

平成 27 年 8 月 1 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 28 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 2 題 (問題 1~2), 選択問題 5 題 (問題 3~7), 合計 7 題である。  
必須問題は 2 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 16 枚である。
- 解答用紙は全部で 4 枚ある。  
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) の解答を  
3 枚目 (白色) の解答用紙には問題 3~7 (選択問題) から選択した 1 題の解答を  
4 枚目 (白色) の解答用紙には問題 3~7 (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。
- 解答用紙は 4 枚すべてを回収するので, すべての解答用紙に受験番号を記入すること。
- 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) を記入すること。  
また, 選択問題調査票には, 選択した問題の番号 (3~7 から二つ) 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。
- 解答用紙には, 日本語または英語で解答すること。

配点：(1) 30点, (2-1) 10点, (2-2) 15点, (2-3) 20点, (2-4) 20点, (3) 30点

図に示す ANSI-C 準拠である C 言語のプログラム (program) は, 配列 (array) values の要素 (element) に格納された非負整数 (non-negative integer) のデータ (data) を,  $r$  進数 ( $r$  base number) ( $2 \leq r \leq 256$ ) とみなして整列 (sort) し, 出力 (output) するプログラムである. 以下の各問に答えよ.

- (1) プログラムの 10~26 行目の for 文の処理において 1 巡目 ( $d=1$  のとき) および 2 巡目 ( $d=2$  のとき) の終了時における配列 buckets の内容を, 解答用紙の太線内の空欄を埋めることにより答えよ. なお, 配列の各要素は 0 で初期化される. 配列の各要素 buckets[ $x$ ][ $y$ ]において,  $x$  は 3 以上の値,  $y$  は 5 以上の値を取りうるが, これらの場合の内容は答えなくてよい.

| buckets[ $x$ ][ $y$ ] | 1 巡目 ( $d=1$ ) |       |       |       |       | 2 巡目 ( $d=2$ ) |       |       |       |       |
|-----------------------|----------------|-------|-------|-------|-------|----------------|-------|-------|-------|-------|
|                       | $y=0$          | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=0$          | $y=1$ | $y=2$ | $y=3$ | $y=4$ |
| $x=0$                 |                |       |       |       |       |                |       |       |       |       |
| $x=1$                 |                |       |       |       |       |                |       |       |       |       |
| $x=2$                 |                |       |       |       |       |                |       |       |       |       |

- (2) プログラムで実現されている整列アルゴリズム (sorting algorithm) に関する以下の各小問に答えよ.
- (2-1) この整列アルゴリズムは, 一般に何と呼ばれているか名称を答えよ.
- (2-2) 整列対象のすべてのデータを  $r$  進数で表現したときの最大の桁数 (maximum digit number) を  $k$  とする. 整列対象のデータの数を  $n$  としたとき, プログラムで実現されている整列アルゴリズムの時間計算量 (time complexity) のオーダー表記 (order notation) を, その理由とともに答えよ.
- (2-3) この整列アルゴリズムは, 整列対象のデータ同士の比較を伴わない整列アルゴリズムである. この整列アルゴリズムでは, どのような工夫によって整列対象のデータ同士の比較を伴わない整列を可能にしているかを答えよ.
- (2-4) 整列対象のデータ同士の比較を伴う整列アルゴリズムと比較して, プログラムで実現されている整列アルゴリズムの特徴を, 時間計算量および空間計算量 (space complexity) の観点から答えよ.
- (3) 配列 values の要素に格納されたデータを降順 (descending order) に整列するように, プログラムを変更することを考える. そのためには, 下線 (ア) および下線 (イ) をどのように記述すればよいか, 適切な式をそれぞれ答えよ.

```

1 #include <stdio.h>
2 #define MAXR 256
3 #define MAXN 1000
4 void sort(int values[], int numvalues, int r, int maxdigit){
5     int rd, d, b, i, n, j;
6     int buckets[MAXR][MAXN];
7     int numbucket[MAXR];
8
9     rd = 1;
10    for(d = 1; d <= maxdigit; d++){
11        for(b = 0; b < r; b++){
12            numbucket[b] = 0;
13        }
14        for(i = 0; i < numvalues; i++){
15            n = (values[i] / rd) % r;
16            buckets[n][numbucket[n]++] = values[i];
17        }
18
19        i = 0;
20        for( b = 0; b < r; b++ ){
21            for( j = 0; j < numbucket[b]; j++ ){
22                values[i++] = buckets[b][j];
23            }
24        }
25        rd *= r;
26    }
27 }
28 int main(){
29     int i;
30     int values[5] = {12, 21, 1, 11, 2};
31
32     sort(values, 5, 10, 2);
33     for(i = 0; i < 5; i++){
34         printf("%d\n", values[i]);
35     }
36     return 0;
37 }

```

図 プログラム

配点： (1-1) 20 点, (1-2-1)~(1-2-3) 各 5 点, (1-2-4) 10 点, (1-3-1) 10 点, (1-3-2) 10 点  
(2-1-1) 10 点, (2-1-2) 15 点, (2-2-1) 10 点, (2-2-2) 10 点, (2-3) 15 点

