平成20年8月2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21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18枚である．
- 解答用紙は全部で7枚である．
  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（黄色）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（黄色）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．
- 留学生特別選抜の受験者は，英語で解答することも可能である．

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

ハッシュ法（hash method）によって，いくつかの非負整数（nonnegative integer）を，添字（index）の範囲が $0 \sim n-1$，要素数が n の配列（array）に格納（store）することを考える．ここで，格納する非負整数を n で割った剰余（remainder）を返すハッシュ関数（hash function）を用いて，ハッシュ値（hash value）を添字とするセル（cell）に非負整数を格納するものとする．なお，衝突（collision）が生じた際には，ハッシュ値に整数 k ($k \geq 1$) を加えて n で割った剰余を添字とするセルに格納する．それでもなお衝突が生じる場合には，非負整数が格納されていないセルが見つかるまで，k を $2k, 3k, 4k, \cdots$ と k の倍数へ順に置き換えて同様の処理を行うものとする．以下の各問に答えよ．

(1) $n = 10, k = 1$ とするハッシュ法を考える．どのセルにも非負整数が格納されていない状態から始めて，まず 21 を添字が 1 のセルに格納し，次に 15 を添字が 5 のセルに格納した状態を考える．これに続けて，28 と 35 をこの順番で格納するとき，それぞれどのセルに格納されるか，その添字を答えよ．

(2) 図 1 に示す C 言語で書かれたプログラムは，あらかじめ 10 個の非負整数を配列に格納しておき，キーボードから入力された非負整数が配列に格納されているかどうかを，ハッシュ法によって調べるプログラムである．以下の各小問に答えよ．

(2-1) このプログラムが 43 行目まで実行された後の，`table[0]`～`table[19]` の値を答えよ．なお，解答にあたっては，解答用紙の欄を用いること．

(2-2) 関数 `search(int *table, int d)` は，非負整数 `d` が配列 `table` に格納されていればその添字を，格納されていなければ `-1` を返す．そのような動作となるように，プログラム中の空欄（ア）～（エ）を適切に埋めよ．

(2-3) このプログラムでは，`SKIP` は 3 となっている．これを 4 に変更すると，45 行目の `printf` 文以降が実行されない．このような問題が生じる理由を簡潔に答えよ．

(2-4) 小問 (2-3) のような問題が生じないようにするためには，一般に `MAX` と `SKIP` の間にどのような関係が成立していなければならないか，簡潔に答えよ．

```
#include <stdio.h>

#define EMPTY -1
#define MAX   20
#define SKIP   3

int hash(int x){
  return x % MAX;
}

int next(int x){
  x = x + SKIP;
  return x % MAX;
}

void store(int *table, int d){
  int h;
  h = hash(d);
  while (table[h] != EMPTY) h = next(h);
  table[h] = d;
}

int search(int *table, int d){
  int h;
  h = （ア） ;
  while (table[h] != EMPTY) {
    if (table[h] == d) return （イ） ;
    h = （ウ） ;
  }
  return （エ） ;
}

int main(){
  int i, query;
  int table[MAX];
  int data[] = {31,45,59,25,95,39,76,27,65,43};

  /* 配列を初期化する */
  for (i=0; i<MAX; i++) table[i] = EMPTY;

  /* 10 個の非負整数を格納する */
  for (i=0; i<10;  i++) store(table, data[i]);

  /* キーボードから非負整数を入力する */
  printf("Query: ");
  scanf("%d", &query);

  /* 入力された非負整数が配列に格納されているかどうかを調べる */
  if (search(table, query) != -1) printf("Found.\n");
  else printf("Not found.\n");

  return 0;
}
```

図 1: 非負整数が格納されているかどうかを調べるプログラム

2 （必須問題:論理回路）

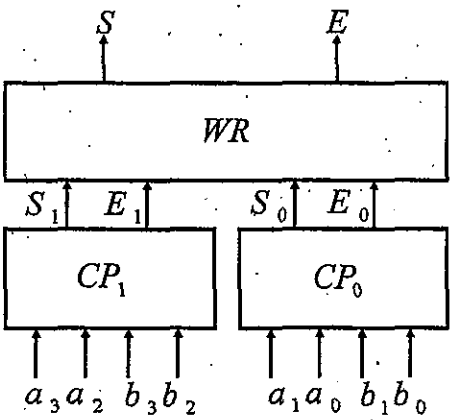
（配点：(1-1)15 点，(1-2)20 点，(1-3)5 点，(2-1)10 点，(2-2)20 点，(2-3)30 点）

(1)

a_0 および b_0 を最下位ビット(LSB: Least Significant Bit)とする符号なし 2 進数(unsigned binary number) $A(a_{n-1}a_{n-2}...a_1a_0)$，$B(b_{n-1}b_{n-2}...b_1b_0)$ の $2n$ ビットの信号を入力とし，$A \geq B$ であれば出力 S を 1，$A < B$ であれば出力 S を 0，$A = B$ であれば出力 E を 1，$A \neq B$ であれば出力 E を 0 とする n ビットの比較器(comparator)を作りたい．以下の各小問に答えよ．

(1-1) $n = 2$ のときの出力 S を最小積和形(最簡積和形，minimal sum-of-products expression)で表せ．カルノー図（Karnaugh map）も示すこと．

(1-2) $n = 2$ のときの 4 入力 2 出力の比較器を CP とし，この比較器 2 つを部品(component)として用いて $n = 4$ の比較器を作ることを考える．2 つの部品 $CP_i (i = 0,1)$ を，入力を $a_{2i+1}, a_{2i}, b_{2i+1}, b_{2i}$，出力を S_i, E_i として下図のような構成で接続する．このとき，回路 WR の出力 S, E を S_1, S_0, E_1, E_0 の最小積和形で表せ．カルノー図も示すこと．ただし，$S_i = 0$ であれば $E_i = 0$ であることに注意せよ．



