令和元年８月３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

令和２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 18 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 1/17）

配点：(1) 12 点，(2) 12 点，(3) 14 点，(4) 25 点，(5-1) 20 点，(5-2) 30 点，(6) 12 点

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）は，複数の整数（integer）のデータ（data）を，二分木（binary tree）を利用して昇順（ascending order）に整列（sort）して出力（output）するプログラムである．図 1 のプログラムでは，配列（array）の添え字（index）が二分木の節点番号（node number）に対応している．ただし，二分木の根（root）の節点番号を 0 とし，節点番号が i の節点に子（child）がある場合，左の子の節点番号を $2i+1$，右の子の節点番号を $2i+2$ とする．また，配列に格納されたデータは，二分木の対応する節点のデータを示している．

整列するデータは図 2 に示すような形式（format）のファイル data.txt で与えられ，1 行目には整列するデータの個数 n（$\geqq 1$），2 行目以降の n 行には整列するデータの値（value）が書かれている．図 3 は，図 2 の data.txt を与えて図 1 のプログラムを実行した場合の，28 行目が実行される直前の配列 d に対応する二分木であり，丸が節点，丸の左側の数字が節点番号，丸の中の数字がデータの値，線分が枝（edge）を示している．図 1 のプログラムに関する以下の各問に答えよ．

(1) 40 行目で呼び出されている関数（function）sort で実現されている整列アルゴリズム（sorting algorithm）は，一般に何と呼ばれているか名称を答えよ．

(2) 図 2 の data.txt を与えてプログラムを実行した場合の，28 行目が実行された直後の配列 d に対応する二分木を図示せよ．ただし，図 3 にならい，丸で節点，丸の左側の数字で節点番号，丸の中の数字でデータの値，線分で枝を示すこと．

(3) 11 行目および 12 行目が実行されることにより，節点番号が current の節点のデータとその子のデータの間に成立する関係を説明せよ．

(4) 関数 sort で実現されている整列アルゴリズムの最悪時間計算量（worst case time complexity）を，整列するデータの個数 n を用いて理由と共にオーダ表記（order notation）で示せ．

(5) 関数 sort において，28 行目の実行時に関数 swap が呼び出される回数を $T(n)$ とする．n は整列するデータの個数である．28 行目を変更し，28 行目の for ループの繰り返し回数と $T(n)$ の最大値をできる限り削減（28 行目の実行に要する最悪時間計算量を削減）することを考える．以下の各小問に答えよ．

(5-1) 下記の（あ）～（え）を埋めて変更後の 28 行目を完成させよ．

```
for (i = [ (あ) ]; 0 <= i; i--) downh( [ (い) ] , [ (う) ] , [ (え) ] );
```

(5-2) 変更後のプログラムにおける $T(n)$ の n に関するオーダ表記を理由と共に示せ．

$\sum_{j=0}^{h} \frac{j}{2^j} = 2 - \frac{2+h}{2^h}$ を用いてよい．

(6) 下線（ア）～（エ）で示す条件式を必要に応じて変更し，データを降順（descending order）に整列して出力することを考える．変更後のプログラムにおける下線（ア）～（エ）の条件式をそれぞれ答えよ．

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void swap(int d[], int p, int q) {
4      int tmp;
5      tmp = d[p]; d[p] = d[q]; d[q] = tmp;
6  }
7  void downh(int d[], int n, int k) {
8      int child, current = k;
9      while (current < n / 2) {
10         child = current * 2 + 1;
11         if ((child + 1 < n) && (d[child] < d[child + 1])) child++;
                    (ア)                    (イ)
12         if (d[current] < d[child]) swap(d, current, child);
                      (ウ)
13         else break;
14         current = child;
15     }
16 }
17 void uph(int d[], int k) {
18     int parent, current = k;
19     while (0 < current) {
20         parent = (current - 1) / 2;
21         if (d[parent] < d[current]) swap(d, parent, current);
                        (エ)
22         else break;
23         current = parent;
24     }
25 }
26 void sort(int d[], int n) {
27     int i;
28     for (i = 1; i < n; i++) uph(d, i);
29     for (i = n - 1; 0 < i; i--) {swap(d, 0, i); downh(d, i, 0);}
30 }
31 int main() {
32     int i, N, *D;
33     FILE* fp;
34     fp = fopen("data.txt", "r");
35     fscanf(fp, "%d", &N);
36     D = (int*) malloc(sizeof(int) * N);
37     for (i = 0; i < N; i++) fscanf(fp, "%d", &D[i]);
38     fclose(fp);
39
40     sort(D, N);
41
42     for (i = 0; i < N; i++) printf("%d ", D[i]);
43     printf("\n");
44     free(D);
45     return 0;
46 }
```

図 1　プログラム

