

平成24（2012）年度  
東京大学大学院学際情報学府学際情報学専攻  
修士課程（総合分析情報学コース）  
入学試験問題  
専門科目

（平成24年1月16日 14：00～16：00）

試験開始の合図があるまで問題冊子を開いてはいけません。開始の合図があるまで、下記の注意事項をよく読んでください。

(Please read the instructions on the backside.)

1. 本冊子は、総合分析情報学コースの受験者のためのものである。
2. 本冊子の本文は18ページである。落丁、乱丁、印刷不鮮明の箇所などがあった場合には申し出ること。
3. 本冊子には、計8問の問題が収録されている。この8問の中から4問を選択して解答すること。
4. 本冊子の問題には、日本語文と英語文があるが、日本語文が正式なもので、英語文はあくまでも参考である。両者に意味の違いがある場合は、日本語文を優先すること。
5. 解答用紙は4枚ある。選択した問題ごとに解答用紙1枚を使用すること。このほかにメモ用紙が1枚ある。なお、解答用紙のみが採点の対象となる。
6. 解答用紙の上方の欄に、選択した問題の番号及び受験番号を必ず記入すること。問題番号及び受験番号を記入していない答案は無効である。
7. 解答には必ず黒色鉛筆（または黒色シャープペンシル）を使用すること。
8. 解答は原則として日本語によるものとする。ただし、英語で解答しても採点の対象とする。
9. 試験開始後は、中途退場を認めない。
10. 本冊子、解答用紙、メモ用紙は持ち帰ってはならない。
11. 次の欄に受験番号と氏名を記入せよ。

受験番号	
氏名	

## 総合分析情報学 第1問 (Question A1)

以下の設問に答えよ。

- (1) 次の極限值を求めよ。

$$\lim_{x \rightarrow 0} \frac{\tan x - \sin x}{3x^3}$$

- (2) 以下の微分方程式を考える。

$$y' - 2xy - 3x^2e^{x^2} = 0$$

- (a) 上式が線形であることを示せ。  
(b) 上式の一般解を求めよ。  
(c) 初期条件  $y(0) = 3$  として、上式を解け。

- (3) 次の式で表される図形の面積を求めよ。

$$x^{2/3} + y^{2/3} = r^{2/3} \quad (r > 0)$$

## Question A1

Answer the following questions.

- (1) Find the following limit.

$$\lim_{x \rightarrow 0} \frac{\tan x - \sin x}{3x^3}$$

- (2) Consider the following differential equation.

$$y' - 2xy - 3x^2e^{x^2} = 0$$

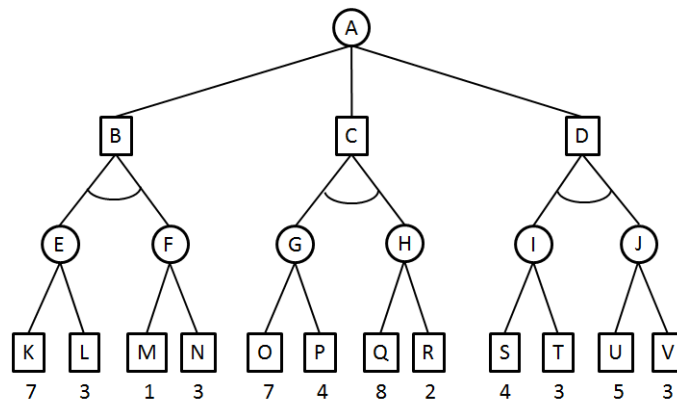
- (a) Determine whether the differential equation is linear.  
(b) Find the general solutions of the differential equation.  
(c) Solve the differential equation with the initial condition  $y(0) = 3$ .
- (3) Calculate the area of the figure described by the following equation.

