平成 26 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 25 年 8 月 20 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．

(2) 4 題すべてに答えよ．問題ごとに指定された解答用紙を使用すること．

(3) 解答用紙および問題冊子は持ち帰らないこと．

Specialized Subjects I

10:00 – 12:30, August 20, 2013

Entrance Examination (AY 2014)
Department of Computer Science, Graduate School of Information Science and Technology
The University of Tokyo

Notice:

(1) Do not open this problem booklet until the start of the examination is announced.

(2) Answer the following 4 problems. Use the designated answer sheet for each problem.

(3) Do not take the problem booklet or any answer sheet out of the examination room.

下欄に受験番号を記入すること．

Write your examinee's number in the box below.

受験番号	No.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

問題 1

n 個の 2 回連続微分可能な n 変数関数 $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ を考える．ここで $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$ である．これらの関数の二乗和を

$$S(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x})$$

として，S を局所的に最小にする $\boldsymbol{x} \in \mathbb{R}^n$ を計算したい．ただし $\mathbb{R}$ は実数全体の集合を表す．

以下の問いに答えよ．

(1) 1 変数関数 $\sigma(\xi)$ に関する方程式 $\sigma(\xi) = 0$ をみたす ξ を求める Newton 法（Newton-Raphson 法ともいう）の原理を示せ．

(2) S の x_j による偏微分

$$g_j(\boldsymbol{x}) = \frac{\partial S}{\partial x_j}$$

を f_i, x_j で表せ．また，S が最小となる $\boldsymbol{x}$ で $g_j(\boldsymbol{x})$ が取る値 $\hat{g}_j$ を答えよ．

(3) $g_j(\boldsymbol{x}) = \hat{g}_j$ $(j = 1, 2, \ldots, n)$ なる連立方程式を Newton 法で解きたい．第 m 反復での x_j の値を $x_j^{(m)}$ として，

$$\boldsymbol{x}^{(m)} = (x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)})^\top$$

とする．Newton 法の反復式を

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(m)} - H^{-1}\boldsymbol{g}(\boldsymbol{x}^{(m)})$$

とする．ここで $\boldsymbol{g} = (g_1, g_2, \ldots, g_n)^\top$ である．H の (j, k) 要素 H_{jk} を求めよ．

(4) 問い (3) で答えた H は，1 階微分と 2 階微分を必要とする．H から 2 階微分の項を省略したものを $\bar{H}$ として，

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(m)} - \bar{H}^{-1}\boldsymbol{g}(\boldsymbol{x}^{(m)})$$

を反復するアルゴリズムを Gauss-Newton 法という．Gauss-Newton 法は，S の最小値が 0 になる場合，解に十分近い初期値 $\boldsymbol{x}^{(0)}$ を与えると，最小値に 2 次収束する理由を説明せよ．ただし解の近くで $\bar{H}$ は正則であるとする．また，Hessian H が正則のときに Newton 法が 2 次収束することは既知とせよ．

Problem 1

Consider n functions $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$, ..., $f_n(\boldsymbol{x})$, each of which has n arguments $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$ and is twice continuously differentiable. We want to compute $\boldsymbol{x} \in \mathbb{R}^n$ that attains a local minimum of S, which is defined as the sum of squares of the functions as follows:

$$S(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}).$$

Here $\mathbb{R}$ denotes the set of all real numbers.

Answer the following questions.

(1) Let $\sigma(\xi)$ be a function of one argument. Describe the Newton method (also called the Newton-Raphson method) that computes a value of ξ that gives $\sigma(\xi) = 0$.

(2) Express

$$g_j(\boldsymbol{x}) = \frac{\partial S}{\partial x_j}$$

in terms of f_i and x_j. Also, determine the value of $g_j(\boldsymbol{x})$ (let $\hat{g}_j$ be the value) for $\boldsymbol{x}$ that minimizes S.

