平成18年度
東京大学大学院情報理工学系研究科
コンピュータ科学専攻
入学試験問題

専門科目 I

平成17年8月 23日
10:00〜12:30

注意事項

1. 試験開始の合図があるまで, この問題冊子を開けないこと.
   Do not open this problem booklet until the start of the examination is announced.
2. 4問すべてに答えよ. 問いごとに指定された解答用紙を使用すること.
   Answer the following four problems. Use the designated answer sheet for each problem.
3. 解答用紙および問題冊子は持ち帰らないこと.
   Do not take the answer sheets and the problem booklet out of the examination room.

下欄に受験番号を記入すること.
Fill the following blank with your examinee's number.

受験番号	No.

問題 1(100 点).
一階論理に関して，以下の問いに答えよ．

(1) 「充足可能な論理式の否定は，充足不可能な論理式となる」は，正しいか？正しいと思う場合は，簡単な証明を与えよ．また，正しくないと思う場合は，反例を与えよ．

(2) 「推論系が完全である」,「推論系が健全である」とは，どういうことかをそれぞれ簡単に述べよ．

(3) 完全な推論系 α がある．推論系 α による，論理式集合 Γ から式 ψ の証明が存在しないとする．このとき，論理式集合 $\Gamma \cup \{\neg\psi\}$ は充足可能となることを示せ．

(4) 一階論理の完全で健全な推論系を一つ挙げて簡単に説明せよ．さらに，それを使って次のことを証明せよ．

(a) $(\exists x \forall y P(x, y)) \rightarrow (\forall x \exists y P(y, x))$

(b) $(\forall x \forall y (P(x, y) \rightarrow P(y, x))) \rightarrow (\neg\exists x \exists y \neg(\neg P(x, y) \vee P(y, x)))$

Problem 1(100 points).
Answer the following questions concerning the first-order logic.

(1) Is "the negation of a satisfiable formula is unsatisfiable" true? If you think that it is, then give a brief proof. If not, then give a counterexample.

(2) Explain briefly what "an inference system is complete" and "an inference system is sound" mean.

(3) Consider a complete inference system α. Suppose that there is no proof of a formula ψ from the set Γ of formulas by using the inference system α. Prove that the set $\Gamma \cup \{\neg\psi\}$ of formulas is satisfiable.

(4) Give a sound and complete inference system for the first-order logic and explain it briefly. Further, prove the following using it.

(a) $(\exists x \forall y P(x, y)) \rightarrow (\forall x \exists y P(y, x))$

(b) $(\forall x \forall y (P(x, y) \rightarrow P(y, x))) \rightarrow (\neg\exists x \exists y \neg(\neg P(x, y) \vee P(y, x)))$

問題 2(100 点).
1 次記憶上に入りきらないため 2 次記憶上に格納してある大きなデータがあったときに，それらを効率よく整列する手法としてマージソートがある．以下の問いに答えよ．アルゴリズムについては，まず概要を述べた後，擬似コードを記せ．

(1) k 本の整列された整数値の列が 2 次記憶上にあるとき，それらをあわせて 1 本の整列された整数値の列として 2 次記憶上に書き出すアルゴリズム (merge) を説明せよ．

(2) merge アルゴリズムを利用して，2 次記憶上にある整列されていない k 本の整数値の列を，1 本の整列された整数値の列として 2 次記憶上に書き出すアルゴリズム (mergesort) を説明せよ．

(3) mergesort アルゴリズムを用いて長さ n の整列されてない整数値の列 k 本を整列する場合に，2 次記憶にアクセスする回数の見積もりを与えよ．またその根拠を説明せよ．

(4) mergesort アルゴリズムを高速化する方法として，まず 1 次記憶上に収まる分量ずつ，クイックソートを用いて 1 次記憶上で整列し，それらを単位としてマージを開始する方法がある．この方法を適用したときの 2 次記憶へのアクセス回数の見積もりを与えよ．ただしクイックソートのために使える 1 次記憶の大きさを m とする．

Problem 2(100 points).
Mergesort is an algorithm suitable for sorting sequences that are larger than a primary memory and are stored in a secondary memory. Answer the following problems on mergesort algorithm. When describing an algorithm, give an overview at the beginning, and then give pseudo-code for it.