$$x^{2/3} + y^{2/3} = r^{2/3} \quad (r > 0)$$

## 総合分析情報学 第2問 (Question A2)

AND/OR木を用いたゲーム木と、その上での検索アルゴリズムに関して、以下の4つの間に答えなさい。

- (1) いくつかのマッチ棒を3つの山に分け、先手と後手の各プレイヤーは、残っている山の1つから、1本または2本のマッチ棒を交互に取り除くとする。手を進めていき、最後の1本を取った方が敗者となるゲームを考える。いま、各山に配置されているマッチ棒の数が  $n_1, n_2, n_3$  であることを  $(n_1, n_2, n_3)$  と表すとし、山は区別する必要がないため、 $n_1 \geq n_2 \geq n_3$  として表示することにする。このとき、このゲームにおけるゲーム木を示しなさい。但し、初期配置を  $(2, 2, 2)$  とする。また、ゲーム木を作る際に、同じ組合せが出てきたら、どちらか一方のみを書くものとする。
- (2) 下図のゲーム木は、ある先手番の局面から3手先の局面の評価値を示している。○の局面は先手番、□の局面は後手番である。このとき、ミニマックス法によりすべての局面の評価値を決定し、ゲーム木上に示しなさい。また、ミニマックス法で決定される局面のたどるパスを求めなさい。

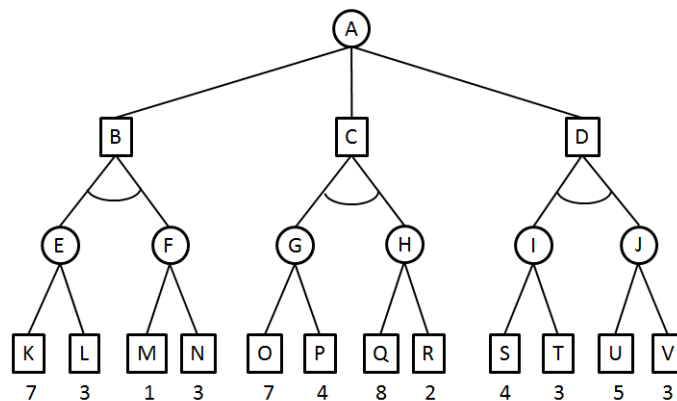


- (3) 上図のゲーム木をアルファベータ枝刈りで探索した場合、枝切りされる局面をゲーム木上に示しなさい。結果だけではなく、その結果が得られた理由も説明しなさい。なお、縦型探索を用いるものとする。
- (4) アルファベータ枝刈りの効率は、木のノード（節点）の評価値がどのような順番で並んでいるかによって決まる。では、アルファベータ枝刈りの効率がもっとも良くなるのは、一般的にどのような場合か。またその場合に、ミニマックス法に比べてどの程度探索が効率的になるか。木の深さを  $d$ 、分岐を  $b$  として、探索されるノードの個数を計算した上で説明しなさい。

## Question A2

Answer the following four questions about a game tree using an and-or tree and the search algorithms on the game tree.

- (1) Suppose that some matchsticks are divided into three heaps and that two players take turns removing one or two matchsticks from one of the three heaps. The player who removes the last matchstick is supposed a loser in this game. The fact that the numbers of matchsticks in the three heaps are  $n_1, n_2, n_3$  is denoted by  $(n_1, n_2, n_3)$ . The three heaps are not needed to be differentiated, so we can assume  $n_1 \geq n_2 \geq n_3$ . Then show the game tree in this game, where the initial numbers of matchsticks in the three heaps are assumed  $(2, 2, 2)$ . If the identical combinations in the number of matchsticks appear when making the game tree, you can show only one of them.
- (2) The below diagram shows the first three levels, or plies, in a game tree; a circle node with an alphabet inside denotes the first player's turn while a square node with an alphabet does the second player's turn. Then evaluate values of all the nodes in the game tree using the simple minimax algorithm and show the values on the tree. And show the minimax path determined by the algorithm on the tree.



- (3) Show the subtrees that need not be explored when the alpha-beta pruning algorithm is applied to the above game tree. You need explain not only the result of pruning but also the reason why the result is obtained. Here you can use depth-first search.
- (4) The efficiency of the alpha-beta pruning algorithm depends on in what order the nodes of a tree are searched. In what case can the algorithm search a tree most efficiently? And in this case, how much efficiently can the alpha-beta pruning algorithm search a tree compared with the simple minimax algorithm, assuming that the tree has an (average or constant) branching factor of  $b$  and a search depth of  $d$  plies?