```
6
40
30
50
10
60
20
```

図 2　data.txt

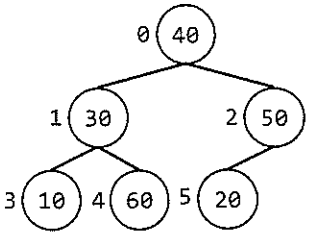


図 3　二分木の例

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 3/17）

配点： (1-1) 10 点，(1-2-1) 15 点，(1-2-2) 15 点，(1-2-3) 15 点，(1-2-4) 10 点，
(2-1) 18 点，(2-2) 20 点，(2-3) 12 点，(2-4) 10 点

(1) 計算機 (computer) における数の表現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) 以下の文章の空欄 [(a)] ～ [(e)] に当てはまる最も適切な語句または数値を，下記の選択肢から選び，記号で答えよ．

数の表現形式として固定小数点 (fixed point number) 表現と浮動小数点 (floating point number) 表現がある．科学技術計算の分野では，[(a)] 等の理由から [(b)] 表現が多く用いられ，小型機器の制御等の分野では，[(c)] 等の理由から [(d)] 表現が多く用いられる．いずれの表現形式においても，指数の底 (base of exponent) が 2 の場合，10 進数 (decimal number) の [(e)] を誤差 (error) 無く表現することはできない．

【選択肢】

(ア) 固定小数点	(イ) 浮動小数点
(ウ) 演算回路 (arithmetic circuit) が簡素になる	(エ) 表現できる数の範囲が広い
(オ) 小数を誤差無く表現できる	(カ) 0.5
(キ) 0.25	(ク) 0.1

(1-2) 以下の浮動小数点表現に関し，(1-2-1)～(1-2-4) に答えよ．(1-2-1)～(1-2-3) については導出過程も示すこと．

- 浮動小数点数 A を $A = (-1)^s \times 2^{e-7} \times (1+m)$ と表し，符号部 s (sign)，指数部 e (exponent)，仮数部 m (significand, mantissa) を最上位ビットから順にメモリに格納する．
- 符号部 s は 1 ビットである．$s \in \{0, 1\}$ をメモリに格納する．
- 指数部 e は 4 ビットである．7 余り表現とし，e を 4 ビットの符号無し 2 進整数 (unsigned binary integer) で表したビット列をメモリに格納する．$e = 0$ および $e = 15$ は $\pm\infty$ などの特殊な数を表現するのに用いる．
- 仮数部 m は 9 ビットである．正規化 (normalization) し，整数部分の 1 は記録しない (隠しビット)．小数部分の m を 2 進数の小数で表し，小数点以下の 9 ビットをメモリに格納する．丸め (rounding) 処理は切り捨て (rounding down) とする．非正規化 (denormalization) 表現は用いない．

(1-2-1) この表現形式で表すことのできる，正の最大値を 10 進数の小数で示せ．但し，無限大 (infinity) は除く．

(1-2-2) この表現形式で表すことのできる，正の最小値を 10 進数の小数で示せ．

(1-2-3) 10 進数 -36.66 をこの形式で表現し，そのビット列を示せ．

(1-2-4) 繰り返し計算を行う際，丸め処理として切り捨てを用いると，問題が生ずる場合がある．どの様な問題か，理由も含めて示せ．

(2)　仮想記憶 (仮想メモリ; virtual memory) に関して以下の各小問に答えよ．

(2-1)　次の説明文を読み，空欄 [(a)] ～ [(i)] にあてはまる最も適切な語句を下の (ア) ～ (ソ) の選択肢から選んで記号で答えよ．一つの選択肢を複数回用いてはならない．

仮想記憶は，主記憶 (primary storage, main memory) と補助記憶 (secondary storage, auxiliary memory) を用いて，主記憶の容量よりも大きい [(a)] を提供する．プロセスが指すアドレスは [(b)] と呼ばれる．プロセスの実行においてメモリアクセスのたびに，アドレス変換テーブル (mapping table) に基づいて [(b)] を [(c)] に変換する．一般的に，アドレス変換テーブルは [(d)] 上に構成する．

アドレス変換を固定長 (fixed length) のブロック単位で行う方式を [(e)] と呼び，可変長 (variable length) のブロック単位で行う方式を [(f)] と呼ぶ．主記憶の利用効率 (memory efficiency) の観点からこれらの方式を比較すると，[(e)] 方式の利点は，主記憶領域の割り付けと解放を繰り返しても [(g)] がほとんど発生しない点である．一方，欠点は，固定長のページ単位で領域を割り付けるために [(h)] が発生しやすい点である．[(h)] は，[(i)] の大きさと比較して使用領域が小さいプロセスを実行する場合に顕著である．

【選択肢】

(ア) 主記憶	(イ) ページ (page)	(ウ) 内部断片化 (internal fragmentation)
(エ) スラッシング (thrashing)	(オ) 実アドレス (real address)	(カ) 仮想アドレス (virtual address)
(キ) アドレス空間 (address space)	(ク) 補助記憶	(ケ) メモリ階層 (memory hierarchy)
(コ) ページング (paging)	(サ) セグメント (segment)	(シ) 外部断片化 (external fragmentation)
(ス) ページフォールト (page fault)	(セ) レジスタ (regsiter)	(ソ) セグメンテーション (segmentation)

(2-2)　プロセスが次に示すページ参照列 S で処理を行う場合を想定する．ページ枠 (page frame) の数が 3 のページング方式を採用した場合に，ページフォールトが発生するページ参照に○を記入せよ．ページ置換アルゴリズムに LRU (least recently used) および FIFO (first in first out) を用いる場合についてそれぞれ回答せよ．但し，初期状態では全てのページ枠は空である．

ページ参照列 S : 0, 1, 2, 0, 3, 1, 4, 3, 2, 3, 1, 2, 4

(2-3)　ページング方式を採用したシステムを想定する．このシステムでは，1 回の主記憶へのアクセスに 2 [マイクロ秒] を要し，ページフォールトが発生するとそれに加えて 8 [ミリ秒] のオーバヘッド (overhead) が発生する．1 命令あたり平均 2 回のメモリアクセスを行うとき，ページフォールトによる性能低下をページフォールトの無い状態と比較して平均 10%以下に抑えるために，許容できるページフォールトの確率 P の上限を求めよ．導出過程も示せ．なお，ページフォールトが発生しない場合のメモリアクセス時間は主記憶へのアクセス時間に等しく，演算時間および他のオーバヘッドは無視できるものとする．