(1) 計算機 (computer), 特に, 中央処理装置 (CPU: central processing unit) に関する以下の各小問に答えよ。解答は全て解答用紙の太線内に書くこと。

(1-1) 以下の文章の空欄 (a)~(e) に当てはまる最も適切な語句を, 下記の選択肢から選び, 記号で答えよ。

計算機の構成方式のことを (a) と呼ぶ。現在, 実用的に利用されている大部分の計算機は, 線形アドレス空間 (linear address space) を有するメインメモリ (main memory) 上にプログラム (program) 及びデータ (data) を置き, プログラムを逐次実行する, 等の (a) を採用している。このような計算機は, (b) 計算機と呼ばれる。

(b) 計算機では, (c) が示すメインメモリアドレスから, プログラムの構成要素である機械語命令を読み出し, これをデコード (decode) して実行する。機械語命令は, その種類を示す (d) 部と, 演算対象のデータである (e) の格納場所を示すアドレス部から成る。

【選択肢】

- |                                       |                                       |
|---------------------------------------|---------------------------------------|
| (ア) データフロー型 (dataflow architecture)   | (イ) オペコード (operation code, opcode)    |
| (ウ) プログラムカウンタ (program counter)       | (エ) アーキテクチャ (architecture)            |
| (オ) VLIW (very long instruction word) | (カ) オペランド (operand)                   |
| (キ) 命令レジスタ (instruction register)     | (ク) メモリデータレジスタ (memory data register) |
| (ケ) スーパー标カラー (superscalar)            | (コ) ノイマン型 (von Neumann architecture)  |

(1-2) 一つの機械語命令 (以下, 命令と略す) の処理を  $n$  段のステージ (stage) に分け, 1 ステージを 1 クロックサイクル (clock cycle) で実行する同期型 CPU を考える。クロック周波数を  $f$  [Hz] とする。以下の (1-2-1)~(1-2-4) に答えよ。

(1-2-1)  $m$  個の命令をパイプライン (pipeline) 処理を用いずに逐次的に実行する場合の実行時間 [s] を示せ。

(1-2-2)  $m$  個の命令をパイプライン処理を用いて実行する場合の実行時間 [s] を示せ。但し, パイプラインストール (pipeline stall) は無いものとする。

(1-2-3)  $m$  個の命令をパイプライン処理を用いて実行する場合の単位時間あたりの命令実行数 [instructions/s] について,  $m \rightarrow \infty$  における極限を示せ。但し, パイプラインストールは無いものとする。

(1-2-4) パイプライン処理を用いた CPU の性能 (単位時間あたりの命令実行数) を高める手法の一つとして, ステージ数  $n$  を増やす方法がある。この方法により CPU の性能を高めることが可能である理由を説明せよ。

(1-3) 一つの機械語命令 (以下, 命令と略す) の処理を, 命令フェッチ (IF: instruction fetch), デコード (D: decode), オペランドフェッチ (OF: operand fetch), 実行 (EX: execution), 結果の格納 (S: store) の 5 つのステージに分け, 1 ステージ 1 クロックサイクルのパイプライン処理を用いた同期型 CPU において, 以下の (1-3-1) および (1-3-2) に示す命令 1~3 を実行することを考える。解答欄の命令 1 の例を参考に, 実行されるステージ名 “IF”, “D”, “OF”, “EX”, “S” を解答欄に記入せよ。パイプラインストールにより完了までに複数クロックサイクルが必要なステージは, 完了するクロックサイクルにステージ名を, それ以外のクロックサイクルに “→” を記入すること。なお, 以下の点を仮定する。

- パイプラインストールの原因としては, 構造ハザード (structural hazard) およびデータハザード (data hazard) を考える。
- 構造ハザードはメモリアクセス (memory access) の競合 (conflict) のみ考える。
- プログラムとデータは同じメインメモリ上にある。
- ある命令でレジスタに書き込まれた値を以降の命令で参照する場合, 前者の命令の格納 (S) が完了した後, 後者の命令のオペランドフェッチ (OF) が可能となる。

(次ページに続く)

- (1-3-1) 命令1: MOV R1, (A) ; メインメモリアドレス A の内容をレジスタ R1 に転送  
 命令2: MOV R2, (B) ; メインメモリアドレス B の内容をレジスタ R2 に転送  
 命令3: ADD R1, R2 ; R1 + R2 の結果を R1 に代入
- (1-3-2) 命令1: MOV R1, (A) ; メインメモリアドレス A の内容をレジスタ R1 に転送  
 命令2: INC R1 ; R1 + 1 の結果を R1 に代入  
 命令3: MOV (B), R1 ; レジスタ R1 の内容をメインメモリアドレス B に転送

|         |                  |    |   |    |    |   |   |   |   |   |   |    |    |
|---------|------------------|----|---|----|----|---|---|---|---|---|---|----|----|
| (1-3-1) | クロックサイクル         | 0  | 1 | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|         | 命令1: MOV R1, (A) | IF | D | OF | EX | S |   |   |   |   |   |    |    |
|         | 命令2: MOV R2, (B) |    |   |    |    |   |   |   |   |   |   |    |    |
|         | 命令3: ADD R1, R2  |    |   |    |    |   |   |   |   |   |   |    |    |
| (1-3-2) | クロックサイクル         | 0  | 1 | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|         | 命令1: MOV R1, (A) | IF | D | OF | EX | S |   |   |   |   |   |    |    |
|         | 命令2: INC R1      |    |   |    |    |   |   |   |   |   |   |    |    |
|         | 命令3: MOV (B), R1 |    |   |    |    |   |   |   |   |   |   |    |    |