## 総合分析情報学 第3問 (Question A3)

(1) 次のCプログラムに関し以下の問いに答えよ。

```
void f(int a[], int n) {
    int tmp;
    if (n <= 1) return;
    tmp = a[0];
    a[0] = a[n-1];
    a[n-1] = tmp;
    f(a+1, n-2);
}
```

(a) 次を実行した後の配列 a の内容を記せ。

```
int a[] = {1, 2, 3, 4, 5};
f(a, 5);
```

(b) f と同じ機能を持つ、再帰呼び出しを含まないプログラムを記せ。

(2) 条件文に関する次のBNF文法について以下の問いに答えよ。

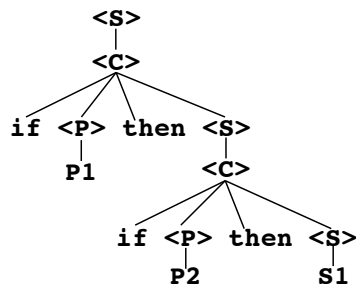
```
<S> ::= <C> | <other statement>
<C> ::= if <P> then <S> | if <P> then <S> else <S>
```

ここで <S>, <C>, <P> は非終端記号であり <P> は条件式である。

(a) この文法は曖昧である。その理由を、以下に対する全ての解析木を図示することで説明せよ。

if P1 then if P2 then S1 else S2      (\*)

例えば、“if P1 then if P2 then S1” に対応する解析木は以下のようになる：



(b) 曖昧性を除去するようにこの文法を書き換えよ。

(c) 前問の文法を使った場合の (\*) に対する解析木を図示せよ。

## Question A3

(1) Consider the following program in C.

```
void f(int a[], int n) {
    int tmp;
    if (n <= 1) return;
    tmp = a[0];
    a[0] = a[n-1];
    a[n-1] = tmp;
    f(a+1, n-2);
}
```

(a) Write the content of array **a** after executing the following code:

```
int a[] = {1, 2, 3, 4, 5};
f(a, 5);
```

(b) Write an equivalent program of function **f** without using recursive calls.

(2) Consider the following BNF grammar for conditional statements.

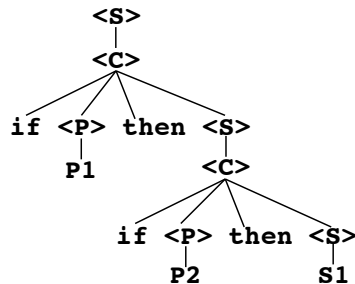
```
<S> ::= <C> | <other statement>
<C> ::= if <P> then <S> | if <P> then <S> else <S>
```

where **<S>**, **<C>**, **<P>** are nonterminal symbols and **<P>** is a predicate.

(a) This grammar is ambiguous. Explain why by showing all the parse trees for:

if P1 then if P2 then S1 else S2      (\*)

For example, “if P1 then if P2 then S1” would yield the following parse tree:



(b) Correct the above grammar so that it is not ambiguous.

(c) Draw a parse tree for (\*) by using the corrected grammar in (b).

## 総合分析情報学 第4問 (Question A4)

(1) 計算機アーキテクチャに関する以下の問に答えよ。

- (a) C 言語で作成したプログラムをコンパイルして実行したところ、segmentation fault というエラーメッセージを出して実行が中断した。このエラーの考えられる原因を具体的に説明せよ。
- (b) スタックとはどのようなデータ構造か説明せよ。
- (c) 一般的な計算機アーキテクチャではハードウェアでスタック構造をサポートしているものが多いが、その機構を説明せよ。
- (d) 前問で説明したスタック構造の代表的な使い方を説明せよ。

(2) 割込に関する以下の問に答えよ。

- (a) 割込とはどのような処理か説明せよ。
- (b) 割込が発生する要因を分類して説明せよ。
- (c) ある処理を実行中に割込が発生してから、対応するサービスルーチンを実行し、元の処理に復帰するまでの一般的な処理の流れを説明せよ。
- (d) 割込ベクタ (Interrupt Vector) の機能と実現方式を説明せよ。



## Question A4

- (1) Answer the following questions on computer architecture.
  - (a) Suppose that you developed a program in C language, compiled it, and executed it. The program stopped running displaying an error message “Segmentation fault”. Explain the possible causes of the execution error.
  - (b) Explain stack data structure.
  - (c) Many computer architectures provide a mechanism supporting the stack data structure by hardware. Explain its mechanism.
  - (d) Explain an example to use the architecture-supported stack data structure explained in the previous question.
  
