平成 30 年 8 月 4 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 31 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 2 題（問題 1～2），選択問題 5 題（問題 3～7），合計 7 題である．
  必須問題は 2 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 17 ページである．
- 解答用紙は全部で 5 枚ある．
  1 枚目（赤色）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(1)の解答を
  3 枚目（緑色）の解答用紙には問題 2（必須問題）の(2)の解答を
  4 枚目（白色）の解答用紙には問題 3～7（選択問題）から選択した 1 題の解答を
  5 枚目（白色）の解答用紙には問題 3～7（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．
- 解答用紙は 5 枚すべてを回収するので，すべての解答用紙に受験番号を記入すること．
- 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）を記入すること．
  また，選択問題調査票には，選択した問題の番号（3～7 から二つ）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．
- 解答用紙には，日本語または英語で解答すること．

1 【必須問題】アルゴリズムとプログラミング　（情報工学 1/16）

配点：(1) 20 点，(2) 35 点，(3) 15 点，(4-1) 15 点，(4-2) 15 点，(5) 25 点

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）は所有している複数のくじ（lottery）のそれぞれが当選（win）しているかを調べて，当選しているくじ番号（lottery number）と等級（grade）をもれなく出力（output）するものである．くじ番号は 10000 未満の自然数（natural number）で定められており，いずれのくじ番号のくじもたかだか一つしか存在しない．所有しているくじ番号が，当選番号（winning number）と一致した場合に，その当選番号に対応する等級に当選したとする．当選番号は 10000 未満の自然数から重複なく選ばれたN個（Nは$3 \leq N \leq 100$の自然数）の数字で，等級は 1 等から 3 等まであり，1 等が 1 本，2 等が 1 本，3 等が$N-2$本である．

当選番号と等級は図 2 に示すような形式（format）のファイル win.txt で与えられ，1 行目に当選番号の総数N，2 行目以降のN行は全ての当選番号とその等級r（rは$1 \leq r \leq 3$の自然数）が書かれている．また，所有しているくじ番号は図 3 に示すような形式のファイル lots.txt で与えられ，所有しているくじ番号が 1 行目から各行に一つずつ書かれている．以下の各問に答えよ．

(1) 図 2 の win.txt，図 3 の lots.txt を与えてプログラムを実行することを考える．プログラムの 36 行目で関数 functionA が呼び出されたときに，プログラム 6～13 行目の for 文処理において，i=1 および i=3 の時に，j に関する for 文が終了した時点で a[0]～a[9]および b[0]～b[9]の値が以下のようになった．8 行目の空欄(A)を配列 a に関する適切な条件式で埋めよ．

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
i=1	5308	900	7888	3500	8	4905	8698	1328	89	9003
i=3	900	3500	8	4905	5308	1328	89	7888	8698	9003

	b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]
i=1	3	3	2	3	3	1	3	3	3	3
i=3	3	3	3	1	3	3	3	2	3	3

(2) 36 行目で呼び出された関数 functionA の処理によって，配列 win および配列 grade はどのようになるか，説明せよ．またその処理の平均時間計算量（average case time complexity）を，変数 n を用いて，オーダ表記（order notation）で表わせ．その理由も答えよ．ただし，win.txt 内では，当選番号は無作為（at random）な順序で並んでいる．

(3) 39 行目で呼び出された関数 functionB はどのような処理をしているのか，配列 win，変数 lot，変数 n を用いて説明せよ．また，関数の戻り値（return value）についても言及すること．

(4) 4〜14 行目で定義されている関数 functionA を以下のように変更することで，36 行目で functionA を実行する時の平均時間計算量を少なくすることを考える．以下の各小問に答えよ．

```
void functionA(int a[], int b[], int t, int w){
 if(t<w){
  int i, j, x, tmp;
  i=t; j=w;
  x=a[t];

  while(1){
   while(a[i]<x) i++;
   while(a[j]>x) j--;
   if(i>=j) break;
   tmp=a[i]; a[i]=a[j]; a[j]=tmp; tmp=b[i]; b[i]=b[j]; b[j]=tmp;
   i++; j--;
  }
  functionA(a, b, (あ), (い));

  functionA(a, b, (う), (え));
 }
}
```

(4-1) 空欄(あ)〜(え)に入るものの組み合わせとして，適切なものを下の(ⅰ)〜(ⅳ)から一つ選び，答えよ．

	(あ)	(い)	(う)	(え)
(ⅰ)	i-1	t	w	j+1
(ⅱ)	t	j-1	i+1	w
(ⅲ)	t	i-1	j+1	w
(ⅳ)	j-1	t	w	i+1

(4-2) 関数 functionA の変更後の平均時間計算量を，変数 n を用いて，オーダ表記で表わせ．ただし，win.txt 内では，当選番号は無作為な順序で並んでいる．

(5) 図 1 のプログラムを，下線（ア）および下線（イ）で示した main 関数中の関数 functionA の引数 (argument) と if 文の条件式のみを変更し，1 等に当選している場合にのみ，当選しているくじ番号と等級を出力するようにする．当選番号と等級，および所有しているくじ番号は図 2 および図 3 と同じ形式で与えられる．39 行目の下線（イ）における判定を平均時間計算量$O(1)$で実現するためには，下線（ア）および下線（イ）をそれぞれどのように変更すればよいか，（ア）には適切な引数を，（イ）には適切な式をそれぞれ答えよ．

```
1  #include<stdio.h>
2  #define MAXN 100
3
4  void functionA(int a[], int b[], int t, int w){
5   int tmp, i, j;
6   for(i=t+1; i<=w; i++){
7    for(j=t; j<=w-i; j++){
8     if(      (A)      ){
9      tmp=a[j+1]; a[j+1]=a[j]; a[j]=tmp;
10     tmp=b[j+1]; b[j+1]=b[j]; b[j]=tmp;
11    }
12   }
13  }
14 }
15
16 int functionB(int a[], int x, int n){
17  int t, w, m;
18  t=0; w=n-1;
19  do{
20   m=(t+w)/2;
21   if(x<a[m]) w=m-1;
22   else t=m+1;
23  } while(t<=w);
24  if(w>=0 && x==a[w]) return w;
25  else return -1;
26 }
27
28 int main(){
29  int n, i, k;
30  FILE *fp;
31  int win[MAXN], grade[MAXN], lot;
32  fp=fopen("win.txt", "r");
33  fscanf(fp,"%d", &n);
34  for(i=0; i<n; i++) fscanf(fp, "%d %d", &win[i], &grade[i]);
35  fclose(fp);
36  functionA(win, grade, 0, n-1);
                  (ア)
37  fp=fopen("lots.txt", "r");
38  while(fscanf(fp, "%d", &lot)!=EOF){
39   if((k=functionB(win, lot, n))!=-1){
                  (イ)
40    printf("***winning number: %4d, grade: %d\n", lot, grade[k]);
41   }
42  }
43  fclose(fp);
44  return 0;
45 }
```

