

平成 17 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 16 年 8 月 10 日 (火)  
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、日本語から母国語への辞書 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 4 枚、草稿用紙 4 枚が配布されていることを確認せよ。
5. 問題は、A、B の各科目群について下記のように解答せよ（合計 4 科目を  
選択して解答せよ）。  
A 群：次の 3 科目から 2 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  
B 群：次の 3 科目から 2 科目を選択して解答せよ。  
(1) 計算機理論 (2) ハードウェア (3) 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。
6. 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。
7. 解答用紙は、試験終了後に 4 枚とも提出せよ。
8. 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

(解の導出過程を書くこと)

[1] 以下の手順に従って、次の微分方程式の一般解を求めよ。

$$xy'' + y' - x = 0 \quad (y' = \frac{dy}{dx})$$

(a)  $p = y'$  とおき、 $p'$  を  $p$  と  $x$  で表せ。

(b) (a) で求めた微分方程式は、 $p = xu$  とおくことで同次形と呼ばれる形になる。これを解き、 $u$  を  $x$  で表せ。

(c) (b) で得られた式に  $(p =) xu = y'$  の関係を適用することで、与えられた微分方程式の一般解を求めよ。

[2] 次の漸化式について、以下の問いに答えよ。

$$x_{n+3} = 2x_{n+2} + 5x_{n+1} - 6x_n$$

ただし、 $n = 0, 1, 2, \dots$ 、また  $x_0 = 3, x_1 = 2, x_2 = 14$  とする。

(a)  $\vec{x}_n = \begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix}$  とするとき、 $\vec{x}_{n+1} = A\vec{x}_n$  となる行列  $A$  を求めよ。

(b) 行列  $A$  の固有値と、各固有値に対応する固有ベクトルを求めよ。

(c)  $\vec{x}_0$  を行列  $A$  の固有ベクトルの線形和で表せ。

(d) (c) の結果を用いて、 $\vec{x}_8$  を求めよ。

## Translations of technical terms

微分方程式 (differential equation)

一般解 (general solution)

同次形 (homogeneous)

漸化式 (recurrence relation)

行列 (matrix)

固有ベクトル (eigenvector)

線形和 (linear combination)

# 確率・統計

(解の導出過程を書くこと)

互いに独立な  $n$  個の確率変数  $X_1, \dots, X_n$  が以下の分布に従うとする.

$$f_{X_i}(x_i) = \begin{cases} 1 & (0 \leq x_i \leq 1) \\ 0 & (\text{その他}) \end{cases} \quad (i=1 \cdots n)$$

[1] 平均  $E[X_i]$  と分散  $V[X_i]$  を求めよ.

また,  $\bar{X} = \frac{1}{n}(X_1 + X_2 + \cdots + X_n)$  とするとき, 分散  $V[\bar{X}]$  を求めよ.

[2]  $X_1$  と  $X_2$  の同時確率密度関数  $f_{X_1, X_2}(x_1, x_2)$  を求めよ.

[3]  $|X_1 - X_2| > d$  となる確率を求めよ.  $d$  は  $0 < d < 1$  とする.

[4]  $|X_1 - X_2| > d$ ,  $|X_2 - X_3| > d$ ,  $|X_3 - X_1| > d$  を同時に満たす確率を求めよ.

$d$  は  $0 < d < \frac{1}{2}$  とする.

[5] 全ての  $i, j$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $i \neq j$ ) に対して,  $|X_i - X_j| > d$  となる確率を求めよ.

$d$  は  $0 < d < \frac{1}{n-1}$  とする.

## 【専門用語の英訳】

独立: independence, 確率変数: random variable, 分布: distribution,

平均: mean, 分散: variance,

同時確率密度関数: joint probability density function

## プログラミング

添字 (index) が 0 から始まる要素  $n$  個の配列 (array)  $a$  上に, 頂点 (node) に値をもつ二分木 (binary tree) を次のように実現する. 根 (root) を  $a[0]$ ,  $a[i]$  の左の子 (left son) を  $a[2i+1]$ , 右の子 (right son) を  $a[2i+2]$  とする.

(1) 以下に示す要素数 10 の配列  $a$  が実現する二分木を図示せよ.

