平成18年8月5日（土）
9:00～12:00

大阪大学大学院情報科学研究科

コンピュータサイエンス専攻
情報システム工学専攻
情報ネットワーク学専攻
マルチメディア工学専攻
バイオ情報工学専攻

平成１９年度　博士前期課程　入試問題

（A）情報工学

【注意事項】

- 問題数は必須問題3題（問題1～3），選択問題8題（問題4～11），合計11題である．必須問題は3題すべて解答すること．また，選択問題は2題を選択して解答すること．
- 問題用紙は表紙を含めて20枚である．
- 解答用紙は全部で7枚である．
  1枚目（黄色）の解答用紙には問題1（必須問題）の解答を
  2枚目（青色）の解答用紙には問題2（必須問題）の小問(1)の解答を
  3枚目（青色）の解答用紙には問題2（必須問題）の小問(2)の解答を
  4枚目（オレンジ色）の解答用紙には問題3（必須問題）の小問(1)の解答を
  5枚目（オレンジ色）の解答用紙には問題3（必須問題）の小問(2)の解答を
  6枚目（白色）の解答用紙には問題4～11（選択問題）から選択した1題の解答を
  7枚目（白色）の解答用紙には問題4～11（選択問題）から選択したもう1題の解答を
  それぞれ記入すること．
  解答用紙は間違えると採点されないことがあるので注意すること．
- 解答用紙の「試験科目」の欄には解答した問題の科目名（「アルゴリズムとプログラミング」など）を，「問」の欄には対応する問題番号（1～11から1つ）を記入すること．また，選択問題調査票上の，選択した問題の番号（4～11から2つ）に○をつけること．
- 解答欄が不足した場合は裏面を使用すること．その際，表面末尾に「裏面に続く」と明記しておくこと．解答用紙の追加は認めない．

1 （必須問題：アルゴリズムとプログラミング）

配点：(1) 10 点, (2-1) 10 点, (2-2) 10 点, (3) 20 点, (4-1) 30 点, (4-2) 20 点

逆ポーランド記法（reverse Polish notation）とは，演算子（operator）を被演算子（operand）の後に記述する記法であり，後置記法（postfix notation）ともいう．演算の順番を一意に定めることができるため，括弧を用いないで記述するという特徴がある．我々が日常で用いる記法である中置記法（infix notation）で「1 + 2」と書く算術式は，逆ポーランド記法では「1 2 +」となる．

この問題では，「+」（加算），「-」（減算），「*」（乗算），「/」（除算）の 4 種類の二項演算子（binary operator），整数（integer），「(」（左括弧），「)」（右括弧）からなる算術式を取り扱う．ただし，0 による除算は生じないものとする．また，二項演算子，整数，左括弧，右括弧をあわせて記号と呼ぶ．記号と記号は空白で区切られているものとする．

ここで，算術式は，内部頂点（internal vertex）を二項演算子とし，その左および右の子を根（root）とする部分木（subtree）を被演算子とする二分木で表現できる．例えば，中置記法による算術式「8 / (4 - 2)」の二分木は，図 1 のように表される．この算術式は逆ポーランド記法では「8 4 2 - /」となる．

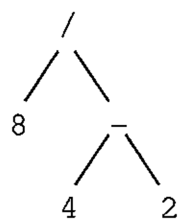


図 1: 算術式「8 / (4 - 2)」を表す二分木

(1) 逆ポーランド記法で記述された算術式「5 4 * 3 2 * -」を表す二分木を，図 1 にならって図示せよ．また，この算術式の演算結果を答えよ．

(2) 中置記法による算術式「(5 + (7 - 3)) * (6 - 4)」について，以下の各小問に答えよ．
(2-1) この算術式を表す二分木を，図 1 にならって図示せよ．
(2-2) この算術式を逆ポーランド記法で記せ．

(3) 一般に，図 1 のような算術式を表す二分木から，逆ポーランド記法による記述を得るためには，どのように木をたどり，どのような順番で頂点を出力すればよいか，簡潔に説明せよ．

(4) 図2は，逆ポーランド記法で記述された算術式を入力として，その演算結果を出力するC言語で書かれたプログラムである．基本的な処理は，以下の通りである．

- 与えられた算術式に対して，先頭から順番に記号（二項演算子または整数）を読み込む．
- その記号が整数であればスタックに積む．
- その記号が二項演算子であれば，その演算に必要な被演算子をスタックから取り出して演算を行う．二項演算子による演算結果はスタックに積む．
- すべての記号が読み込まれた後に，スタックに積まれている整数が算術式の演算結果となる．

なお，図2では関数 ReadToken() の定義は省略されている．ReadToken() は，算術式の記号を先頭から順番に読み込み，その記号を格納した構造体へのポインタを返す関数であり，記号をすべて読み終えた場合には NULL を返すものとする．

以下の各小問に答えよ．
(4-1) プログラム中の（ア）～（カ）を，スタックに関する操作で適切に埋めよ．
(4-2) 逆ポーランド記法で記述された算術式「4 5 6 * +」を入力として，このプログラムを実行したときに，スタックの中身はどのように変化していくか，以下の例にならって図示せよ．この例は，push(1)，push(2)，push(3)，pop() がこの順番で呼ばれたときの，スタックの中身の変化を表している．なお，括弧でスタックを表しており，スタックが空のときは「[ ]」と書く．スタックが空でないときは，左端の要素がスタックの先頭（top）を表している．

例： [ ] -> [1] -> [2 1] -> [3 2 1] -> [2 1]

```
#include <stdio.h>
#include <stdlib.h>

#define STACKMAX 1000

typedef enum { /* 記号の種類を示す列挙型 */
  NUMBER,      /* 整数     */
  PLUS,        /* 加算「+」*/
  MINUS,       /* 減算「-」*/
  ASTERISK,    /* 乗算「*」*/
  SLASH        /* 除算「/」*/
} TokenType;

struct Token {
  TokenType type; /* 記号の種類 */
  int number;     /* 記号が整数であった場合の値 */
};

static int stack[STACKMAX], sp=0;
struct Token *ReadToken(void);

void push(int x){
  stack[sp++]=x;
}

int pop(void){
  return stack[--sp];
}

int main(){
  int a, b;
  struct Token *token;
  while ( (token = ReadToken()) != NULL ){
    if (token->type == NUMBER)  (ア) ;
    else{
      a = pop();
      b = pop();
      switch (token->type){
      case PLUS:
         (イ) ; break;
      case MINUS:
         (ウ) ; break;
      case ASTERISK:
         (エ) ; break;
      case SLASH:
         (オ) ; break;
      }
    }
  }
  printf("%d\n", (カ) );
  exit(0);
}
```