図 1 プログラム

```
10
5308    3
7888    2
900     3
8698    3
3500    3
8       3
4905    1
9003    3
1328    3
89      3
```

図 2 win.txt

```
9003
7888
356
28
2457
4905
43
29
3500
81
48
444
314
1028
777
```

図 3 lots.txt

2 【必須問題】計算機システムとシステムプログラム （情報工学 4/16）

配点： (1-1) 10 点，(1-2) 12 点，(1-3-1) 12 点，(1-3-2) 8 点，(1-4-1) 6 点，(1-4-2) 12 点
(2-1) 20 点，(2-2) 10 点，(2-3-1) 15 点，(2-3-2) 20 点

(1) 計算機 (computer) における整数 (integer) の表現 (representation) と算術演算 (arithmetic operation) に関する以下の各小問に答えよ．解答は，全て解答用紙の太線枠内に書くこと．

(1-1) 10 進数 (decimal number) の -15 を 8 [ビット] の符号絶対値表現 (signed magnitude representation) および 1 の補数表現 (one's complement representation) で表した場合のビット列 (bit string) を示せ．

(1-2) 8 [ビット] の 2 の補数表現 (two's complement representation) で表すことのできる正 (positive) の最大値 (maximum value) および負 (negative) の最小値 (minimum value) のビット列を示せ．また，それらを 10 進数の数値で示せ．

(1-3) 計算機で符号付き整数 (signed integer) の加算を行なう時，多くの場合，2 の補数表現が用いられる．2 の補数表現を用いた加算に関する以下の (1-3-1), (1-3-2) に答えよ．

(1-3-1) $43+(-5)$ の加算を 8 [ビット] の 2 の補数表現を用いて行う過程を示せ．最上位ビットからの桁上げ (carry) 出力の扱いも記すこと．

(1-3-2) 計算機上で符号付き整数の加算に 2 の補数表現を用いる利点を一つ示せ．

(1-4) 二つの整数 A, B を加算し，整数 S を得る演算を考える．A, B, S は同じビット長とする．A, B, S のビット列の最上位ビットをそれぞれ a, b, s とし，加算器の最上位ビットからの桁上げ出力を c とする．このとき，以下の (1-4-1), (1-4-2) に答えよ．

(1-4-1) A, B, S が符号無し整数の場合，オーバーフロー (overflow) の発生を判定する方法を，変数 a, b, s, c を用いて示せ．なお，必要の無い変数は使わなくて良い．

(1-4-2) A, B, S が 2 の補数表現で表された符号付き整数の場合，オーバーフローの発生を判定する方法を，変数 a, b, s, c を用いて示せ．なお，必要の無い変数は使わなくて良い．

(2) 単一プロセッサ (single processor) のマルチタスク (multitask) 環境における排他制御 (exclusive control) に関する以下の各小問に答えよ．解答は，全て解答用紙の太線枠内に書くこと．

(2-1) 排他制御を実現する際には，デッドロック状態 (deadlock) や飢餓状態 (starvation) になることを避ける必要がある．それぞれの状態について説明せよ．

(2-2) 以下に示すプロセス 1 とプロセス 2 は，右図に示すようなスタック (stack) の操作を実現する．プロセス 1 はスタックへのデータのプッシュ (push) を行い，プロセス 2 はスタックからのデータのポップ (pop) を行う．スタックには事前にデータ $d_1, d_2, \cdots, d_n$ が格納されている．top はスタックの先頭 (stack top) のアドレスを表す変数であり，二つのプロセスが共有している．top は事前に x に設定されている．stack() は引数 (argument) で指定されたアドレスのデータを表す．push_item と pop_item はデータを表す変数であり，push_item には事前に値が設定されている．スタックはプロセス 1 とプロセス 2 のみが共有しており，スタックの操作を実行するプロセスは他に存在しない．

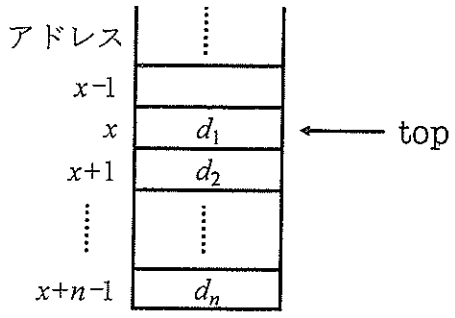


図: スタック

```
（ア） top = top - 1;                （ウ） pop_item = stack(top);
（イ） stack(top) = push_item;       （エ） top = top + 1;
       プロセス 1                           プロセス 2
```

プロセス 1 とプロセス 2 が並行に動作した時に，二つのプロセスの命令実行とプロセス切り換えのタイミングによっては，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作のいずれか，あるいは両方が正しく行われないことがある．そのような (ア)(イ)(ウ)(エ) の実行順序 (execution order) のうち，(ア) から開始されるものを一つ示せ．

(2-3) セマフォ (semaphore) を用いて，(2-2) におけるプロセス 1 とプロセス 2 が並行に動作した時においても，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作が正しく行われるようにすることを考える．ここで，セマフォは 0 または 1 の値を取るものとする．P() はセマフォを引数とし，セマフォが 0 であれば P() を実行したプロセスを休止させ，1 であればセマフォを 1 から 0 にする不可分命令 (atomic operation, atomic transaction) である．V() はセマフォを引数とし，同じセマフォを引数として実行された P() によって休止されたプロセスがあれば起動し，なければセマフォを 0 から 1 にする不可分命令である．このとき，以下の (2-3-1), (2-3-2) に答えよ．