(1) Describe an algorithm (merge) for merging k sorted sequences of integers in the secondary memory into one sorted sequence of integers in the secondary memory.

(2) Describe an algorithm (mergesort) for sorting k unsorted sequences of integers in the secondary memory and generating one sorted sequence of integers in the secondary memory using the above algorithm (merge).

(3) Estimate the number of accesses to the secondary memory when sorting k unsorted sequences of integers with length n using the mergesort algorithm and explain it.

(4) One method to improve the efficiency of the mergesort algorithm is to sort each possible partial sequence fitting in the primary memory by quicksort and to use a sorted partial sequence as the unit for mergesort. Estimate the number of accesses to the secondary memory using this method. Let m be the size of the primary memory to be used for quicksort.

問題 3(100 点).
コンパイラの生成するコードの最適化について，以下の問いに答えよ．

(1) データフロー解析をせず，生成されたコードを局所的に走査しながら，無駄な操作を削減する方法を一つ説明せよ．

(2) 次の最適化について，それぞれを行うために必要なデータフロー解析と，そこから得られた情報に基づいてコード変換を行う手法を説明せよ．

  (a) 定数伝播

  (b) 共通部分式の削除

  (c) 無用命令の削除

Problem 3(100 points).
Answer the following questions concerning optimization of code generated by a compiler.

(1) Describe and explain a method for eliminating unnecessary operations by locally scanning the generated code without resorting to any data flow analysis.

(2) For each of the following optimization methods, describe and explain a data flow analysis necessary for this and a code transformation scheme based on the information obtained from the analysis.

  (a) constant propagation

  (b) common sub-expression elimination

  (c) dead code elimination

問題 4(100 点).
プロセスの排他制御問題を扱うために，mutex_lock，mutex_unlock 関数を以下の通り定義する．

- void mutex_lock(MID m)
  MID(Mutex 識別子) 型変数 m で示される Mutex オブジェクトをロックする．既に他のプロセスによってロックされている場合には，アンロックされるまで待たされる．実行が再開すると，再度 Mutex オブジェクトのロックを試みる．

- void mutex_unlock(MID m)
  Mutex 識別子 m で示される Mutex オブジェクトをアンロックする．他のプロセスがアンロックされるのを待たされている場合には，そのプロセスの実行を再開する．

以下の問いに答えよ．

(1) 以下のプログラムはデッドロックする可能性がある．3 つのプロセスがどのような実行順序で実行されると，デッドロックが生じるか示せ．

```
Process 1
1: mutex_lock(mtxA);
2: mutex_lock(mtxB);
   /* 資源 A と資源 B を
      排他利用する */
8: mutex_unlock(mtxB);
9: mutex_unlock(mtxA);
```

```
Process 2
1: mutex_lock(mtxB);
2: mutex_lock(mtxC);
   /* 資源 B と資源 C を
      排他利用する */
8: mutex_unlock(mtxC);
9: mutex_unlock(mtxB);
```

```
Process 3
1: mutex_lock(mtxC);
2: mutex_lock(mtxA);
   /* 資源 C と資源 A を
      排他利用する */
8: mutex_unlock(mtxA);
9: mutex_unlock(mtxC);
```

(2) プロセスがデッドロックしないように上記プログラムを書き直せ．

(3) あらかじめ決められた複数の資源を扱うプロセス群のデッドロックを防止するための手法を述べよ．

(4) 与えられたプロセスが，デッドロック状態にあることを検知するプログラムを以下の関数を用いて記述せよ．

- MID waitFor(PID p)
  p で示されるプロセスが mutex_lock 関数で待ち状態となっている場合には，その Mutex オブジェクトの持つ MID を返す．そうでなければ，0 を返す．有効な MID は 0 以外の値を持つ．