(2) キャッシュメモリ (cache memory) 及びメインメモリ (main memory) で階層を形成しているメモリシステムを持つ計算機を考える。以下の各小問に答えよ。解答は全て解答用紙の太線内に書くこと。

(2-1) キャッシュメモリに存在する命令あるいはデータを読み出す際のアクセス時間 (access time) が 2 [ns] であり、キャッシュメモリに存在しない命令あるいはデータをメインメモリから読み出す際の、キャッシュメモリ及びメインメモリへのアクセス時間の和が 50 [ns] であるとする。また、メインメモリの容量はプログラムに対して十分大きく、プログラムの命令及びデータは全てメインメモリに格納されているものとする。以下の (2-1-1) 及び (2-1-2) に答えよ。

(2-1-1) プログラムを実行した結果、プログラムの命令あるいはデータを読み出す際の平均のキャッシュヒット率 (cache hit ratio, cache hit rate) が 80% であった。この時の、プログラムの命令あるいはデータを読み出す際の平均アクセス時間を求めよ。導出根拠も示せ。

(2-1-2) キャッシュメモリを、命令あるいはデータを読み出す際のアクセス時間が 2 [ns] のものから 4 [ns] のものに変更する。ただし、キャッシュメモリに存在しない命令あるいはデータをメインメモリから読み出す際の、キャッシュメモリ及びメインメモリへのアクセス時間の和は 50 [ns] のままであるとする。この時、命令あるいはデータを読み出す際の平均アクセス時間の増加を防止するためには、キャッシュメモリの容量を大きくするなどの方法により、キャッシュヒット率を高める必要がある。(2-1-1) で得られた平均アクセス時間を維持するために必要となるキャッシュヒット率を求めよ。導出根拠も示せ。

(2-2) 一般に、プログラムを実行するためにアクセスされる命令及びデータには、参照局所性 (locality of reference) がある。次の 2 種類の参照局所性のそれぞれについて説明せよ。

(2-2-1) 空間的参照局所性 (spatial locality of reference)

(2-2-2) 時間的参照局所性 (temporal locality of reference)

(2-3) キャッシュメモリとメインメモリで階層を形成することの利点をその理由と共に述べよ。

配点：(1-1)~(1-5) 各 6 点, (1-6) 7 点, (1-7) 7 点, (1-8-1) 6 点, (1-8-2) 15 点  
 (2-1) 12 点, (2-2) 15 点, (2-3) 20 点, (2-4) 13 点

(1) 情報論理 (mathematical logic) に関する以下の各小問に答えよ。ただし、論理式 (logic formula) の記述には以下の記号を用いる。→, ∧, ∨, ¬ はそれぞれ含意 (implication), 論理積 (conjunction, and), 論理和 (disjunction, or), 否定 (negation, not) を表す論理演算子とする。また、必要に応じて  $\bigwedge_{1 \leq i \leq n} T_i$  (あるいは  $\bigvee_{1 \leq i \leq n} T_i$ ) の表記を用いる。これは論理式  $T_i$  の  $i$  を 1 から  $n$  ( $n$  は正整数) まで順次変えながら論理積 (あるいは論理和) で結合した式を意味している。なお、 $\bigwedge_{1 \leq i < j \leq n} T_{ij}$  は、 $\bigwedge_{1 \leq i \leq n-1} (\bigwedge_{i+1 \leq j \leq n} T_{ij})$  を意味する。

縦  $n^2$  マス×横  $n^2$  マスで構成されるパズル (puzzle) を考える。本パズルにおける規則は以下の通りである。

- 空いているマスに 1 から  $n^2$  までの整数を入れる。
- 各列、各行および、太線で囲まれた  $n \times n$  の各ブロック内に同じ整数を複数用いてはいけない。

図 1 は、 $n = 3$  における本パズルの問題例である。以降、このパズル問題を解くための制約式を和積形 (CNF: Conjunctive Normal Form) の命題論理式 (propositional formula) で与えることを考える。なお、和積形とは、一つ以上の和項の論理積で表される論理式である。また、和項とは、一つ以上のリテラル (literal) の論理和で表される論理式である。ここでリテラルとは、命題変数 (propositional variable) または命題変数の否定を意味する。 $i$  ( $1 \leq i \leq n^2$ ) 行  $j$  ( $1 \leq j \leq n^2$ ) 列のマスを  $c(i, j)$  で表記する (行は上から順に  $1, \dots, n^2$  行目, 列は左から順に  $1, \dots, n^2$  列目である)。