(3) Let us define a set of simultaneous equations $g_j(\boldsymbol{x}) = \hat{g}_j$ ($j = 1, 2, \ldots, n$); we want to solve this system using the Newton method. Let $x_j^{(m)}$ be the value of x_j at the m-th iteration, and define

$$\boldsymbol{x}^{(m)} = (x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)})^\top.$$

The iteration of the Newton method is represented as

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(m)} - H^{-1}\boldsymbol{g}(\boldsymbol{x}^{(m+1)}),$$

where $\boldsymbol{g} = (g_1, g_2, \ldots, g_n)^\top$. Describe H_{jk}, the (j,k) element of H.

(4) Note that H answered in Question (3) contains terms with first order and second order derivatives. Let $\bar{H}$ be obtained from H by removing the terms with the second order derivatives. The Gauss-Newton method is an iterative solver with iteration

$$\boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(m)} - \bar{H}^{-1}\boldsymbol{g}(\boldsymbol{x}^{(m+1)}).$$

Explain why the Gauss-Newton method gives second order convergence, when the minimum value of S is 0 and the initial value $\boldsymbol{x}^{(0)}$ is close enough to the solution. Here assume that $\bar{H}$ is not singular around the solution. Also assume that the following fact is known: the Newton method converges in second order when the Hessian H is not singular.

問題 2

文脈自由文法 G の（1ステップの）最右導出（最も右側の非終端記号の書き換え）の関係を $\longrightarrow_G$，その反射推移閉包を $\longrightarrow_G^*$ で表す．

例えば S を開始記号とする以下の文脈自由文法 G_0：

$$S \to E\$ \qquad E \to F \qquad E \to E+F \qquad F \to a \qquad F \to F*a$$

の場合，

$$S \longrightarrow_G E\$ \longrightarrow_G E+F\$ \longrightarrow_G E+a\$ \longrightarrow_G F+a\$ \longrightarrow_G a+a\$$$

という最右導出列が得られる．

次の問いに答えよ．

(1) 上の文法 G_0 による $a*a+a\$$ の最右導出列を書け．

いま，G が S を開始記号，$\mathbf{N}$ を非終端記号の集合，$\mathbf{T}$ を終端記号の集合としてもつ文脈自由文法であるとする．各書き換え規則 $A \to \alpha$（$A \in \mathbf{N}$，$\alpha \in (\mathbf{N} \cup \mathbf{T})^*$）について集合 $\mathit{Left}_G(A)$（$\subseteq (\mathbf{N} \cup \mathbf{T})^*$）および $L_{G,A\to\alpha}$（$\subseteq (\mathbf{N} \cup \mathbf{T})^*$）を以下によって定義する．

$$\mathit{Left}_G(A) = \{\beta \mid \exists w \in \mathbf{T}^*.\ S \longrightarrow_G^* \beta A w\} \qquad L_{G,A\to\alpha} = \{\beta\alpha \mid \beta \in \mathit{Left}_G(A)\}$$

以下の問いに答えよ．

(2) 上の文法 G_0 について，$\mathit{Left}_{G_0}(F)$ を求めよ．

(3) すべての文脈自由文法 G とその規則 $A \to \alpha$ について，$L_{G,A\to\alpha}$ が $\mathbf{N} \cup \mathbf{T}$ をアルファベットとする正規言語であることを示せ．

ただし解答には次の事実を用いて良い：左線形の文脈自由文法が生成する言語は正規言語である．ここで，文脈自由文法 G が左線形であるとは，G のすべての書き換え規則 $A \to \alpha$ において $\alpha \in \mathbf{T}^* \cup (\mathbf{N}\mathbf{T}^*)$ が成り立つことをいう．

(4) LR(0) 構文解析において reduce/reduce conflict が起きないための文脈自由文法 G の必要十分条件を，$L_{G,A\to\alpha}$ を用いて表せ．

Problem 2

For a context-free grammar G, let $\longrightarrow_G$ be the (one-step) rightmost derivation relation (denoting a rewriting of the rightmost non-terminal symbol), and $\longrightarrow_G^*$ be its reflexive and transitive closure.

For example, for the context-free grammar G_0 (where S is the start symbol):

$$S \to E\$ \qquad E \to F \qquad E \to E+F \qquad F \to a \qquad F \to F * a,$$

we have the following rightmost derivation sequence.

$$S \longrightarrow_G E\$ \longrightarrow_G E + F\$ \longrightarrow_G E + a\$ \longrightarrow_G F + a\$ \longrightarrow_G a + a\$.$$

Answer the following question.