(2-3-1) プロセス 1 とプロセス 2 が並行に動作した時においても，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作が正しく行われるように，P(), V(), 及びセマフォ a を用いて，プロセス 1 とプロセス 2 に適切な命令を追加したものを示せ．また，a の適切な初期値 (initial value) も示せ．

(2-3-2) 以下に示すプロセス 3 とプロセス 4 は，変数 m を共有している．プロセス 3 は m の値をインクリメント (increment) し続けるプロセスであり，プロセス 4 は m の値を表示し続けるプロセスである．m は事前に 1 に初期化されており，m を共有するプロセスは他に存在しない．

```
while (true) {                  while (true) {
 m = m + 1;                      print(m);
}                               }
プロセス 3                      プロセス 4
```

プロセス 3 とプロセス 4 が並行に動作した時に，プロセス 4 における m の値の表示が 1, 2, 3, … となるようにしたい．P(), V(), 及び二つのセマフォ b, c を用いて，プロセス 3 とプロセス 4 に適切な命令を追加したものを示せ．また，b, c の適切な初期値も示せ．

3 【選択問題】離散構造

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

(1) 述語論理（predicate logic）に関する以下の各小問に答えよ．ただし，論理式（logic formula）の記述には以下の記号を用いる．$\forall$，$\exists$ はそれぞれ全称作用素（universal quantifier），存在作用素（existential quantifier）であり，$\Leftrightarrow$，$\rightarrow$，$\wedge$，$\vee$，$\neg$ はそれぞれ等価（equivalence），含意（implication），論理積（conjunction，and），論理和（disjunction，or），否定（negation，not）を表す論理演算子である．また，論理式集合（a set of logical formulae）Γ から論理式 P が証明可能（provable）であるとき，$\Gamma \vdash P$ と書くこととする．特に断りのない限り，x を変数（variable）記号，p，q を述語（predicate）記号として用いる．

論理式 $E = (\forall x(p(x) \rightarrow q(x)) \wedge \exists x\ p(x)) \rightarrow \exists x\ q(x)$ が恒真（valid）であることを，2 通りの方法で証明する．以下の各小問に答えよ．

(1-1) 公理系（axioms）に基づいた証明を考える．ここでは，次の述語論理の公理系 S を用いる．

［公理系 S］ P, Q, R で任意の論理式を表す．また，$P(x)$ は論理式 P 内に出現する可能性のある変数 x を明記した記法であり，$P(t)$ はその変数 x に項（term）t を代入（substitute）して得られる論理式を表す．S の公理（axiom）は次の $A1$〜$A4$ の 4 つである．

$A1$： $P \rightarrow (Q \rightarrow P)$

$A2$： $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$

$A3$： $(\neg P \rightarrow \neg Q) \rightarrow ((\neg P \rightarrow Q) \rightarrow P)$

$A4$： $\forall x P(x) \rightarrow P(t)$　　ただし，t は $P(x)$ において変数 x に代入可能な項．

また，S の推論規則（inference rule）は次の $B1$，$B2$ である．

$B1$： P と $P \rightarrow Q$ から Q を導く．

$B2$： $P \rightarrow Q$ から $P \rightarrow \forall x Q$ を導く．ただし，P は自由変数 x を含まない．

本小問の証明では，以下の定理 $T1$〜$T3$ と演繹定理（deduction theorem）$D1$ は証明無しに用いてもよい．

$T1$： $(P \wedge Q) \vdash P$

$T2$： $(P \wedge Q) \vdash Q$

$T3$： $\vdash (P \rightarrow Q) \rightarrow (\exists x P \rightarrow \exists x Q)$

$D1$： P が閉論理式（closed formula）のとき，$\Gamma, P \vdash Q$ ならば $\Gamma \vdash (P \rightarrow Q)$

以下の (1-1-1) および (1-1-2) に答えよ．

(1-1-1) 公理系 S のもとで，次の定理 $T4$ を証明せよ．

$T4$： $\forall x P(x) \vdash P(x)$

(1-1-2) 公理系 S のもとで，論理式 E を証明せよ．ただし，定理 $T4$ も証明無しに用いてもよい．

(1-2) 論理式 E が恒真であることを示すため，まず，E の否定をスコーレム化（Skolemization）し，次いで，導出原理（resolution principle）を適用し，その式が充足不能であることを示す．以下の (1-2-1)〜(1-2-3) に答えよ．

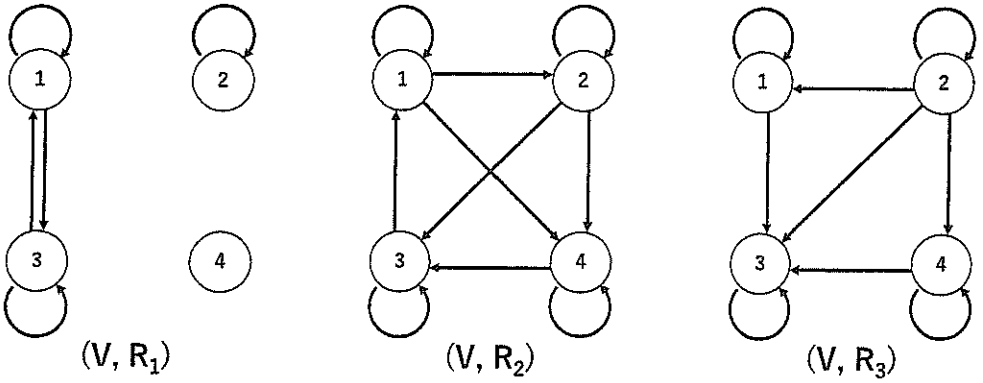
(1-2-1) $\neg E$ の冠頭標準形（prenex normal form）を示せ．冠頭標準形は，すべての限定作用素（quantifier）が先頭にある閉論理式である．ただし，閉論理式は連言標準形（和積標準形：conjunctive normal form）で示すこと．

(1-2-2) 上記の (1-2-1) で得た論理式のスコーレム連言標準形（Skolem conjunctive normal form）$\neg E'$ を求めよ．

(1-2-3) 導出原理を用いて $\neg E'$ が充足不能であることを示せ．

(2) 空でない有限集合（finite set）上の二項関係（binary relation）に関し，以下の各小問に答えよ．

(2-1) 有向グラフ（directed graph）(V,R)（$R \subseteq V \times V$）において，$R = \{(s,t) \mid s,t \in V; s$ から t への有向辺（directed edge）が存在 $\}$ であるとき，R は有限集合 V 上の二項関係を表す．以下の有向グラフで表される $V = \{1,2,3,4\}$ 上の二項関係 R_1，R_2 および R_3 について，以下の (2-1-1) および (2-1-2) に答えよ．