(2-4)　ページング方式を採用し，ページ置換アルゴリズムに LRU を用いたシステムにおいて，ページフォールトを削減する方法を具体的かつ簡潔に記述せよ．但し，実行する各プロセスのページ参照列は変更しないものとする．

3 【選択問題】離散構造 （情報工学 5/17）

配点：(1-1) 16 点，(1-2-1) 8 点，(1-2-2) 15 点，(1-2-3) 16 点，(1-2-4) 10 点
(2-1) 10 点，(2-2-1) 25 点，(2-2-2) 25 点

(1) 一階述語論理（first-order 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（logical operator）である．

ロボット（robot），バッテリー（battery），移動可能な台（stepladder）が部屋にあり，ロボットは台を移動させることができる．部屋の中の位置 *goal* に台を移動させ設置し，台に登ることで，ロボットはバッテリーを握って取得できる．ロボットが実行可能な動作を論理式で記述し，それらの論理式のもとで，ロボットがバッテリーを取得した状態を表す論理式が成り立つことを示すことで，ロボットがバッテリーを取得できることを示したい．ロボットは動作をすることで，ロボットや台の状況を表す状態を遷移させることができる．状態は，ロボットが台に登っているかどうか，ロボットがバッテリーを取得しているかどうか，および台の位置で決まるものとする．本問では，ロボットの動作を表す関数（function）として，遷移前の状態を s としたときに，遷移後の状態 s' を返すような以下の関数を用いる．

- $move(s, x)$：状態 s からロボットが台を位置 x に移動させ設置した状態 s' を返す．
- $climb(s)$：状態 s からロボットが台に登った状態 s' を返す．
- $grasp(s)$：状態 s からロボットがバッテリーを握って取得した状態 s' を返す．

また，以下の述語（predicate）を用いる．

- $on(s)$：状態 s においてロボットが台に乗っている．
- $sl(s, x)$：状態 s において台が位置 x に設置されている．
- $have(s)$：状態 s においてロボットがバッテリーを取得している．

本問では，以下の閉論理式（closed formula）A〜D が成り立つことを仮定する．

- $A = \forall s\ \forall x(sl(s, x) \rightarrow sl(climb(s), x))$：どのような状態で，どのような位置に台が設置されていても，ロボットはその位置において台に登ることができ，登った後の状態に遷移する．
- $B = \forall s((sl(s, goal) \wedge on(s)) \rightarrow have(grasp(s)))$：どのような状態においても，位置 *goal* に設置されている台にロボットが乗っていれば，ロボットはバッテリーを握ることができ，取得した状態に遷移する．
- $C = \forall s\ \forall x(\neg on(s) \rightarrow sl(move(s, x), x))$
- $D = \forall s\ on(climb(s))$

以下の各小問に答えよ．

(1-1) 閉論理式 C および D が表していることをそれぞれ説明せよ．

(1-2) 閉論理式 A〜D を用いて，ロボットがバッテリーを取得できることを次の手順で示す．以下の各問に答えよ．

(1-2-1) 「ロボットがバッテリーを取得している状態が存在する」ことを表す閉論理式 E を示せ．

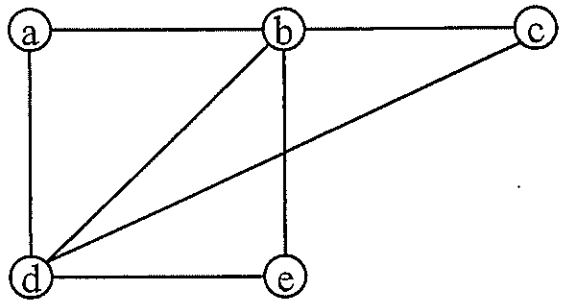
(1-2-2) 論理式 $F = (A \wedge B \wedge C \wedge D) \rightarrow E$ としたとき，$\neg F$ の冠頭標準形（prenex normal form）$\neg F_p$ を示せ．冠頭標準形は，すべての限定作用素（quantifier）が先頭にある閉論理式である．ただし，$\neg F_p$ は連言標準形（和積標準形：conjunctive normal form）で示し，最終的に得られる $\neg F_p$ には，記号 A, B, C, D, E を用いないこと．

(1-2-3) $\neg F_p$ をスコーレム化し，導出原理を用いて，充足不能であることを示そうとしたが，空節を導くことができなかった．これは，ロボットの初期条件に関する論理式 I が考慮されていないためである．導出原理を適用して，E を成り立たせるための論理式 I を示せ．導出過程も示せ．ただし，論理式 I は初期状態 s_0 について記述したものであり，台の初期位置は $goal$ ではないこととする．

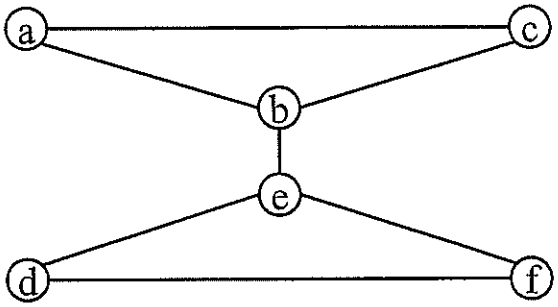
(1-2-4) 論理式 I を F の前提部に追加，つまり $F = (A \wedge B \wedge C \wedge D \wedge I) \rightarrow E$ と修正し，再度導出原理を適用することで，空節を導くことができた．このとき，論理式 E に該当する節の変数への代入から，状態の遷移を知ることができるが，この状態の遷移を，代入された項を示すことにより答えよ．

(2) 頂点（node）の集合，および辺（edge）の集合がいずれも空（empty）でない無向グラフ（undirected graph）に関する以下の各小問に答えよ．

(2-1) 以下のグラフ G_1 および G_2 それぞれについて，それらはどの一辺を取り除いても連結（connected）であるか，そうでないかを答えよ．そうでない場合は，どの辺を取り除くと連結でなくなるかも答えよ．



G_1



G_2

(2-2) 各頂点の次数（degree）が偶数（even number）であり，かつ連結である任意のグラフ G について，以下の各問に答えよ．ただし，G は同じ頂点を結ぶ辺を持たず，かつ，各頂点組に対し，それらの頂点を結ぶ辺は高々一つであるとする．なお，隣接する頂点の系列を，それに対応する辺の系列により表したものを経路（path）とよび，始点と終点が同じ経路を閉路（cycle）とよぶ．例えば小問 (2-1) の G_1 の経路 $(b,c),(c,d),(d,b),(b,e),(e,d),(d,a),(a,b)$ は閉路である．

(2-2-1) G から任意の辺 (u,v) を取り除いて得られるグラフ G' は，頂点 v から頂点 u への経路を持つことを示せ．G' が連結でないと仮定し，頂点の次数に関する矛盾（contradiction）を以下の補題（lemma）を用いて導くとよい．

（補題）任意のグラフにおいて，すべての頂点の次数の和は偶数である．