マス  $c(i, j)$  に整数  $k$  ( $1 \leq k \leq n^2$ ) が入っているとき、かつ、そのときのみ真となる命題変数  $x_{ijk}$  を導入する。以下の各小問に答えよ。

|   |   | 列 |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
|   |   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 7 | 3 | 2 |   | 1 |   |   |   |   |   |
| 2 | 4 |   |   |   |   |   |   | 2 |   |   |
| 3 |   | 5 | 1 |   |   |   | 3 | 8 |   |   |
| 4 |   |   |   |   |   |   |   |   | 1 |   |
| 5 |   | 8 |   |   |   |   | 2 |   |   |   |
| 6 | 9 |   |   |   |   |   |   |   | 7 |   |
| 7 |   |   |   |   |   |   |   |   |   |   |
| 8 |   | 2 |   | 6 |   |   |   | 3 |   |   |
| 9 |   |   |   |   | 7 |   |   |   | 9 |   |

図 1:  $n = 3$  の問題例

- (1-1) 図 1 において命題変数  $x_{115}, x_{214}, x_{841}$  の真偽をそれぞれ答えよ。
- (1-2) 縦  $n^2$  マス×横  $n^2$  マスで構成される本パズル問題における命題変数  $x_{ijk}$  の総数はいくつか。  $n$  を用いて示せ。
- (1-3) 「1 行 1 列目のマスには、1 以上 9 以下の整数が少なくとも一つ入る」ことを表す論理式を記述せよ。

- (1-4) CNF 式  $A(i, j)$  を「 $i$  行  $j$  列目のマスには、1 以上  $n^2$  以下の整数が少なくとも一つ入る」ことを表す論理式とする。論理式  $A(i, j)$  を記述せよ。
- (1-5)  $A(i, j)$  を用いて、「どのマスについても、1 以上  $n^2$  以下の整数が少なくとも一つ入る」ことを表す論理式  $A$  を示せ。
- (1-6)  $\neg p \vee \neg q$  は、「 $p, q$  がともに真になるということはない」ことを表す論理式である。これを用いて「どの行についても各整数は高々1回しか現れない」ことを表す CNF 式  $B$  を作りたい。以下の空欄を埋めることで  $B$  を完成させよ。

$$B = \bigwedge_{1 \leq i \leq n^2} \bigwedge_{1 \leq j < l \leq n^2} \bigwedge_{1 \leq k \leq n^2} (\quad)$$

- (1-7) 「どの列についても各整数が高々1回しか現れない」ことを表す CNF 式  $C$  を示せ。
- (1-8) この小問では、 $n = 2$  として、図 2 の問題に着目する。

|   |   |   |   |
|---|---|---|---|
|   |   | 1 | 2 |
| 1 | 2 | 4 |   |
|   | 1 | 2 | 4 |
|   |   |   | 3 |

図 2:  $n = 2$  の問題

図 2 の問題では、空いたマス（例えば 2 行 4 列目）にどのように整数を埋めても、パズルを解くことができない。ここでは、(命題 P) 「図 2 の問題では、空いたマスにどのように整数を埋めても、パズルを解くことができない」ことを示したい。この命題 P を示すには、「太線で囲まれた  $n \times n$  のどのブロックについても各整数が高々1回現れる」ことを意味する CNF 式を  $D$ 、図 2 における整数の配置を表す論理式を  $Assign$  としたときに、(方針)  $A \wedge B \wedge C \wedge D \wedge Assign$  の CNF 式が充足不能 (unsatisfiable) であることを示せばよい。このとき、以下の (1-8-1)~(1-8-2) に答えよ。

- (1-8-1) 論理式  $Assign$  を示せ。
- (1-8-2) 上記の方針に沿って、命題 P が成り立つことを導出原理 (resolution principle) を用いて具体的に示せ。

- (2) 空でない有限集合 (finite set)  $V$  上の二項関係 (binary relation)  $E$  について、次のように二項関係  $R_i$  と  $S_i$  ( $i = 0, 1, 2, \dots$ ) を定義する。また、 $V$  の各要素を頂点とし、 $E$  の各要素を有向辺とする有向グラフ (directed graph) を  $G$  とする。ただし  $(u, v) \in E$  に対しては、 $u$  を有向辺の始点とし、 $v$  を有向辺の終点とする。