(2-1-1) R_1，R_2 および R_3 はそれぞれ
(a) 反射律（reflexive law）
(b) 対称律（symmetric law）
(c) 反対称律（antisymmetric law）
(d) 推移律（transitive law）
のうち，どの性質を満たすか．それぞれについて，記号 (a)~(d) の中から満たすものをすべて答えよ．

(2-1-2) R_1，R_2 および R_3 のうち，順序関係（partial order relation）であるものをすべて答えよ．

(2-2) 順序関係が定義された有限集合を順序集合（partially ordered set）とよび，順序関係 $\preceq$ が定義された有限集合 V を $(V,\preceq)$ で表す．任意の順序集合 $(V,\preceq)$ と 集合 A（$A \subseteq V$）に対し，以下を定義する．

$upper(A) = \{t \mid \forall s \in A; s \preceq t\}$ を A の上界（upper bound）とよぶ．

$upper(A)$ に含まれる A の要素が存在すれば，それを A の最大元（maximum）とよぶ．

$lower(A) = \{s \mid \forall t \in A; s \preceq t\}$ を A の下界（lower bound）とよぶ．

$lower(A)$ に含まれる A の要素が存在すれば，それを A の最小元（minimum）とよぶ．

ある有限集合 X のべき集合（power set）$\mathcal{P}(X)$，およびその上での二項関係 $\preceq_{\mathcal{P}(X)} = \{(s,t) \mid s,t \in \mathcal{P}(X);\ s \subseteq t\}$ について，以下の (2-2-1)~(2-2-3) に答えよ．なお，$\cap$ および $\cup$ は，それぞれ二つの集合の共通集合（intersection）および和集合（union）を求める演算子である．

(2-2-1) $(\mathcal{P}(X), \preceq_{\mathcal{P}(X)})$ が順序集合であることを示せ．

(2-2-2) 任意の $s,t \in \mathcal{P}(X)$ に対し，$w \in upper(\{s,t\})$ であり，かつ要素数が最小の w を，s および t を用いて表せ．また，$z \in lower(\{s,t\})$ であり，かつ要素数が最大の z を，s および t を用いて表せ．

(2-2-3) 任意の $u \in \mathcal{P}(X)$ に対し，$c(u)$ は $\mathcal{P}(X)$ の要素のうち，$u \cup c(u)$ が $\mathcal{P}(X)$ の最大元，$u \cap c(u)$ が $\mathcal{P}(X)$ の最小元であるものとする．このとき，任意の $s,t \in \mathcal{P}(X)$ に対し，$c(s \cup t) = c(s) \cap c(t)$ であることを示せ．

4 【選択問題】計算理論

配点：(1-1) 5 点, (1-2) 5 点, (1-3) 35 点, (1-4) 20 点, (2-1) 35 点, (2-2) 25 点

(1) アルファベット (alphabet) $\Sigma_{ab} = \{a, b\}$ で構成される回文 (palindrome) を考える．回文とは前から読んでも後ろから読んでも同じ文字列 (string) のことである．例えば，b, aa, aba, $baab$ は回文だが，ab, abb, $baaa$ は回文ではない．なお，文字列 w の長さを $|w|$ と表す．例えば，$|b| = 1$，$|abb| = 3$ である．また，空列 (empty string) ε ($|\varepsilon| = 0$) は回文である．

回文を受理するオートマトン (automaton) に関する以下の各小問に答えよ．

(1-1) 決定性有限オートマトン (deterministic finite automaton) は $(Q, \Sigma, \delta, q_0, F)$ で表される．Q は状態 (state) の有限集合，Σ は入力アルファベット (input alphabet)，δ は遷移関数 (transition function)，q_0 は開始状態 (start state)，F は最終状態 (final state) の集合である．言語 (language) $\{p \in \Sigma_{ab}^* \mid |p| = 2,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンの状態遷移図 (state transition diagram) を書け．なお，下図に示すように開始状態には太い矢印 (thick arrow) を付与し，最終状態は二重丸 (double circle) で表現すること．下図では，開始状態は q_0，最終状態は q_2 である．

(1-2) 言語 $\{p \in \Sigma_{ab}^* \mid |p| = 3,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンの状態遷移図を書け．なお，下図に示すように開始状態には太い矢印を付与し，最終状態は二重丸で表現すること．

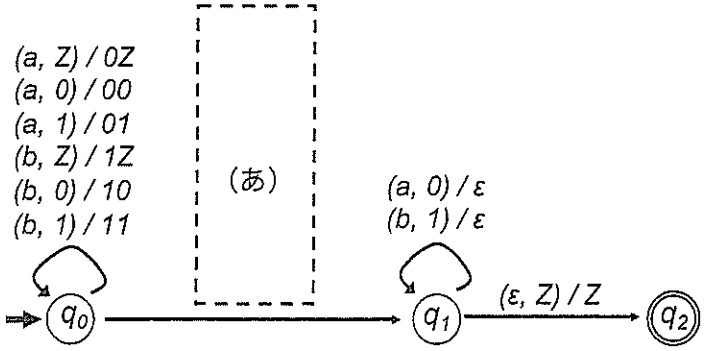
(1-3) 言語 $\{p \in \Sigma_{ab}^* \mid |p| \geq 0,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンは存在しないこと，すなわち，この言語は正則言語 (regular language) ではないことを背理法 (proof by contradiction) により証明せよ．なお，以下の正則言語に対する反復補題 (pumping lemma) を用いること．

正則言語に対する反復補題

L を正則言語とすると，次の条件を満たす正の整数 n が存在する：$v \in L$, $|v| \geq n$ なら v は $v = xyz$ ($|xy| \leq n, |y| \geq 1$) と分解でき，任意の $k \geq 0$ に対して $xy^k z \in L$ である．

