

修士課程 社会情報学専攻入学者選抜試験問題
(情報学基礎)

Entrance Examination for Master's Program
(Fundamentals of Informatics)

Department of Social Informatics

平成 24 年 8 月 6 日 10 : 00 ~ 12 : 00

August 6, 2012 10:00 - 12:00

【注意】

- ・ 試験開始の合図があるまで中を見てはいけない。
- ・ 問題用紙は表紙を含めて 7 枚である。試験開始後、枚数を確認し、落丁または印刷の不鮮明なものがあれば直ちに申し出ること。
- ・ 問題は 5 題である。このうち 3 題を選択し、解答しなさい。
- ・ 解答用紙の表紙に記載されている注意事項についても留意すること。
- ・ 問題 1 題につき、解答用紙 1 枚を使用すること。解答用紙は裏面を使用しても構わないが、使用する場合は裏面に継続することを明記すること。

NOTES

- ・ Do not open the pages before a call of starting.
- ・ This is the Question Booklet in 7 pages including this front cover.
After the call of starting, check all pages are in order and notify proctors immediately if missing pages or unclear printings are found.
- ・ There are 5 questions. Choose and answer 3 questions in total.
- ・ Read carefully the notes on the front cover of the Answer Sheets, too.
- ・ Use one sheet for each question. If necessary, the reverse side may be used, stating "Over" at the end of the page.

【情報学基礎】問題番号： 1

(1) ゴルフをプレイするか、プレイしないかを、以下のように推薦する決定木を示せ。

- ・曇りの時は、プレイする。
- ・晴れで、かつ、湿度が高い時はプレイしない。
- ・晴れで、かつ、湿度が高くない時はプレイする。
- ・雨で、かつ、風が吹いている時はプレイしない。
- ・雨で、かつ、風が吹いていない時はプレイする。

Design a decision tree that recommends whether it is good to play golf or not as follows.

- If it is overcast, we should play golf.
- If it is sunny and humid, we should not play golf.
- If it is sunny but not humid, we should play golf.
- If it is rainy and windy, we should not play golf.
- If it is rainy but not windy, we should play golf.

(2) 以下のプログラム GolfAdvice では、(1)の決定木を実現しようとしている。このプログラムを完成させるため、空欄 (2) にいくつかの Pascal の文を書け。

The following program GolfAdvice implements the decision tree in (1).

Fill in the blank (2) with Pascal statements to complete the program.

```
program GolfAdvice;
var
    ans1, ans2: string;
begin
    writeln('Is it overcast, sunny, or rainy today?');
    readln(ans1);
    if ans1 = 'overcast' then
        begin
            writeln('Play golf.');
```

```

        writeln('Do not play golf.');
```

end

```

    else
        begin
            writeln('Play golf.');
```

end;

```

    end;
    if ans1 = 'rainy' then
        begin
            (2)
```

end;

```

    end.
```

- (3) 以下のプログラム ReadData では、キーボードでタイプされた入力を読み取り、コンピュータのメモリにデータを記憶し、そしてそのデータを画面に表示している。以下の全ての語を用いながら、プログラム ReadData の全ての Pascal の文について詳しく説明せよ。

語：コンピュータのメモリ、データ、メモリ内の記憶場所、変数

The following program ReadData collects typed input data from the keyboard, stores the data in the computer memory, and prints the data on the screen. Explain the details of all the Pascal statements in the program ReadData. Note that all of the following terms should be used.

Terms: computer memory, data, locations in memory, variable

```

program ReadData;
var
    position1: string;
begin
    readln(position1);
    writeln(position1);
end.
```

- (4) コンピュータがなかったところに比べて、コンピュータが可能にした仕事の拡大について、複数の具体例をあげて説明せよ。

Explain the work amplification that computers make possible by comparing with the time period when computers were not available. Give more than one concrete examples.

【情報学基礎】問題番号：Ⅱ

- (1) コンピュータで実数を記憶する際の表現方法について説明せよ。また、その表現方法のメリットについて説明せよ。さらに、デメリットについて、具体例をあげて説明せよ。
Explain how to represent a real number to store in a computer and the advantages of the representation. Also, explain disadvantages giving concrete examples.

- (2) コンピュータを利用して数値計算を行うことのメリットについて説明せよ。
Explain advantages of numerical computation using computers.