(2-2-2) 任意の頂点 u に対し，G は u を含み，各辺を高々一つだけ含む閉路を持つことを示せ．(2-2-1) は成り立つとしてよい．

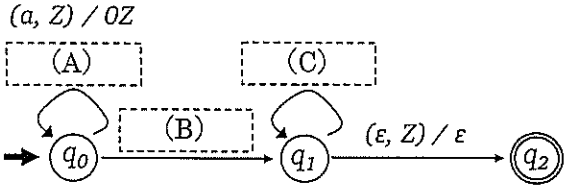
4 【選択問題】計算理論　（情報工学 7/17）

配点：(1-1) 15 点, (1-2-1) 10 点, (1-2-2) 25 点, (1-3) 10 点, (2-1) 40 点, (2-2) 25 点

(1) アルファベット (alphabet) $\Sigma_{ab} = \{a, b\}$ で構成される言語 (language) $L_{ab} = \{a^n b^n \mid n$ は正の整数 $\}$ を考える．なお，a^n は a が n 個連結した文字列 (string) を表す．例えば，$a^3 = aaa$ である．また，ε は空文字列 (empty string) を表す．言語 L_{ab} を認識 (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) 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 (state) の有限集合，Σ は入力アルファベット (input alphabet)，Γ はスタックアルファベット (stack alphabet)，δ は遷移関数 (transition function)，q_0 は開始状態 (start state)，Z はスタックの開始記号，F は最終状態 (final state) 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して，言語 L_{ab} を認識することを考える．

下図は言語 L_{ab} を認識する非決定性 (non-deterministic) プッシュダウンオートマトンの状態遷移図 (state transition diagram) であり，開始状態は q_0，最終状態は q_2 である．状態遷移図の空欄 (A), (B), (C) に必要な動作を，$(r, s)/t$ の形式で答えよ．ただし，$(r, s)/t$ $(r \in \Sigma \cup \{\varepsilon\}, s \in \Gamma, t \in \Gamma^*)$ は，入力から読み出す記号が r でスタックから取り出す記号が s のときに，スタックからその 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．$(a, Z)/0Z$ は，a を読み出すときにスタックから 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．入力アルファベットは $\Sigma = \Sigma_{ab}$，スタックアルファベットは $\Gamma = \{Z, 0\}$ とする．



(1-2) 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 は状態の有限集合，Σ は入力アルファベット，δ は遷移関数，q_0 は開始状態，F は最終状態の集合である．入力アルファベットは $\Sigma = \Sigma_{ab}$ とする．このとき，(1-2-1) および (1-2-2) に答えよ．

(1-2-1) 言語 $L_2 = \{a^n b^n \mid n = 2\}$ を認識する決定性有限オートマトン，および，言語 $L_3 = \{a^n b^n \mid n = 3\}$ を認識する決定性有限オートマトンの状態遷移図をそれぞれ書け．なお，上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．また，全ての状態において，全ての入力記号に対する遷移を表記すること．

(1-2-2) 言語 L_{ab} を認識する決定性有限オートマトンは存在しないことを背理法 (proof by contradiction) により証明したい．以下の空欄 (D) に適切な文章を記述し，証明を完成させよ．

証明：言語 L_{ab} を認識する決定性有限オートマトンが存在すると仮定し，そのオートマトンの状態数を k とする．文字列 $a^k b^k$ が与えられたとき，a を i 個 $(0 \leq i \leq k)$ 読み終えた時点での状態を p_i とすると，

(D)

よって仮定と矛盾するため，言語 L_{ab} を認識する決定性有限オートマトンは存在しない．　(証明終)

(1-3) 言語 L_{ab} を認識する決定性有限オートマトンは存在しないが，言語 $L'_{ab} = \{a^n b^m \mid n$ および m は正の整数 $\}$ を認識する決定性有限オートマトンは存在する．言語 L'_{ab} を認識する決定性有限オートマトンの状態遷移図を書け．なお，上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．また，全ての状態において，全ての入力記号に対する遷移を表記すること．

(2) 文脈自由文法 (context-free grammar) は，変数 (variable) の有限集合 V, 終端記号 (terminal symbol) の有限集合 T, 生成規則 (production rule) の有限集合 P, および開始記号 (start symbol) S の組により定められる．変数は非終端記号 (nonterminal symbol) と呼ぶこともある．Chomsky 標準形 (Chomsky normal form) とは，文脈自由文法のうち，生成規則が，(i) $X \to YZ$, (ii) $X \to a$ のいずれかの形をしたものである．ただし，X, Y, Z は変数，a は終端記号である．

ここで，Chomsky 標準形で与えられる文脈自由文法 $G = (V, T, P, S)$ および文字列 $w = a_1 a_2 \cdots a_n$（ただし，各 a_i は T に属する）に対し，w が G によって生成されるか否かを，動的計画法 (dynamic programming) によって効率的に判定するアルゴリズムを考える．まず，整数 i, j のうち $1 \le i \le j \le n$ であるものに対し，集合

$$M[i,j] = \{X \in V \mid \text{文法 } G \text{ において，} w \text{ の連続した部分文字列 } a_i a_{i+1} \cdots a_j \text{ を変数 } X \text{ から導出できる}\}$$

を定める．以下に擬似コードで示したアルゴリズムでは各 $M[i,j]$ を $M[1,1], M[2,2], \ldots, M[n,n]$, $M[1,2], M[2,3], \ldots, M[n-1,n], M[1,3], \ldots$ の順に $M[1,n]$ まで計算している．最後の判定では，w が G によって生成されるとき，かつそのときに限り $S \in M[1,n]$ という性質を利用している．