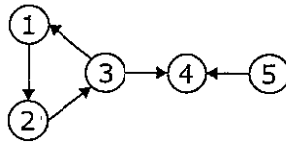
$$R_0 = S_0 = \{(v, v) \mid v \in V\}$$

$$R_{i+1} = \{(u, v) \mid (\exists w \in V) [uR_i w \wedge wEv]\} \quad (i \geq 0)$$

$$S_{i+1} = S_i \cup R_{i+1} \quad (i \geq 0)$$

以下の各小問に答えよ。

- (2-1) 有向グラフ  $G = (V, E)$  が下図の時、 $R_3$  を求めよ。

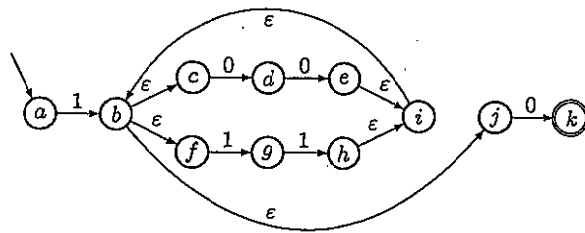


- (2-2) ある非負の整数  $n$  が存在して  $S_n = S_{n+1}$  であることを、背理法 (proof by contradiction) を用いて証明せよ。
- (2-3) 小問 (2-2) の  $n$  について、 $S_n$  の逆関係 (inverse relation) を  $S_n^{-1} = \{(u, v) \mid vS_n u\}$  とし、 $S = S_n \cap S_n^{-1}$  とする。 $S$  が同値関係 (equivalence relation) であることを証明せよ。ただし、任意の非負整数  $i, j$  について、 $((u, v) \in R_i) \wedge ((v, w) \in R_j)$  ならば  $(u, w) \in R_{i+j}$  が成立することは、証明なしで用いてよい。
- (2-4) 小問 (2-3) の  $S$  について、 $v \in V$  の  $S$  による同値類 (equivalence class) を  $[v]_S$  とする。 $[v]_S$  が、グラフ  $G$  について何を表すか簡潔に述べよ。



配点: (1-1) 5 点, (1-2) 5 点, (1-3) 5 点, (1-4) 25 点, (1-5) 20 点, (2-1) 25 点, (2-2) 20 点, (2-3) 20 点

- (1) 有限オートマトン (finite automaton)  $M$  は 5 項組  $M = (Q, \Sigma, \delta, q_0, F)$  で与えられる。ここで,  $Q, \Sigma, \delta, q_0, F$  は, それぞれ, 状態 (state) の有限集合, 入力記号 (input symbol) の有限集合 (アルファベット (alphabet)), 状態遷移関数 (state transition function), 初期状態 (initial state) ( $q_0 \in Q$ ), 受理状態 (accepting state) の集合 ( $F \subseteq Q$ ) である。また,  $M$  が受理 (accept) する言語 (language) (認識する言語) を  $L(M)$  と表す。下の状態遷移図 (state transition diagram) に示す非決定性 (non-deterministic) 有限オートマトン  $M_1$  について, 以下の各小問に答えよ。なお,  $Q = \{a, b, c, d, e, f, g, h, i, j, k\}$ ,  $\Sigma = \{0, 1\}$ ,  $q_0 = a$ ,  $F = \{k\}$  である。



- (1-1)  $M_1$  が受理する言語 (language) を正規表現 (正規表現, regular expression) で示せ。
- (1-2)  $M_1$  が受理する語 (word) を, 1 文字目を最上位ビット (most significant bit) とする符号なし 2 進数 (unsigned binary number) とみなす。  $M_1$  が受理するすべての語の中で, 7 番目に小さな数となる語を示せ。
- (1-3)  $M_1$  の状態  $i$  の  $\epsilon$ -閉包 ( $\epsilon$ -closure) を示せ。
- (1-4)  $L(M_1) = L(M_2)$  を満たす, 決定性 (deterministic) 有限オートマトン  $M_2$  をサブセット構成 (subset construction) 法を用いて求め, 状態遷移図で示せ。同値 (equivalent) な状態があっても全て残すこと。状態名は  $A, B, C, \dots$  とすること。導出過程と結果を示すこと。
- (1-5)  $M_2$  の状態数を最小化した決定性有限オートマトン  $M_3$  を求め, 状態遷移図で示せ。状態名は  $A', B', C', \dots$  とすること。

(次ページに続く)

(2) 文脈自由文法 (context-free grammar) により生成される文脈自由言語 (context-free language) の閉包性 (closure property) を考える。文脈自由文法  $G$  は 4 項組  $G = (V, T, P, S)$  で表される。ただし,  $V$  は変数 (variable, あるいは非終端記号 non-terminal symbol) の集合,  $T$  は終端記号 (terminal symbol) の集合,  $P$  は生成規則 (production rule) の集合,  $S$  は開始記号 (start symbol) で  $S \in V$  である。なお, 生成規則の集合は  $\epsilon$ -規則 ( $\epsilon$ -rule) を含んでよいものとし, 変数の集合と終端記号の集合は共通の記号を含まないものと仮定する。文法  $G$  が生成する言語を  $L(G)$  と表記する。

任意の文脈自由文法  $G_1 = (V_1, T_1, P_1, S_1)$  と  $G_2 = (V_2, T_2, P_2, S_2)$  が与えられる (ただし  $V_1 \cap V_2 = \emptyset$  である)。

文脈自由言語は和 (union) 演算に関して閉じている。なぜなら, 文脈自由文法  $G_3$  を以下の通りに構成すると, 証明は省略するが,  $L(G_1)$  と  $L(G_2)$  の和の言語  $L_3 = L(G_1) \cup L(G_2)$  は  $G_3$  によって生成される。すなわち  $L_3$  は文脈自由言語だからである。