図 2: 算術式の演算結果を表示するプログラム

2 （必須問題:論理回路）

（配点：(1-1)15 点，(1-2)15 点，(1-3)10 点，(2-1)10 点，(2-2)10 点，(2-3)20 点，(2-4)20 点）

(1)

4 ビットの符号なし 2 進数(binary number)表現を入力として，入力値が素数(prime number)であれば出力 f が 1 となり，素数でなければ f が 0 となる回路を作りたい．ただし，素数には 1 が含まれないことに注意せよ．以下の各小問に答えよ．

(1-1)　入力が 0～15 を表す 4 桁の 2 進数 $a_3a_2a_1a_0$ であった場合の f を最小積和形（最簡積和形, minimal sum-of-products expression）で示せ．ただし，a_0 を最下位ビットとする．また，導出の過程（カルノー図(Karnaugh map)または式の変形など）は省略しないこと．

(1-2)　4 ビットの BCD $a_3a_2a_1a_0$ で表現された 10 進数 1 桁の数のみを入力とした場合の f を最小積和形（最簡積和形）で示せ．ただし，a_0 を最下位ビットとする．ここで，BCD（2 進化 10 進, binary coded decimal）とは，10 進数(decimal number)1 桁を 4 ビットの 2 進数列のコード（符号, code）として表現する方式である．

(1-3)　(1-2)の回路を 2 入力の NAND ゲート 4 つのみを用いて構成し，図示せよ．

(2)

1本の信号線から入力されるビット系列に対して3ビット幅のウインドウ(window)を設け，ウインドウ内の連続した時刻$t-2, t-1, t$における各ビットの値に応じて時刻tでの出力を確定するような順序回路(sequential circuit)を考える．以下では，時刻tに入力信号xの値を新たに受け取る毎にウインドウ$[t-2, t-1, t]$内の"1"の数を出力する順序回路を設計する．ただし，ウインドウの初期状態（時刻 −2，−1，0のビットパターン）は，"000"であるものとする．また，出力信号は，"1"の数を2進数で表現し，(z_1, z_0)とする．ただし，z_0が最下位ビットである．

具体的な入出力例を以下に示す．

【入出力例】

t(時刻)	:	0 1 2 3 4 5 6 7 8 9 ・・・・・・・・・・・・・・・・・
x	:	0 0 1 0 1 0 0 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0 ・・・
z_1	:	0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 1 0 0 0 0 0 ・・・
z_0	:	0 0 1 1 0 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 1 0 ・・・

以下の各小問に答えよ．

(2-1) ウインドウ$[t-2, t-1, t]$に対応するビットパターンを状態にとり（つまり，状態数は 8）順序回路を設計することを考える．上記の順序回路の状態遷移図(state transition diagram)を Mealy 型で記述したい．解答用紙の図に状態遷移と出力を付加し，状態遷移図を完成させよ．ただし，Mealy 型順序回路は出力が現状態と入力に依存して決まる順序回路である．

(2-2) ウインドウ$[t-2, t-1, t]$に対応するビットの順に状態変数q_2, q_1, q_0を割当て，この順序回路を設計するものとする．状態変数割当て後の状態遷移表(state transition table)ならびに，出力表(output table)を作成せよ．

(2-3) この順序回路を3個のDフリップフロップ(D flip-flop)を用いて設計する．状態変数q_2, q_1, q_0に対応するDフリップフロップの入力をd_2, d_1, d_0とする．(2-2)の状態遷移表ならびに出力表を元に，カルノー図を利用して，d_2, d_1, d_0およびz_1, z_0をそれぞれx, q_2, q_1, q_0の論理関数で表せ．結果は最小積和形（最簡積和形）で示すこと．なお，設計の過程がわかるように解答すること．

(2-4) (2-1)で設定した状態集合に対して等価な状態を検査し，等価な状態があれば併合した最小状態数の状態遷移図を Mealy 型で記述せよ．ただし，等価な状態の検査過程も記述すること．

(配点比率 (1-1)10 点, (1-2-1)5 点, (1-2-2)15 点, (1-3-1)15 点, (1-3-2)15 点, (2-1)12 点, (2-2)14 点,(2-3)14 点)

(1) 機械語(machine word)の 1 命令が 5 ステージで実行されるプロセッサを考える. 5 つのステージとは IF (Instruction Fetch) ステージ, ID (Instruction Decode) ステージ, EX (EXecution) ステージ, MEM (MEMory) ステージ, WB (Write Back) ステージである. 以下の各小問に答えよ.

(1-1) 以下の空欄 (a) ～ (e) に入る適切な語句を①～⑧より選択せよ.

パイプライン処理(pipeline processing)は, 複数の命令を少しずつずらして (a) に実行する方式である. パイプラインを構成する 5 つのステージの役割は以下の通りである.

IF ステージでは, プログラムカウンタ中のアドレスを用いてメモリから命令が読み出される. ID ステージでは, 命令の (b) とレジスタ (c) が行われる. それに続く EX ステージでは命令操作の実行またはアドレスの (d) が行われる. MEM ステージではメモリ中のオペランドへのアクセスが行われる. WB ステージでは実行結果のレジスタ (e) が行われる.

①生成	②消滅	③エンコード	④デコード
⑤読み出し	⑥書き込み	⑦同時並列的	⑧順次直列的

(1-2) 5 つのステージを処理する機能ブロックの遅延時間(delay time)を表 1 に示す. 表 1 を用いて以下の各小問に答えよ. ただし, 表 1 に記載されている機能ブロック以外の遅延時間は考慮しないものとする.

表 1 : 各ステージを処理する機能ブロックの遅延時間

	IF	ID	EX	MEM	WB
遅延時間	2ns	1ns	2ns	2ns	1ns

$(n = 10^{-9})$

(1-2-1) これらの機能ブロックを用いて逐次処理(sequential processing)を行う (パイプライン処理を行わない) プロセッサを構成した場合において, 1 命令の実行時間(execution time)を示せ.

(1-2-2) これらの機能ブロックを用いてパイプライン処理を行うプロセッサを構成した場合において, 1 命令の実行時間を示せ. また, 1 秒あたりの最大処理命令数とその理由を述べよ.