(1-3) 回路 WR の出力 S を実現する回路を 4 つの 2 入力 NAND ゲートのみを用いて設計せよ．

(2)
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Moore型の順序回路(sequential circuit)として設計したい．カウンタは入力信号x_1およびx_0を受けつけ，信号fを出力する．$x_1 = x_0 = 0$のとき，カウンタは現在の値に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．$x_1 = 0$，$x_0 = 1$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．$x_1 = x_0 = 1$のとき，現在の値を次の時刻でも保持する．$x_1 = 1$，$x_0 = 0$は禁止入力であり，カウンタに入力されない．出力信号fは，カウンタの値が0のときに1であり，それ以外の値では0である．なお，Moore型順序回路は出力が現状態のみに依存して決まる順序回路である．以下の各小問に答えよ．

(2-1) カウンタの状態遷移図(state transition diagram)を示せ．ただし，カウンタの初期値(initial value)は0とする．また，入力はx_1x_0の順で記述せよ．

(2-2) カウンタの値を2進数として状態変数(state variable)Q_1，Q_0で表現したときの状態遷移表(state transition table)および出力表(output table)を示せ．ただし，Q_0を最下位ビットとする．

(2-3) (2-2)の状態遷移表および出力表をもとに，2個のDフリップフロップ(D flip-flop)を用いてカウンタを設計したい．状態変数Q_1，Q_0に対応するDフリップフロップの入力D_1，D_0，および出力信号fを，Q_1，Q_0，x_1，x_0の論理関数で表せ．結果は最小積和形で示すこと．また，D_1およびD_0についてはカルノー図も示すこと．

（配点：(1-1) 21 点，(1-2) 10 点，(1-3-1) 4 点，(1-3-2) 15 点，(2-1) 18 点，(2-2) 32 点）

(1) 計算機におけるメモリアクセスに関して以下の各小問に答えよ．

(1-1) (a)〜(g)にあてはまる語句を下の(ア)〜(コ)の選択肢から選んで記号で答えよ．ただし，同じ選択肢を複数回用いてもよい．

主記憶 (main memory) 内のデータへのアクセスは，語（ワード，word）を単位とした格納位置を示すアドレス (address) を指定することによって行われる．このとき，アクセスされたデータの写しを小容量であるが高速のキャッシュメモリ (cache memory)（以下，キャッシュ）に一時的に作っておき，次に同じデータがアクセスされたときにキャッシュからデータの写しを取り出すことにより，見かけ上主記憶のアクセス速度を高速化することが広く行われている．これは，一度アクセスされたデータは繰り返し利用される確率が高いという [(a)]，近いアドレスに格納されたデータはまとまってアクセスされる確率が高いという [(b)] を利用するものである．アクセスされるデータの写しがキャッシュに存在する確率をヒット率 (hit ratio) という．

主記憶のデータの写しは，一定の大きさの連続したアドレスに格納されたデータを単位として作られる．このとき，写しを作る大きさで主記憶を先頭から区切ったときの各々の区分をブロック (block) と呼ぶ．また，キャッシュをブロックと同じ大きさに分けた時の各々の区分をブロック枠 (block frame) という．ブロックに対しその写しを作るブロック枠を割り当てることを，ブロックのマッピング (mapping) と呼ぶ．ブロックのマッピングを行うための方式には，任意のブロック枠を割り当てることができる [(c)] 方式，ブロック枠の割り当てが一意的に決められている [(d)] 方式がある．[(d)] 方式は，[(c)] 方式よりも，キャッシュが同じ容量を持つときのヒット率は一般に [(e)] く，キャッシュの使用効率も [(e)] い．また，これら 2 つの方式を組み合わせた方式として，ブロック枠を同数の集合に分け，それぞれの集合に対し [(f)] 方式を適用する [(g)] 方式がある．

選択肢

(ア) 空間的局所性 (spatial locality)	(イ) セット連想（群連想）マッピング (set associative mapping)
(ウ) 時間的局所性 (temporal locality)	(エ) アドレスマルチプレクス (address multiplex)
(オ) 直接マッピング (direct mapping)	(カ) 完全連想マッピング (fully associative mapping)
(キ) ライトスルー (write through)	(ク) ファイルマッピング (file mapping)
(ケ) 高	(コ) 低

(1-2) キャッシュに必要な写しが存在しない新たな主記憶のブロックがアクセスされたとき，キャッシュに空きブロック枠がないならば，ブロック枠のどれかを選択してデータを入れ替える必要がある．このとき，入れ替えるべきブロック枠を選択するアルゴリズムをブロック置き換えアルゴリズム (block replacement algorithm) という．ブロック置き換えアルゴリズムとしては，LRU (Least Recently Used)，FIFO (First-In First-Out) などがある．この LRU および FIFO について，

- 実装の容易さ
- ブロック枠の数とヒット率の関係

の観点で比較した上で，それぞれの特徴をその特徴を有する理由とともに述べよ．

(1-3) 1 ブロックの大きさが 4 語，ブロック枠の数が 4 のキャッシュについて以下の各小問に答えよ．ただし，主記憶のアドレスは 8 ビットで指定されるものとする．また，ブロックのマッピング方式としては，セット数 2 のセット連想マッピング方式が用いられるものとする．

(1-3-1) 主記憶のアドレスのうち，セット内で各ブロックを識別するタグ (tag) に何ビット用いられるか答えよ．

(1-3-2) 64 個の語を格納する配列 a[i] (0 ≦ i ≦ 63) を考える．a[i] は，主記憶上のアドレス 00 ～ 3F (16 進数表記)に a[0] から，a[63] まで順に格納され，常駐しているものとする．また，主記憶上の各ブロックにはブロック番号が 0 から順に割り振られるものとする．すなわち，主記憶上のアドレス 00～03 はブロック番号 0, 04～07 はブロック番号 1, ... , 3C～3F はブロック番号 15 となる．

