
（d）恒真ではないが充足可能

真にする解釈　$g(a,b) = true,\ g(b,a) = true,\ g(a,a) = true,\ g(b,b) = true$
偽にする解釈　$g(a,b) = true,\ g(b,a) = true,\ g(a,a) = false,\ g(b,b) = false$

(2)

(2-1)

$E = (A \wedge B \wedge C) \Rightarrow D$ より $E = \neg(A \wedge B \wedge C) \vee D$ であるから

$$\neg E = (A \wedge B \wedge C \wedge \neg D)$$

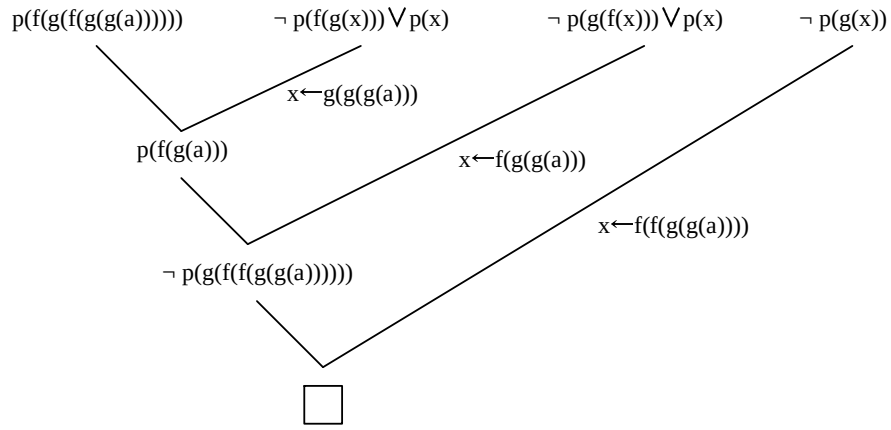
また，各論理式の冠頭標準形は以下のようになる．
$A = \forall x\{p(f(g(f(g(g(a))))))\}$
$B = \forall x\{\neg p(f(g(x))) \vee p(x)\}$
$C = \forall x\{\neg p(g(f(x))) \vee p(x)\}$
$\neg D = \forall x\{\neg p(g(x))\}$
以上より

$\neg E = \forall x\{[p(f(g(f(g(g(a))))))] \wedge [\neg p(f(g(x))) \vee p(x)] \wedge [\neg p(g(f(x))) \vee g(x)] \wedge [\neg p(g(x))]\}$

となり，これは限量子を含まないため E' である．

(2-2)

図に示す．



(3)

反射律　任意の $t \in X \cup Y$ について，$(t,t) \in R$ であるので反射律を満たす.

反対称律　命題 $\forall x, y \in X \cup Y\{(x,y) \in R \wedge (y,x) \in R \Rightarrow x = y\}$ の真偽を以下のように調べる.

(i) $x \in X, y \in Y$ のとき

$(y,x) \notin R$ であるので命題は真.

(ii) $x \in Y, y \in X$ のとき

$(x,y) \notin R$ であるので命題は真.

(iii) $x \in X, y \in X$ のとき

$(x,y) \in R \wedge (y,x) \in R$ が成立するのは $x = y$ のときのみであるので命題は真.

(iv) $x \in Y, y \in Y$ のとき

(iii) と同様.

以上より命題 $\forall x, y \in X \cup Y\{(x,y) \in R \wedge (y,x) \in R \Rightarrow x = y\}$ は真であるので，反対称律を満たす.

推移律　命題 $\forall x, y \in X \cup Y\{(x,y) \in R \wedge (y,z) \in R \Rightarrow (x, z \in R)$ の真偽を調べる．$(x,y) \in R \wedge (y,z) \in R$ が成立するのは以下の場合のみである.

(i) $x = y \in X, z \in Y$ のとき

$(x,z) \in R$ であるので命題は真.

(ii) $x = \in X, y = z \in Y$ のとき

$(x,z) \in R$ であるので命題は真.

以上より命題 $\forall x, y \in X \cup Y\{(x,y) \in R \wedge (y,z) \in R \Rightarrow (x,z) \in R$ は真であるので，推移律を満たす.