- $G_3 = (V_3, T_3, P_3, S_3)$
- $V_3 = V_1 \cup V_2 \cup \{S_3\}$ , ただし  $S_3 \notin (V_1 \cup V_2)$
- $T_3 = T_1 \cup T_2$
- $P_3 = \{S_3 \rightarrow S_1, S_3 \rightarrow S_2\} \cup P_1 \cup P_2$

文脈自由言語は接続 (concatenation) 演算に関して閉じていることを, 文法の構成により示したい。以下の各小問に答えよ。

(2-1)  $L(G_1)$  と  $L(G_2)$  を接続した言語  $L_4 = \{w_1w_2 : w_1 \in L(G_1), w_2 \in L(G_2)\}$  を生成する文脈自由文法  $G_4 = (V_4, T_4, P_4, S_4)$ , すなわち  $L_4 = L(G_4)$  である  $G_4$  を構成せよ。

(2-2) 上記 (2-1) で構成した  $G_4$  によって生成される任意の語  $w$  は, 上記 (2-1) で定義した  $L_4$  に属することを証明せよ。

(2-3) 上記 (2-1) で定義した  $L_4$  に属する任意の語  $w$  は, 上記 (2-1) で構成した  $G_4$  により生成されることを証明せよ。

配点: (1)15点, (2-1)15点, (2-2)25点, (2-3)20点,  
(3-1)7点, (3-2)10点, (3-3)11点, (3-4)12点, (3-5)10点

データリンク層 (data link layer) では, 端末 (host) がフレーム (frame) を送信する際にプリアンブル (preamble) と呼ばれる特定のビット列を付与している. プリアンブルとフレームの構成は, 図1に示すものとする. 以下の各問に答えよ.

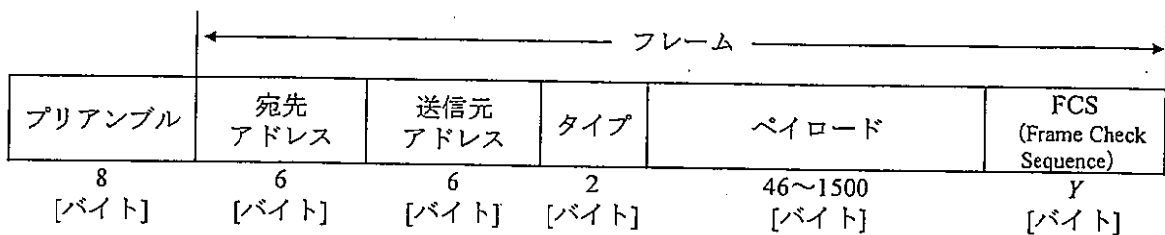


図1 プリアンブルとフレーム構成

- (1) 伝送レート (transmission rate)  $X$  [bps] の伝送媒体 (transmission medium) を用いた場合に, 上位層に対して提供できる最大の転送レート (transfer rate) を,  $X$  および図1中の  $Y$  を用いて示せ. なお, インターフレームギャップ (inter-frame gap) は考えなくて良い.
- (2) プリアンブルに関する以下の各小問に答えよ.
  - (2-1) プリアンブルをフレームに付与する目的を述べよ.
  - (2-2) プリアンブルのビット列として最も適切なものを, 以下の四つの選択肢から一つ選び, 記号を答えよ. また, 選んだビット列が, 最も適切である理由を説明せよ.
 

選択肢 A : 00000111 00000110 00000101 00000100 00000011 00000010 00000001 00000000

選択肢 B : 10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101011

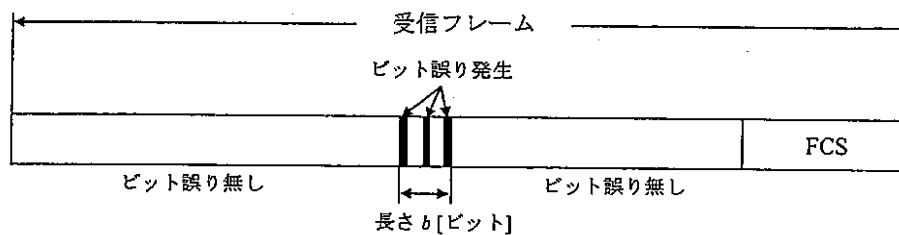
選択肢 C : 11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000

選択肢 D : 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
  - (2-3) プリアンブルと同一のビット列がペイロード (payload) に含まれる場合に生じる問題を述べよ. また, その問題を回避するためにイーサネット (Ethernet) で行われる方策を説明せよ.
- (3) 図1で与えられるフレームには, 伝送誤り (transmission error) 検出を行うための FCS (frame check sequence) が含まれている. 生成多項式  $G(x)$  を用いた巡回冗長検査 (cyclic redundancy check) によりフレームの伝送誤りが検出される. すなわち, 生成多項式  $G(x)$  で定義される擬巡回符号 (pseudo-cyclic code) の符号語の冗長記号部分が FCS となり, 誤り検出が行われる. 以下の各小問に答えよ. ただし, 生成多項式  $G(x)$  は次の多項式とする.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