このとき，あるプログラムが下記の順で配列にアクセスを行ったとする．

a[0], a[4], a[8], a[53], a[54], a[55], a[56], a[4], a[20], a[21]

ブロック置き換えアルゴリズムとして LRU を用いた場合の，上記アクセスにおけるヒット率を求めよ．ただし，プログラム実行前にはキャッシュの内容は空に初期化されているものとする．

導出過程として，各アクセスの結果，ブロック枠に割り当てられるブロックのブロック番号を解答用紙の対応するセット番号の欄に記入せよ．また，アクセスされるデータの写しがキャッシュに存在するとき，「キャッシュヒット」欄に○を記入せよ．

(2) 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関して以下の各小問に答えよ．

(2-1) 次の説明文および図 1 の(a)～(f)にあてはまる語句を下の(ア)～(ソ)の選択肢から選んで記号で答えよ．

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，実行中のプログラムの実体であるプロセスは実行（running）状態，実行可能（ready）状態，待ち（waiting または blocked）状態の 3 状態を遷移する．

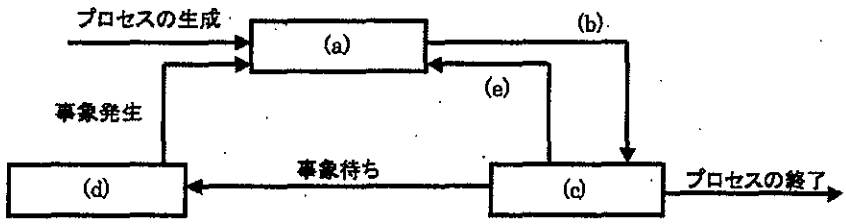


図 1： プロセスの状態遷移

生成されたプロセスはまず (a) となる．その後，そのプロセスは (b) され， (c) に遷移する． (c) にあるプロセスは，入出力処理の完了を待つなどの事象を待つ操作を開始することにより， (d) に遷移する．あるいは， (c) にあるプロセスは，強制的に (e) され， (a) に遷移する． (d) にあるプロセスは，入出力操作の完了や何らかの事象の発生により， (a) に遷移する．このように，生成されたプロセスは 3 状態を遷移し， (c) を経て終了する．なお，実行中のプロセスを次に実行すべきプロセスと切り替えることを (f) という．

選択肢

(ア) レイテンシ (latency)	(イ) コンテクストスイッチ (context switch)	(ウ) 待ち状態
(エ) コンパイル (compile)	(オ) ソースプログラム (source program)	(カ) 実行可能状態
(キ) ディスパッチ (dispatch)	(ク) 実行可能プログラム (executable program)	(ケ) リンク (link)
(コ) ライブラリ (library)	(サ) インターセクション (intersection)	(シ) 実行状態
(ス) スループット (throughput)	(セ) オブジェクトプログラム (object program)	(ソ) プリエンプション (preemption)

(2-2) 単一プロセッサをもつマルチプログラミングシステムでは，新たに生成されたプロセス，および実行可能状態になったプロセスは，レディキュー（ready queue）の末尾に格納され，レディキューの先頭よりプロセッサにディスパッチされ実行される．この際，レディキューにどのような順序でプロセスを格納し，どのタイミングでプロセッサにディスパッチするかを決定するスケジューリングアルゴリズムは，プロセスの実行順序，ターンアラウンドタイム（turn around time），応答時間（response time）に影響を与える．

いまレディキューに格納されたプロセスが全くない状態を初期状態とし，プロセス A, B, C, D が表 1 に示される生成時刻で生成され，同じく表 1 に示される処理時間で処理が完了すると仮定する．

表 1：プロセスの生成時刻とその処理時間

プロセス	生成時刻（T）	処理時間
A	0	6
B	1	1
C	4	4
D	9	2

このとき，スケジューリングアルゴリズムとして FIFO (First-In First-Out) 方式，RR (Round Robin) 方式を採用した場合，プロセス A, B, C, D がプロセッサで処理される様子をそれぞれの方式ごとに解答用紙に図示せよ．また，プロセス A, B, C, D の平均ターンアラウンドタイムおよび平均応答時間についてもそれぞれの方式について求めよ．

ただし，下記の点に留意すること．

1．　ある時刻に生成あるいはプリエンプションされたプロセスは，その時刻に直ちにレディーキューに格納される．生成されたプロセスとプリエンプションされたプロセスが，レディキューに同時刻に格納される場合は，生成されたプロセスが先にレディキューに格納される．

2．　あるプロセスの処理が完了あるいはプリエンプションされた時刻から次のプロセスがプロセッサにディスパッチされ実行される時刻までに要する処理時間（プロセスの切り替えに要する処理時間）を 1 とする．ただし，時刻 T＝0 でプロセス A がプロセッサにディスパッチされ実行されるまでに要する処理時間については 0 とする．

3．　プロセスがディスパッチされ実行されてからプリエンプションされるまでの時間をタイムスライス（time slice）とよび，それを 2 とする．

4．　タイムスライス中にプロセスが終了した時は，次のプロセスがディスパッチされる．

(配点：(1-1) 15 点，(1-2) 25 点，(2-1)〜(2-4) 各 15 点)

以下の各問に答えよ．

(1) 図 1 に示すように，円柱座標系 (circular-cylindrical coordinates) (r, φ, z) における z 軸上に太さの無視できる無限に長い直線状導体 (infinitely long straight conductor) があり，z 軸正の方向に電流 (electric current) $\boldsymbol{I}$ が流れている．この時，導体外の任意の点における磁界 (magnetic field) $\boldsymbol{H}$ を求める．以下の二つの方法を用い，それぞれ導出過程を示せ．なお，各軸方向の単位ベクトル (unit vector) を $\boldsymbol{i}_r, \boldsymbol{i}_\varphi, \boldsymbol{i}_z$ とせよ．