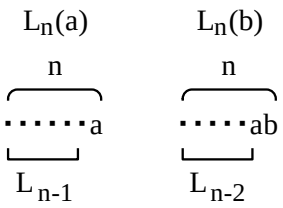
したがって，R は順序関係である.

(4)

(4-1)

右図より

$$\begin{aligned} |L_n(a)| &= |L_{n-1}| \\ |L_n(b)| &= |L_{n-2}| \end{aligned}$$



(4-2)

(4-1) より

$$L_n = L_n(a) \cup L_n(b)$$

また，$L_n(a) \cap L_n(b) = \phi$ であるから $n > 2$ のとき

$$|L_n| = |L_n(a)| + |L_n(b)| = |L_{n-1}| + |L_{n-2}|$$

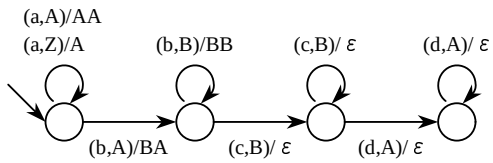
さらに，$L_1 = \{a, b\}$，$L_2 = \{aa, ab, ba\}$ であるので以上より

$$|L_n| = \begin{cases} 2 & (n = 1) \\ 3 & (n = 2) \\ |L_{n-1}| + |L_{n-2}| & (n > 2) \end{cases}$$

4. 計算理論 （作成者：秦野）

(1)

L_1 ×
L_2 ◯ 例：$S \to aSa|bSb|\epsilon$
L_3 ×
L_4 ◯ 例：空スタック受理の PDA



L_5 ◯ 例：$S \to aAbcBd$, $A \to aAb|\epsilon$, $B \to cBd|\epsilon$
L_6 ×
∗ 文脈自由言語でないものは繰り返し定理によりその証明が可能.

(2)

$E \to E + T$, $E \to T + T$, $E \to (E)$, $E \to a$, $E \to b$,
$E \to Ia$, $E \to Ib$, $E \to T * F$, $E \to F * F$

(3)

文脈自由文法は，正規文法より表現できる言語が多いため，プログラマが使用できる記述方法が多くなる．また，文脈依存文法より表現できる言語を制限しているため，コンパイラがプログラムの構文解析を行う処理が単純になる.

(4)

(4-1)

（ア）b （イ）f （ウ）f （エ）e （オ）f （カ）e （キ）e （ク）b

(4-2)

（サ）(0,A)/AA （シ）(1,A)/ϵ （ス）(0,Z)/Z

5. ネットワーク （作成者：柏原）

(1)

(1-1)

$C_0(010) = \phi$
$C_1(101) = 111$

(1-2)

復号失敗とする

(1-3)

シンドローム

(1-4)

ある符号語 $\overline{u}$ と，$\Delta(\overline{u}, \overline{v}) \leq t$ となる語 $\overline{v}$ を考える．ここで，$\Delta(\overline{u}, \overline{w}) = d$ となる符号語 $\overline{w}$ を考える．$\Delta(\overline{w}, \overline{v}) \leq t$ とすると，三角不等式より

$$\begin{aligned} d = \Delta(\overline{u}, \overline{w}) &< \Delta(\overline{u}, \overline{v}) + \Delta(\overline{w}, \overline{v}) \leq t + t = 2t \\ \therefore\ d &< 2t \end{aligned}$$

となり，$2t + 1 \leq d$ と矛盾する．よって，$C_t(v)$ の要素数はたかだか 1 つである．

(1-5)

生成行列 $G = [111]$ より，$G = [I_k A^T]$ とすると，

$$I_k = [1] \qquad A^T = [11]$$

$H = [AI_m]$ より

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

(2)

(2-1)

- 応答確認や送り直しをしない分，実装が単純である．
- コネクションを設定しないため，送受信が速い．

(2-2)

- 送信側ポート番号
- 受信側ポート番号
- シーケンス番号
- 確認応答番号

(2-3)

- 別のコネクションであることを見分ける手段がない.

図 2 において，同一のシーケンス番号を用いると別のコネクションからのデータであることを見分ける方法がない．シーケンス番号の初期値をランダムにすることで，高確率でコネクションが異なることを判別できるようになる.

