平 成 ２０ 年 度

名古屋大学大学院情報科学研究科
情報システム学専攻
入 学 試 験 問 題

専　　　門

平成１９年８月９日（木）
１２：３０～１５：３０

注 意 事 項

１．試験開始の合図があるまでは、この問題冊子を開いてはならない。

２．試験終了まで退出できない。

３．外国人留学生は、和英辞書などの辞書を１冊に限り使用してよい。
電子辞書の持ち込みは認めない。

４．問題冊子、解答用紙３枚、草稿用紙３枚が配布されていることを確認せよ。

５．問題は、下記の６科目から３科目を選択して解答せよ。

(1) 解析・線形代数　(2) 確率・統計　(3) プログラミング
(4) 計算機理論　(5) ハードウェア　(6) ソフトウェア

なお、選択した科目名を解答用紙の指定欄に記入せよ。
計算機理論を選択する場合は、計算機理論Iと計算機理論IIのいずれか一方を選択して解答せよ。計算機理論Iと計算機理論IIの両方を選択することはできない。解答用紙の指定欄には計算機理論Iまたは計算機理論IIと記入せよ。

６．ＩＴスペシャリストコースへの配属を希望する者は、必ず、「プログラミング」を選択して回答すること。

７．解答用紙は、指定欄に受験番号を必ず記入せよ。解答用紙に受験者の氏名を記入してはならない。

８．解答用紙は、試験終了後に３枚とも提出せよ。

９．問題冊子、草稿用紙は、試験終了後に持ち帰ってよい。

解析・線形代数

(解の導出過程も書くこと)

［1］ 次(つぎ)の微分方程式(びぶんほうていしき)の一般解(いっぱんかい)を求(もと)めよ．

$$\frac{d^2y}{dx^2} + 4\frac{dy}{dx} + 4y = e^{-x}$$

［2］ 極座標(きょくざひょう) (r, θ) で表(あらわ)したときに，$r(\theta)$ が次の式となる曲線(きょくせん)を考える．

$$r^2(\theta) = \cos 2\theta \quad (-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4},\ \ r \geq 0)$$

(1) $\frac{dr}{d\theta}$ を求めよ．

(2) $\frac{dr}{d\theta} = 0$ となる θ の値(あたい)と，それに対応(たいおう)する $r(\theta)$ を求めよ．

(3) 直交(ちょっこう)座標 (x, y) で表したときに，$\frac{dy}{dx} = 0$ となる θ の値と，それに対応する $r(\theta)$ を求めよ．

(4) この曲線で囲(かこ)まれる部分(ぶぶん)の面積(めんせき)を求めよ．

［3］ 次の行列(ぎょうれつ)A と E について，以下(いか)の問(と)いに答(こた)えよ．

$$\mathrm{A} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 2 \\ -1 & 1 & 3 \end{pmatrix} \qquad \mathrm{E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(1) 行列 A の固有値(こゆうち)と，それに対応する単位(たんい)固有ベクトルを求めよ．

(2) 行列 A に対して，A^2 の値を求めよ．

(3) 行列 A に対して，$\mathrm{A}^3 = c_2\mathrm{A}^2 + c_1\mathrm{A} + c_0\mathrm{E}$ と表すとき，係数(けいすう) c_2, c_1, c_0 の値を，それぞれ求めよ．

注:

微分方程式(びぶんほうていしき) *differential equation*	一般解(いっぱんかい) *general solution*
極座標(きょくざひょう) *polar coordinates*	直交座標(ちょっこうざひょう) *orthogonal coordinates*
面積(めんせき) *area*	行列(ぎょうれつ) *matrix*
固有値(こゆうち) *eigenvalue*	単位固有ベクトル(たんい) *unit eigenvector*
係数(けいすう) *coefficient*	

確率・統計（解の導出過程も書くこと.）

次の問いに答えなさい.