(1-1) 適当な積分経路 C をとり，アンペールの周回積分の法則 (Ampère's circuital law) を適用して磁界 $\boldsymbol{H}$ を求める過程を示せ．

(1-2) ビオサバールの法則 (Biot-Savart law) を用いて磁界 $\boldsymbol{H}$ を求める過程を示せ．なお，以下の関係式を用いて良い．

$$\int \frac{dx}{(x^2+a)^{\frac{3}{2}}} = \frac{x}{a\sqrt{x^2+a}} + c$$

但し，c, a は定数 (constant)．

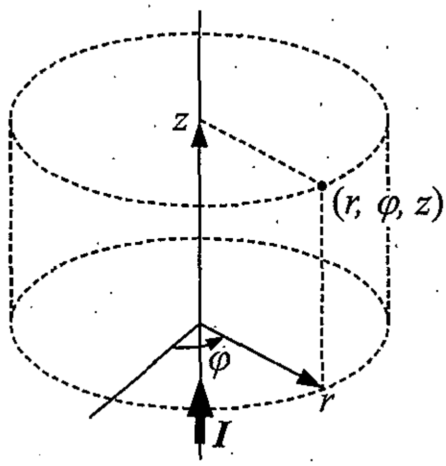


図 1

(2) 図 2 に示すように，直交座標系 (orthogonal coordinates) (x, y, z) における $z=0$ の面で，透磁率 (permeability) μ_0 の真空領域 (free space) $(z>0)$ と透磁率 μ_1 の常磁性体 (paramagnet) 領域 $(z<0)$ が接している．また，$z=d$ の面には，厚さの無視できる無限に広い導体があり，y 軸正の方向に面電流密度 (surface current density) $\boldsymbol{K}_f$ の自由電流 (free current) が流れている．このとき，以下の各小問に答えよ．いずれの問題も，導出過程を示すこと．なお，各軸方向の単位ベクトルを $\boldsymbol{i}_x, \boldsymbol{i}_y, \boldsymbol{i}_z$ とせよ．

(2-1) 領域 $0<z<d$ および $z>d$ における磁界 $\boldsymbol{H}$，磁束密度 (magnetic flux density) $\boldsymbol{B}$ を求めよ．

(2-2) 領域 $z<0$ における磁界 $\boldsymbol{H}$，磁束密度 $\boldsymbol{B}$ を求めよ．

(2-3) 領域 $z<0$ および $z>0$ における磁化ベクトル (magnetization vector) $\boldsymbol{M}$ を求めよ．

(2-4) 領域 $z<0$，$z>0$ および面 $z=0$ に現れる磁化電流密度 (magnetization current density) $\boldsymbol{K}_m$ を求めよ．

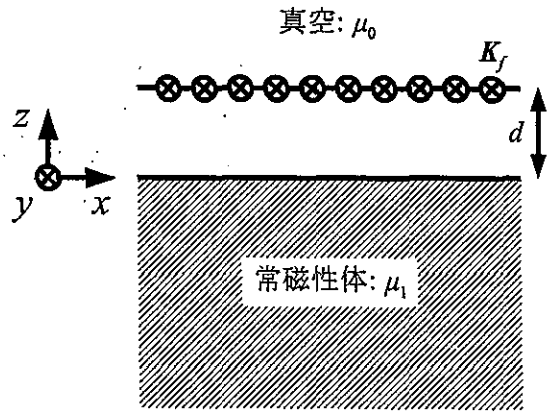


図 2

5 (選択問題：回路理論)

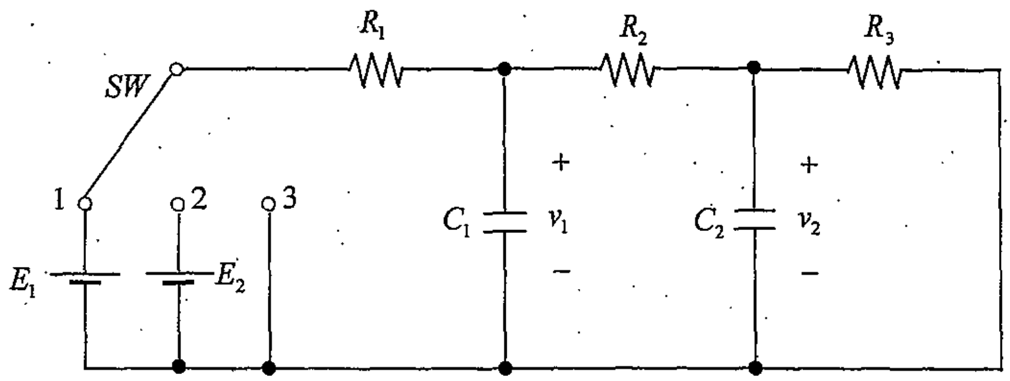
(配点：(1)20 点，(2)50 点，(3)30 点)

抵抗(resistance)とキャパシタ(capacitor)からなる下図の回路(circuit)において，時刻$t<0$でスイッチSWは端子1(terminal 1)側に閉じており定常状態(steady state)にあるものとする．以下の各問に答えよ．ただし，計算過程も記すこと．なお，$R_1=R_3=1[\Omega], R_2=\frac{1}{2}[\Omega]$，$C_1=C_2=1[\mathrm{F}]$，$E_2>E_1>0[\mathrm{V}]$とし，キャパシタ$C_1, C_2$の両端に現れる電圧をそれぞれ$v_1, v_2[\mathrm{V}]$とする．また，スイッチの切り替えに要する時間は十分に小さいものとする．

(1) キャパシタC_1, C_2に蓄えられている電荷$q_1, q_2[\mathrm{C}]$を求めよ．

(2) 時刻$t=0$にスイッチSWを端子2(terminal 2)側に切り替える．キャパシタC_2の両端に現れる電圧$v_2[\mathrm{V}]$をtの関数として求めよ．ただし，ラプラス変換(Laplace transform)を用いること．