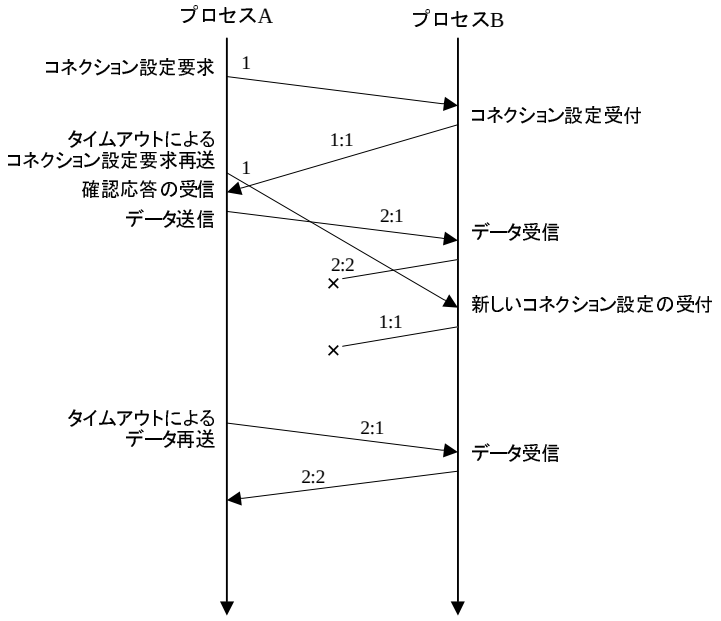


図 2: プロセス A は 1 回，プロセス B は 2 回コネクションを行ったことになっている

- 同時に 2 つ以上のコネクションを設定したとき，それぞれのコネクションを見分ける方法がない．シーケンス番号の初期値をランダムに選択することで，高確率で複数のコネクションを判別できるようになる.

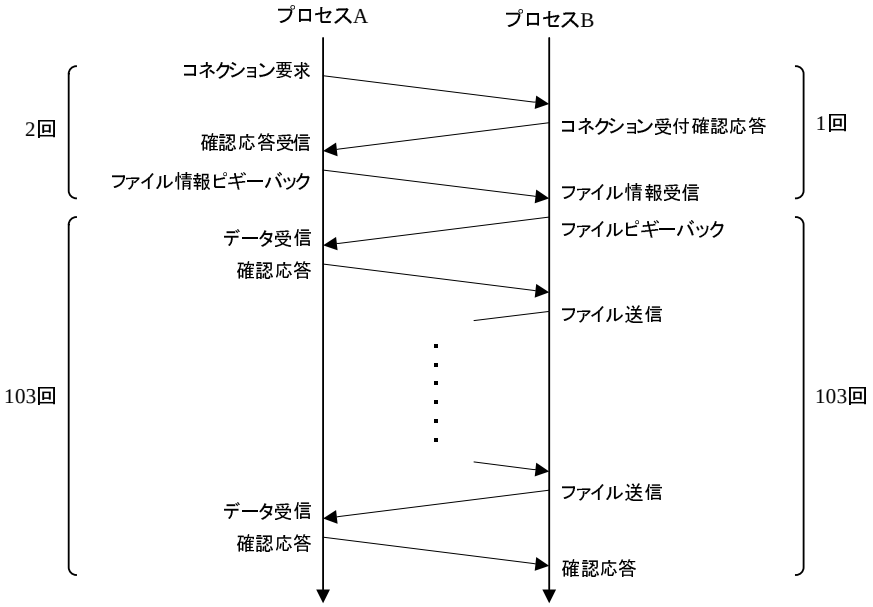
(2-4)

ヘッダのサイズ 20 バイト，最大セグメントサイズ 1000 バイトより，1 セグメントで送信できる最大データ量は 1000 - 20 = 980 バイトとなる．転送するデータは 100100 バイトなので，100100 / 980 = 102.14··· より，100100 バイトのデータ送信に 103 セグメント必要である．また，100 バイトのデータ送信には，1 セグメント必要である．