(1-3) 加算命令 add と減算命令 sub とレジスタ Rn (n=1,2,...) を用いた以下の 3 命令を実行した.

add R1,R2,R3 : R2＋R3 の結果を R1 に格納する.

sub R4,R1,R5 : R1－R5 の結果を R4 に格納する.

add R6,R7,R8 : R7＋R8 の結果を R6 に格納する.

そのとき, 図 1 に示すような 3 クロックサイクル(6ns)のハザード(hazard, パイプラインの乱れ)が発生した. 以下の各小問に答えよ.

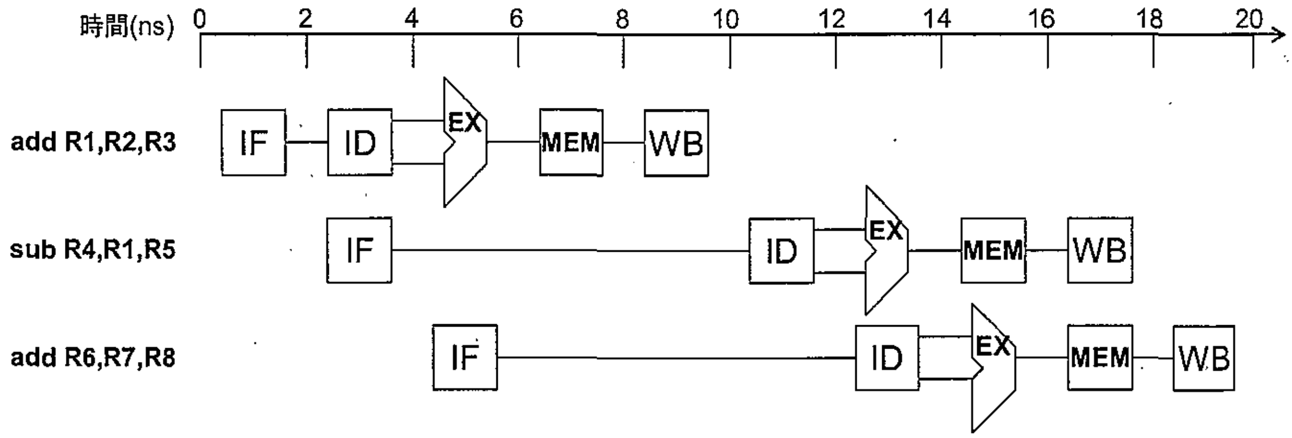


図１：3命令のパイプライン処理過程の模式図

(1-3-1)　3クロックサイクルにわたるハザードが発生した原因を簡潔に述べよ．なお，次の3単語を用いること．

「add 命令」，「sub 命令」，「WB ステージ」

(1-3-2)　このハザードを回避または削減するためにはどうすればよいか．回避または削減のいずれか一方の手法について簡潔に述べよ．またその結果，上記3命令の実行に要する時間が 20ns から何 ns に変化するかを答えよ．

(2) 以下の各小問に答えよ．

(2-1)　次の説明文の空欄 (a) ～ (l) に当てはまる適当な語句を下記の選択肢から１つ選び，その番号を答えよ．ただし，選択肢からは複数回選んでもよい．

単一プロセッサをもつマルチプログラミングシステムにおいて，プロセスは，実行状態(running state)，実行可能状態(ready state)，待ち状態(wait state または blocked state)の３つの状態を持ち，プロセスが生成されてから消滅するまでその３状態間を遷移する．生成されたプロセスは，まず初期状態として (a) となりキュー(queue)につながれる．プロセッサに空きが生じると (b) にあるプロセスから１つのプロセスが選択され，その選択されたプロセスはプロセッサに割り付けられ， (c) に遷移する． (d) にあるプロセスは，処理の完了によって消滅する．あるいはプリエンプション(preemption)により， (e) へ遷移しキューにつながれる． (f) にあるプロセスが入出力待ちとなった場合には， (g) へ遷移し，入出力が行われた後， (h) へ遷移する．

プロセスをプロセッサに割り付けるスケジューリングアルゴリズムとして，いくつかの方式が考えられる． (i) 方式では，プロセスがキューに到着した順に割り付けられる． (j) 方式では，予想される処理時間の短い順に割り付けられる．これらの方式では，プリエンプションは行われない． (k) 方式では，プロセスがキューに到着した順に割り付けられるが，一定時間（タイムスライス）が経過した後にプリエンプションされ，再びキューの最後につながれる． (l) 方式は，新しく到着したプロセスの予想される処理時間とその時点で実行中のプロセスに必要な残りの処理時間を比較し，新しく到着したプロセスの処理時間の方が短ければ，プリエンプションにより新しく到着したプロセスを割り付ける．

選択肢

①　カーネルモード　②　実行状態　③　FCLS (First Come Last Served)　④　ユーザモード
⑤　優先度順　⑥　FCFS (First Come First Served)　⑦　実行可能状態　⑧　シングルタスク
⑨　RR (Round Robin)　⑩　マルチプロセッサ　⑪　SRT (Shortest Remaining Time First)
⑫　待ち状態　⑬　EDF (Earliest Deadline First)　⑭　SPT (Shortest Processing Time First)

(2-2)　プロセス P1, P2, P3, P4 が表２に示す時刻にそれぞれ生成され，キューにつながれたとする．また，それぞれのプロセスの処理時間は表２に示すとおりである．なお，入出力処理はないものとする．

FCFS 方式のプロセス処理の進む様子を図２に示している．FCFS 方式と同様の形式で，RR 方式，SPT 方式におけるプロセス処理の進む様子を解答用紙の図に記入せよ．さらに，RR 方式，SPT 方式の平均ターンアラウンド時間を求めよ．ただし，プロセス切り替え時間（プロセスの切り替えに要する制御時間）は無視できるものとする．また，タイムスライスを４とし，タイムスライス中に処理が完了した時は次のプロセスに切り替わるものとする．