```
入力 w = a_1 a_2 ... a_n に対し，集合 M[i,j] (1 ≤ i ≤ j ≤ n) の初期値をすべて空集合 ∅ とする．
for i = 1 to n do                    // 部分文字列の長さが 1 の場合の計算．
  (X → a_i) ∈ P であるすべての X ∈ V を M[i,i] に追加．
end for
for ℓ = 2 to n do                    // 他のすべての M_{i,j} の計算．ℓ は部分文字列の長さを表す．
  for i = 1 to n − ℓ + 1 do          // i は部分文字列の先頭位置を表す．
    j = i + ℓ − 1 とする．            // j は部分文字列の終了位置を表す．
    for k = i to j − 1 do            // k は部分文字列の分割位置の一つ前を表す．
      ある Y ∈ M[i,k], Z ∈ M[k+1,j] に対し，(X → YZ) ∈ P であるすべての X ∈ V を M[i,j] に追加．
    end for
  end for
end for
S ∈ M[1,n] であれば Yes を，そうでなければ No を出力して終了．        // 判定
```

文脈自由文法 $G_1 = (V_1, T_1, P_1, S_1)$ を，$V_1 = \{A, B, C, D, S\}, T_1 = \{a, b\}, S_1 = S$,

$$P_1 = \{S \to AB, A \to CA, A \to AA, A \to a, B \to DB, B \to b, C \to a, D \to b\},$$

と定める．例えば，文法 G_1 と文字列 bab に対し，上記のアルゴリズム実行した場合の計算は以下のように行われる．

- まず，$M[1,1], M[2,2], M[3,3]$ が決まる．
- 次に，部分文字列の長さが $\ell = 2$ の場合，先頭位置は $i = 1, 2$ の二つの場合があり，
  - $i = 1$ ならば $j = 2$ であり，$k = 1$ として，$Y \in M[1,1], Z \in M[2,2]$ を調べることで $M[1,2]$ が決まる．
  - $i = 2$ ならば $j = 3$ であり，$k = 2$ として，$Y \in M[2,2], Z \in M[3,3]$ を調べることで $M[2,3]$ が決まる．
- 最後に，部分文字列の長さが $\ell = 3$ の場合，先頭位置は $i = 1$ の場合だけであり，
  - $i = 1$ ならば $j = 3$ である．このとき，$k = 1$ として，$Y \in M[1,1], Z \in M[2,3]$ を調べ，$k = 2$ として，$Y \in M[1,2], Z \in M[3,3]$ を調べることで，$M[1,3]$ が決まる．

終了時における $M[i,j]$ の内容は以下の表のとおりである．この場合，$S \notin M[1,3]$ であるため No が出力される．

$M[i,j]$	$j=1$	$j=2$	$j=3$
$i=1$	$\{B,D\}$	$\emptyset$	$\emptyset$
$i=2$	—	$\{A,C\}$	$\{S\}$
$i=3$	—	—	$\{B,D\}$

以下の各小問に答えよ．

(2-1) 文法 G_1 および文字列 $w_1 = aaab$ に対し，上記のアルゴリズムを実行した場合，終了時における $M[i,j]$ の内容を，上記の例のように，表として示せ．

(2-2) 文脈自由文法 $G_2 = (V_2, T_2, P_2, S_2), V_2 = \{A, B, S\}, T_2 = \{a, b\}, S_2 = S,$

$$P_2 = \{S \to AB, S \to SA, S \to BS, S \to \varepsilon, A \to a, B \to b\}$$

を考える．ここで，ε は空文字列 (empty string) である．文法 G_2 は，上で説明した Chomsky 標準形の制約を満たしていないため，上記のアルゴリズムに対して正しく動作しない場合がある．つまり，ある入力文字列に対しては間違った判定をする．そのような入力文字列として長さ 2 以上のものを具体的に示し，正しく動作しないことを説明せよ．

配点: (1-1) 10 点，(1-2-1) 15 点，(1-2-2) 25 点，(1-3) 15 点，(1-4)15 点，
(2-1) 20 点，(2-2-1) 10 点，(2-2-2) 5 点，(2-2-3) 10 点

(1) ビット誤り (bit error) が発生することがある伝送路 (transmission line) を用いて，送信端末 (terminal) の応用プログラム (application program) がファイル (file) を受信端末の応用プログラムに転送するプロトコル (protocol) を考える．プロトコルでは，応用プログラムがファイルをデータ (data) に分割し，送信端末が分割されたデータをデータパケット (packet) を用いて転送する．以下の各小問に答えよ．なお，端末は受信したパケットのビット誤りを検出できるものとする．さらに，小問 (1-1) から (1-3) では，パケットは紛失 (loss) しないことを仮定する．