- (2) Answer the following questions on interrupts.
  - (a) Explain the mechanism of interrupts.
  - (b) Classify the causes to start the interrupts, and explain them.
  - (c) Explain the process flow of interrupt processing in which an interrupt occurs, launches associated routines, and returns to the original execution process.
  - (d) Explain the function and implementation methods of interrupt vectors.

## 総合分析情報学 第5問 (Question A5)

共有メモリ型マルチプロセッサシステムにおいて、生産者プロセスが消費者プロセスに、共有するバッファを介してデータ（文字）を受け渡す有限バッファ問題について考える。生産者プロセスが生成したデータは、バッファに格納され、それを消費者プロセスが取り出す。バッファは、長さNの環状バッファで、バッファにデータを入れる位置を示すポインタ in、バッファからデータを取り出す位置を示すポインタ out をもつ。以下のプログラムは、この有限バッファ問題を記述したものであるが、同期をとるためのコードが抜けているため、正しく動作しない。生産者プロセスは関数 producer、消費者プロセスは関数 consumer をそれぞれ1回ずつ実行するものとする。

```
#define N 16

char buffer[N];
int count = 0;
int in = 0;
int out = 0;

void producer()
{
    char c;

    do {
        while (count == N);
        文字 c を生成する;
        count = count + 1;
        buffer[in] = c;
        in = (in + 1) % N;
    } while (1);
}

void consumer()
{
    char c;

    do {
        while (count == 0);
        c = buffer[out];
        out = (out + 1) % N;
        count = count - 1;
        文字 c を処理する;
    } while (1);
}
```

```
} while (1);  
}
```

以下の問いに答えよ。

- (1) このプログラムが正しく動作しない例を示せ。
- (2) 計数セマフォ `s` は、カウンタ `counter` と待ち行列 `queue` をもつ構造体として定義される。計数セマフォに対する P 命令 (wait 命令) は、例えば、以下のように定義することができる。

```
P(semaphore s) {  
    if (s.counter > 0) s.counter = s.counter - 1;  
    else 呼出しプロセスを s.queue の最後尾に置く;  
}
```

- (a) この P 命令のコードは、それ自身、相互排除しなければならない。その理由を述べよ。また、こうした相互排除の実現を支援するマシン命令の例を挙げ、その仕様を述べよ。
  - (b) この P 命令に対応する V 命令 (signal 命令) を定義せよ。
- (3) (2) で定義した P 命令と V 命令を用いて、2つのプロセスが正しく動作するように、上記のプログラムを書き換えよ。上記のプログラムで、バッファ内の文字数のカウントは、変数 `count` の代わりに、セマフォ内部のカウンタを使って実現せよ。
- (4) 生産者プロセスが複数存在する場合、生産者プロセス間で相互排除が必要となる。その場合でも正しく動作するよう、(2) で定義した P 命令と V 命令を用いて、(3) のプログラムをさらに書き換えよ。

## Question A5

Consider the bounded buffer problem in which a producer process passes data to a consumer process via a common fixed-size buffer. The producer process deposits a data item (character) into the buffer while the consumer process retrieves the data item from the buffer. The buffer is a circular buffer of size  $N$  with two pointers `in` and `out` to indicate the next available position for depositing data and the position that contains the next data to be retrieved. The following program describes this bounded buffer problem but does not run correctly due to the lack of the synchronization code. The producer process executes a function `producer()` once and the consumer process executes a function `consumer()` once.

```
#define N 16

char buffer[N];
int count = 0;
int in = 0;
int out = 0;

void producer()
{
    char c;

    do {
        while (count == N);
        Generate a character c;
        count = count + 1;
        buffer[in] = c;
        in = (in + 1) % N;
    } while (1);
}

void consumer()
{
    char c;

    do {
        while (count == 0);
        c = buffer[out];
        out = (out + 1) % N;
        count = count - 1;
    } while (1);
}
```

```
    Process a character c;  
} while (1);  
}
```

Answer the following problems.