表２：プロセスの生成時刻と処理時間（その１）

プロセス	生成時刻	処理時間
P1	0	4
P2	2	8
P3	4	2
P4	10	4

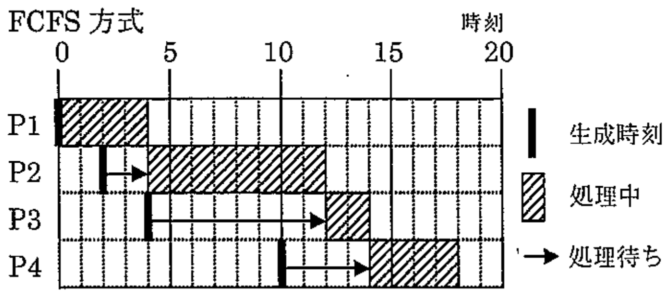


図２：FCFS 方式のプロセス処理の進む様子

(2-3)　プロセス P1, P2 が表 3 に示す時刻にそれぞれ生成され，キューにつながれたとする．SPT 方式に比べ RR 方式の平均ターンアラウンド時間が短くなる場合の例を考え，その場合のプロセス P3, P4 の生成時刻と処理時間を解答用紙の表に埋めよ．さらに，その場合のプロセス処理の進む様子を(2-2)と同様の形式で解答用紙の図に記入せよ．ただし，4 つの全プロセスは時刻 25 までに終了すること．タイムスライスは 4 とする．

表 3：プロセスの生成時刻と処理時間（その 2）

プロセス	生成時刻	処理時間
P1	0	4
P2	2	8
P3		
P4		

（配点：(1-1) 20 点, (1-2-1) 15 点, (1-2-2) 25 点, (2-1) 15 点, (2-2) 25 点）

(1)

図 1 に示す中心点 O，半径 a の円形コイル(coil)に電流(current) I が流れているときの磁界(magnetic field)を考える．以下の各小問に答えよ．

(1-1)　微小電流要素 Ids によって，中心軸上の点 P に生じる微小磁界の大きさ dH を示せ．

(1-2)　点 P における磁界 $\boldsymbol{H}$ を求めたい．(1-2-1), (1-2-2)の問いに答えよ．

(1-2-1) 点 P における磁界 $\boldsymbol{H}$ の向きが中心軸と平行である理由を 50 字以内で簡単に説明せよ．

(1-2-2) コイルの円周に沿って積分することにより磁界の大きさ H を求めよ．

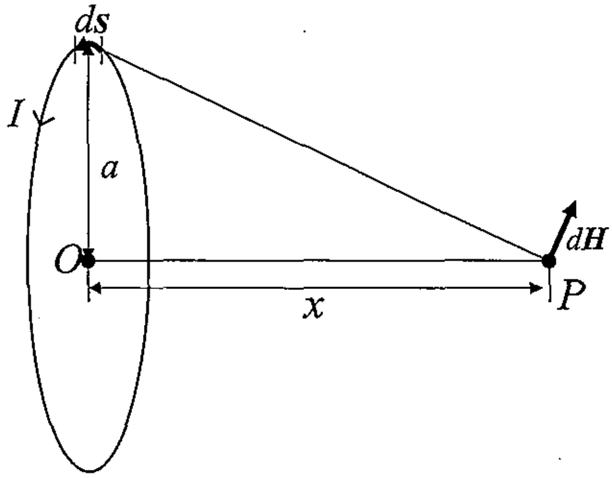


図 1

(2)

図 2 のように半径 a の 2 個の円形コイルが中心軸を共通にして間隔 $2d$ で対置されている．両コイルに同一方向に同じ大きさの電流 I が流れている．点 Q は 2 つの円形コイルの中心点の中点である．点 R は点 Q から距離 z 離れた中心軸上の点である．以下の各小問に答えよ．

(2-1)　点 R の磁界の大きさ H を求めよ．

(2-2)　Q 点付近の磁界がほぼ一様(homogeneous)となるために必要な a と d の関係を求めよ．

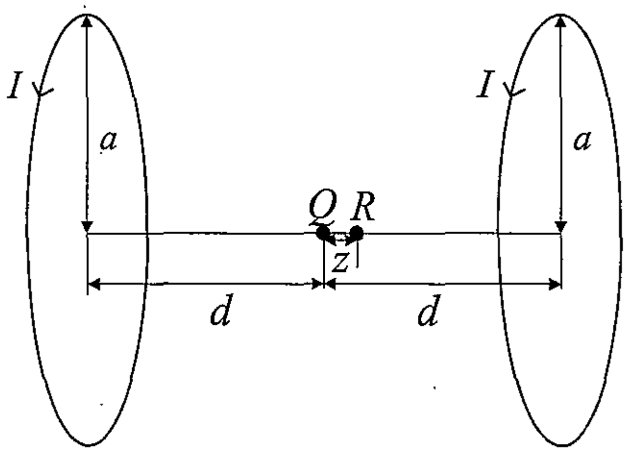


図 2

（配点：(1) 25 点，(2) 25 点，(3-1) 20 点，(3-2) 30 点）

ラプラス変換（Laplace transform）領域において，以下の各問に答えよ．

(1) 図 1 の 2 ポート回路（two-port circuit）の一次側の電圧 $V_1(s)$，電流 $I_1(s)$ と二次側の電圧 $V_2(s)$，電流 $I_2(s)$ を関係付ける式(1)において，

$$\begin{pmatrix} I_1(s) \\ I_2(s) \end{pmatrix} = \begin{pmatrix} y_{11}(s) & y_{12}(s) \\ y_{21}(s) & y_{22}(s) \end{pmatrix} \begin{pmatrix} V_1(s) \\ V_2(s) \end{pmatrix} \tag{1}$$

$y_{11}(s)$，$y_{12}(s)$，$y_{21}(s)$，$y_{22}(s)$ を要素とする行列をアドミタンス行列（admittance matrix）$\mathbf{Y}(s)$ と呼ぶ．
このとき，行列 $\mathbf{Y}(s)$ の各要素を，V_1，I_1，V_2，I_2 を用いて表せ．

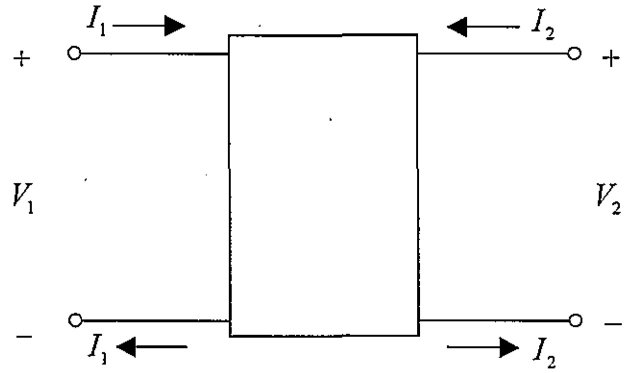


図 1

(2) 図 2 のように，2 つの部分回路の並列接続（parallel connection）からなる 2 ポート回路全体のアドミタンス行列を $\mathbf{Y}(s)$ とし，それぞれの部分回路のアドミタンス行列を $\mathbf{Y}'(s)$ と $\mathbf{Y}''(s)$ で表すとき，$\mathbf{Y}(s)$ を $\mathbf{Y}'(s)$ と $\mathbf{Y}''(s)$ で表せ．導出過程も示せ．