［1］ ある母集団から，6個の標本を無作為抽出したとき，標本平均が 1.0，標本分散（標本を全データとして計算した分散）が 2.0 であった．母分散の不偏推定量を求めよ．また，母分散の不偏推定量と標本分散に違いが発生する理由を数行で簡単に説明せよ．

［2］ 事象Aの発生確率が p である試行を繰り返すとき，次の問いに答えよ．

(1) $x+1$ 回目にはじめて事象 A が発生する確率 $\Pr\{X=x\}$ $(x=0,1,2,\cdots)$ を求めよ．

(2) 確率変数 X,Y が独立で，それぞれが(1)の確率分布に従うとき，$\Pr\{X+Y=k\}$ を求めよ．

(3) 上記(1)の確率分布のモーメント母関数 $m(t)=E[e^{tX}]$ を求めよ．

(4) モーメント母関数を用いることにより，X の期待値 $E[X]$ を求めよ．（途中の式変形を省略しないこと）

【注意】 (2),(4)の解答では，Σを残したままにしないこと．次の公式を利用してよい．

$$(-1<r<1)\text{ のとき，}\ \sum_{i=0}^{\infty} r^i = \frac{1}{1-r}$$

［3］ 確率変数 X,Y が $f_{X,Y}(x,y)=\begin{cases} e^{-x-y} & (0\le x,\quad 0\le y) \\ 0 & (otherwise) \end{cases}$ に従うとき，

(1) 条件付確率密度関数 $f(y\,|\,x)$ を求めよ．

(2) また $Z=X+Y$ としたときの確率密度関数 $f_Z(z)$ を求めよ．

【専門用語の英訳】

母集団 population，標本 sample，無作為 random，平均 mean，分散 variance，母平均 population mean，
母分散 population variance，不偏推定 unbiased estimate，確率変数 random variable，事象 event，
試行 trial，独立 independence，確率分布 probability distribution，母関数 generating function，
期待値 expectation，条件付確率 conditional probability，密度関数 density function

プログラミング

次の (1)〜(4) に答えよ．

(1) リスト1は、以下で定義されるフィボナッチ (Fibonacci) 数 F_n を計算する C 言語のプログラムである．

$$\begin{cases} F_1 & = & 1 \\ F_2 & = & 1 \\ F_{n+2} & = & F_{n+1} + F_n \quad (n \geq 1) \end{cases}$$

このプログラムを実行し、標準入力 (standard input) から 6 を入力したとき、func(1), func(2),...,func(6) が呼び出される回数をそれぞれ示せ．また、その導出過程も示せ．

```
#include <stdio.h>

int func(int n)
{
  if (n <= 2) {
    return 1;
  } else {
    return func(n - 1) + func(n - 2);
  }
}

int main()
{
  int n, fn;
  printf("Input a positive integer: ");
  scanf("%d", &n);
  if (n > 0) {
    fn = func(n);
    printf("func(%d) = %d\n", n, fn);
    return 0;
  } else {
    printf("Error!\n");
    return -1;
  }
}
```

リスト 1

(2) リスト1の3行目から10行目を以下のリスト2で置き換え、リスト1と同じ計算をしたい.

(ア) リスト2中の (a)～(d) に適切な式を入れよ.

(イ) このプログラムを実行し、標準入力から 6 を入力したとき、
リスト2の8行目の printf 文の出力結果を順に示せ.

(ウ) このように置き換えたとき、リスト1と比べ、どのような点で優れているか述べよ.

```
1   int func(int n)
2   {
3     return func_sub(n, 1, 0);
4   }
5
6   int func_sub(int n, int a1, int a2)
7   {
8     printf("func_sub(%d, %d, %d)\n", n, a1, a2);
9     if ( (a) ) {
10      return  (b) ;
11    } else {
12      return func_sub(n - 1,  (c) ,  (d) );
13    }
14  }
```