- PID used(MID m)
  m で示される Mutex オブジェクトがプロセスによってロックされている時，そのプロセスの PID(プロセス識別子) を返す．当該 Mutex がロックされていない場合には，0 を返す．有効な PID は 0 以外の値を持つ．

Problem 4(100 points).
Let us define functions `mutex_lock` and `mutex_unlock` to handle the mutual exclusion problem in processes as follows:

- `void mutex_lock(MID m)`
  It locks the Mutex object specified by `m` whose data type is `MID` (mutex object identifier). If the object has been locked, the process waits for the object to be unlocked. As soon as the object is unlocked, the process is resumed and tries to lock the object again.

- `void mutex_unlock(MID m)`
  It unlocks the Mutex object specified by the `m` variable. If some processes have waited for the object to be unlocked, those processes are resumed.

Answer the following questions.

(1) In the following program, deadlock may occur. Show the execution sequence of the three processes that leads to deadlock.

```
Process 1
1: mutex_lock(mtxA);
2: mutex_lock(mtxB);
  /* Resources A and B
  are exclusively used*/
8: mutex_unlock(mtxB);
9: mutex_unlock(mtxA);
```

```
Process 2
1: mutex_lock(mtxB);
2: mutex_lock(mtxC);
  /* Resources B and C
  are exclusively used*/
8: mutex_unlock(mtxC);
9: mutex_unlock(mtxB);
```

```
Process 3
1: mutex_lock(mtxC);
2: mutex_lock(mtxA);
  /* Resources C and A
  are exclusively used*/
8: mutex_unlock(mtxA);
9: mutex_unlock(mtxC);
```

(2) Rewrite the above program so as to prevent deadlock.

(3) Describe a method to prevent the deadlock of processes which share predefined resources.

(4) Write a program to detect whether a given process is in the deadlock state using the following functions:

- `MID waitFor(PID p)`
  It returns the MID of the `Mutex` object for which the process denoted by `p` is waiting at the `mutex_lock` function. It returns 0 if the process is not waiting for any `Mutex` object. Note that an effective `MID` does not have the zero value.

- `PID used(MID m)`
  It returns the PID (process identifier) of the process which has locked the `Mutex` object denoted by `m`. It returns 0 if no process has locked the Mutext object. Note that an effective `PID` does not have the zero value.

平成18年度
東京大学大学院情報理工学系研究科
コンピュータ科学専攻
入学試験問題

専門科目 II

平成17年8月 23日
14:00〜16:30

注意事項

1. 試験開始の合図があるまで, この問題冊子を開けないこと.
   Do not open this problem booklet until the start of the examination is announced.
2. 4問すべてに答えよ. 問いごとに指定された解答用紙を使用すること.
   Answer the following four problems. Use the designated answer sheet for each problem.
3. 解答用紙および問題冊子は持ち帰らないこと.
   Do not take the answer sheets and the problem booklet out of the examination room.

下欄に受験番号を記入すること.
Fill the following blank with your examinee's number.

受験番号	No.

問題 1(100 点).
負の枝長の枝は存在するが，負の経路長のサイクルは持たない強連結な有向グラフ $G = (V, E)$ を考える．ここで，$l(u, v)$ は枝 $(u, v) \in E$ の長さ，$d(v, w)$ は点 $v \in V$ から点 $w \in V$ への最短路長を表すものとする．以下の問いに答えよ．

(1) グラフ G において，任意の点 $v \in V$ 及び任意の枝 $(u, w) \in E$ に対し，$l(u, w) + d(v, u) - d(v, w) \geq 0$ が成り立つことを証明せよ．