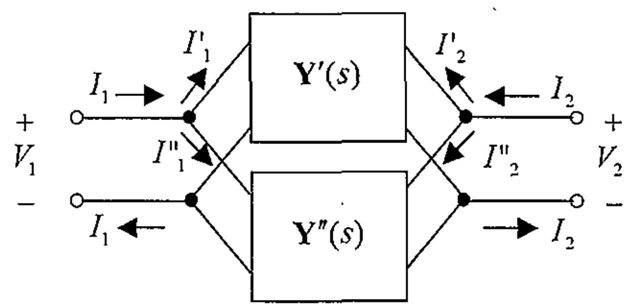


図 2

(3) 図 3 の 2 ポート回路について，以下の各小問に答えよ．

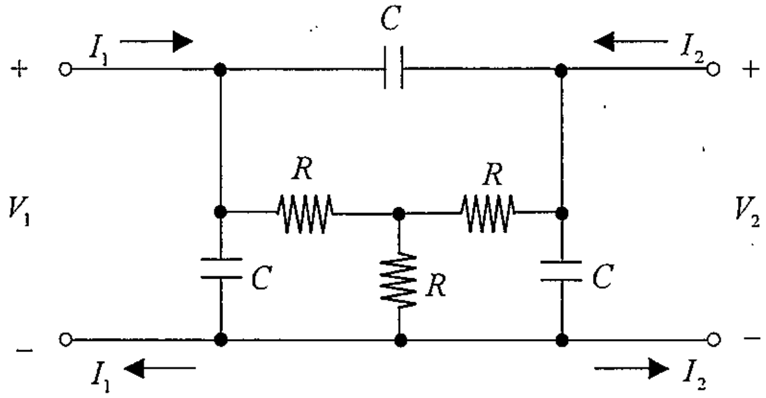


図 3

(3-1) 図 3 の 2 ポート回路を，図 2 のように部分回路の並列接続で表すことを考える．その際，抵抗器 (resistor) だけの 2 ポート部分回路と，キャパシタ (capacitor) だけの 2 ポート部分回路との並列接続とした場合について，それぞれの部分回路を図示せよ．

(3-2) 図 3 の 2 ポート回路の，アドミタンス行列 $\mathbf{Y}(s)$，ならびに二次側を開放したときの電圧伝達関数 (voltage transfer function) $T(s)=V_2(s)/V_1(s)$ を求めよ．

（配点：(1) 50 点，(2-1) 10 点，(2-2) 30 点，(2-3) 10 点）

以下の各問に答えよ．

(1) ラプラス変換 (Laplace transform) を利用して次の連立微分方程式を解き，$x(t)$ を求めよ．$y(t)$ を求める必要はない．

$$\dot{x}(t) + 2\dot{y}(t) + x(t) - y(t) = 1$$

$$2\dot{x}(t) + y(t) = e^t$$

ただし，$x(0) = 0$，$y(0) = 1$ とする．

(2) 以下の各小問に答えよ．

(2-1) 区間 $[-\pi, \pi]$ で定義される実数関数 $f(x)$ が偶関数のとき，$f(x)$ のフーリエ級数展開 (Fourier series expansion) の式およびその係数を求める式を示せ．

(2-2) 上記の $f(x)$ が $f(x) = e^{|x|}$ のとき，$f(x)$ をフーリエ級数展開せよ．

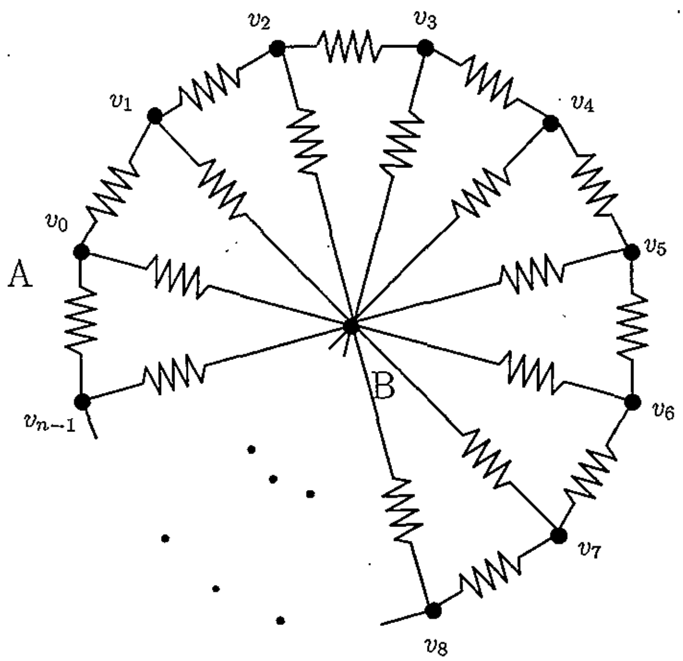
(2-3) (2-2) で得られた結果を利用して，

$$\sum_{n=1}^{\infty}[e^{\pi}(-1)^n - 1]\frac{2}{1+n^2}$$

の値を求めよ．

配点：(1)15点，(2)15点，(3)20点，(4)50点

1オームの抵抗 $2n$ 本を図のように接続した（外周上に n 本，中心と外周の間に n 本ある）．ただし，$n \geq 3$ とする．外周上の端子を任意に一つ選んで端子Aと呼び，中心の端子を端子Bと呼ぶ．



端子Aと端子Bに1アンペアの定電流源を接続し，端子Aに1アンペアの電流を流し込み，端子Bから引き出した．このとき端子Bの電位を0，外周上の端子の電位を端子Aから時計回りの順に $v_0, v_1, \cdots, v_{n-1}$ とする．