リスト 2

(3) リスト1の3行目から10行目を、以下のリスト3で置き換え、リスト1と同じ計算をしたい.
リスト3中の (e)～(i) に適切な式を入れよ. ただし、再帰呼び出し (recursive call) を
用いてはならない.

```
int func(int n)
{
  int res = 1;
  int a3 = 1;
  int a4 = 0;
  while ( (e) ) {
     (f)  =  (g) ;
    a4 =  (h) ;
    a3 =  (i) ;
    n = n - 1;
  }
  return res;
}
```

リスト 3

（４）リスト１の３行目から 10 行目を、以下のリスト４で置き換え、以下で定義されるトリボナッチ (Tribonacci) 数 T_n を計算したい.

$$\begin{cases} T_1 & = 1 \\ T_2 & = 1 \\ T_3 & = 2 \\ T_{n+3} & = T_{n+2} + T_{n+1} + T_n \ (n \geq 1) \end{cases}$$

リスト４中の (j)～(u) に適切な式を入れよ.

```
int func(int n)
{
  int res = 1;
  int a6 = (j) ;
  int a7 = (k) ;
  int a8 = (l) ;
  while ( (m) ){
    (n) = (o) ;
    (p) = (q) ;
    (r) = (s) ;
    (t) = (u) ;
    n = n - 1;
  }
  return res;
}
```

リスト 4

計算機理論 I

計算機理論 I を選択する場合には、解答用紙の指定欄に計算機理論 I と記入せよ。なお、計算機理論 I と計算機理論 II の両方を選択することはできない。

[1]

(1) 0,1 によって 2 進数表現 (binary expression) された非負整数 (non-negative integer) を上位から入力して、その数を 3 で割ったときの剰余 (remainder) が 1 となるとき、かつ、そのときに限り、その 2 進数表現を受理 (accept) する決定性有限オートマトン (deterministic finite automaton) の状態遷移図 (state transition diagram) を示せ。状態遷移図には、初期状態 (initial state) と受理状態 (acceptance state) を明示すること。ここで、入力とする 2 進数表現には、00111 のように、上位に 0 の並びを含んでもよいとする。

(2) この決定性有限オートマトンが受理する言語 (language) を長さが 25 を越えない正則表現 (regular expression) で示せ。アルファベット $\{0,1\}$ に対する正則表現 R は以下の拡張 BNF(extended BNF) で定義されるものとする。

$$R \quad ::= \quad \varepsilon \mid \emptyset \mid 0 \mid 1 \mid R \cdot R \mid R \; + \; R \mid (R) \mid R^*$$

ここで、$\cdot$ は連接 (concatenation)、$+$ は和 (set union)、$*$ は閉包 (closure) を表し、演算子の結合の強さは、閉包＞連接＞和とする。正則表現の長さ (length) とは正則表現に現れるすべての記号の数とする。また、連接記号については、明らかな場合は省略してもよいが、長さには含める。(例：$a \cdot b$ は ab と省略してよいが、長さはいずれも 3 である。)

[2]

$L_0 = \{1^n 0^n \mid n \geq 0\}$ を受理言語とする有限オートマトンが存在しないことを示す。ここで a^m はシンボル a の m 個の連接を表す。

L_0 を受理言語とする有限オートマトン A が存在するとするとし、その状態数を $k > 0$ とする。すると、$1^k 0^k \in L_0$ であるので、A は語 $1^k 0^k$ を受理する。このときの、A の状態の系列を $q_0, q_1, \ldots, q_k, \ldots, q_{2k}$ とする。ここで、q_0 は A の初期状態 (initial state) であり、q_{2k} は受理状態である。

(1) A は、L_0 に含まれない語も受理しなければならないことを示せ。

(2) 有限オートマトンの受理言語に対する以下の性質を証明せよ。

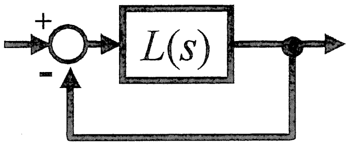
> 言語 L が有限オートマトンの受理言語であるならば以下の条件をみたす正整数 p が存在する。L に属する長さ p 以上の語 x は $x = uvw$ として分解することができ、ここで、$|v| > 0$ かつ、すべての非負整数 n に対して $uv^n w \in L$ である。(v^n は語 v の n 回の連接を表す。)