- (3) 関数とは、入力を受け取り、出力を返す機能である。関数を表すには、自然言語、数式、コンピュータプログラム、表、グラフという5種類の方法がある。ある入力に対し、対応する出力を求める方法が説明できるものであれば、それは関数と定義することができる。例えば、円の半径が与えられたときに、その円の面積を求める関数 F について、日本語による記述で表現すると以下ようになる。

F が半径の数値を受け取ると、その数値を二乗したものに円周率 π を乗じた数値を返す。

数式、コンピュータプログラム、表、グラフの4つの方法で、どのようにこの関数 F を表すか述べよ。コンピュータプログラムで表す際には、 π を 3.14159 と近似して扱っても良い。プログラミング言語は既存の任意の言語を用いること。

A function receives inputs and yields, for each input, a uniquely defined output. There are five methodologies for describing functions: natural language, mathematical notation, computer programs, tables, and graphs. Any technique that tells how to find the appropriate output for each given input can define a function. For example, the following English description represents the function F that computes the area of a circle from its radius.

If F receives a radius, it will yield the mathematical constant π times the square of the radius.

Show how to express the function F with the other four representations, i.e., mathematical notation, computer programs, tables, and graphs. In computer programs, the mathematical constant π can be approximated by 3.14159. You can use any existing programming language.

【情報学基礎】問題番号：Ⅲ

以下の生成規則 R1, R2, R3 と R4 を用いて, 次の文

$Z := X + Y$

を P88 マシンのアセンブリ言語に翻訳せよ. P88 マシンの命令セットは表 1 に示す.

Translate the following statement into assembly language of P88 Machine by using the production rules R1, R2, R3 and R4 shown below. The instructions of P88 machine are shown in Table 1.

$Z := X + Y$

構文 Syntax	意味 Semantics
R1: $\langle i \rangle_j \rightarrow w$	$M(\langle i \rangle_j) = w$
R2: $\langle e \rangle_i \rightarrow \langle i \rangle_j$	$M(\langle e \rangle_i) = M(\langle i \rangle_j)$ $\text{code}(\langle e \rangle_i) = \text{nothing}$
R3: $\langle s \rangle_k \rightarrow \langle i \rangle_j := \langle e \rangle_i$	$\text{code}(\langle s \rangle_k) = \text{code}(\langle e \rangle_i)$ COPY AX, M($\langle e \rangle_i$) COPY M($\langle i \rangle_j$), AX
R4: $\langle e \rangle_i \rightarrow (\langle e \rangle_j + \langle e \rangle_k)$	$M(\langle e \rangle_i) = \text{createname}$ $\text{code}(\langle e \rangle_i) = \text{code}(\langle e \rangle_j)$ code($\langle e \rangle_k$) COPY AX, M($\langle e \rangle_j$) ADD AX, M($\langle e \rangle_k$) COPY M($\langle e \rangle_i$), AX

ただし, $\langle i \rangle_j$, $\langle e \rangle_i$, $\langle e \rangle_j$, $\langle e \rangle_k$, $\langle s \rangle_k$ は文法変数である.

w は識別子の文字列である. M は意味関数である.

where, $\langle i \rangle_j$, $\langle e \rangle_i$, $\langle e \rangle_j$, $\langle e \rangle_k$, and $\langle s \rangle_k$ are grammar variables.

w is some identifier string. The notation M stands for the concept of "meaning".

表 1 命令セット

TABLE 1: Instruction Set

命令 Instruction	フォーマット Format	アクション Action
Copy from mem	COPY AX, mem	AX:=mem
Copy to mem	COPY mem, AX	mem:=AX
Add	ADD AX, mem	AX:=AX+mem
Subtract	SUB AX, mem	AX:=AX-mem
Multiply	MUL AX, mem	AX:=AX*mem
Divide	DIV AX, mem	AX:=AX div mem
Compare	CMP AX, mem	If AX < mem then CF:=B else CF:=NB
Jump	JMP lab1	Go to the instruction with label lab1
Jump if not below	JNB lab1	Go to the instruction with label lab1 if CF=NB. Otherwise go to next instruction
Jump if below	JB lab1	Go to the instruction with label lab1 if CF=B. Otherwise go to next instruction
Input	IN AX	Input an integer into register AX
Output	OUT AX	Output an integer from register AX

【情報学基礎】問題番号：Ⅳ

コンピュータの計算速度を極限まで高め、計算困難な問題を一瞬で計算可能となっても、それはあらゆる計算問題の解決を意味しない。その理由を述べよ。

Explain why unlimited computation speed, which lets us execute intractable computations, is not enough for solving any possible problems.

【情報学基礎】問題番号：Ⅴ

以下の事柄から4つを選び、説明せよ。

- ・ 意味ネットワーク
- ・ 巡回セールスマン問題
- ・ アルゴリズム
- ・ コードパターン
- ・ プログラムのライフサイクル
- ・ フォン・ノイマン型アーキテクチャ

Choose four terms from the following list and explain them.

- Semantic network
- Traveling salesperson problem
- Algorithm
- Code patterns
- Program lifecycle
- Von Neumann architecture