(1) $i = 0, 1, \cdots, n-3$ について，v_i, v_{i+1}, v_{i+2} の間の関係式を書け．

(2) v_0, v_1, v_{n-1} の間の関係式および v_0, v_{n-1}, v_{n-2} の間の関係式を書け．

(3) 問(1)の解答として与えた $n-2$ 個の関係式をまとめて式(1)と呼び，式(1)を満たすような $v_0, v_1, \cdots, v_{n-1}$ を並べたベクトル $[v_0, v_1, \cdots, v_{n-1}]$ を式(1)の解ベクトルと呼ぶことにする．このとき，式(1)の一般解を与えよ．ここで，式(1)の一般解とは，いくつかのパラメータを含んだ記述で，各パラメータに実数値を代入すると実数ベクトルが得られ，

a) 各パラメータにどのような実数値を代入したものも式(1)の解ベクトルになり，かつ，

b) 各パラメータに適切な値を代入することにより，式(1)の任意の解ベクトルに等しくできる

ようなものをいう．

(4) v_0 を求めよ．なお，代数方程式の根は α，β など適当な記号で表記してよい．

(配点: (1) 45 点, (2-1) 10 点, (2-2) 10 点, (2-3) 8 点, (2-4-1) 15 点, (2-4-2) 3 点, (2-4-3) 4 点, (2-5) 5 点)

以下の各問において $\forall, \exists, \wedge, \vee, \neg, \rightarrow, (,)$ は, それぞれ, 全称記号, 存在記号, 論理積, 論理和, 否定, 含意, 補助記号としての "(", ")" とする. それ以外の論理記号は解答において用いてはいけない.

(1) 以下で定まる一階述語論理式 E について $\neg E$ が充足不能 (infeasible) であることを導出原理 (resolution principle) を用いて示せ. 導出過程も示すこと. なお, a, b は定数記号, f, g は関数記号, P は述語記号を表す.

$$A = \forall x \forall y(P(x, f(y, x)) \vee \neg P(x, y)), \quad B = \forall x \forall y(P(g(x), y) \vee \neg P(x, y)), \quad C = P(a, b),$$
$$D = P(g(g(a)), f(b, g(a))), \quad E = \neg(A \wedge B \wedge C) \vee D$$

(2) 次の論理問題に関する以下の各小問に答えよ.

> ある庭に花が咲いている. A が言った.「この庭の花の色は赤, 紫, 黄のいずれかである. この 3 色のいずれの色の花も咲いている.」B が言った.「この庭から異なる花を 3 本どのように選んでも紫色の花が少なくとも 1 本は含まれる. 」C が言った.「この庭から異なる花を 3 本どのように選んでも黄色の花が少なくとも 1 本は含まれる. 」A,B,C の陳述が全て真であるとき, 「この庭から異なる花を 3 本どのように選んでも赤色の花が少なくとも 1 本は含まれる. 」は真か?

ある花 x が赤色, 紫色, 黄色であることを表す述語をそれぞれ $R(x), P(x), Y(x)$ とする.
また, $x \neq y$ は x と y が等しくないことを表す述語とする.
各述語変数のドメインは「この庭の花」に限定する. A,B,C の陳述内容を表す一階述語論理式をそれぞれ A, B, C とする.

(2-1) 次の文章の内容を正確に表す一階述語論理式 D を, $R(\cdot)$, $P(\cdot)$, $Y(\cdot)$ を用いて表せ. D の式は複数考えられるが, 冗長な部分式を含まないこと.
「どの花についても次が成り立つ.『その花が同時に赤色かつ紫色である, ということはない.』そのようなことが赤, 紫, 黄, 各色のいずれの組みあわせについても言える. 」

(2-2) A を $R(\cdot), P(\cdot), Y(\cdot)$ を用いて表せ.

(2-3) C を次のように表した. 空欄を埋めよ. 述語として $R(\cdot), P(\cdot), Y(\cdot)$, および, $\neq$ を用いてよい. $C = \forall x \forall y \forall z(($ (a) $) \rightarrow ($ (b) $))$

(2-4) $D \wedge A \wedge B \wedge C$ を真にする解釈 (interpretation) を考えたい.
値集合 V を $V = \bigcup_{i \in \mathbb{N}}\{a_i\} \cup \bigcup_{i \in \mathbb{N}}\{b_i\} \cup \bigcup_{i \in \mathbb{N}}\{c_i\}$ とする ($\mathbb{N}$ は自然数全体の集合).
この小問では各述語記号の解釈 ($\mathbb{I}_p$) を次に固定する.
$R(x), P(x), Y(x)$ は, x が それぞれ a_i, b_i, c_i $(i \in \mathbb{N})$ のとき真それ以外は偽.
$x \neq y$ は, $x, y \in V$ なる x, y が相異なる要素であるとき真それ以外は偽.

このとき, 例えば, V と $\mathbb{I}_p$ からなる解釈は $D \wedge A$ を真にする.

以下の各小問に答えよ.

(2-4-1) $V_1 = \{a_1\} \cup \{b_1\} \cup \bigcup_{i \in \mathbb{N}}\{c_i\}$ とする. V_1 と $\mathbb{I}_p$ からなる解釈は $D \wedge A \wedge C$ を真にすることを説明せよ.

(2-4-2) 同様に V_2 と $\mathbb{I}_p$ からなる解釈のもとで $D \wedge A \wedge B$ を真にするような値集合 V_2 を 1 つ与えよ. 理由の説明は不要.

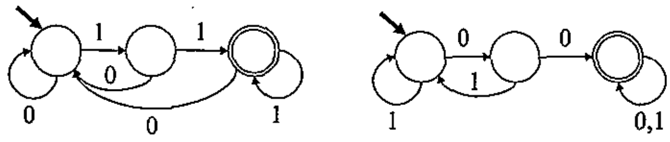
(2-4-3) 同様に $D \wedge A \wedge B \wedge C$ を真にするような V_3 を具体的に示せ (要素を列挙せよ).

(2-5) D の前提下でこの論理問題の真偽を答えよ. 理由の説明は不要.