(1-4) プッシュダウンオートマトン (pushdown automaton) は $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ で表される．Q は状態の有限集合，Σ は入力アルファベット，Γ はスタックアルファベット (stack alphabet)，δ は遷移関数，q_0 は開始状態，Z はスタックの開始記号，F は最終状態の集合である．最終状態による受理 (acceptance by final state) を行うプッシュダウンオートマトンを利用すれば，言語 $\{p \in \Sigma_{ab}^* \mid |p| \geq 0,\ p \text{ は回文}\}$ を受理できる．以下はそのような非決定性プッシュダウンオートマトンの状態遷移図である．状態遷移図の (あ) に必要な動作を，$(r, s)/t$ の形式ですべて答えよ．

ただし，$(r, s)/t$ は，入力から r を読み出すときにスタックの先頭にある記号 s を取り去り，t をスタックに押し込むことを意味する．t が複数の記号の場合は右側の記号から順にスタックに押し込む．例えば，下図の $(a, Z)/0Z$ は，a を読み出すときにスタックの先頭から Z を取り去り，スタックに Z と 0 をこの順で押し込むことを意味する．t が ε の場合は，スタックに記号を押し込まない．r が ε の場合は，入力から記号を読み出さないで遷移を行う．スタックアルファベットは $\Gamma = \{Z, 0, 1\}$ とする．



(2) 文脈自由文法 (context-free grammar) は一般に $G = (V, T, P, S)$ で定められる．ここで V は変数 (variable; 非終端記号 nonterminal symbol) の集合，T は終端記号 (terminal symbol) の集合，P は生成規則 (production rule) の集合，S は出発記号 (start symbol) であり，$S \in V$ である．文脈自由文法 $G_1 = (V_1, T_1, P_1, S_1)$ を $V_1 = \{A\}$，$T_1 = \{\mathtt{a}, \mathtt{b}\}$，$P_1 = \{A \to \mathtt{a}A\mathtt{b}A,\ A \to \mathtt{b}A\mathtt{a}A,\ A \to \varepsilon\}$，$S_1 = A$ と定める．L_1 を G_1 により生成される言語 (language) とする．L_1 は同じ個数 (the same number) の a と b を含む文字列 (string) すべてからなる言語であることを証明したい．以下の各小問に答えよ．

【注】文字列とは終端記号の列である．任意の文字列 w に対し，記法 $|w|$ は w の中の終端記号すべての個数を，$|w|_{\mathtt{a}}$ は w の中の終端記号 a の個数を，$|w|_{\mathtt{b}}$ は w の中の終端記号 b の個数を，それぞれ表す．例えば $w = \mathtt{aaba}$ に対して，$|w| = 4$，$|w|_{\mathtt{a}} = 3$，$|w|_{\mathtt{b}} = 1$ である．ε は空列 (empty string) を表す．記法 $\alpha \Rightarrow \beta$ は文法 G_1 の生成規則を 1 回適用することで文形式 (sentential form) α から文形式 β が得られることを表す．

(2-1) L_1 に含まれている任意の文字列 w に対して w は同じ個数の a と b を含んでいること，すなわち必要条件 $\forall w \in T_1^*[w \in L_1$ ならば $|w|_{\mathtt{a}} = |w|_{\mathtt{b}}]$ を証明したい．空欄（ア）を埋めよ．

証明: 生成規則を適用する回数 k に関する数学的帰納法 (mathematical induction) で証明する．

- 基底段階 (base case): $k = 1$ の場合．1 回の生成規則の適用で文字列を得る導出 (derivation) は $A \Rightarrow \varepsilon$ のみである．$w = \varepsilon$ とすれば $|\varepsilon|_{\mathtt{a}} = |\varepsilon|_{\mathtt{b}} = 0$ なので，$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ が成立する．
- 帰納段階 (inductive step): $k > 1$ の場合．$k-1$ 回以下の生成規則の適用で得られる文字列 v はいずれも $|v|_{\mathtt{a}} = |v|_{\mathtt{b}}$ が成立すると仮定する．すると以下の理由により $|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ が成立する．

理由：［　　（ア）　　］

(証明終)

(2-2) 同じ個数の a と b を含んでいる任意の文字列 w に対して w は L_1 に含まれていること，すなわち十分条件 $\forall w \in T_1^*[|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ ならば $w \in L_1]$ を証明したい．空欄（イ）～（エ）を埋めよ．

証明: 文字列 w の長さ $|w|$ に関する帰納法で証明する．$|w|$ は 0 以上の偶数 (even number) なのは明らかである．

- 基底段階: $|w| = 0$ の場合．以下の理由により $w \in L_1$ が成立する．

理由：［　　（イ）　　］

- 帰納段階: $|w| > 0$ かつ偶数の場合．長さが偶数で $|w|$ 未満の文字列 v が $|v|_{\mathtt{a}} = |v|_{\mathtt{b}}$ を満たすならば，$v \in L_1$ であると仮定する．w の最初（左端）の終端記号により以下の場合分けをする．
  - a の場合．$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ のとき，長さが $|w|$ 未満のある文字列 $v_1, v_2 \in L_1$ が存在して $w = \mathtt{a}v_1\mathtt{b}v_2$ と書ける事が示せる（証明略）．従って，導出 $A \Rightarrow$［　（ウ）　］$\Rightarrow \cdots \Rightarrow \mathtt{a}v_1\mathtt{b}v_2 = w$ により $w \in L_1$ が成立する．
  - b の場合．$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ のとき，長さが $|w|$ 未満のある文字列 $v_1, v_2 \in L_1$ が存在して $w = \mathtt{b}v_1\mathtt{a}v_2$ と書ける事が示せる（証明略）．従って，導出 $A \Rightarrow$［　（エ）　］$\Rightarrow \cdots \Rightarrow \mathtt{b}v_1\mathtt{a}v_2 = w$ により $w \in L_1$ が成立する．

(証明終)

配点: (1-1-1) 6 点，(1-1-2) 6 点，(1-2-1) 15 点，(1-2-2) 6 点，(1-2-3) 12 点，(1-3) 15 点
(2-1) 20 点，(2-2) 15 点，(2-3-1) 10 点，(2-3-2) 10 点，(2-3-3) 10 点