$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$	$a[8]$	$a[9]$
4	3	13	8	2	11	5	1	16	2

次に, 以下のような C 言語プログラムを与える.

```
#include <stdio.h>

int a[]={4,3,13,8,2,11,5,1,16,2};

void proc1(int i,int j) {
    int temp,k;

    k=i;
    if(i*2+1<=j) {
        if(i*2+2<=j) {
            if(a[i*2+1]>a[i]||a[i*2+2]>a[i]) {
                if(a[i*2+1]>a[i*2+2]) k=i*2+1; else k=i*2+2;
            }
        } else if(a[i*2+1]>a[i]) k=i*2+1;
        if(k!=i) {
            temp=a[i]; a[i]=a[k]; a[k]=temp;
            proc1(k,j); }
    }
}

void proc(int i,int j) {
    int k;
    for(k=j;k>=i;k--) proc1(k,j);
}

main() {
    int i;

    proc(0,9);
    for(i=0;i<10;i++) printf("a[%d]=%d\n",i,a[i]);
}
```

- (2) 上記プログラムの出力を示せ。さらに、関数 main の実行が終了する時点で、配列 a が実現する二分木を図示せよ。
- (3) (2) で得られた二分木において、任意の頂点の値とその子頂点の値との間に成立する関係を述べよ。
- (4) 上記プログラムの main() 関数を以下のように変更し、配列 a の内容を小さい順に整列 (sort) するプログラムを作成した。なぜこのような結果が得られるのか、(3) の性質を用いて説明せよ。

```
#include <stdio.h>

int a[]={4,3,13,8,2,11,5,1,16,2};

void proc1(int i,int j) {
    int temp,k;

    k=i;
    if(i*2+1<=j) {
        if(i*2+2<=j) {
            if(a[i*2+1]>a[i]||a[i*2+2]>a[i]) {
                if(a[i*2+1]>a[i*2+2]) k=i*2+1; else k=i*2+2;
            }
        } else if(a[i*2+1]>a[i]) k=i*2+1;
        if(k!=i) {
            temp=a[i]; a[i]=a[k]; a[k]=temp;
            proc1(k,j); }
    }
}

void proc(int i,int j) {
    int k;
    for(k=j;k>=i;k--) proc1(k,j);
}

main() {
    int i,temp;

    for(i=9;i>0;i--) {
        proc(0,i);
        temp=a[i]; a[i]=a[0]; a[0]=temp;
    }
    for(i=0;i<10;i++) printf("a[%d]=%d\n",i,a[i]);
}
```

## 計算機理論

[1] 以下の命題 (proposition) の真偽 (true/false) を判定せよ。

判定の理由も述べよ。ここに,  $R$  は実数全体の集合とする。

- 1)  $(\exists x \in R)(\forall y \in R)(\exists z \in R) x^2 = y^2 + z^2$
- 2)  $(\exists y \in R)(\forall z \in R)(\exists x \in R) x^2 = y^2 + z^2$
- 3)  $(\exists z \in R)(\forall x \in R)(\exists y \in R) x^2 = y^2 + z^2$

[2] 次の遷移表 (state transition table) で表される決定性有限オートマトン (deterministic finite automaton) について, 下の問に答えよ。

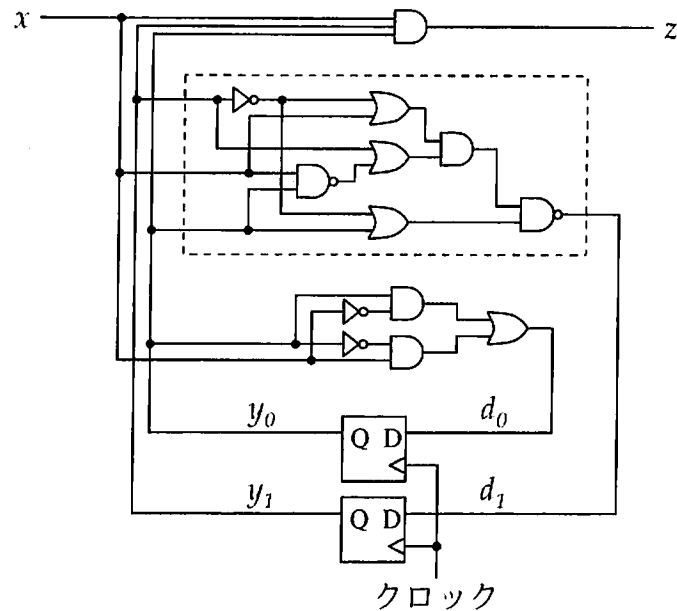
	0	1
→A	A	B
B	C	D
*C	A	E
*D	C	D
E	C	D
F	B	F

ここに, →は初期状態 (initial state), \*は受理状態 (accepting state) を表す。

- 1) この有限オートマトンを最小化せよ。
- 2) この有限オートマトンが受理する言語を, 正則表現 (正規表現, regular expression) で表せ。ただし, 表現の長さ (記号数) を 15 以内にすること。
- 3) この有限オートマトンと等価な, 状態数が 3 の非決定性有限オートマトン (non-deterministic finite automaton) を示せ。