(1) For the grammar G_0 above, write a rightmost derivation sequence to generate $a * a + a\$$.

Now let G be a context-free grammar that has start symbol S, the set $\mathbf{N}$ of non-terminal symbols, and the set $\mathbf{T}$ of terminal symbols. For each rewriting rule $A \to \alpha$ ($A \in \mathbf{N}, \alpha \in (\mathbf{N} \cup \mathbf{T})^*$), we define the sets $\mathit{Left}_G(A)$ ($\subseteq (\mathbf{N} \cup \mathbf{T})^*$) and $L_{G,A\to\alpha}$ ($\subseteq (\mathbf{N} \cup \mathbf{T})^*$) by:

$$\mathit{Left}_G(A) = \{\beta \mid \exists w \in \mathbf{T}^*.\ S \longrightarrow_G^* \beta A w\} \qquad L_{G,A\to\alpha} = \{\beta\alpha \mid \beta \in \mathit{Left}_G(A)\}.$$

Answer the following questions.

(2) For the grammar G_0 above, give $\mathit{Left}_{G_0}(F)$.

(3) Show that, for every context-free grammar G and every rule $A \to \alpha$ of G, $L_{G,A\to\alpha}$ is a regular language over the alphabet $\mathbf{N} \cup \mathbf{T}$.

You may use the following fact: the language generated by a left-linear context-free grammar is regular. Here a context-free grammar G is said to be left-linear if $\alpha \in \mathbf{T}^* \cup (\mathbf{N}\mathbf{T}^*)$ holds for every rewriting rule $A \to \alpha$ of G.

(4) Express, in terms of $L_{G,A\to\alpha}$, a sufficient and necessary condition for no reduce/reduce conflict occurring during LR(0) parsing for a context-free grammar G.

問題 3

x 座標が相異なる n 個の点からなる x-y 平面上の点集合 S を考える．$\mathrm{dist}(v,w)$ は 2 点 v, w 間のユークリッド距離を表すものとする．このとき，以下の問いに答えよ．ただし丸め誤差は考えないものとする．

(1) S の n 個の点を x 座標について昇順にソートするアルゴリズムを 1 つ示し，その時間計算量を述べよ．

なお，各点は x 座標を表す浮動小数点数と y 座標を表す浮動小数点数からなる構造体として表現され，点集合 S はその構造体の長さ n の配列として与えられているものとする．

(2) S 中の相異なる 2 点からなるすべての組 $\langle v,w\rangle$ $(v \neq w)$ のうち，$\mathrm{dist}(v,w)$ が最小である組の距離の値を ℓ_S とする．このとき，平面上のいかなる点 (x^*,y^*) に対しても，$x^* \le x \le x^* + \ell_S/2$, $y^* \le y \le y^* + \ell_S/2$ で表される正方形内に S の点は高々1 つしかないことを示せ．

(3) S の n 個の点を x 座標について昇順にソートしたものを v_1, v_2, …, v_n とする．このとき，

$$P = \{v_1, v_2, \ldots, v_{\lfloor n/2 \rfloor}\}\ , \qquad Q = \{v_{\lfloor n/2\rfloor+1}, v_{\lfloor n/2\rfloor+2}, \ldots, v_n\}$$

とする．ただしここで $\lfloor x \rfloor$ は $x-1 < \lfloor x \rfloor \le x$ なる唯一の整数を表す．

P 中の相異なる 2 点からなる組 $\langle s,t\rangle$ $(s \neq t)$ のうち，$\mathrm{dist}(s,t)$ が最小である組の距離の値を ℓ_P とする．点 $q \in Q$ を一つ固定したとき，$\mathrm{dist}(p,q) < \ell_P$ を満たすような点 $p \in P$ の個数を考えると，この個数は 8 個を超えないことが問い (2) を用いてわかる．このことを証明せよ．

(4) P の点 $p \in P$ の中で，点 $q \in Q$ からの距離 $\mathrm{dist}(p,q)$ が最小となるもの（複数ある場合はそのうちの 1 つ）を $\mathrm{nearest}_P(q)$ とおく．