(3) L_0 には、上記の条件を満たす p は存在しないことを示せ。

計算機理論 II

計算機理論 II を選択する場合には、解答用紙の指定欄に計算機理論 II と記入せよ。なお、計算機理論 I と計算機理論 II の両方を選択することはできない。

[1] 下図及び下式において、$L(s)$は開ループ一巡伝達関数である。$L(j\omega)$は、$L(s)$の周波数伝達関数であるが、$\omega=0\sim\infty$ としたとき、$L(j\omega)$の複素平面上でのベクトル軌跡について、下記の質問に答えよ。なお解答に際して解の導出過程も明記すること。



$$L(s)=\frac{Kab}{s(s^2+as+b)}\quad (K,a,b>0)$$

(1) $L(j\omega)$のベクトル軌跡の始点近傍 $(\omega\approx 0)$ における $L(j\omega)$ の実部を求めよ。

(2) $L(j\omega)$のベクトル軌跡が終点 $(\omega=\infty)$ に漸近するとき、$L(j\omega)$ の偏角を求めよ。

(3) $L(j\omega)$ の偏角が $-180°$となる周波数、すなわち位相交差周波数ω_{cp}と、そのときの$L(j\omega_{cp})$ を求めよ。また、この値と(1), (2) の結果を用いて、$L(j\omega)$のベクトル軌跡、すなわちナイキスト線図の概形を描き、安定となるKの範囲を求めよ。

[2] 下式の伝達関数$G(s)$ を有するシステムに関して、下記の質問に答えよ。なお解答に際して解の導出過程も明記すること。

$$G(s)=\frac{(s+c)(s+d)}{(s+a)(s+b)}\quad (a,b>0)$$

(1) 入力 $v_i(t)=\sin\omega t$ に対する定常状態における出力を求めよ。

(2) 次に (1)の結果を用いて、入力が正弦波の場合、入力と定常状態における出力の間の関係（振幅比、位相差）が$G(j\omega)$ を用いて表されることを示せ。

Translations of technical terms

開ループ一巡伝達関数:open loop transfer function
周波数伝達関数:frequency domain transfer function
複素平面:complex plane
ベクトル軌跡:vector locus
偏角:argument
位相交差周波数:phase crossover frequency
ナイキスト線図:Nyquist diagram
安定:stability
伝達関数:transfer function
定常状態:steady state
正弦波:sine wave
振幅比:amplitude ratio
位相差:phase difference

ハードウェア

[1] プロセッサアーキテクチャにおけるスーパースカラ (superscalar) 方式および VLIW 方式について説明し，両者を比較し得失を述べよ．

[2] 以下の動作をする同期式順序回路 (synchronous sequential circuit) について，各問に答えよ．

- 入力は x_1, x_2，出力は z_1, z_2.
- 0〜2 の整数値を保持するカウンタを持ち，
  $x_1 = 1, x_2 = 0$ であれば法 3 (modulo 3) のもとで 1 を加算し，
  $x_1 = 1, x_2 = 1$ であれば法 3 (modulo 3) のもとで 1 を減算し，
  $x_1 = 0$ であれば現在の値を保持する．
- カウンタ値が 2 から 0 になるとき $z_1 = 1, z_2 = 0$,
  カウンタ値が 0 から 2 になるとき $z_1 = 0, z_2 = 1$,
  その他の場合は $z_1 = 0, z_2 = 0$ を出力する．

(1) この順序回路の状態遷移図 (state transition diagram) および状態遷移表 (state transition table) を示せ．