(3) 前問(2)の状態から十分な時間が経過して定常状態に達した後，スイッチSWを端子3(terminal 3)側に切り替える．この時刻を新たに$t=0$として，抵抗R_2を流れる電流$i_2[\mathrm{A}]$をtの関数として求めよ．また，その時定数(time constant) $\tau[\mathrm{s}]$を求めよ．



（配点: (1-1)40 点, (1-2)20 点, (2)40 点）

以下の各問に答えよ．

(1)

微分方程式 (differential equation)

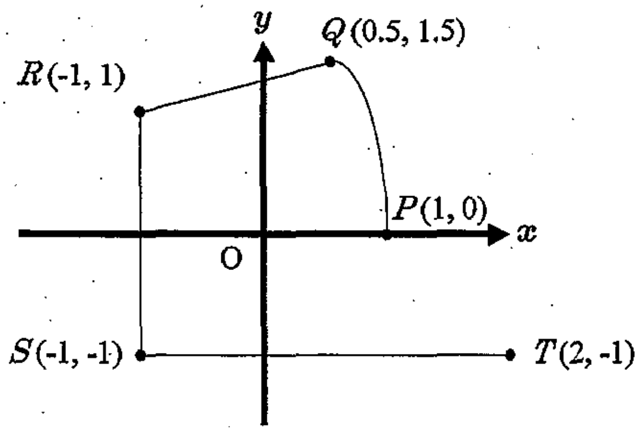
$\dfrac{dy}{dx} = 3x^2 y\,(y+1)$ に関して，以下の各小問に答えよ．

(1-1) $y\,(x)$ の一般解 (general solution) を求めよ．

(1-2) $x=0$ のとき $y=1$ となる解を求めよ．

(2)

複素関数 (complex function) $1/z$ を下図 に示す $P(1,0)$ から $Q(0.5,1.5)$ ， $R(-1,1)$, $S(-1,-1)$, $T(2,-1)$ の順にたどる積分路 (contour) C に沿って積分 (integrate) した値を求めよ．ただし，図において，x 軸は実軸，y 軸は虚軸である．



（配点: (1-1) 15 点, (1-2) 35 点, (2-1) 25 点, (2-2) 25 点）

(1) 高さ h $(h \geq 1)$ の根付き木 (rooted tree) T を考える．T の節点 (node) の集合 V に対して，次の条件を満たす関係 (relation) $R \subseteq V \times V$ を導入する．ここで，$V \times V$ は V の直積集合 (Cartesian product) を表す．

条件「T の節点 u が子 v を持つとき，そのときに限って $(u, v) \in R$」

さらに，R^i $(i = 1, 2, \cdots)$ を次のように定義する．

$$R^1 = R,$$
$$R^{i+1} = \{(u, w) | \exists v \left((u, v) \in R^i \text{ かつ } (v, w) \in R\right)\} \ (i = 1, 2, 3, \cdots).$$

以下の各小問に答えよ．

(1-1) ある $i \geq 1$ に対して，$(u, v) \in R^i$ となる u と v が存在するとき，u と v は根付き木 T 上でどのような関係にあるか，簡潔に述べよ．

(1-2) R^i が空集合となる必要十分条件を示せ．また，必要十分条件となる理由を簡潔に説明せよ．

(2) 以下では，N は 1 以上の整数の集合，$\mathcal{P}(N)$ は N のべき集合 (power set) をそれぞれ表す．順序対 (x, y) と (u, v) について，「“$(x, y) = (u, v)$” $\Leftrightarrow$ “$x = u$ かつ $y = v$”」と定義する．

このとき，次のように 4 つの集合 X_1, X_2, X_3, X_4 それぞれの上で定義された関係 R_1, R_2, R_3, R_4 を考える．

$X_1 = \mathcal{P}(N)$, $\quad R_1 = \{(x, y) \mid x \text{ は } y \text{ の部分集合}\}$
$X_2 = N \times N$, $\quad R_2 = \{((x, y), (z, w)) \mid x \text{ と } y \text{ の最大公約数と，} z \text{ と } w \text{ の最大公約数が等しい}\}$
$X_3 = N \times N$, $\quad R_3 = \{((x, y), (z, w)) \mid x \text{ と } y \text{ の積が，} z \text{ と } w \text{ の積の約数}\}$
$X_4 = N \times N$, $\quad R_4 = \{((x, y), (z, w)) \mid x \text{ が } z \text{ の約数，かつ } w \text{ が } y \text{ の約数}\}$

以下の各小問に答えよ．

(2-1) $R_i (i = 1, 2, 3, 4)$ のうち，反対称性 (anti-symmetry) を持たない関係はどれか．すべて挙げよ．また，それぞれについて反対称性を満足しない例を示せ．

(2-2) $R_i (i = 1, 2, 3, 4)$ のうち，同値関係 (equivalence relation) はどれか．すべて挙げよ．また，それぞれについて同値関係であることを証明せよ．

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

一階述語論理式 (first-order logic formula) として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，$\rightarrow, \vee, \wedge, \neg$ はそれぞれ，含意 (implication)，選言 (disjunction, or)，連言 (conjunction, and)，否定 (negation, not) の各論理演算子とする．

また，断りのない限り，u, w, x, y, z で変数 (variable) 記号，a, b で定数 (constant) 記号，f で関数 (function) 記号，p, q で述語 (predicate) 記号を表わす．

(1) 一階述語論理式の解釈 (interpretation) I は (D, C, F, P) の 4 項組で与えられる．ここで，D は値集合，C は各定数記号への D の要素の割り当て，F は各 n 引数関数記号への $\mathrm{D}^n \rightarrow \mathrm{D}$ の要素の割り当て，P は各 n 引数述語記号への $\mathrm{D}^n \rightarrow \{0,1\}$ の要素の割り当てである．ここで $0,1$ はそれぞれ偽 (false)，真 (true) を表わす．