（配点：(1-1) 10 点, (1-2) 40 点, (2-1) 20 点, (2-2-1) 15 点, (2-2-2) 15 点）

(1) 決定性有限オートマトン (deterministic finite automaton) に関する以下の各小問に答えよ．決定性有限オートマトンは 5-組 $(Q, \Sigma, \delta, q, F)$ で表される．ここで Q は状態 (state) の有限集合，Σ は入力記号 (input symbol) の有限集合，$\delta : Q \times \Sigma \to Q$ は状態遷移関数 (state transition function)，q は初期状態 (initial state)，F は受理状態 (accepting state. 最終状態 (final state) と呼ぶ場合もある) の集合である．以下，Σ は $\{0,1\}$ とする．

下の (i), (ii) は，2 つの決定性有限オートマトンの状態遷移図 (state transition diagram) を表している．始点を持たない太い矢印が指し示している状態が初期状態，2 重丸で表されている状態が受理状態である．(i), (ii) が表している有限オートマトンを，それぞれ A, B とする．



(i) 有限オートマトン A　　(ii) 有限オートマトン B

(1-1) A と B それぞれについて，受理される長さ 4 以下の語 (word) をすべて書け．

(1-2) 決定性有限オートマトン $M = (Q, \Sigma, \delta, q, F)$ が与えられたとき，$\mathcal{U}(M)$ を次のようなオートマトンと定義する．

$$\mathcal{U}(M) = (Q, \Sigma, \delta, q, Q - F)$$

また，2 つの決定性有限オートマトン $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ が与えられた場合，$\mathcal{T}(M_1, M_2)$ を次のようなオートマトンと定義する．

$$\mathcal{T}(M_1, M_2) = (Q, \Sigma, \delta, (q_1, q_2), F_1 \times F_2)$$

ここで，$Q = Q_1 \times Q_2$ であり，すべての $((p_1, p_2), x) \in Q \times \Sigma$ について，以下の様に δ を定める．

$$\delta((p_1, p_2), x) = (\delta_1(p_1, x), \delta_2(p_2, x))$$

今，次の 5 つの有限オートマトン C_1, C_2, C_3, C_4, C_5 を考える．

$$C_1 = \mathcal{U}(A) \quad C_2 = \mathcal{U}(\mathcal{U}(A)) \quad C_3 = \mathcal{T}(A, B) \quad C_4 = \mathcal{T}(\mathcal{U}(B), A) \quad C_5 = \mathcal{T}(B, \mathcal{U}(B))$$

これらのオートマトンそれぞれについて，そのオートマトンが認識する言語 (language) を表している正規表現 (regular expression) を，1〜10 の中から 1 つ選べ．同じものを複数回選んでもよい．ただし，ϕ は空集合 (empty set)，ε は空語 (empty word) を表すものとする．

1. $(0+1)^*(00+10+01)$　2. $(0+1)^*$　3. ϕ　4. $(1+01)(1+01)^*1$　5. $(0+1)(11^*0)^*11$
6. $\varepsilon+1+(0+1)^*(0+01)$　7. $(0+1)^*11$　8. 0^*1^*　9. $(0+1)^*00(0+1)^*$　10. $(0+1)^*00(0+1)^*11$

(2) 文脈自由文法 (context-free grammar) に関する以下の各小問に答えよ．以下では文脈自由文法を 4-組 (N, T, P, S) で表す．ここで N は非終端記号 (non-terminal symbol) の有限集合，T は終端記号 (terminal symbol) の有限集合，P は生成規則 (generating rule, production rule) の有限集合，S は始記号 (start symbol) である．

(2-1) 2 つの文脈自由文法 $G_1 = (N_1, T_1, P_1, S_1)$, $G_2 = (N_2, T_2, P_2, S_2)$ について考える．N_1, N_2, T_1, T_2, P_1, P_2, S_1, S_2 は次の通りとする．ただし，ε は空語を表すものとする．

$N_1 = N_2 = \{block, seq, stmt\}$, $T_1 = T_2 = \{$ `expr`, `while`, `other`, `[` , `]` , `;` $\}$,

$P_1 = \{block \to$ `[` seq `]` , $seq \to stmt$, $seq \to stmt\ seq$, $stmt \to$ `while` `expr` $block$, $stmt \to$ `while` `expr` $stmt$, $stmt \to$ `other` `;` $\}$,

$P_2 = \{block \to$ `[` seq `]` , $seq \to stmt$, $seq \to stmt$ `;` seq , $stmt \to$ `while` `expr` $block$, $stmt \to$ `other` , $stmt \to \varepsilon$ $\}$,

$S_1 = S_2 = block$

以下の 1～6 のうち，G_1 によって生成される文 (sentence) をすべて挙げよ．また，G_2 によって生成される文をすべて挙げよ．

1. `[ while expr [ ] ]`
2. `[ other ; other ; other ; ]`
3. `[ other ; while expr other ; ]`
4. `[ other ; while expr [ other ] ]`
5. `[ while expr [ while expr other ; ] ]`
6. `[ while expr [ while expr [ other ; ] ] ]`

(2-2) 文脈自由文法 $G_3 : (\{S\}, \{a, +, -, (,)\}, P, S)$ について考える．ただし，P は以下の通りとする．

$$P = \{S \to S + S,\quad S \to -S,\quad S \to (\ S\ ),\quad S \to a\}$$

(2-2-1) 文 $(\ a\ ) + a + -a$ について，G_3 によって得られる異なる導出木 (derivation tree) を 2 つ示せ．

(2-2-2) 文脈自由文法 $G_4 : (\{S, A\}, \{a, +, -, (,)\}, P', S)$ を考える．G_4 が G_3 と同じ言語を生成し，かつ，あいまいでない (unambiguous) 文脈自由文法となるように，以下の $\boxed{1}$～$\boxed{4}$ の空欄を埋めよ．ただし，各空欄には 1 つの終端記号か，1 つの非終端記号が入るものとする．

$$P' = \{\ S \to A,\quad \boxed{1} \to S + \boxed{2},\quad A \to -A,\quad A \to (\ \boxed{3}\ ),\quad A \to \boxed{4}\ \}$$

10 (選択問題:情報理論)

(配点: (1)20 点, (2)30 点, (3-1)10 点, (3-2)10 点, (3-3)20 点, (4)10 点)

最適な情報源符号化に関して述べた次の文章を読み，以下の各問に答えよ．

S は記憶のない情報源 (memoryless source) であるとし，記号 0 または 1 をそれぞれ確率 1/2 で発生するものとする．このとき S のエントロピー (entropy) $H(S)$ は $\boxed{(ア)}$ であり，これは情報源記号数 (the number of source symbols) が 2 の情報源がもちうるエントロピーの最大値である．また，T を，情報源記号集合を $\{a_1, \ldots, a_6\}$ とする記憶のない情報源とし，各 a_i $(1 \leq i \leq 6)$ の生起確率 p_i はそれぞれ $p_1 = 1/2$，$p_2 = p_3 = p_4 = 1/8$，$p_5 = p_6 = 1/16$ であるとする．このとき T のエントロピー $H(T)$ は $\boxed{(イ)}$ である．