コネクション設定と取得するファイル情報の送信にプロセス A で 2 セグメント，プロセス B で 1 セグメント送信する．

プロセス A からファイル情報を取得してプロセス B がファイルを送信するのに確認応答とデータ送信のためにそれぞれ 103 セグメント送信する．

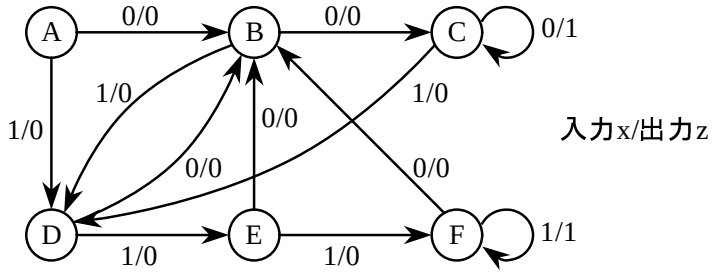
以上より，プロセス A で 105 セグメント，プロセス B で 104 セグメント．



6. 電子回路と論理設計 （作成者：秦野）

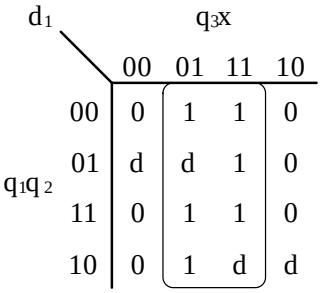
(1)

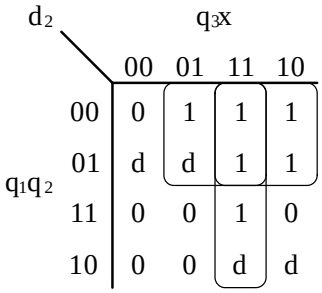
(1-1)

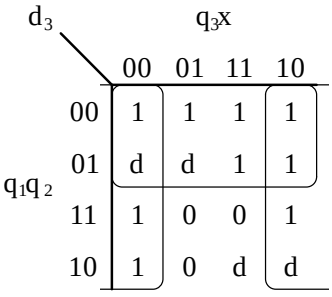


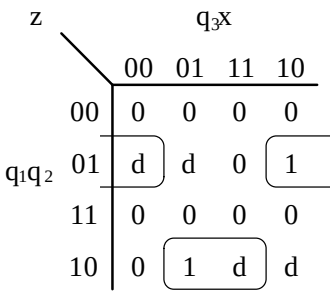
(1-2)

d_1, d_2, d_3, z のカルノー図は以下のようになる.









最小積和形は次の通り.

$d_1 = x$

$d_2 = q_3 x \vee \overline{q_1} x \vee \overline{q_1} q_3$

$d_3 = \overline{q_1} \vee \overline{x}$

$z = q_1 \overline{q_2} x \vee \overline{q_1} q_2 \overline{x}$

(1-3)

(1-2) の最小積和形を以下のように変形する.

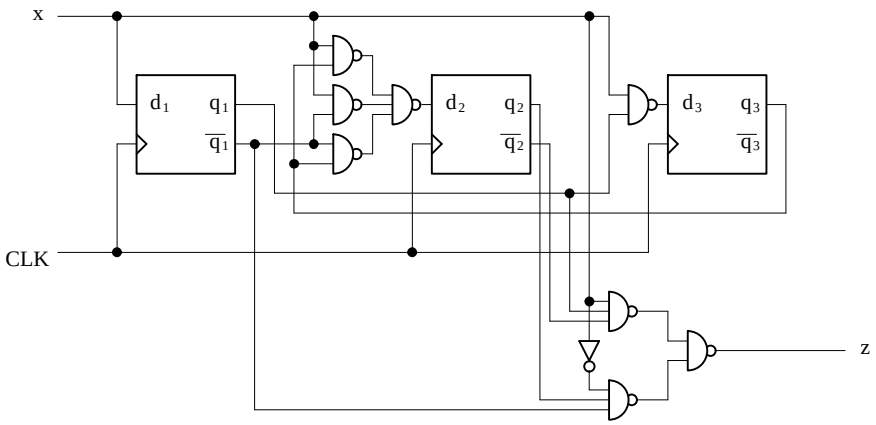
$d_1 = x$

$d_2 = \overline{q_3 x} \cdot \overline{\overline{q_1} x} \cdot \overline{q_1 q_3}$

$d_3 = \overline{q_1 x}$

$z = \overline{q_1 \overline{q_2} x} \cdot \overline{\overline{q_1} q_2 \overline{x}}$

回路図は以下の通り.



(2)