表1  $G(x)$  を生成多項式とする巡回符号の次数最小の符号語を表す多項式  $c(x)$ 

| ハミング重み | $c(x)$  |
|--------|---|
| 1      | なし  |
| 2      | なし  |
| 3      | $1+x^{41678}+x^{91639}$                         |
| 4      | $1+x^{2215}+x^{2866}+x^{3006}$                  |
| 5      | $1+x^{89}+x^{117}+x^{155}+x^{300}$              |
| 6      | $1+x^{79}+x^{85}+x^{123}+x^{186}+x^{203}$       |
| 7      | $1+x^{45}+x^{53}+x^{74}+x^{80}+x^{120}+x^{123}$ |

図2 長さ  $b$  [ビット] のバースト誤りの例

- (3-1) FCSのサイズ  $Y$  [バイト] の値を答えよ。
- (3-2) 受信フレーム (誤り検出における受信語) の多項式表現を  $R(x)$  とする。巡回冗長検査において誤り無しと判断する条件式を、 $R(x)$  と  $G(x)$  を用いて書け。
- (3-3) 生成多項式  $G(x)$  を用いて検出できないビット誤り (bit error) の最小個数  $r$  を考える。フレーム長が 1000 [バイト] である時の  $r$  を求めよ。求めた過程も簡単に書け。必要に応じて、 $G(x)$  を生成多項式とする巡回符号の次数最小の符号語を表す多項式 (符号多項式)  $c(x)$  をハミング重みごとにまとめた表1を用いてよい。
- (3-4) 図1のフレームの最大ペイロード長は 1500 [バイト] である。最大ペイロード長をその10倍の 15000 [バイト] に増やすことの利点と欠点を、巡回冗長検査の観点から簡潔に説明せよ。
- (3-5) 生成多項式  $G(x)$  を用いたバースト誤り (burst error) 検出に関する以下の文章中で、 $b$  にあてはまる数値と空欄  にあてはまる多項式を書け。

フレーム内で、図2に示すような1つのバースト誤りが発生したとする。その長さが  $b$  [ビット] 以下であれば、確実にそのバースト誤りを検出できる。しかし、長さが  $(b+1)$  [ビット] のときは、検出できない場合がある。例えば、送信フレーム (符号語) を表す多項式が0と等しく小問(3-2)の多項式  $R(x)$  が  と等しいときに、そのバースト誤りを検出できない。

配点: (1-1) 25点, (1-2) 15点, (1-3) 15点, (2-1) 15点, (2-2) 24点, (2-3) 16点, (3) 15点

- (1) 二つの  $n$  ビット符号無し 2 進整数 (unsigned binary integer)  $X = (x_{n-1}, \dots, x_0)$ ,  $Y = (y_{n-1}, \dots, y_0)$  が入力され,  $X > Y$  のとき出力  $out$  に 1 を出力し,  $X \leq Y$  のとき 0 を出力する比較器 (comparator) を設計する. 1 ビットの比較器を図 1 のように直列 (serial) に接続した構造を考える. 以下の各小問に答えよ.
- (1-1) 1 ビットの比較器は, 各桁において  $x_i$ ,  $y_i$  と下位桁の比較結果  $c_i$  を入力して, 比較結果  $c_{i+1}$  を出力する. この 1 ビット比較器の真値表 (truth table) を示せ.
- (1-2)  $x_i$ ,  $y_i$ ,  $c_i$  を変数とする  $c_{i+1}$  の最小積和形 (最簡積和形; minimal sum-of-products expression) を求めよ.
- (1-3) (1-1)(1-2) で考えた 1 ビット比較器を NAND ゲートと NOT ゲートのみを用いて実現せよ.

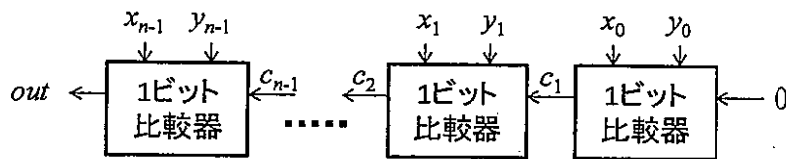


図 1

(2) エッジトリガ型 D フリップフロップ (edge-triggered D flip-flop) を 3 個用いた順序回路 (sequential circuit) について考える. 3 個の D フリップフロップのそれぞれの入力を  $D_2, D_1, D_0$ , 出力を  $Q_2, Q_1, Q_0$  とする. ここでは,  $(Q_2, Q_1, Q_0)$  が  $(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (0, 0, 0), \dots$  と遷移するカウンタ (counter) を設計する.  $(Q_2, Q_1, Q_0)$  の初期値 (initial value) は  $(0, 0, 0)$  とする. 以下の各小問に答えよ.