(1-1) ビット誤りが発生したデータパケットを再送 (retransmit) するプロトコルを考える．このプロトコルをプロトコル 1 と呼び，その送信端末の状態遷移図 (state transition diagram) を図 1 に示す．状態遷移では，発生したイベント (event) に基づきアクション (action) が実行され，次の状態に遷移する．イベントとアクションの組は図 1 の破線の枠内に示す通りに記載する．使用するイベント，アクションとそれらに付随する変数 (variable) を図 2 に示す．プロトコル 1 では，送信端末はシーケンス番号 (sequence number) に 0 を設定してデータパケットを送信する．一方，受信端末は，データパケットを受信すると，ビット誤りが無い場合は確認応答 (positive acknowledgement) パケットを返送し，ビット誤りを検出すると否定確認応答 (negative acknowledgement) パケットを返送する．以降，これら二つのパケットを応答パケットと総称する．なお，プロトコル 1 では，応答パケットのシーケンス番号には 0 を設定し，応答パケットにはビット誤りが発生しないものとする．図 1 の状態遷移図の空欄 [（あ）]，[（い）] に，それぞれ一つのアクションを埋めよ．

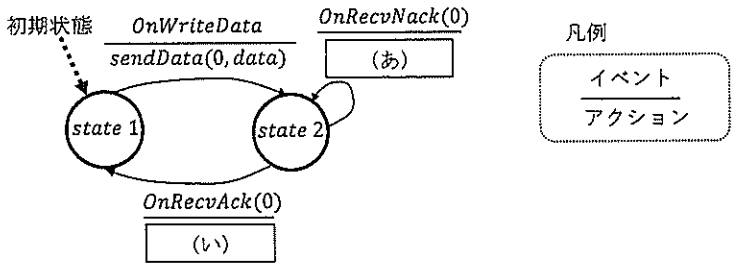


図 1: プロトコル 1 の送信端末の状態遷移図

変数　*data*
イベント　*OnWriteData*：応用プログラムが転送するデータを受領し，変数 *data* に格納．
　OnRecvAck(0 または 1)：シーケンス番号が 0 または 1 に設定された確認応答パケットを受信．
　OnRecvNack(0 または 1)：シーケンス番号が 0 または 1 に設定された否定確認応答パケットを受信．
アクション　Λ：何もしない．
　sendData(0 または 1, *data*)：シーケンス番号が 0 または 1 に設定され，変数 *data* の中身をペイロード (payload) として持つデータパケットを受信端末に送信する．

図 2: 送信端末の変数，イベント，アクション

(1-2) 次に，プロトコル 1 を改良し，応答パケットにビット誤りが発生したとき，データパケットを再送するプロトコルを考える．ここで，図 3 に示すイベントを追加する．(1-2-1) および (1-2-2) に答えよ．

(1-2-1) 送信端末の状態遷移図を図 4 とするプロトコルをプロトコル 2 と呼ぶ．しかし，プロトコル 2 では，送信端末の応用プログラムが転送したファイルの中身と，受信端末の応用プログラムが受領したファイルの中身が一致しないことがある．この問題が発生する条件を一つ示し，ファイルの中身が一致しない理由を説明せよ．

イベント　*OnRecvBiterr*：ビット誤りがある応答パケットを受信．

図 3: 追加するイベント

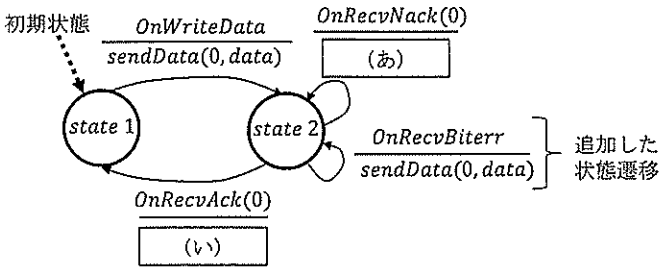


図 4: プロトコル 2 の送信端末の状態遷移図

(1-2-2) 応答パケットのビット誤りに対処できるように，送信端末がデータパケットに，0 と 1 のシーケンス番号を交互に設定し，最初のデータパケットには 0 を設定する方針でプロトコル 3 を設計する．受信端末は，データパケットを受信すると，ビット誤りの有無にかかわらず，確認応答パケットを返送する．データパケットにビット誤りが無いとき，次に受信することを期待するシーケンス番号 (next expected sequence number) を確認応答パケットに設定する．一方，ビット誤りを検出すると，受信することを期待していたシーケンス番号を確認応答パケットに設定する．図 5 の空欄 (う) ～ (き) に，それぞれ一つのイベントあるいはアクションを埋めることで，プロトコル 3 の送信端末の状態遷移図を完成させよ．

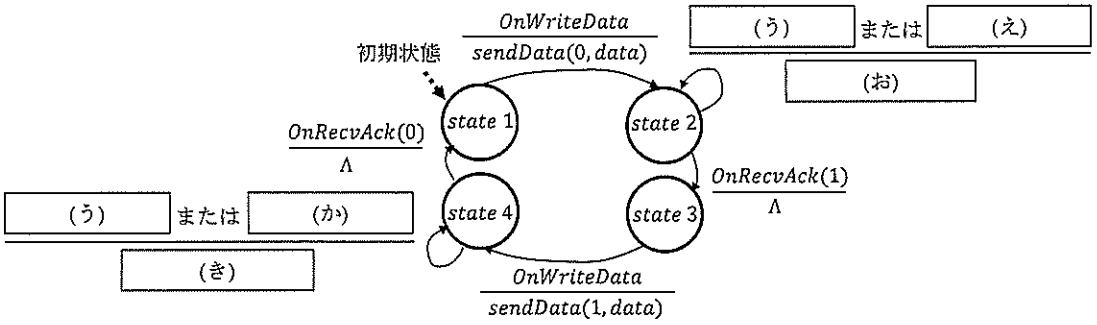


図 5: プロトコル 3 の送信端末の状態遷移図

(1-3) 伝送路の帯域 (bandwidth) は 1600 [ビット/秒]，伝搬遅延 (propagation delay) は 10 [ミリ秒]，送信端末と受信端末のプロトコルの処理遅延 (processing delay) は 0 [ミリ秒]，確認応答パケットは 8 ビット長と仮定する．このとき，24 ビット長のデータパケットを転送する場合の，プロトコル 3 の送信端末が，1 秒間に転送できる最大のデータパケット数を求めよ．あわせて，計算過程も示せ．

(1-4) 本小問では，伝送路でパケットが紛失することがあると仮定する．プロトコル 3 において，受信端末の動作を変えることなく，送信端末がデータパケットや確認応答パケットの紛失に対処する方法を一つ説明せよ．説明にあたっては，状態遷移図を示す必要はない．