(2) グラフ G において，V のすべての点 v に対して数値 $s(v)$ が与えられているとする．さらに，すべての枝 $(u, v) \in E$ についてその長さを $l(u, v) + s(u) - s(v)$ に変換したグラフ G' を考える．この時，G' 上において任意の 2 点 w, x 間の最短路であるパスは，G においても同じく w, x 間の最短路であることを証明せよ．

(3) グラフ G において，$v \in V$ を始点とする最短路木を求めるアルゴリズムを記述し，その計算量を述べよ．

(4) グラフ G において，$v \in V$ を始点とする最短路木が与えられている時に，別の点 $w \in V$ を始点とする最短路木を求めるアルゴリズムを記述し，その計算量を述べよ．

Problem 1(100 points).
Consider a strongly connected directed graph $G = (V, E)$, which has negative-length edges, but has no negative-length cycles. Let $l(u, v)$ denote the length of an edge $(u, v) \in E$, and let $d(v, w)$ denote the shortest path distance from vertex v to vertex w. Answer the following questions.

(1) Prove that the inequality $l(u, w) + d(v, u) - d(v, w) \geq 0$ holds for any vertex $v \in V$ and any edge $(u, w) \in E$ on the graph G.

(2) Assume that a value $s(v)$ is attached to each vertex $v \in V$ on the graph G. Consider a new graph G' resulting from transforming G by replacing the length of each edge $(u, v) \in E$ with $l(u, v) + s(u) - s(v)$. Prove that the shortest path on the graph G' between $w \in V$ and $x \in V$ is also the shortest path between w and x on the graph G.

(3) Describe an algorithm that computes the shortest path tree from a vertex $v \in V$ on the graph G, and discuss the computational complexity of the algorithm.

(4) Given the shortest path tree from $v \in V$, describe an algorithm that computes the shortest path tree from another vertex $w \in V$ $(w \neq v)$ on the graph G, and discuss the computational complexity of the algorithm.

問題 2(100 点).
二つの状態 1 と 2 を持つ機械がある．初期状態は 1 である．状態 1 において，文字 a を出力して状態 1 に遷移する (留まる) 確率が 1/2，文字 b を出力して状態 2 に遷移する確率が 1/4，停止する確率が 1/4 とする．状態 2 において，文字 b を出力して状態 1 に遷移する確率が 1/2，停止する確率が 1/2 とする．

(1) この機械が状態 i から $aa+b$ という正則表現に属する文字列を出力して状態 j に遷移する確率を p_{ij} とする．行列 $\begin{pmatrix} p_{11} & p_{21} \\ p_{12} & p_{22} \end{pmatrix}$ を求めよ．

(2) 上の行列を対角化せよ．

(3) 初期状態から動作を始めて，この機械が停止するまでに出力する文字列が，$(aa+b)^*$ という正則表現に属する確率を求めよ．

Problem 2(100 points).
Consider a machine having two states, 1 and 2. The initial state is 1. At state 1, it outputs character a and moves to (stays at) state 1 with probability 1/2, or it outputs character b and moves to state 2 with probability 1/4, or it halts with probability 1/4. At state 2, it outputs b and moves to state 1 with probability 1/2, or it halts with probability 1/2.

(1) p_{ij} denotes the probability that the machine moves from state i to state j after outputting a string belonging to the regular expression $aa+b$. Compute the matrix $\begin{pmatrix} p_{11} & p_{21} \\ p_{12} & p_{22} \end{pmatrix}$.

(2) Diagonalize the above matrix.

(3) Compute the probability that the output string (the string that the machine outputs after starting at the initial state until it halts) belongs to the regular expression $(aa+b)^*$.

問題 3(100 点).
球をスクリーンに投影して表示する問題に関して, 以下の問いに答えよ.