- (1) Find a counterexample which demonstrates that this program is incorrect.
- (2) A counting semaphore  $s$  is defined as a structure which consists of a counter (counter) and a queue (queue). The synchronization primitive  $P$  (wait) of the semaphore can be defined as follows.

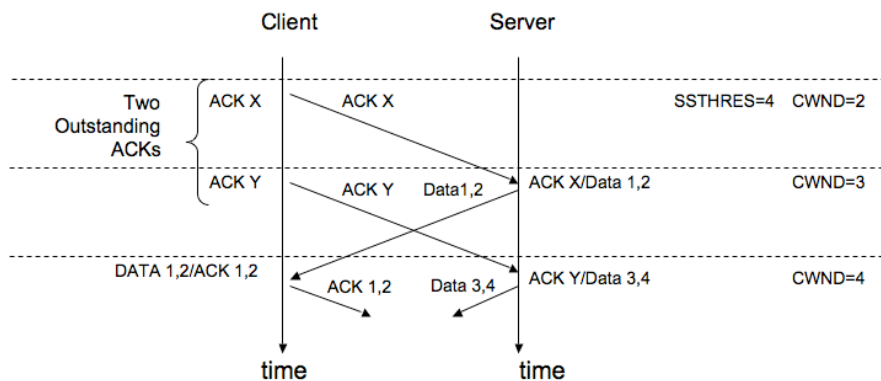
```
P(semaphore s) {  
    if (s.counter > 0) s.counter = s.counter - 1;  
    else the calling process is placed at the end of s.queue;  
}
```

- (a) Explain the reason why the above code of  $P$  primitive must be mutually exclusive. Give an example of machine instruction that supports this mutual exclusion, and explain its specification.
  - (b) Define  $V$  (signal) primitive that corresponds to the above  $P$  primitive.
- (3) Rewrite the above program using the  $P$  and  $V$  primitives defined in problem (2) so that two processes run correctly. The number of characters in the buffer should be counted by using the internal counter of the semaphore in place of the variable `count` in the above program.
- (4) Consider the case that there are multiple producer processes. In this case some mutual exclusion is needed among producer processes. When we implement this mutual exclusion with semaphores, rewrite the program of (3) using the  $P$  and  $V$  primitives defined in problem (2).