(2) 伝送路における誤り検出 (error detection) に関する以下の各小問に答えよ．

(2-1) 以下の文章の空欄 [（あ）] ～ [（お）] を埋めよ．

ある情報が送信された後，伝送路上において電気的雑音 (noise) により受信された情報に誤りが発生する場合がある．これに対処するためには，誤り検出が必要となる．

誤り検出の例として，[（あ）] が挙げられる．[（あ）] とは，元の情報ビットに対し，1 の総数が奇数または偶数となるようなチェックビット (check bit) を付与する方法である．例えば，2^7 個の文字や記号を 7 ビットで表現する符号の場合，符号語 (codeword) 間の最小ハミング距離 (minimum Hamming distance) は 1 であるが，チェックビットを付与することによって符号語間の最小ハミング距離は [（い）] になる．一般に，最小ハミング距離が [（い）] の符号は，[（う）] ビットの誤りをすべて検出できる．

2 ビット以上の誤りや，あるいは符号語の連続したビットに発生する誤りである [（え）] にも対処できるように，巡回符号 (cyclic code) が広く用いられる．巡回符号は，[（え）] の検出に優れており，生成多項式 (generator polynomial) の次数が r であるとき，[（お）] ビット以下の [（え）] を必ず検出できる．

(2-2) 2 元巡回符号 (binary cyclic code) を用いた誤り検出について，(2-2-1)～(2-2-3) に答えよ．

(2-2-1) 生成多項式 $G(x)$ の 2 元巡回符号を考える．$G(x)$ の次数を r とし，符号長を n とする．このとき，情報ビットの長さ m は $n-r$ である．情報ビット “$a_{m-1}a_{m-2}\cdots a_1a_0$” を多項式

$$M(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0$$

で表現するとき，$M(x)$ から符号語多項式 (code polynomial) $F(x)$ を生成する方法を説明せよ．

(2-2-2) 生成多項式 $G(x)$ として

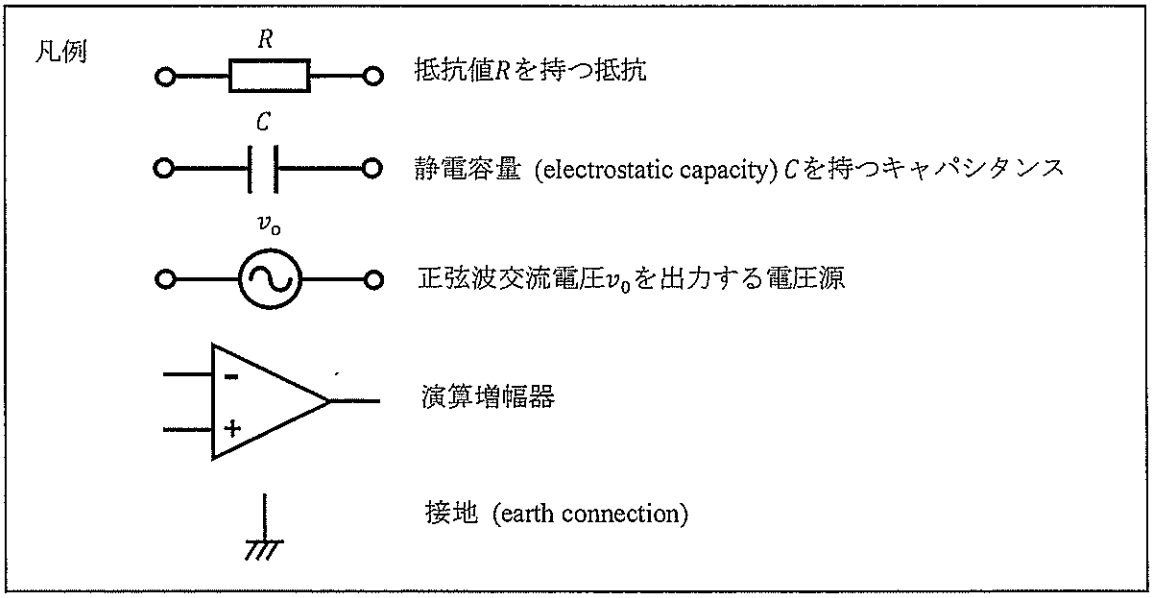
$$G(x) = x^4 + x + 1$$

を用いる．符号長 n を 15 とするとき，情報ビット “00011010011” を送信する場合の符号語多項式 $F(x)$ を，(2-2-1) で説明した方法をもとに生成せよ．

(2-2-3) 受信側において，受信語 (received word) を表す多項式に基づいて誤りを検出する方法を一つ説明せよ．

配点：(1-1) 10 点，(1-2) 20 点，(1-3) 30 点，(2-1) 10 点，(2-2) 30 点，(2-3) 15 点，(2-4) 10 点

(1) 演算増幅器 (operational amplifier)，抵抗 (resistance)，キャパシタンス (capacitance) から構成される回路 (circuit) について以下の各小問に答えよ．なお，図中の記号は以下の凡例に従うとする．また，正弦波交流電圧 (sinusoidal AC voltage) は複素数 (complex numbers) 表示されており，その絶対値は実効値 (effective value) を表すとし，演算増幅器の利得 (gain) 及び入力インピーダンス (input impedance) は無限大，出力インピーダンス (output impedance) は 0 であるとする．虚数単位 (imaginary unit) が必要な場合には j を用いること．



(1-1) 図 1 に示す非反転増幅器 (non-inverting amplifier) の利得 $A = V_{\mathrm{out}}/V_{\mathrm{in}}$ を求めよ．なお α は 0 または正の実数である．

(1-2) 図 2 に示す回路において，角周波数 (angular frequency) ωの正弦波交流電圧 v_o を印加した．回路の利得を $\beta = v_{\mathrm{fbk}}/v_o$ としたとき，β の絶対値を最大とする角周波数 ω_c を R, C の式として示すとともに，$\omega = \omega_c$ の時の入力電圧 v_o に対する出力電圧 v_{fbk} の位相差 (phase difference) を求めよ．

(1-3) 図 1 の回路に帰還回路 (feedback circuit) として図 2 の回路を追加した図 3 の回路を考える．今，α を 0 から連続的に増加させながら出力 V_{out} を観測したところ，ある α の時に発振 (oscillation) を開始した．この時の α 及び発振周波数 (oscillation frequency) を R, C の式として示せ．

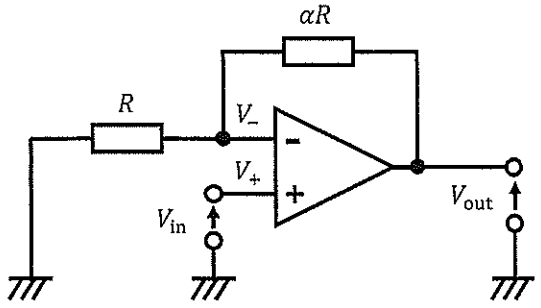


図 1

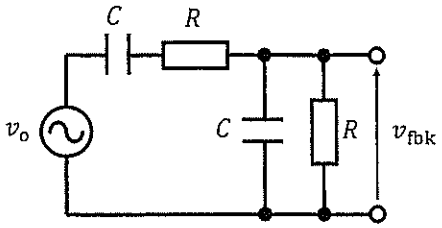


図 2

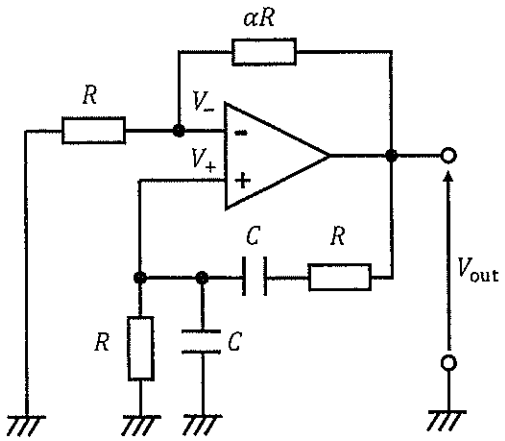