(1) レイトレーング (光線追跡法) といわれる描画法で球をスクリーンに描画することを考える (図 1 参照). 半径 r で中心 $\boldsymbol{c}$ の球の表面上の任意の点 $\boldsymbol{p}$ は下式を満たす.

$$|\boldsymbol{p} - \boldsymbol{c}| - r = 0$$

また, 点 $\boldsymbol{e}$ を通過し, 単位方向ベクトル $\boldsymbol{d}$ の直線上の点 $\boldsymbol{s}$ はパラメータ t を用いて下式で表わされる.

$$\boldsymbol{s} = \boldsymbol{e} + \boldsymbol{d}t$$

球と直線の交点の有無は $\boldsymbol{p} = \boldsymbol{s}$ を満たす t が存在するかどうかで判断できる. この条件を $\boldsymbol{e}$, $\boldsymbol{c}$, $\boldsymbol{d}$ を用いて表わせ.

(2) 球と直線が交差する場合について，点 $\boldsymbol{e}$ に近い交点 $\boldsymbol{Q}$ を $\boldsymbol{e}$, $\boldsymbol{c}$, および $\boldsymbol{d}$ を用いて表現せよ.

(3) 交点 $\boldsymbol{Q}$ における球の単位法線ベクトルを求めよ. また, 球の表面は鏡のように光を反射するとして，点 $\boldsymbol{Q}$ に単位ベクトル $\boldsymbol{a}$ の方向から光が入射した場合の反射方向ベクトル $\boldsymbol{b}$ を計算せよ.

(4) 原点 $\boldsymbol{o}$ を通過し x 軸 y 軸を含む面をスクリーンとし, 視点 $\boldsymbol{e}$ は z 軸上にあるものとする. 視点を投影中心にして球をスクリーンに投影する場合を考える. すなわち, 視点 $\boldsymbol{e}$ を通過する直線と球との交点をスクリーンに投影する. なお, スクリーンをはさみ球全体と視点 $\boldsymbol{e}$ は反対側にあるものとする. 点 $\boldsymbol{e}$ および点 $\boldsymbol{c}$ の座標を $(0, 0, z_e)$ および (x_c, y_c, z_c) とし，まず，球の中心をスクリーンに投影した点の座標 x, y を求めよ. 次に，スクリーン上に投影された球が存在する y の最小値 y_{min} と最大値 y_{max} を算出せよ.

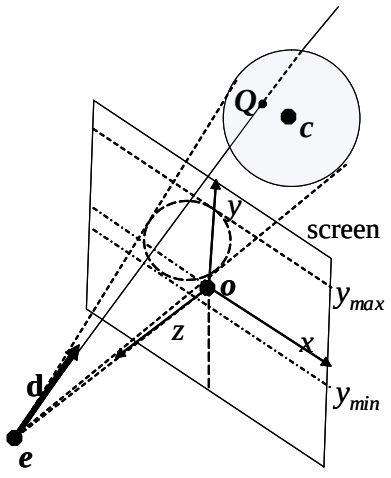

図 1

Problem 3(100 points).
Answer the following questions on rendering a sphere by projecting the sphere onto a screen.

(1) Consider rendering a sphere on the screen by using the rendering algorithm called raytracing (see Figure 1). A point $\boldsymbol{p}$ on a sphere of radius r about center point $\boldsymbol{c}$ is given by

$$|\boldsymbol{p} - \boldsymbol{c}| - r = 0.$$

A point $\boldsymbol{s}$ on the line which passes through a point $\boldsymbol{e}$ with unit vector $\boldsymbol{d}$ is expressed by using parameter t, and given by

$$\boldsymbol{s} = \boldsymbol{e} + \boldsymbol{d}t.$$

The intersection of the line and the sphere can be tested by checking the existence of t that satisfies $\boldsymbol{p} = \boldsymbol{s}$. Describe this condition by using $\boldsymbol{e}$, $\boldsymbol{c}$, and $\boldsymbol{d}$.