## 総合分析情報学 第6問 (Question A6)

インターネットにおける通信について以下の問いに答えよ。

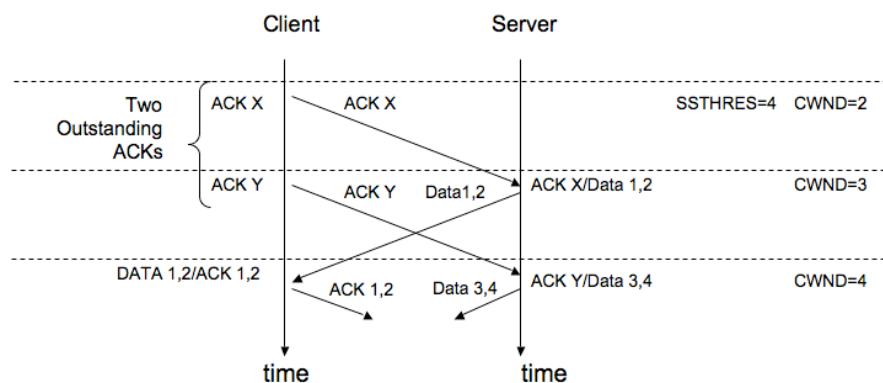
- (1) OSI 参照モデルとは何か簡潔に説明せよ。
- (2) OSI 参照モデルにおける第 2,3,4,7 層の各層に対応するインターネットで用いられるプロトコルについて簡潔に説明せよ。
- (3) OSI 参照モデルにおける第 4 層に対応する、あるプロトコル P についての以下の設問について答えよ。
  - (a) P がウィンドウベースのフロー制御と輻輳制御を行うことがわかっている場合、考えられる P の候補を一つあげよ。
  - (b) (a) の P が行うフロー制御と輻輳制御を簡潔に説明せよ。
  - (c) (a) の P を用いてクライアントとサーバーが通信を行う場合を考える。輻輳ウィンドウサイズを CWND、その閾値を SSTHRESH とする。簡単のために、パケット長は MSS バイトとし、CWND と SSTHRESH は任意のバイト数ではなく固定長の MSS の倍数であるとせよ。ただし、MSS は最大セグメント長とする。回答するにあたって必要な前提を仮定する場合は、それらを明記せよ。
    - (i) CWND はどのように変化するか簡潔に述べよ。
    - (ii) 現在、下図のように、ACK を受けていないパケットが 2 つあり、CWND は 2 MSS、SSTHRESH は 4 MSS であるとする。図では MSS は省略して記述してある。今後、最初の 4 パケットは正常に受信され ACK も完了するが、5 パケット目がパケットロスし、その後の 5 パケットは正常に受信され ACK も完了すると仮定する。これらの 10 パケットの通信の様子を図に描き、適切な説明を加えよ。



## Question A6

Answer the following questions regarding the Internet communication.

- (1) Concisely explain the OSI reference model.
- (2) In the OSI reference model, what are the protocols used in the Internet corresponding to the 2nd, 3rd, 4th and 7th layers, respectively? Concisely explain them.
- (3) Answer the following questions regarding a protocol P corresponding to the 4th layer of the OSI reference model.
  - (a) Suppose P implements window-based flow and congestion controls. Give one candidate protocol of P.
  - (b) Explain the flow and congestion controls performed by the protocol P in (a).
  - (c) Suppose a client and a server communicate with each other via the protocol P in (a). Let CWND be congestion window size and SSTHRESH be its threshold. For the simplicity's sake, every packet is MSS bytes long, and CWND and SSTHRESH are not of arbitrary length but of a multiple of MSS. Note that MSS is Maximum Segment Size. Explicitly write down what assumptions are made in the answers.
    - (i) Briefly explain how CWND changes over time.
    - (ii) Suppose there exist two outstanding ACKs as shown in the following figure. CWND is 2 MSS and SSTHRESH is 4 MSS. Note that MSS is omitted in the figure. From now on, suppose the first 4 packets are successfully received and their ACKs are also received, but the 5th packet gets lost. Also suppose the successive 5 packets and their ACKs are successfully received. Draw a diagram to depict how these 10 packets are transmitted between the client and the server with appropriate explanations in the diagram.



## 総合分析情報学 第7問 (Question A7)

- (1) 論理順序回路に関する以下の問に答えよ。
- (a) 4ビット整数の2入力・1出力の全加算器を設計し、真偽値表と論理回路図を示せ。
  - (b) キャリルックアヘッド回路を用いて、8ビット整数の2入力・1出力の全加算器を設計し、論理回路図を示せ。
- (2) 次の論理関数  $F(x, y, z, w)$  を簡約化し、簡約化された関数の not-and-or 形式で表記と、その簡約化の過程を示せ。
- (a)  $F(0, 0, 0, 1) = F(0, 1, 1, 1) = F(1, 0, 0, 1) = F(1, 0, 1, 0) = F(1, 0, 1, 1) = F(1, 1, 1, 1) = 1$ 、その他の場合は0
  - (b)  $F(0, 0, 0, 0) = F(0, 0, 0, 1) = F(0, 0, 1, 1) = F(0, 1, 0, 0) = F(0, 1, 0, 1) = F(0, 1, 1, 1) = F(1, 1, 1, 1) = F(1, 1, 1, 0) = F(1, 0, 0, 1) = F(1, 0, 1, 1) = F(1, 0, 1, 0) = 1$ 、その他の場合は0

not-and-or 形式とは、 $xyz + x'zw + \dots$  といった論理関数で表され、論理回路を使うと NOT 回路、AND 回路、OR 回路の順番で並べて実現できる形式をいう。