ℓ_P の値が与えられているものとするとき，

$$\mathrm{dist}(\mathrm{nearest}_P(q), q) < \ell_P$$

を満たすようなすべての $q \in Q$ の集合を $\{q_1, q_2, \ldots, q_m\}$ とする．これらの点に対して

$$\mathrm{nearest}_P(q_1),\ \mathrm{nearest}_P(q_2),\ \ldots,\ \mathrm{nearest}_P(q_m)$$

をすべて求めたい．そのための効率のよいアルゴリズムとその時間計算量を示せ．

(5) S に対し ℓ_S を求めるアルゴリズムとその時間計算量を示せ．

Problem 3

Consider a set S of n points on an x-y plane, whose x coordinates are all different. Let $\mathrm{dist}(v, w)$ denote the Euclidean distance between the two points v and w. Answer the following questions. You can assume that computations have no round-off errors.

(1) Give an algorithm that sorts all the n points in S so that their x coordinates increase. Describe its time complexity.

Assume here that each point is defined using a structure that consists of a floating point number variable that represents its x coordinate and another floating point number variable that represents its y coordinate. The set S of points is given as a length-n array of the structures.

(2) Let ℓ_S denote the minimum value among the values $\mathrm{dist}(v, w)$ for all pairs $\langle v, w\rangle$ of distinct points in S (i.e. $v \neq w$). Show that, for any point (x^*, y^*) on the plane, there exists at most one point of S in the square defined by $x^* \leq x \leq x^* + \ell_S/2$ and $y^* \leq y \leq y^* + \ell_S/2$.

(3) Let v_1, v_2, ..., v_n be the points of S sorted so that their x coordinates increase. Let

$$P = \{v_1, v_2, \ldots, v_{\lfloor n/2 \rfloor}\} \ , \quad \text{and} \quad Q = \{v_{\lfloor n/2 \rfloor + 1}, v_{\lfloor n/2 \rfloor + 2}, \ldots, v_n\} \ ;$$

here $\lfloor x \rfloor$ denotes the unique integer such that $x - 1 < \lfloor x \rfloor \leq x$.

Let ℓ_P denote the minimum value among the values $\mathrm{dist}(s, t)$ for all pairs $\langle s, t\rangle$ of distinct points in P (i.e. $s \neq t$). Fix $q \in Q$ and consider the number of points p in P such that $\mathrm{dist}(p, q) < \ell_P$; using Question (2) it can be seen that the number does not exceed 8. Prove this.

(4) For a point $q \in Q$, let $\mathrm{nearest}_P(q)$ denote the point $p \in P$ that has the minimum $\mathrm{dist}(p, q)$ value (if there are multiple such $p \in P$, let $\mathrm{nearest}_P(q)$ be one of them).

Let $\{q_1, q_2, \ldots, q_m\}$ be the set of all those points $q \in Q$ which satisfy

$$\mathrm{dist}(\mathrm{nearest}_P(q), q) < \ell_P \ .$$

For those points we want to compute all of

$$\mathrm{nearest}_P(q_1),\ \mathrm{nearest}_P(q_2),\ \ldots,\ \mathrm{nearest}_P(q_m) \ .$$

Describe an efficient algorithm and show its time complexity.

(5) Describe an algorithm that computes ℓ_S for S, and show its time complexity.

問題 4

以下にスタック計算機シミュレータの一部を示す.

```
typedef unsigned char ub;
typedef signed char sb;
sb sim(ub *code, ub *dp, ub *sp)
{
 sb  disp; sb  run = 1, tmp;
 while (run) {
   switch(*code++) {
   case 00: /* push data */
     disp = *code++; *--sp = *(dp + disp);
     break;
   case 01: /* dup */
     --sp; *sp = *(sp + 1);
     break;
   case 02: /* swap */
     tmp = *sp;
     *sp = *(sp + 1); *(sp + 1) = tmp;
     break;
   case 03: /* add */
     tmp = (sb)*sp + (sb) *(sp + 1);
     *++sp = tmp;
     break;
   case 04: /* ... */
     tmp = *sp == *(sp + 1);
     *++sp = tmp;
     break;
   case 05: /* ... */
     tmp = *sp; disp = *code++;
     if (tmp == 0) code += disp;
     break;
   case 06: /* ... */
     disp = *code++; code += disp;
     break;
   case 07: /* stop */
     run = 0; break;
   case 255:
     abort(); /* never returns */
     break;
   }
 }
 return (sb) *sp;
}
```

以下の問いに答えよ.