例えば一階述語論理式 $\forall x\, p(f(b,x),a)$ に対して，解釈 I_0 として
- D を非負整数 (nonnegative integer) 全体からなる集合とし，
- C として a, b それぞれへ非負整数値 0,1 を割り当て，
- F として 2 引数関数記号 $f(u,w)$ へ非負整数上の加算 $u+w$ を割り当て，
- 2 引数述語記号 $p(u,w)$ へ非負整数上の比較演算 $u > w$ を割り当てたとき (例えば $4 > 3$ の値は 1 である)，

$\forall x\, p(f(b,x),a)$ の，解釈 I_0 のもとでの評価値は 1 となる．

以下の各小問に答えよ．なお，F, P として，簡単な演算 (四則演算，大小比較等) への割り当てを考えること．

(1-1) この小問 (1-1) では D を $\{0,1,2\}$ とする．以下の各論理式ごとに，それぞれの評価値を 1 とする具体的な解釈を 1 つ挙げよ．また，その評価値を 1 とする解釈がない場合は充足不能 (unsatisfiable) と答えよ (理由説明は不要)．

(a) $\forall x\ (p(x) \rightarrow p(a))$

(b) $\forall x\ p(f(x))$

(c) $\forall x\ (p(x) \rightarrow (\neg p(a) \vee \neg p(b)))$

(1-2) この小問 (1-2) では D を非負整数全体からなる集合とする．以下のすべての論理式に対してそれらの評価値を同時に 1 とする具体的な解釈を 1 つ挙げよ．そのような解釈が存在しない場合は充足不能と答えよ (理由説明は不要)．

(a) $\forall x\ (p(x) \vee q(x))$

(b) $\neg\forall x\ (p(x) \wedge q(x))$

(c) $\neg\exists x\ (\neg p(x) \wedge \neg q(x))$

(1-3) この小問 (1-3) では D を非負整数全体からなる集合とする．
$\forall x\ (p(x) \vee q(x)) \not\models \forall x\, p(x) \vee \forall x\, q(x)$ であること，すなわち，$\forall x\ (p(x) \vee q(x))$ の評価値を 1 にし，$\forall x\, p(x) \vee \forall x\, q(x)$ の評価値を 0 にする解釈が存在することを示せ．

(2) 以下で与えられる論理式 A が恒真 (valid) であることを，(i) A の否定のスコーレム化 (Skolemization)，(ii) 導出原理 (resolution) の適用を行い，(i) の充足不能性を示すことによって，示したい．以下の各小問に答えよ．各小問において導出過程も示すこと．

$A = (\forall y \forall x\, p(x,y) \wedge \exists y \forall x\ (p(x,y) \rightarrow q(x))) \rightarrow \forall z\ q(z)$

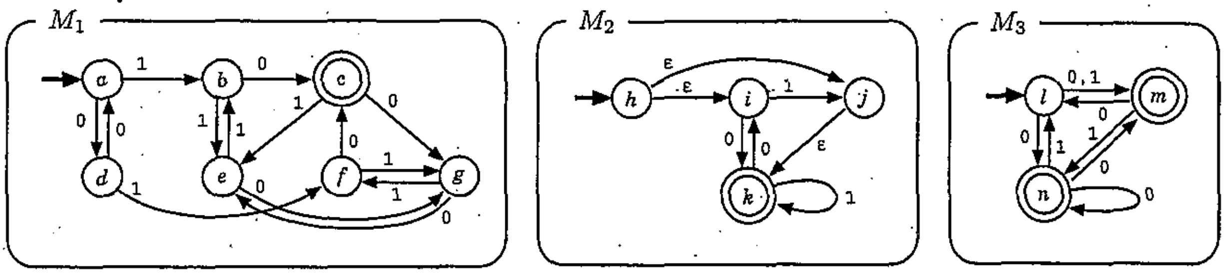
(2-1) $\neg A$ の冠頭標準形 (prenex normal form) を示せ．
冠頭標準形はすべての限定作用素が先頭にある閉論理式 (closed formula) である．

(2-2) 小問 (2-1) で得た論理式をスコーレム化せよ (スコーレム連言標準形 (Skolem conjunction normal form) を求めよ)．

(2-3) 小問 (2-2) で得た論理式 A' をもとに，導出原理を用いて A' が充足不能であることを示せ．

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

(1) 有限オートマトン (finite automaton) に関して以下の各小問に答えよ．なお有限オートマトン M_1, M_2, M_3 それぞれを以下の状態遷移図 (state transition diagram) の通り定める．各有限オートマトンの初期状態 (initial state) は a, h, l であり，受理状態 (accepting state) は c, k, m, n である．また，入力記号 (input symbol) の集合は $\Sigma = \{0, 1\}$ とする．



(1-1) 有限オートマトン M_1 が受理 (accept) する語 (word) の内，長さ 4 以下のもの全てを示せ．

(1-2) 有限オートマトン M_2 は ε-動作 (ε-move) を有する．最初に，M_2 の各状態に対する ε-閉包 (ε-closure) を示せ．次に，求めた ε-閉包から ε-動作のない非決定性 (non-deterministic) 有限オートマトン M_2' を構成し，状態遷移図で示せ．ただし M_2' は M_2 と同じ状態集合を有するものとし，状態遷移図には状態名も記入せよ．なお，ある状態 x に対する ε-閉包とは，x そのものと x から ε-動作のみで到達できる状態全ての集合をいう．

(1-3) 有限オートマトン M_3 と同じ言語 (language) を受理する状態数が最小 (minimum) の決定性有限オートマトン M_3' を求めて，状態遷移図で示せ．なお M_3' を求める過程は示さずに，結果のみを示せ．