(2) Assuming that the line intersects with the sphere, express the intersection point $\boldsymbol{Q}$ close to point $\boldsymbol{e}$ by using $\boldsymbol{e}$, $\boldsymbol{c}$, and $\boldsymbol{d}$.

(3) Calculate the unit normal vector of the sphere at point $\boldsymbol{Q}$. Assuming that the sphere surface reflects light as a mirror, give the reflection vector $\boldsymbol{b}$ at the intersection point $\boldsymbol{Q}$ when the light is incident to the point with unit vector $\boldsymbol{a}$.

(4) Consider the screen on the x-y plane, and the eye point $\boldsymbol{e}$ lying on the z-axis. The sphere is projected onto the screen with the projection center being the eye point $\boldsymbol{e}$. That is, the projection is done by projecting the intersection points between the sphere and the lines through the eye point on the screen. The whole sphere and the point $\boldsymbol{e}$ are placed on the opposite sides of the screen. Let the coordinates of points $\boldsymbol{e}$ and $\boldsymbol{c}$ be expressed as $(0, 0, z_e)$ and (x_c, y_c, z_c), respectively. Calculate the x and y coordinates of the projected point of the center $\boldsymbol{c}$ of the sphere on the screen. Calculate the minimum and maximum values of the y component, y_{min} and y_{max}, of the projected sphere.

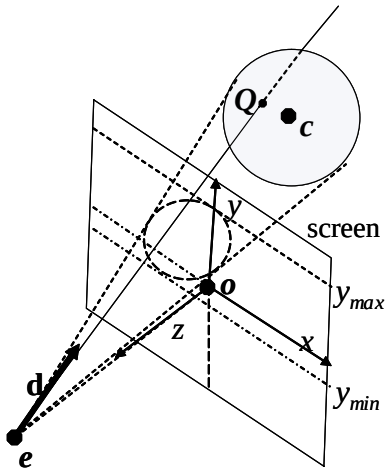

Figure 1

問題 4(100 点).
D フリップフロップ，AND ゲート，OR ゲート，NOT ゲートを用いて，以下の手順に従って 0, 1, 2, ..., 9, 10, 11 までの値を順次取り，11 の次は再び 0 に戻り動作を続ける同期式カウンタを設計せよ．なお，実装するカウンタはバイナリカウンタでも，ジョンソンカウンタでも良いが，リングカウンタであってはならない.

入力信号の仕様

入力信号	意味
クロック	同期式順序回路のクロック
カウント	1 の場合，カウントアップ，0 の場合値を保持
リセット	1 の場合，カウンタの値を 0 にする

(1) 状態遷移表および状態遷移図を作成せよ．状態遷移図 (State Diagram) はミーリ形でもムーア形でも良い.

(2) 同期式の論理回路を設計し，図示せよ.

(3) 仕様に含まれない状態 (例えば，4 ビットで設計する場合は，12, 13, 14, 15) に回路エラーで突入した場合，リセットを 1 にする以外は正常な状態に戻らないように設計を変更せよ.

Problem 4(100 points).
Consider designing a synchronous counter specified below using AND, OR, and NOT gates, and D-type flip-flops. The counter counts up as 0, 1, 2, ..., 9, 10, and 11. After 11, the counter returns to 0, and counts up again. The designed counter can be either a binary counter or a Jonson counter, but must not be a ring counter.

Specification of the input signals

Input signal	Description
Clock	A clock to a synchronous counter
Count	When Count is 1, the counter counts up.
	Otherwise, the counter holds its value.
Reset	When Reset is 1, the counter is set to 0.

(1) Describe a state transition table and a state diagram. The state diagram can be either a Mealy or a Moore.

(2) Design a synchronous circuit and show it as a circuit diagram.

(3) Modify the circuit so that when the counter enters unspecified state (for example, when 4 bit binary counter is used, 12, 13, 14, 15) due to some circuit errors, it does not return to normal states unless the reset signal is asserted to 1.