さて，T の 2 元ハフマン符号化 (binary Huffman encoding) の一つとして，以下の符号化 C がある．

$$
\begin{aligned}
C(a_1) &= \boxed{(ウ 1)}, & C(a_2) &= 101, \\
C(a_3) &= \boxed{(ウ 2)}, & C(a_4) &= \boxed{(ウ 3)}, \\
C(a_5) &= \boxed{(ウ 4)}, & C(a_6) &= 1000.
\end{aligned}
$$

U を，T が発生する記号を C により符号化した $\{0, 1\}$ 上の記号列を発生する，情報源記号数 2 の情報源とする．

このとき，U の振舞いは S のそれと区別ができない．より正確には，U が i 番目に発生する記号を表す確率変数を U_i とおくと，任意の i について，<u>(A) $U_i = 0$ となる確率は 1/2</u> であり，かつ，<u>(B) U_i は結合確率変数 (joint random variable) $U_1 \cdots U_{i-1}$ と独立</u> である．このことより，U のエントロピーは $H(S)$ と等しいことが導ける．すなわち，U は最大のエントロピーをもつ情報源記号数 2 の情報源であり，これは，C により T が最適に 2 元情報源符号化されたことを意味する．実際，$H(T)$ を符号化 C の $\boxed{(エ)}$ で割った値は $H(S)$ と一致する．

(1) 空欄 (ア), (イ) を適切な値で埋めよ．ただし，対数の底は 2 とし，導出過程も記述すること．

(2) 空欄 (ウ 1) 〜 (ウ 4) を適切な記号列で埋めよ．符号化 C が一意に定まらず適切な記号列の組合せが複数存在する場合は，<u>それらの組合せすべて</u>を答えること．導出過程を記述する必要はない．

(3) ハフマン符号は瞬時に復号可能な符号 (instantaneous code) であるため，$U_1 \cdots U_{i-1}$ の値が与えられると，U_i が C における符号語 (codeword) の何番目の記号なのかが一意に定まる．たとえば，$U_1 \cdots U_5 = 10110$ だとすると，符号語の切れ目は /101/10 のようになるため，U_6 は符号語の 3 番目の記号であると一意に決定できる (なお，U_1 と U_4 は符号語の先頭の記号，U_2 と U_5 は符号語の 2 番目の記号，U_3 は符号語の 3 番目の記号である)．下線部 (A), (B) は，U_i の符号語における位置に基づく場合分けにより証明できる．いくつかの場合に相当する以下の各小問に答えよ．必要ならば,「3 つの確率変数 X，Y，Z について，結合確率変数 YZ が X と独立かつ Z が Y と独立ならば，Z は XY と独立」という事実を用いてよい．

(3-1) U_i が符号語の先頭の記号であるとき，$U_i = 0$ となる確率が 1/2 であることを示せ．

(3-2) $i \geq 2$ とする．U_i が符号語の先頭の記号であるとき，U_i は $U_1 \cdots U_{i-1}$ と独立であることを示せ．

(3-3) $i \geq 4$ とする．U_i が符号語の 3 番目の記号であるとき，U_i は $U_1 \cdots U_{i-1}$ と独立であることを示せ．

(4) 空欄 (エ) を適切な語句 (数値や式ではない) で埋めよ．

(配点：(1) 30 点，(2) 20 点，(3) 30 点，(4) 20 点)

トランスポート層プロトコルである TCP (Transmission Control Protocol) に関する以下の各問に答えよ．

(1) TCP について述べた以下の文中の空欄にあてはまる最も適切な用語を選択肢 (1)ー(20) よりそれぞれ選べ．

TCP は，(a) 間で (b) 型通信サービスを実現するトランスポート層プロトコルである．上位層に対し信頼性のある通信路を提供するため，TCP はフロー制御，輻輳制御，再送制御，順序制御を行う．フロー制御により，(c) におけるバッファあふれによるセグメントの損失を回避し，輻輳制御により，(d) におけるバッファあふれによるセグメント損失を抑制，回避する．フロー制御には (e) や (f) などの方式があるが，TCP では (e) が用いられている．

選択肢：

(1) プロトコル　(2) P2P (Peer to Peer)　(3) ストップアンドウェイト　(4) アドホック
(5) ルータ　(6) 誤り制御　(7) AIMD (Additive Increase and Multiplicative Decrease)
(8) スライディングウィンドウ　(9) コネクション　(10) ビット誤り　(11) リンク
(12) エンドホスト　(13) パケット　(14) コネクションレス　(15) 受信側ノード
(16) フレーム　(17) 無線通信　(18) ACK (確認応答: acknowledgement)
(19) UDP (User Datagram Protocol)　(20) 片方向リアルタイム

(2) 一般的に，テレビ会議やインターネット電話などのアプリケーションでは，トランスポート層プロトコルに TCP ではなく UDP (User Datagram Protocol) が用いられることが多い．その理由を 150 文字程度で述べよ．

(3) スロースタートフェーズ (slow start phase)，輻輳回避フェーズ (congestion avoidance phase)，しきい値 (slow start threshold) の三つの用語を用いて，コネクション確立後から，輻輳回避フェーズにおいて初めてパケット損失が発生するまでの輻輳ウィンドウ (congestion window) の変化について 200 文字程度で述べよ．

(4) TCP ではセグメントの送信から ACK (確認応答) を受信するまでの時間からラウンドトリップ時間 (round trip time) の観測値を得て，ラウンドトリップ時間の推定値を算出し，輻輳検知のためのタイムアウト時間を決定する．ラウンドトリップ時間の推定値を A，平均偏差を D，およびタイムアウト時間を R とした時，あるセグメントに対する ACK を受信し，ラウンドトリップ時間の観測値 M が得られると，新しい推定値 A'，平均偏差 D'，タイムアウト時間 R' はそれぞれ次式で求められる．なお，α=1/8 である．

$$E = M - A$$

$$A' = A + \alpha E$$

$$D' = D + \alpha(|E| - D)$$

$$R' = A' + 4D'$$

長期間，ラウンドトリップ時間の観測値がある一定の値であった結果，ラウンドトリップ時間の推定値だけでなく，タイムアウト時間もラウンドトリップ時間の観測値に近付いた．このことにより，以後，どのような問題が生じる可能性があるかを 100 文字程度で述べよ．