(1)　次のような記憶がない情報源 (memoryless information source) S を考える．情報源 S の記号数 s は 2 以上とし，情報源記号は $a_1, a_2, \ldots, a_s$，各 i $(1 \leq i \leq s)$ について記号 a_i の生起確率 (occurence probability) は 2^{-m_i} で表される．ここで，$m_1, m_2, \ldots, m_s$ は，$1 \leq m_1 \leq m_2 \leq \cdots \leq m_s$ を満たす整数である．このとき，各 i $(1 \leq i \leq s)$ について記号 a_i に対する符号語長 (code length) が m_i である瞬時に復号可能な符号化 (instantly decodable source coding) が存在することについて以下の各小問に答えよ．

(1-1)　特別な場合として，$s = 6$，$m_1 = 1$，$m_2 = 2$，$m_3 = m_4 = m_5 = m_6 = 4$ である情報源 S_0 を考える．

(1-1-1)　この場合に題意を満たす符号化 (下線部) の復号木 (decoding tree) を一つ示せ．通信路記号は 0, 1 とせよ．なお，復号木は符号の木 (code tree) とも呼ばれている．

(1-1-2)　上で復号木を示した符号化の平均符号語長，および情報源 S_0 の 2 を底とするエントロピー (entropy) を求めよ．

(1-2)　一般の場合に題意を満たす符号化 (下線部) が存在することを，記号数 s に関する数学的帰納法 (mathematical induction) により証明したい．

(1-2-1)　証明の準備として，以下の文章の各空欄を埋めることにより，常に，$m_{s-1} = m_s$ であることを示せ．なお，空欄 （い） ～ （え） には，それぞれ「偶数」または「奇数」のいずれかの語を埋めよ．

$2^{m_s} \sum_{i=1}^{s} 2^{-m_i}$ = （あ） である．$m_s \geq 1$ なので，$2^{m_s} \sum_{i=1}^{s} 2^{-m_i}$ は （い） である．$m_{s-1} \neq m_s$ と仮定すれば，$2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i}$ は （う） であり，これらの差 $2^{m_s} \sum_{i=1}^{s} 2^{-m_i} - 2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i}$ は （え） となるが，差の値

$$2^{m_s} \sum_{i=1}^{s} 2^{-m_i} - 2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i} = \boxed{\text{（お）}}$$

であり矛盾する．

(1-2-2)　証明の基底段階 (base case) を示せ．すなわち，$s = 2$ のときに題意を満たす符号化が存在することを示せ．

(1-2-3)　証明の帰納段階 (inductive step) を示せ．

(1-3)　以下の文章の各空欄を埋めよ．なお，空欄 （う） には人名 (姓 family name) を埋めよ．また，空欄 （お） には「がある」または「はない」のどちらかを埋めよ．

一般の場合に，情報源 S の 2 を底とするエントロピーは （あ） である．小問 (1-2) で存在を示した符号化の平均符号語長は （い） であり，（う） の （え） 定理から，S の n 次拡大 (n-th extension) に対する符号化を考えた場合に，一記号あたりの平均符号語長がより小さい符号化が存在する可能性 （お）．

(2)　図 1 に概要を示すディスタンスベクトル型ルーティングアルゴリズム (distance vector routing algorithm) を用いて最小コスト経路 (minimum cost route) を決めることを考える．これについて以下の各小問に答えよ．

記号

$\mathcal{N}$　全ノードの集合．

$c(i,j)$　ノード i からノード j へのリンクのコスト．$c(i,j) \geq 0$，$c(i,i) = 0$ とする．また，ノード i からノード j へリンクがない場合は $c(i,j) = \infty$ とする．

$d_i(j)$　ノード i において，その時点でノード i が知るノード i からノード j までの最小コスト経路のコストを保持する変数．

D_i　ノード i の持つディスタンスベクトル．$\mathcal{N} = \{x, y, z\}$ の場合，D_i は以下の通りとする．

$D_i = [d_i(x)$ の値$, d_i(y)$ の値$, d_i(z)$ の値$]$

初期状態

- 各ノード i において $c(i,j)$ は既知である $(\forall j \in \mathcal{N})$.
- 各ノード i において $d_i(j) = c(i,j)$ である $(\forall j \in \mathcal{N})$.

動作

全ノードは，各ステップで以下の動作 I，動作 II のそれぞれを同期的 (synchronously) に実行し，そのステップを繰り返し実行する．

動作 I.　各ノード i は，隣接 (adjacent) ノードにディスタンスベクトル D_i を送信し，自身に送信されたディスタンスベクトルを受信する．

動作 II.　各ノード i は，受信した全てのディスタンスベクトルにもとづいて，自身のディスタンスベクトル D_i を更新する．

経路の収束 (convergence)

前ステップと現ステップの間で D_i $(\forall i \in \mathcal{N})$ の値に変化がなくなることを経路が収束すると定義する．最小コスト経路は，収束したときの各ノードが持つ情報から導出できる．

図 1: ディスタンスベクトル型ルーティングアルゴリズムの概要

(2-1)　図 1 の動作 II の下線部における D_i の更新方法を数式で記述せよ．記述に必要な記号を新たに定義しても構わないが，定義した記号の説明も解答に含めよ．

(2-2)　図 2 に示すネットワークを考える．リンクのコストは，それぞれ，$c(x,y) = c(y,x) = 2$，$c(x,z) = c(z,x) = 8$，$c(y,z) = c(z,y) = 3$ である．また，動作 I では送信したディスタンスベクトルは紛失なく届くものとする．このとき，図 1 の初期状態の直後の更新をステップ 1 として，経路が収束するまでの各ステップについて，動作 I においてノード x が受信したディスタンスベクトル D_y と D_z，および，動作 II において更新した後のノード x のディスタンスベクトル D_x を記述せよ．ただし，D_i $(i \in \{x,y,z\})$ は，$D_i = [d_i(x)$ の値, $d_i(y)$ の値, $d_i(z)$ の値$]$ の表記法で記述せよ．

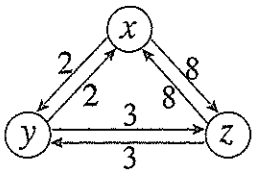


図 2: ネットワーク

(2-3)　ディスタンスベクトル型ルーティングアルゴリズムでは，リンクが切断した際に経路の収束までに非常に長い時間を要するという問題 (count-to-infinity 問題) がある．この問題について以下の (2-3-1)，(2-3-2)，(2-3-3) に答えよ．ただし，ここでは，リンクが切断された場合，ディスタンスベクトルは切断されたリンクを介して到達しないものとする．