(2) 文脈自由文法 (context-free grammar) に関して以下の各小問に答えよ．なお文法 G_1, G_2 それぞれを以下の通り定める．

$G_1(V_1, T_1, P_1, S_1)$

- 非終端記号の集合 (a set of non-terminal symbols) $V_1 = \{A\}$
- 終端記号の集合 (a set of terminal symbols) $T_1 = \{$ **if, then, else, e, s** $\}$
- 開始記号 (start symbol) $S_1 = A$
- 生成規則の集合 (a set of generating rules) $P_1 = \{\ A \rightarrow$ **if e then** $A,\quad A \rightarrow$ **if e then** A **else** $A,\quad A \rightarrow$ **s** $\}$

$G_2(V_2, T_2, P_2, S_2)$

- 非終端記号の集合 $V_2 = \{B, C, D\}$
- 終端記号の集合 $T_2 = \{$ **if, then, else, e, s** $\}$
- 開始記号 $S_2 = B$
- 生成規則の集合 $P_2 = \{\ B \rightarrow C,\quad B \rightarrow D,\quad C \rightarrow$ **if e then** $B,\quad D \rightarrow$ **if e then** D **else** $B,\quad D \rightarrow$ **s** $\}$

(2-1) 文法 G_1 における文 "**if e then s else if e then s**" に対する最左導出 (leftmost derivation) および最右導出 (rightmost derivation) における導出の過程をそれぞれを示せ．

(2-2) 文法 G_1 における文 "**if e then if e then s else s**" に対する異なる 2 通りの導出木 (derivation tree) を示せ．

(2-3) 文法 G_2 は曖昧である．具体的な根拠を示せ．

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

S は記憶のない情報源（memoryless source）であるとし，その情報源記号集合（source alphabet）を $A = \{a_1, \ldots, a_s\}$ $(s \geq 2)$ とする．各 a_i $(1 \leq i \leq s)$ の生起確率（occurrence probability）を p_i とおく（ただし $p_i > 0$ であるとする）．S を通信路記号集合（channel alphabet）$B = \{0, 1\}$ 上の系列に情報源符号化（source coding）することを考える．ここで，B^+ を B 上の空系列以外の有限系列全体の集合とし，$f : A \to B^+$ を符号化関数（coding function）とすると，A 上の系列 $a_{i_1} a_{i_2} \cdots a_{i_k}$ （$i_1, \ldots, i_k$ は 1 以上 s 以下の整数）は，対応する符号語（codeword）を連接（concatenation）した系列，すなわち $f(a_{i_1}) f(a_{i_2}) \cdots f(a_{i_k})$ に符号化されるものとする．

以下の各問に答えよ．

(1) $s = 4$ とする．以下の符号化関数 f_1 による符号化は一意に復号可能（uniquely decodable）ではないことを示せ．

$$f_1(a_1) = 00, \quad f_1(a_2) = 01, \quad f_1(a_3) = 10, \quad f_1(a_4) = 100$$

(2) $s = 5$ とする．また，符号化関数 f_2，f_3 を以下のように定義する．

$$f_2(a_1) = 00, \quad f_2(a_2) = 10, \quad f_2(a_3) = 011, \quad f_2(a_4) = 0111, \quad f_2(a_5) = 1111$$

$$f_3(a_1) = 00, \quad f_3(a_2) = 01, \quad f_3(a_3) = 110, \quad f_3(a_4) = 1110, \quad f_3(a_5) = 1111$$

以下の各小問に答えよ．

(2-1) 符号化により得られた B 上の系列は，その左端の記号から順に通信路（channel）に送り出されるとする．このとき，復号手続きの簡潔さの観点から，f_2 と f_3 のどちらによる符号化のほうが優れているか，理由とともに答えよ．

(2-2) 以下の 2 条件を満たす符号化関数 $f_4 : A \to B^+$ を一つ答えよ．そして，その f_4 が実際にそれらの条件を満たしていることを説明せよ．

- f_4 による符号化は一意に復号可能である．
- p_i $(1 \leq i \leq 5)$ の値に関わらず，f_4 による符号化の平均符号語長（average codeword length）は，f_2 や f_3 による符号化の平均符号語長よりも短い．

(3) $N_f(x)$ （x は正整数）は，符号化関数 $f : A \to B^+$ による符号語のうち長さが x に等しいものの個数を表すとする．このとき以下の各小問に答えよ．

(3-1) 符号化関数 $f_5 : A \to B^+$ による符号化が S の 2 元ハフマン符号化（binary Huffman coding）の一つであり，かつ $N_{f_5}(s-1) \neq 0$ であるとする．このとき $N_{f_5}(1)$ の値を答えよ（必要ならば s の値で場合分けせよ）．そして，その値になる理由を説明せよ．

(3-2) $s = 4$ とする．以下の 2 条件を満たす符号化関数 $f_6 : A \to B^+$，$f_7 : A \to B^+$ が存在するような $p_1, \ldots, p_4$ の値を一組答えよ．そして，そのときの f_6，f_7 を一組答えよ．

- f_6，f_7 による符号化はいずれも S の 2 元ハフマン符号化である．
- ある正整数 c が存在して $N_{f_6}(c) \neq N_{f_7}(c)$ である．