図 3

(2) 自動券売機 (automatic ticket machine) の制御回路 (control circuit) を同期式順序回路 (synchronous sequential circuit) として設計する．この自動券売機は 300 円のチケットのみを販売しており，100 円硬貨のみを受け付ける．100 円硬貨は 1 枚ずつ投入され，300 円分の硬貨が投入されると直ちにチケットが発券される．設計する制御回路は 1 ビットの入力 (input) x_0 と 1 ビットの出力 (output) y_0 を持ち，100 円硬貨が投入されたときに x_0=1，それ以外は x_0=0 が 1 クロックの間保持される．チケットを発券するときは y_0=1，それ以外は y_0=0 が出力される．また，表 1 のように A，B，C の 3 つの状態 (state) を持ち，状態変数 (state variable) k_0，k_1 を用いた状態割り当て (state assignment) が行われている．以下の各小問に答えよ．

表 1

状態	状態割当(k_0, k_1)	意味
A	(0, 0)	合計投入金額が 0 円
B	(0, 1)	合計投入金額が 100 円
C	(1, 0)	合計投入金額が 200 円

(2-1) この制御回路を Mealy 型順序回路 (Mealy sequential circuit) として設計するとき，状態遷移図 (state transition diagram) を示せ．状態遷移図には初期状態 (initial state) と入出力 (input and output) を明示すること．

(2-2) (2-1) の制御回路において，現在の状態 (current state) (k_0, k_1) に対する次状態 (next state) を (k_0^+, k_1^+) で表すとき，k_0^+，k_1^+，y_0 の最簡積和形 (最小積和形，minimal sum-of-products expression) の論理式 (logic expression) を導出せよ．

(2-3) この自動券売機にある機能を追加するため，制御回路に 1 ビットの入力 x_1 と 1 ビットの出力 y_1 を追加した．そして表 1 の状態割り当てのまま図 4 の制御回路を設計した．この制御回路の状態遷移図を示せ．状態遷移図には初期状態と入出力を明示すること．ここで，この制御回路は表 1 に示した状態からのみ遷移し，未定義の状態からの遷移は発生しないとする．また，図中の CLK はクロック信号を表す．

(2-4) 図 4 の制御回路はどのような機能が付いた自動券売機の制御回路であると考えられるか答えよ．またその理由も簡潔に述べよ．

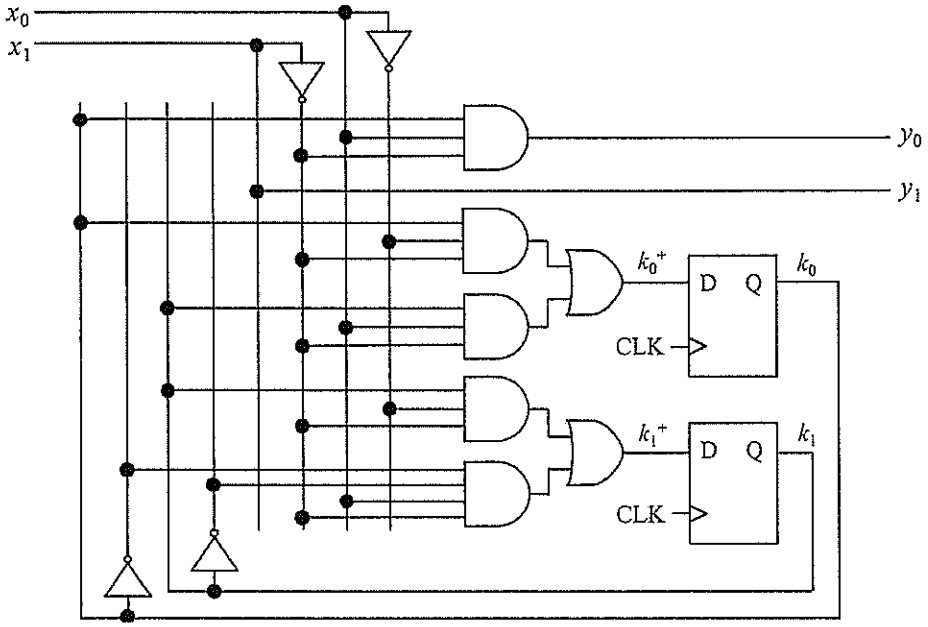


図 4

7 【選択問題】数学解析と信号処理

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

以下の各問に答えよ.

(1) 以下の各小問に答えよ. ただし導出の過程も示すこと.

(1-1) 下記の関数のフーリエ級数 (Fourier series) を求めよ.

$$\begin{aligned} f(x) &= x^2 \\ f(x) &= f(x+2\pi) \qquad\qquad (-\pi \leq x < \pi) \end{aligned}$$

(1-2) (1-1) の結果を用いて下記の値 u を求めよ. ただし n は正の整数 (positive integer) とする.

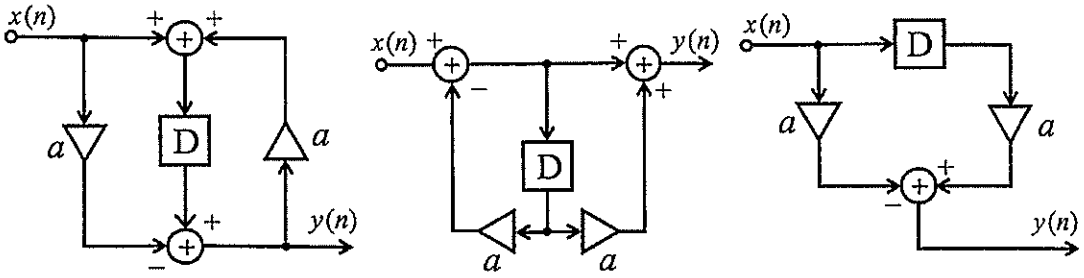
$$u = \sum_{n=1}^{\infty} \frac{1}{n^4}$$

(2) z 変換 (z-transform) された以下の伝達関数 (transfer function)

$$H(z) = \frac{1 - az}{z - a} \quad (a \text{ は実数で } -1 \leq a < 1)$$

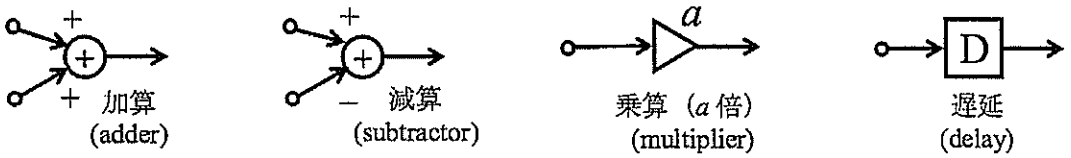
について，以下の各小問に答えよ．ただし，導出の過程も示すこと．

(2-1) $H(z)$ を実現するフィルタ回路 (filter circuit) を下記図 (あ)～(う) から 1 つを選択し，さらに，入力と出力の関係を式で表せ．ただし，入力を $x(n)$，出力を $y(n)$ とする．



(あ)　(い)　(う)

ここで，図中の回路の要素は以下の通りである．



(2-2) $a = 0, -1$ のそれぞれのとき，$H(z)$ を実現するフィルタ回路はどのような特性となるか，下記の選択肢から選んでその理由を説明せよ．

[選択肢]

加算，遅延，反転(inverter)，増幅(amplifier)， ハイパス(high-pass)，ローパス(low-pass)，そのまま通過(all-pass)

(2-3) $H(z)$ を実現するフィルタ回路について，振幅特性 (amplitude characteristics) と位相特性 (phase characteristics) を求めよ．