(1) メモリアドレス 0 番地から以下のような機械語コード及びデータが格納されている. 各要素は 1 byte である.

0 番地: 00, 01, 01, 03, 07, 00, 00, 00, 00, 00,
10 番地: 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
20 番地: 00, 10, -1, 00, 00, 00, 00, 00, 00, 00

以下のように実行した時の返り値を求めよ.

`sim((ub*) 0, (ub*) 20, (ub*) 20)`

(2) アセンブリ言語を設計し, 以下 (†) の機械語コード及びデータをアセンブリ言語で記述せよ.

(†) 0 番地: 00, 01, 00, 01, 00, 02, 03, 05, 07, 02,
10 番地: 00, 03, 03, 02, 06, -12, 02, 07, 00, 00,
20 番地: 00, 10, -1, 01, 00, 00, 00, 00, 00, 00

(3) 上記 (†) を以下 (‡) のように実行するとアボートする. 理由を述べよ.

(‡) `sim((ub*) 0, (ub*) 20, (ub*) 20)`

(4) 上記 (†) をアボートを避けながら期待通りに動作させるためには, (†) または (‡) をどう修正すれば良いか述べよ.

(5) 問い (3) のような問題が発生しないようにするために, スタック計算機シミュレータをどのように改良すれば良いか議論せよ.

Problem 4

The following is a part of a stack machine simulator.

```
typedef unsigned char ub;
typedef signed char sb;
sb sim(ub *code, ub *dp, ub *sp)
{
 sb  disp; sb  run = 1, tmp;
 while (run) {
   switch(*code++) {
   case 00: /* push data */
     disp = *code++; *--sp = *(dp + disp);
     break;
   case 01: /* dup */
     --sp; *sp = *(sp + 1);
     break;
   case 02: /* swap */
     tmp = *sp;
     *sp = *(sp + 1); *(sp + 1) = tmp;
     break;
   case 03: /* add */
     tmp = (sb)*sp + (sb) *(sp + 1);
     *++sp = tmp;
     break;
   case 04: /* ... */
     tmp = *sp == *(sp + 1);
     *++sp = tmp;
     break;
   case 05: /* ... */
     tmp = *sp; disp = *code++;
     if (tmp == 0) code += disp;
     break;
   case 06: /* ... */
     disp = *code++; code += disp;
     break;
   case 07: /* stop */
     run = 0; break;
   case 255:
     abort(); /* never returns */
     break;
   }
 }
 return (sb) *sp;
}
```

Answer the following questions.

(1) Assume that the following machine code and data are stored in the memory area starting at the address 0. Each element represents 1-byte data.

```
Address  0: 00, 01, 01, 03, 07, 00, 00, 00, 00, 00,
Address 10: 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
Address 20: 00, 10, -1, 00, 00, 00, 00, 00, 00, 00
```

Answer the return value of the following function call.

```
sim((ub*) 0, (ub*) 20, (ub*) 20)
```

(2) Design an assembly language, and express the following machine code and data (†) using the assembly language.

(†)

```
Address  0: 00, 01, 00, 01, 00, 02, 03, 05, 07, 02,
Address 10: 00, 03, 03, 02, 06, -12, 02, 07, 00, 00,
Address 20: 00, 10, -1, 01, 00, 00, 00, 00, 00, 00
```

(3) When the above (†) is executed as the following (‡), it aborts. Describe the reason.

(‡)

```
sim((ub*) 0, (ub*) 20, (ub*) 20)
```

(4) Describe how to modify (†) or (‡) in order to make the above (†) work without abort while maintaining the expected behaviors.

(5) Discuss how the stack machine simulator can be modified in order to avoid such problems as in Question (3).

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.