(2-3-1)　図 3 のネットワークを考える．リンクのコストは，$c(x,y) = c(y,x) = c(y,z) = c(z,y) = 1$ である．このネットワーク上で経路が収束している状態でのノード y のディスタンスベクトル D_y は，以下の通りである．

$$D_y = [d_y(x) \text{ の値},\ d_y(y) \text{ の値},\ d_y(z) \text{ の値}] = [1, 0, 1]$$

経路が収束した状態でノード x とノード y 間のリンクが切断した $(c(x,y) = c(y,x) = \infty)$ とする．リンク切断が発生した直後の更新をステップ 1 として，図 1 の動作にしたがって経路を更新したときのステップ 1 からステップ 3 までの各ステップ終了時の $d_y(x)$ の値を示せ．

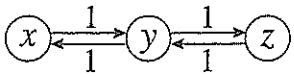


図 3: ネットワーク

(2-3-2)　(2-3-1) の $d_y(x)$ をふまえて，ディスタンスベクトル型ルーティングアルゴリズムで，count-to-infinity 問題が発生する理由を説明せよ．

(2-3-3)　リンクステート型ルーティングアルゴリズム (link state routing algorithm) では，count-to-infinity 問題が発生しない理由を説明せよ．

6 【選択問題】電子回路と論理設計 （情報工学 13/16）

配点：(1-1) 20 点, (1-2) 25 点, (2-1) 15 点, (2-2) 20 点, (3-1) 15 点, (3-2) 12 点, (3-3) 18 点

(1) 図 1 に示す MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) を使った 2 入力 (input)1 出力 (output) の CMOS 型回路 (CMOS circuit) では以下を満たすとする．

- $V_{\mathrm{DD}}(>0)$ は電源電圧 (source voltage) である．
- C_1，C_2 は負荷容量 (load capacitance) である．
- x，y は入力，z は出力である．
- x，y は V_{DD} もしくは 0 の値しか取らない．
- nMOS はゲート・ソース間電圧 (gate-source voltage)$V_{\mathrm{GS}} \geq 0.5V_{\mathrm{DD}}$ のときオンになり，$V_{\mathrm{GS}} < 0.5V_{\mathrm{DD}}$ のときオフとなる．
- pMOS はゲート・ソース間電圧 $V_{\mathrm{GS}} \geq 0.5V_{\mathrm{DD}}$ のときオフになり，$V_{\mathrm{GS}} < 0.5V_{\mathrm{DD}}$ のときオンとなる．
- すべての MOSFET のドレイン (drain)・ソース (source) 間の抵抗値 (resistance) はオンにすると瞬時に R_{MOS} となり，オフにすると瞬時に ∞ になる．
- 配線 (wire) の負荷容量，抵抗値は 0 とする．

以下の各小問に答えよ．

(1-1) 図 1 の CMOS 型回路は，ある論理ゲート (logical gate) を実現したものである．図 1 の CMOS 型回路に相当する論理ゲートを図 2 の 1〜5 から 1 つ選択せよ．

(1-2) 図 1 の CMOS 型回路において，x が 0 で y が V_{DD} になって十分に長い時間が経った後に，x が V_{DD} に変化してから z における出力電圧 (output voltage) が $0.5V_{\mathrm{DD}}$ になるまでの時間 T を V_{DD}，R_{MOS}，C_1，C_2 を用いて示せ．ただし，V_{DD}，R_{MOS}，C_1，C_2 の中で不要な変数は用いなくてよい．

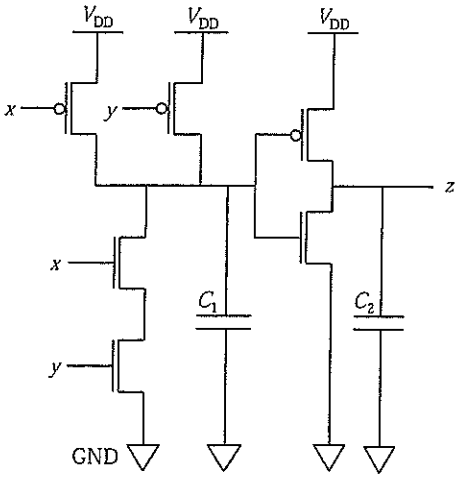


図 1: 2 入力 1 出力 CMOS 型回路

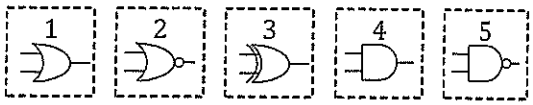


図 2: 論理ゲート

(2) $K(0 \leqq K \leqq 3)$ ビットの左論理シフト (logical left shift) を行う組み合わせ回路 (combinational logic circuit) を設計する．図 3 に設計する回路の回路図を示す．この回路は以下を満たすとする．

- $X = (x_3, x_2, x_1, x_0)$ は 4bit 符号なし 2 進数 (unsigned binary number) であり，入力である．
- $Y = (y_3, y_2, y_1, y_0)$ は 4bit 符号なし 2 進数であり，出力である．
- x_0，y_0 を最下位ビット (LSB: Least Significant Bit) とする．
- シフト量 K は p_3，p_2，p_1，p_0 の 4 個のスイッチによって選択可能であり，p_K をオン $(= 1)$ にすると K ビットシフトする．ただし，複数のスイッチがオンになった場合は，オンになった p_K のうち最大の K をシフト量とする．
- セレクタ 1 は，選択信号 s_0 が 0 の時は A_0 を，1 の時は B_0 を出力する回路である．
- セレクタ 2 は，選択信号 s_1 が 0 の時は A_1 を，1 の時は B_1 を出力する回路である．
- 左 1 ビット論理シフタと左 2 ビット論理シフタでは，シフトによって空いたビットに 0 を格納する．
- 論理回路 1 は K ビットシフトを行うために，スイッチからの入力 p_3，p_2，p_1，p_0 をセレクタ 1，2 への選択信号 s_1，s_0 に変換する回路である．

以下の各小問に答えよ．

(2-1) $X = (0,0,0,1)$ に対して，スイッチを $(p_3, p_2, p_1, p_0) = (0,0,1,1)$ とすると，$Y = (0,0,1,0)$ となる．$X = (0,0,0,1)$ に対して，スイッチを $(p_3, p_2, p_1, p_0) = (0,1,1,0)$ とした時の Y を示せ．