- (2-1) カウンタの状態遷移表 (state transition table) を作成せよ.  
 (2-2)  $D_2, D_1, D_0$  を  $Q_2, Q_1, Q_0$  の最小積和形で表せ.  
 (2-3) (2-1)(2-2)で設計したカウンタにおいて, 一時的な誤動作 (temporal error) により時刻  $T$  にフリップフロップの記憶値が  $(Q_2, Q_1, Q_0) = (1, 1, 0)$  となった. これ以降 3 クロックサイクルにおける  $(Q_2, Q_1, Q_0)$  の値の変化を示せ. 但し, 時刻  $T$  以外において順序回路は設計された通りに動作している. 表 1 にならって解答すること.

表 1

| 時刻    | $(Q_2, Q_1, Q_0)$ |
|-------|-------------------|
| $T$   | $(1, 1, 0)$       |
| $T+1$ |                   |
| $T+2$ |                   |
| $T+3$ |                   |

- (3) 図 2 の CMOS 回路の出力  $z$  を入力  $a, b, c, d$  を用いた論理式で表せ. 図中の  $V_{dd}$  は電源電位,  $Gnd$  はグラウンド電位に接続されている.

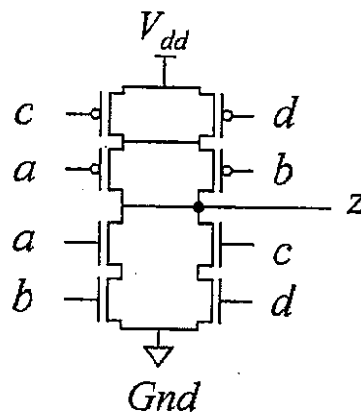


図 2

配点: (1-1)30点, (1-2)25点, (2-1)35点, (2-2)35点

以下の各問に答えよ。

(1) 以下の各小問に答えよ。

(1-1) ある関数  $f(t)$  のラプラス変換 (Laplace transform) を  $\mathcal{L}[f(t)] = F(s)$  とし、

$$g(t) = \begin{cases} f(t-a) & t \geq a \\ 0 & t < a \end{cases}$$

であるとき、 $\mathcal{L}[g(t)] = e^{-as}F(s)$  であることを証明せよ。ただし  $a \geq 0$  とする。

(1-2) 以下の式のラプラス変換を求めよ。

$$h(t) = \begin{cases} \cos(t - 2\pi/3) & t \geq 2\pi/3 \\ 0 & t < 2\pi/3 \end{cases}$$

(次ページへ続く)

(2) 図1に示す周期  $T$  である周期信号 (periodic signal)  $x(t)$  に関する以下の各小問に答えよ. ただし  $n$  は整数 (integer number) である. また  $0 < \tau < \frac{T}{2}$  である.

$$x(t) = \begin{cases} 0 & (n-1/2)T < t \leq nT - \tau/2 \\ V & nT - \tau/2 < t \leq nT + \tau/2 \\ 0 & nT + \tau/2 < t \leq (n+1/2)T \end{cases}$$

(2-1)  $x(t)$  の複素型フーリエ級数 (complex form of Fourier series) を求めよ.

(2-2)  $x(t)$  のパワー (power)  $P$  を,  $P \equiv \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt$  の定義に基づき計算すると,

$\frac{V^2 \tau}{T}$  となる. これは, (2-1) で求めた複素型フーリエ級数を用いることでも求められる. これを以下の二つの式を用いて証明せよ. ただし,  $c_k$  は  $x(t)$  を複素型フーリエ級数表現したときの複素型フーリエ係数 (complex Fourier coefficient) とし ( $k = 0, \pm 1, \pm 2, \dots$ ),  $0 \leq a \leq \pi$  とする.

$$\frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt = \sum_{k=-\infty}^{+\infty} |c_k|^2,$$

$$\sum_{k=1}^{+\infty} \frac{\sin^2(ka)}{k^2} = \frac{1}{2} a(\pi - a)$$

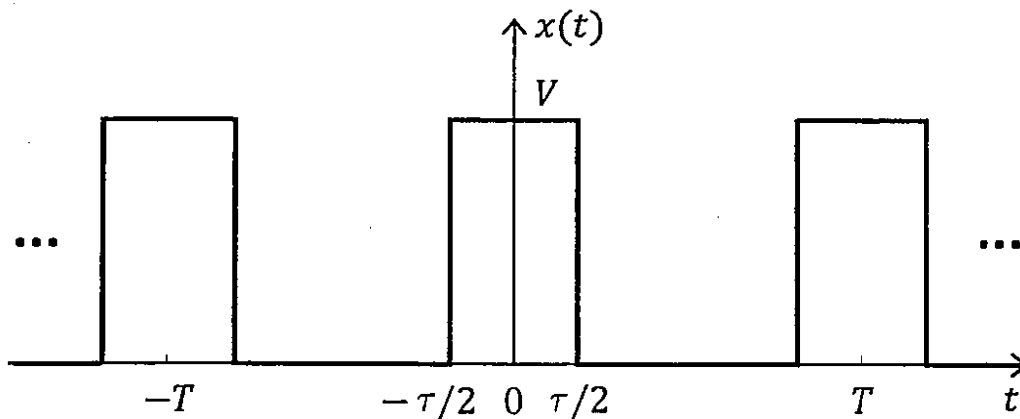


図1: 周期  $T$  の信号  $x(t)$