

平成15年度 京都大学大学院情報学研究科
修士課程 通信情報システム専攻入学資格試験問題

専門基礎B

II群問題

平成14年8月22日(木)9:00 - 12:00

注意

1. 解答開始の合図があるまで中を見てはいけない。
2. 「I群」および「II群」の2種類の問題が用意されている。いずれかの群の問題のみを解答すること。両群の問題を解答した場合、専門基礎Bの得点は0点とする。
3. これは「専門基礎B II群」の問題用紙で、表紙共に6枚ある。解答開始の合図があった後、枚数を確認、落丁または不鮮明なものがあれば直ちに申し出ること。
4. 問題は3つの分野から各2問 合計6問(BII-1a, BII-1b, BII-2a, BII-2b, BII-3a, BII-3b)ある。**BII-1, BII-2, BII-3,の各分野から1問以上合計4問を選択して解答すること。**答案用紙の問題番号欄に問題番号を記入すること。
5. 解答は問題ごとに答案用紙1枚を使うこと。答案用紙1枚に2問以上の解答もしくは1問の解答を2枚以上の答案用紙に書いた場合は無効にすることがある。なお、必要な場合「裏に続く」と明記した上で裏面を使用してもよい。
6. 答案用紙は4枚綴じたまま使用し、切り離さないこと。
7. 答案用紙の綴じ込みがはずれた場合は、直ちに申し出ること。
8. 解答は日本語で行うこと。

専門基礎B

BII-1, BII-2, BII-3 の3つの分野から各2問 合計6問 (BII-1a, BII-1b, BII-2a, BII-2b, BII-3a, BII-3b) がある。BII-1, BII-2, BII-3, の各分野から 1 問以上 合計 4 問を選択して解答せよ。

分野1(BII-1)

BII-1a

グラフのハミルトン閉路とは、各頂点を正確に1回通るような閉路のことである。以下の問に答えよ。

(1) 6頂点のグラフでハミルトン閉路の存在しないグラフの例をあげよ。但し、次数(頂点から出る枝の数)1の頂点を含んではいけない。更にできるだけ枝数の多いグラフを求めること(枝数は採点に影響する)。

(2) 与えられたグラフがハミルトン閉路を持つかどうかを判定する問題は NP 完全である(このことは証明なしで利用して良い)。さて、 n 頂点のグラフに対して、 $n-1$ 頂点を正確に1回通る閉路を準ハミルトン閉路と呼ぶことにする。与えられたグラフが準ハミルトン閉路を持つかどうかの判定問題は NP 完全であることを証明せよ。なお、変換の正しさを証明する時に、必要性和十分性を怠ること無くしっかりと述べること。

(3) (2) の判定問題は、与えられるグラフを「次数1の頂点を持たないグラフ」と制限しても NP 完全になることを示せ。(ヒント: 1個の頂点 v を2個の頂点 v_1, v_2 に分け、更に新しい頂点 x を導入して、枝 $(v_1, x), (x, v_2)$ を引く。この新しいグラフがハミルトン閉路を持つなら、閉路はこの新しい頂点 x を必ず通る。従ってこの部分に適当な細工を加えれば良い。)

なお、質問は一切受け付けない。問題に不審のある場合は、それを明記した上で、適切な仮定を設定して解答すること。

BII-1b

1 ビットの信号 X を入力とし、1 ビットの信号 Z を出力とする Mealy 型 同期式順序回路 (synchronous sequential circuit) を設計する。この回路を動作させたとき、 Z は最初の入力 X と等しい値を出力する。その後、出力の値と異なる値が 2 回続けて入力された時にのみ、 Z の値は変化する。これ以外の場合には、以前と同じ値を出力する。例えば、001011100 と入力されたとき、その出力は 000001110 となる (図 (a) 参照)。すなわち、最初の入力が 0 であるので、最初の出力は 0 となる。その後、1 が 2 回続けて入力されたことになる 6 クロック目に 1 を出力する。それまでは 0 を出力している。7 クロック目以降は、9 クロック目に 0 が 2 回続けて入力されるまで出力は 1 を出し続け、9 クロック目に 0 に反転する。以下の各問に答えよ。

- (1) 入力が 1011000101 の場合の出力を示した以下の表を完成させなさい。

入力 X	1	0	1	1	0	0	0	1	0	1
出力 Z	1									

- (2) この回路の動作を表す 状態遷移図 (state transition diagram) (図 (b)) を完成させなさい。ここで、状態 a は初期状態であり、入力が 0 であれば、0 を出力して状態 b に遷移し、入力が 1 であれば 1 を出力して状態 c に遷移する。枝のラベルは、入力 / 出力を表す。
- (3) 求めた状態遷移図に基づいて、入力が 011001011 の場合の出力が 001100001 となることを確かめよ。どのように確かめたかを説明すること。
- (4) この回路について、状態数 (the number of states) を最小化した 状態遷移表 (state transition table) と 出力表 (output table) を求めよ。
- (5) この回路を最も少ない数の D フリップフロップ (D flip-flop) を用いて実現する。状態割当て (state assignment) として、初期状態である状態 a には全て 0、状態 b には LSB のみが 1 で他は 0、状態 c には MSB のみが 1 で他は 0 を割当てよ。フリップフロップの出力を表す論理変数を Q 、入力を表す論理変数を D として、各フリップフロップは添字で区別することとする。添字は MSB から LSB に向かって 1, 2, とふるものとする。すなわち、MSB に対応するフリップフロップの出力は Q_1 であり、入力は D_1 である。
- 各フリップフロップの入力を与える 論理式 (logic equation) を 簡単化 (minimize) して求めよ。
- (6) この回路の出力 Z を与える論理式を簡単化して求めよ。