(2-2) 図 3 の論理回路 1 の出力 s_1，s_0 を，入力 p_3，p_2，p_1，p_0 を用いた最簡積和形 (最小積和形, minimal sum-of-products expression) の論理式 (logical expression) で示せ．

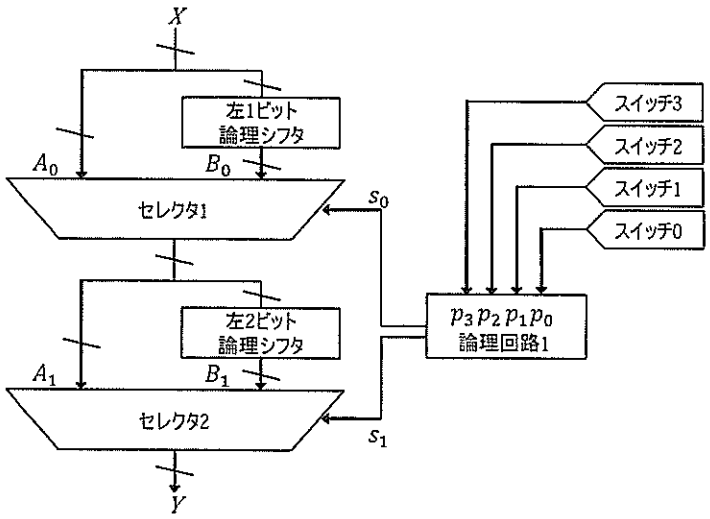


図 3: 回路図

(3) 0〜3 のいずれかの値を保持する 4 進のアップダウンカウンタ (up/down counter) を設計する．作成するカウンタは以下を満たすとする．

- Moore 型の順序機械 (Moore machine) とする．
- 2 個のエッジトリガ型の D フリップフロップ (edge-triggered D flip-flop) を用いる．
- 各フリップフロップの入力をそれぞれ D_1，D_0，出力を Q_1，Q_0 とする．
- この順序機械の状態 (state) は，カウンタ値が 0 の時には S_0，1 の時には S_1，2 の時には S_2，3 の時には S_3 が割り当てられている．
- 制御信号 (x_1, x_0) とクロック clk を入力，カウンタ値 (q_1, q_0) を出力とする．
- カウンタ値 (q_1, q_0) を 2 進数として q_0 を最下位ビットで表現する．
- カウンタ値の初期値 (initial value) は 0 とする．
- Q_1 と q_1，Q_0 と q_0 がそれぞれ接続されている．
- $(x_1, x_0) = (0, 0)$ のとき，現在の値を次の時刻でも保持する．
- $(x_1, x_0) = (0, 1)$ のとき，現在の値に 1 を加算 (add) した値を次の時刻で保持する．ただし，現在の値が 3 のときは次の時刻で 0 を保持する．
- $(x_1, x_0) = (1, 0)$ のとき，現在の値から 1 を減算 (subtract) した値を次の時刻で保持する．ただし，現在の値が 0 のときは次の時刻で 3 を保持する．
- $(x_1, x_0) = (1, 1)$ は禁止入力である．

以下の各小問に答えよ．

(3-1) カウンタの状態遷移図 (state transition diagram) を示せ．ただし，太い矢印が指し示している状態が開始状態 (start state) であるとする．

(3-2) (3-1) で作成した状態遷移図を基に，カウンタの状態遷移表 (state transition table) を作成する．表 1 に状態遷移表を示す．表 1 の (A)〜(L) に入る状態をそれぞれ S_0〜S_3 で示せ．

(3-3) (3-2) で作成した状態遷移表を基に，各フリップフロップの入力 D_1，D_0 を，カウンタへの入力 x_1，x_0 および各フリップフロップの出力 Q_1，Q_0 を用いた最簡積和形の論理式で示せ．

表 1: 状態遷移表

現在の状態	次の状態		
	$(x_1, x_0) = (0, 0)$	$(x_1, x_0) = (0, 1)$	$(x_1, x_0) = (1, 0)$
S_0	(A)	(B)	(C)
S_1	(D)	(E)	(F)
S_2	(G)	(H)	(I)
S_3	(J)	(K)	(L)

7 【選択問題】数学解析と信号処理 （情報工学 16/16）

配点: (1-1)25 点, (1-2)20 点, (1-3)30 点, (2-1)15 点, (2-2)15 点, (2-3)20 点

以下の各問に答えよ.

(1) 以下の各小問に答えよ.

(1-1) 以下の通り定義されるガンマ関数 (Gamma function) を考える.

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$$

ただし, z は実部が正 (positive real part) である複素数 (complex number) とする. このとき $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ であることを証明せよ.

(1-2) $t > 0$, s を複素変数 (complex variable) とする.

$$f(t) = \frac{1}{\sqrt{t}}$$

であるとき, $f(t)$ のラプラス変換 (Laplace transform) $\mathcal{L}[f(t)]$ が $\sqrt{\frac{\pi}{s}}$ であることを証明せよ.

(1-3) 以下の式の逆ラプラス変換 (inverse Laplace transform) $\mathcal{L}^{-1}[G(s)]$ を求めよ.

$$G(s) = \frac{1}{\sqrt{s}(s-1)}$$

(2) 離散時間 (discrete-time) の線形時不変システム (linear time-invariant system) について, 以下の各小問に答えよ. ただし, n は整数 (integer), z は複素変数 (complex variable) である.

(2-1) 以下の離散時間信号 (discrete-time signal) $x(n)$ の z 変換 (z-transform) $X(z)$ を求めよ. ここで $\delta(n)$ は単位インパルス (unit impulse) を意味する.

$$x(n) = \delta(n) - \delta(n-1) + 2\delta(n-2)$$

(2-2) あるシステムに離散時間信号 $x(n)$ を印加したときの応答 (response) が以下の $y(n)$ で与えられた. このシステムの伝達関数 (transfer function) $H(z)$ を求めよ.

$$y(n) = x(n) + 2x(n-1) - 3x(n-2)$$

(2-3) 以下の伝達関数 $H(z)$ を有するシステムの振幅特性 (amplitude characteristics) と位相特性 (phase characteristics) を求めよ.

$$H(z) = 1 + 2z^{-1} + z^{-2}$$