## ハードウェア

- [1] 下図に示す順序回路 (sequential circuit) について、以下の問に答えよ。なお、下部の2つの四角はDフリップフロップ (D flip flop) を表している。



- 1) 破線で囲まれた組み合わせ回路 (combinational circuit) が計算する3変数論理関数のカルノー図 (Karnaugh map) を示せ。
  - 2) 1) の論理関数の最小積和形表現 (minimum sum-of-product form) を示せ。
  - 3) この順序回路が計算する有限状態機械 (finite state machine) の状態遷移図 (state transition diagram) を示せ。
- [2] プロセッサ P は、 $I_A$  と  $I_B$  の2種類の命令を持つ。命令  $I_A$  の CPI (clock cycles per instruction) は3、命令  $I_B$  の CPI は5である。このプロセッサについて、以下の問に答えよ。
- 1) あるプログラムをプロセッサ P 上で実行したとき、全実行命令に占める命令  $I_A$  の割合が  $x$  であった。このときの平均 CPI を求めよ。
  - 2) プロセッサ P と同一の命令セットを持つプロセッサ P' を設計した。プロセッサ P' は、プロセッサ P よりクロック周波数が25%高く、命令  $I_A$ 、命令  $I_B$  ともに CPI が5である。2つのプロセッサ P と P' の性能の優劣を議論せよ。
- [3] キャッシュ (cache) の1つのブロック (ラインとも言う) の大きさと、キャッシュの性能との関係を議論せよ。ただし、キャッシュの容量 (タグ領域は除く) や連想度 (associativity) は一定とする。

## ソフトウェア

[1] 図1はコンパイラ (compiler) の概念図であり, 表1はコンパイラに関する用語のリストである. 図1の(1)~(11)に, 表1から適当な用語を選んで入れて, コンパイラ概念図を完成せよ. 同じ番号の場所には同じ用語を入れるものとする.

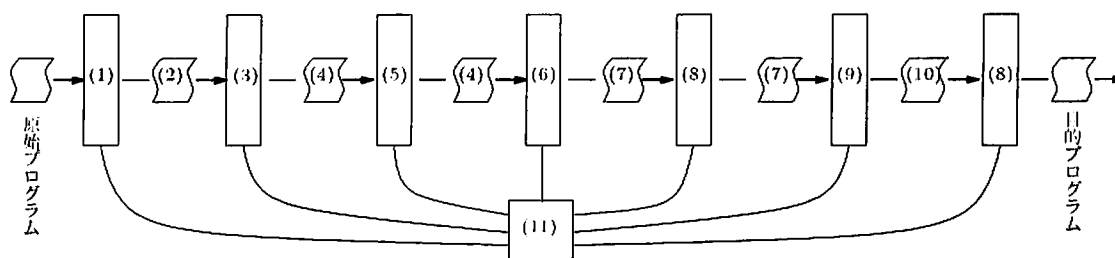


図1 コンパイラ概念図

表1

型変換 (type conversion), 記号表 (symbol table), 駆動レコード (activation record), 原始プログラム (source program), 構文解析 (parsing), 構文解析木 (parse tree), コード生成 (code generation), 最適化 (optimization), 字句解析 (lexical analysis), 字句列 (token sequence), スコープ (scope), 静的検査 (static checking) [意味解析 (semantic analysis) とも言う], 中間コード (intermediate code), 中間コード生成 (intermediate code generation), ディスプレイ (display), 動的リンク (dynamic link), フレーム (frame), 目的プログラム (object program)

[2] コンパイラに関する次の用語 (a)~(c) について以下の問に答えよ.

- (a) 実行時環境 (run-time environment)
- (b) 上昇型構文解析 (bottom-up parsing)
- (c) 字句解析 (lexical analysis)

(1) (a)~(c) について, 表2の中から関係の深い用語をそれぞれ3つ選び出せ.

(2) (1) で選んだ9つの用語をそれぞれ100字以内で説明せよ.

表2

LL 構文解析 (LL-parsing), LR 構文解析 (LR-parsing), 移動還元構文解析 (shift-reduce parsing), 演算子順位構文解析 (operator precedence parsing), 再帰下降構文解析 (recursive descent parsing), 字句 (token), 正規表現 (regular expression), 静的リンク (static link) [アクセスリンク (access link) とも言う], 退避情報 (saved information), データフロー解析 (data-flow analysis), バックパッチング (backpatching), 戻り番地 (return address), 有限オートマトン (finite automaton)