入力 X 0 0 1 0 1 1 1 0 0
 ↓ ↓ ↓
 出力 Z 0 0 0 0 0 1 1 1 0

図 (a) 入出力の例

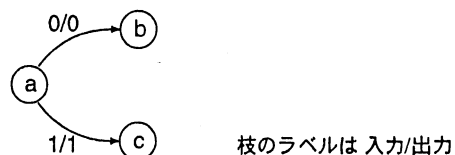


図 (b) 状態遷移図 (一部)

分野2(BII-2)

BII-2a

種々の分岐予測方式について述べよ。分岐予測テーブルの実際の実現方式はどのようなになっているか。また、20段の命令パイプラインが採用され、分岐予測成功時ペナルティ(性能低下)0、予測ミス時ペナルティ19サイクルのとき、分岐予測がCPIに及ぼす値を計算せよ。ただし、分岐予測ミス10%、分岐命令頻度15%とする。

BII-2b

キャッシュメモリについて下記の問に答えよ。

(1) 参照の局所性、セットアソシアティブ、ストアイン・ストアスルー、LRUなどの用語を説明しつつ、図を使って基本構成方式について説明せよ。

(2) 2階層キャッシュメモリの構成方式について述べ、キャッシュメモリへの平均アクセス時間を表す式を示せ。また、パラメータを適当に与えて、1階層キャッシュメモリとの性能比較を数値的に示せ。

(3) ノンブロッキングキャッシュとは何か。どのような効果が期待できるか。

分野3(BII-3)

BII-3a

コンパイラの最適化処理に関する以下の問に答えよ。

1. 基本ブロック (basic block) の概念を説明し、コンパイラの最適化処理において、基本ブロックがどのように利用されるかを示せ。
2. フローグラフ (flow graph) の概念を説明し、コンパイラの最適化処理において、フローグラフがどのように利用されるかを示せ。
3. 次の一連の文 (C 言語の表記を用いている) において、ラベル L1 ~ L11 は、この一連の文以外からは参照されていないものとする。この一連の文 (C 言語の表記を用いている) に対するフローグラフをかけ。基本ブロックを四角で表し、フローを矢印で表すこと。

```
L1:  j = 10;  
L2:  i = -8;  
L3:  i = i + 1;  
L4:  j = j - 1;  
L5:  b = (j != 0);  
L6:  if (b) goto L3;  
L7:  j = 1/2;  
L8:  b = (i < 8);  
L9:  if (b) goto L11;  
L10: i = 2;  
L11: goto L4;
```

4. 生死解析 (liveness analysis) の概念を説明し、コンパイラの最適化処理において、生死解析がどのように利用されるかを示せ。さらに、前問の一連の文に対し、変数 i と j について生死解析を行え。

BII-3b

集合 $\{a_1, \dots, a_n\}$ をリスト $(a_1 \dots a_n)$ で表現することになると、ある値 x がある集合 S の要素かどうか（つまり、 $x \in S$ かどうか）を判定する Lisp 関数 `member?` は、次のように定義できる。

```
> (define (member? x S)
  (if (null? S)
      #f
      (if (eqv? x (car S))
          #t
          (member? x (cdr S))))))
```

この例にならって、以下の各問の Lisp 関数を定義せよ。定義中には、この `member?` の定義で使われている関数の他に、次の関数を使用してよい。

`(cons x y)` x を car 部、 y を cdr 部とするセルを生成する

1. 値 x を集合 S の要素として追加した集合を求める Lisp 関数 `add` を定義せよ。 x を第 1 引数、 S を第 2 引数とし、 x がすでに S の要素であれば、 S そのものを返すものとする。定義中に、上記の関数 `member?` を使用してよい。`add` の使用例を次に示す。

```
> (add 2 '(1 5 3))
(2 1 5 3)
> (add 2 '(1 2 3))
(1 2 3)
```

2. 二つの集合 S_1 と S_2 の和集合 ($S_1 \cup S_2$) を求める Lisp 関数 `union` を定義せよ。 S_1 と S_2 を引数とし、 S_1 の要素で S_2 の要素でないものだけを S_2 に追加することによって、和集合を求めるものとする。定義中に、前問の関数 `add` を使用してよい。`union` の使用例を次に示す。

```
> (union '(1 2 3) '(0 2 4))
(3 1 0 2 4)
```

3. 二つの集合 S_1 と S_2 の積集合 ($S_1 \cap S_2$) を求める Lisp 関数 `product` を定義せよ。 S_1 と S_2 を引数とし、 S_1 の要素で S_2 の要素でもあるものを集めることによって、積集合を求めるものとする。定義中に、関数 `member?` を使用してよい。`product` の使用例を次に示す。

```
> (product '(1 2 3) '(0 2 4))
(2)
> (product '(1 2 3) '(0 4 8))
()
```