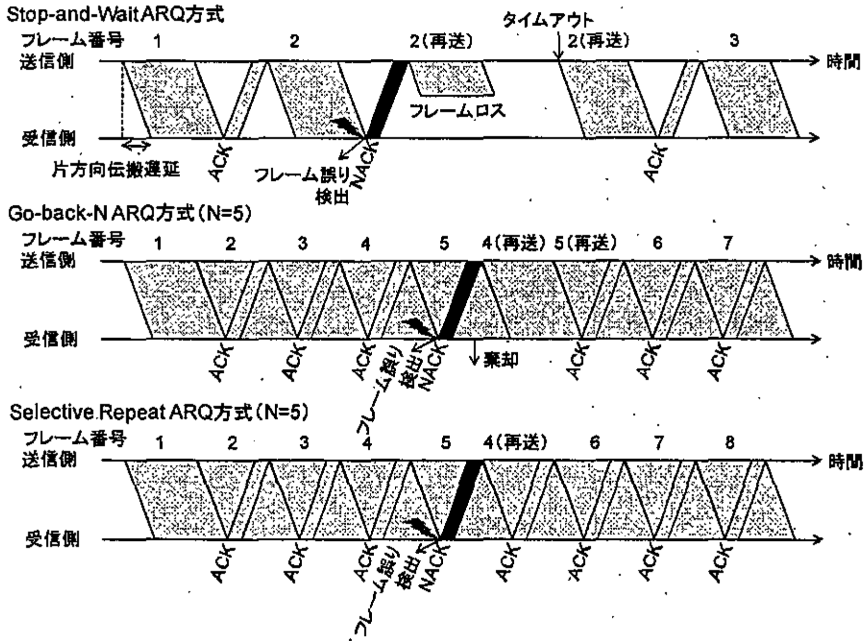
（配点：(1) 20 点，(2) 20 点，(3) 20 点，(4) 20 点，(5) 20 点）

伝送媒体上で隣接するノード (node)（ルータ (router) やホスト (host)）間の誤りのない全二重伝送路 (full duplex channel) を実現するものとして，ARQ（Automatic Repeat-reQuest）がある．ARQ には，Stop-and-Wait ARQ 方式，Go-back-N ARQ 方式，Selective Repeat ARQ 方式がある．以下の説明を読み，各問に答えよ．

Stop-and-Wait ARQ 方式では，送信側ノード（sender）はデータフレーム（data frame）を一つ送信すると，タイマ（timer）を起動させ，受信側ノード（receiver）からの応答を待つ．受信側ノードは，正しくデータフレームを受信した場合には ACK（ACKnowledgement）フレームを，受信したデータフレームに修正できないビット誤り（bit error）がある場合（フレーム誤り（frame error）と呼ぶ）には NACK（Negative ACK）フレームを送信側ノードに返信する．送信側ノードは，ACK フレームを受信した場合には次の新たなデータフレームを送信し，NACK フレームを受信した場合，またはデータフレームが失われること（フレームロス（frame loss）と呼ぶ）によってタイムアウト（timeout）が発生した場合にはそのデータフレームを再送する．

Go-back-N ARQ 方式では，送信側ノードは ACK フレームを受け取ることなく同時に最大 N 個のデータフレームを送信することができる．送信側ノードは，送信済みのデータフレームに対して ACK フレームを受信すると新たなデータフレームを送信するが，NACK フレームを受信，またはタイムアウトが発生した場合には，誤りの生じた，または失われたデータフレーム以降のすべてのデータフレームを再送する．受信側ノードでは，誤りの生じた，または失われたデータフレームを正しく受信できるまで，受信したデータフレームをすべて棄却する．

また，Selective Repeat ARQ 方式では，Go-back-N ARQ 方式と同様に ACK フレームを受け取ることなく最大 N 個のデータフレームを送信するが，送信側ノードは NACK フレームを受信した，あるいはタイムアウトが発生したデータフレームのみを再送する．それぞれの動作の例を下図に示す．



(1) 送受信ノードにおいて,

I) Stop-and-Wait ARQ 方式には必要なく，Go-back-N ARQ 方式には必要な機能,

II) Go-back-N ARQ 方式には必要なく，Selective Repeat ARQ 方式には必要な機能

を，それぞれ 2 つずつ以下から選んで記号を書き，それぞれについて簡潔に理由を述べよ．3 つ以上の選択肢が該当する場合には，そのうちの 2 つについて解答すればよい.

(a) データフレーム番号によるデータフレーム識別　(b) 受信側ノードでの受信データフレームの並べ替え

(c) 受信側ノードでの受信済みデータフレーム番号の管理　(d) 送信側ノードでの送信データフレーム数の管理

(e) 送信側ノードでの最大 N 個分のデータフレームバッファの管理

(f) 受信側ノードでの最大 N 個分のデータフレームバッファの管理

(2) 片方向伝搬遅延（one-way propagation delay）40 ms，通信速度 1000 bps の，フレーム誤りもフレームロスも発生しない伝送媒体において，長さ 100 bit のデータフレームを 99 個送信した際の，Stop-and-Wait ARQ 方式と Go-back-N ARQ 方式それぞれの平均スループットを求めよ．計算過程も示せ．なお，ACK フレームと NACK フレームの長さを 20 bit とし，Go-back-N ARQ 方式における N を 3 とする．平均スループットは，送信側ノードにおける一つ目のデータフレーム送信開始から最後のデータフレームに対する ACK フレーム受信完了までの単位時間あたりの送信データ量であり，単位は bps である．なお，問(2)～問(4)においては，ACK フレーム，NACK フレームともに誤り，損失は発生しないものとする.

(3) 問(2)において，片方向伝搬遅延を 200 ms とし，それ以外の条件を同じとした場合の，Stop-and-Wait ARQ 方式と Go-back-N ARQ 方式それぞれの平均スループットを求めよ．計算過程も示せ.

(4) 問(2)において，伝送媒体でのフレームロスは発生しないが，確率 0.1 でフレーム誤りが発生するものとし，それ以外の条件を同じとした場合の，Stop-and-Wait ARQ 方式と Go-back-N ARQ 方式それぞれの平均スループットを求めよ．計算過程も示せ.

(5) 下図は，片方向伝搬遅延 40 ms，通信速度 1000 bps の伝送媒体において，長さ 100 bit のデータフレームを順次送信し続けた際の，Stop-and-Wait ARQ 方式，Go-back-N ARQ 方式，Selective Repeat ARQ 方式の平均スループットを示したものである．なお，ACK フレームと NACK フレームの長さを 20 bit とし，Go-back-N ARQ 方式，Selective Repeat ARQ 方式における N を十分大きいものとする．横軸が表す指標を答えよ．また，グラフ内の各線が上から順にそれぞれどの方式を表しているか答え，その理由を簡潔に述べよ.