## Question A7

- (1) Answer the following questions on logic circuits.
  - (a) Design a full adder with two inputs and one output of 4bit integers, and show its true-false table and a figure of the logic gates structure.
  - (b) Design a full adder with two inputs and one output of 8 bit integers with a carry look ahead circuit, and show a figure of the logic gates structure.
- (2) Simplify the following boolean functions  $F(x, y, z, w)$  and show the simplified form in the NOT-AND-OR form and the simplifying process.
  - (a)  $F(0, 0, 0, 1) = F(0, 1, 1, 1) = F(1, 0, 0, 1) = F(1, 0, 1, 0) = F(1, 0, 1, 1) = F(1, 1, 1, 1) = 1$  and otherwise  $F(x, y, z, w) = 0$
  - (b)  $F(0, 0, 0, 0) = F(0, 0, 0, 1) = F(0, 0, 1, 1) = F(0, 1, 0, 0) = F(0, 1, 0, 1) = F(0, 1, 1, 1) = F(1, 1, 1, 1) = F(1, 1, 1, 0) = F(1, 0, 0, 1) = F(1, 0, 1, 1) = F(1, 0, 1, 0) = 1$ , and otherwise  $F(x, y, z, w) = 0$

Note that NOT-AND-OR form is represented in the form of boolean function such as  $xyz + x'zw + \dots$ , and is implemented by a logic circuit which is organized by NOT gates-AND gates-OR gates in this order.

## 総合分析情報学 第8問 (Question A8)

空間事象を理解しようとするとき、事象間の空間的近接性は重要な概念である。一般に、近接性の指標として直線距離（ユークリッド距離）が広く用いられているが、扱う空間事象やそのデータ表現によっては、直線距離では空間関係を適切に評価できない場合がある。そのような例を2つ挙げ、それぞれについて、(i) その事象の理解に直線距離が適切でない理由を述べ、(ii) 適切と思われる近接性指標を提案・説明せよ。但し、提案する指標はそれぞれ異なるものとする。こと。（つまり、各例につき1つずつ、計2つの異なる指標を提案すること。）

## Question A8

In order to understand spatial phenomena, spatial proximity is an important concept. While the straight-line distance (i.e. Euclidean distance) is broadly used as an index of spatial proximity, spatial relationships in some type of spatial phenomena and/or their data representation may not appropriately be evaluated by the straight-line distance. Give two examples of situations where the straight-line distance is an inappropriate index of spatial proximity. Then, for each situation, (i) explain why it is inappropriate and (ii) propose and describe an appropriate index. Note that indices you propose should be different for the two examples. (That is, you need to propose two different indices, one for each.)

**Entrance Examination for Masters Program  
in Applied Computer Science Course,  
Graduate School of Interdisciplinary Information Studies,  
The University of Tokyo.  
Academic Year 2012  
(14:00-16:00, January 16th, 2012)**

**Directions:** Do not open this booklet before the examination begins.  
Read the following instructions carefully.

1. This booklet is for the examinees in Applied Computer Science Course, Graduate School of Interdisciplinary Information Studies.
2. This booklet includes eighteen pages. Report missing, misplaced, and imperfect pages to the instructor.
3. This booklet includes eight questions. Select any four questions and answer only those four.
4. Each question is described both in Japanese and in English. Use the Japanese version primarily; the English version is provided for the reference purpose only.
5. There are four answer sheets and a scratch paper. Use one answer sheet per question. A scratch paper is provided for calculation. Only the answer sheets will be considered valid.
6. Write a question number and your examinee's number in the designated boxes located at the top of each answer sheet. The answer missing a question number and/or an examinee's number will not be considered valid.
7. Use only black pencils (or black mechanical pencils).
8. Answer the questions in Japanese as a general rule, although you are also allowed to answer in English.
9. Do not leave the room until the examination is finished.
10. Do not take away this booklet, the answer sheets, and the scratch paper.
11. Write your examinee's number and your name in the designated boxes below.

Examinee's Number	
Name	