(2) D フリップフロップ (D flip-flop) を 2 個用いて，この順序回路を設計する．状態変数を y_1 および y_2 とし，カウンタ値 0, 1, 2 にそれぞれ $(y_2, y_1) = (0,0), (0,1), (1,1)$ を割り当てる．このときの状態変数関数 (state variable function) と出力変数関数 (output variable function) を，x_1, x_2 および y_1, y_2 のできるだけ簡単な積和形論理式 (sum-of-products expression) で表せ．

ソフトウェア

［1］ 文脈自由文法(context-free grammar) G = (N, T, P, S) について以下の問いに答えよ．ここで，N は非終端記号(non-terminal symbol)の集合，T は終端記号(terminal symbol)の集合であり，以下のように定義する．

N = { S, E, F }

T = { i, +, *, [,] }

P は生成規則(production rule)の集合であり，以下の規則からなる．

S → E | S+E

E → F | E*F

F → i | [S] | i[S]

S は開始記号(start symbol)である．

(1) 以下の文(statement)の構文木(parse tree)を示せ．

(A) i+i*i

(B) i*i*i

(C) i[i+i]*i

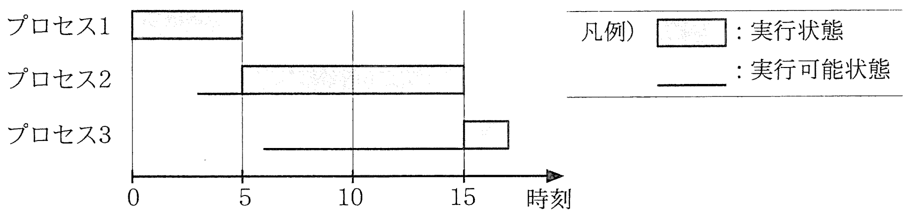
(2) 文法 G の各非終端記号について，First 集合と Follow 集合を求めよ．

(3) 文法 G が LL(1)文法でないことを示せ．

(4) 文法 G と等価な LL(1)文法 G’ = (N’, T, P’, S) を定義せよ．

［2］表に示す3つのプロセスを，到着順（FCFS; First Come First Served）スケジューリングによって実行したところ，図に示すようにスケジュールされた．

	処理時間	到着時刻
プロセス1	5秒	0秒後
プロセス2	10秒	3秒後
プロセス3	2秒	6秒後



(1) この図において，プロセスの平均応答時間（プロセスが到着してから完了するまでの平均時間）を求めよ．

(2) 同じプロセスセットを，ラウンドロビン（round robin）スケジューリングにより実行した時のスケジュールを図示し，プロセスの平均応答時間を求めよ．ただし，タイムスライス（time slice）は1秒とし，プロセス切換えにかかる時間は無視するものとする．また，新たに到着したプロセスは，すでに到着しているプロセスより先に実行するものとする．

(3) ラウンドロビンスケジューリングにより実行するよりも，到着順スケジューリングによって実行した方が，プロセスの平均応答時間が短くなるプロセスセットを1つ示せ．両スケジューリング方式により実行した場合のスケジュールを図示し，プロセスの平均応答時間を求めよ．

(4) 到着順スケジューリングとラウンドロビンスケジューリングの利点と欠点を議論せよ．

(5) 表のプロセスセットを，以下に説明する多段フィードバックキュー（multilevel feedback queues）により実行した時のスケジュールを図示し，プロセスの平均応答時間を求めよ．ただし，プロセス切換えにかかる時間は無視するものとする．

- 高優先度キュー（high priority queue）と低優先度キュー（low priority queue）を用意し，その時点までの合計実行時間が3秒未満のプロセスは高優先度キューで，それ以外のプロセスは低優先度キューで管理する．
- 高優先度キューにプロセスがある場合は，それらをラウンドロビンスケジューリングにより実行し，高優先度キューにプロセスがない場合は，低優先度キューにあるプロセスをラウンドロビンスケジューリングにより実行する．
- タイムスライスは1秒とする．
- 新たに到着したプロセスは，すでに到着しているプロセスより先に実行する．