

平成 31 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 30 年 8 月 20 日
13:45 – 15:15

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) 3 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

13:45 – 15:15, August 20, 2018

Entrance Examination (AY 2019)

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 3 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

この問題では n ビットによって 0 から $2^n - 1$ までの整数を表現する。以下の問いに答えよ。

- (1) Half adder および Full adder を、それぞれ、AND, OR, NOT ゲートのみを用いて設計せよ。
- (2) 3つの 16 ビット整数 a, b, c を入力として、 $a + b + c = d + e$ となるような 17 ビット整数 d, e を出力する回路を設計せよ。ただし使えるのは Half adder と Full adder のみ（個数に制限はない）であり、それらを直列につなぐことはできない。
- (3) 4つの 16 ビット整数 a_1, a_2, a_3, a_4 を入力として、 $a_1 + a_2 + a_3 + a_4 = b_1 + b_2$ となるような 17 ビット整数 b_1, b_2 を出力する回路を設計せよ。ただし使えるのは Half adder と Full adder のみ（個数に制限はない）であり、直列につながる adder の段数を最小にせよ。
- (4) 2つの 8 ビット整数 a, b を入力として、 $a \times b = c + d$ を満たす整数 c, d を出力する回路を設計せよ。ただし使えるのは 8×8 個の 1 ビット乗算器に加えて Half adder と Full adder のみ（個数に制限はない）である。

Problem 1

In this problem, integers ranging from 0 to $2^n - 1$ are represented by n bits. Answer the following questions.

- (1) Design a Half adder and a Full adder. You can use only AND, OR, and NOT gates.
- (2) Design a circuit as follows. Its inputs are three 16-bit integers a , b , and c . The outputs are two 17-bit integers d and e such that $a + b + c = d + e$. Here, you can use only Half adders and Full adders (the number of the adders is not restricted), and no adders can be connected in series.
- (3) Design a circuit as follows. Its inputs are four 16-bit integers a_1 , a_2 , a_3 , and a_4 . The outputs are two 17-bit integers b_1 and b_2 such that $a_1 + a_2 + a_3 + a_4 = b_1 + b_2$. Here, you can use only Half adders and Full adders (the number of the adders is not restricted), and the maximum number of adders connected in series must be minimized.
- (4) Design a circuit as follows. Its inputs are two 8-bit integers a and b . The outputs are two integers c and d such that $a \times b = c + d$. Here, you can use only Half adders and Full adders (the number of the adders is not restricted) in addition to 8×8 one-bit multipliers.

問題 2

32KB の物理メモリを有する 32 ビットのマシン上でオペレーティングシステムがページング機能を提供する場合を考える。ページサイズは 4KB、仮想メモリのサイズは 4GB であり、キャッシュメモリは存在しない。以下の問いに答えよ。なお、1KB は 1024 バイトであるとする。

- (1) ページング機能に関する以下の用語の意味をそれぞれ簡潔に説明せよ。
- ページ
 - ページテーブル
 - ページ置換
 - ページフォールト
 - アドレス変換バッファ (TLB)
- (2) 以下のページテーブルが与えられた際の仮想アドレス 2A0F (16 進数) に対応する物理アドレスを 16 進数で求めよ。

ページ番号 (10 進数)	フレーム番号 (2 進数)	有効ビット
0	111	1
1	000	0
2	110	1
3	000	0
4	101	1
5	000	0
6	000	0
7	000	0
8	000	0
9	001	1
10	100	1
11	000	1
12	011	1
13	000	0
14	000	0
15	010	1

- (3) C 言語で書かれた以下の 2 つのプログラムコードをそれぞれ実行した際に発生するページフォールトの回数を求めよ。

プログラムコード 1:

```
for (j = 0; j < 1024; j++)
  for (i = 0; i < 1024; i++)
    sum += A[i * 1024 + j];
```

プログラムコード 2:

```
for (i = 0; i < 1024; i++)
  for (j = 0; j < 1024; j++)
    sum += A[i * 1024 + j];
```

なお、各プログラムコードを実行する際には以下の仮定が成り立っているものとする。

- i , j , sum は 32 ビット整数型の変数である。A は 1024×1024 個の要素をもつ 32 ビット整数型の 1 次元配列である。sum と A の各要素の値はすべて設定されている。
- コンパイラによるプログラムコード最適化は無効である。
- 初期状態において、A に割り当てられたページはすべて有効ではない。A 以外のデータ (プログラムコードとページテーブルを含む) は有効な 1 ページに割り当てられており、それはページアウトされない。
- A の先頭アドレスに対するページ割当てはページ境界にアラインされている。
- ページ置換は Least Recently Used (LRU) 方針に基づく。

Problem 2

We consider a 32-bit machine with 32KB physical memory, upon which the operating system supports the paging functionality. The page size is 4KB, the virtual memory size is 4GB, and there is no cache memory. Answer the following questions. Note that 1KB is equivalent to 1024 bytes.

- (1) Explain each of the following terms regarding the paging functionality, briefly.
- Page
 - Page Table
 - Page Replacement
 - Page Fault
 - Translation Look-aside Buffer (TLB)
- (2) Obtain the physical address in hexadecimal corresponding to the virtual address of 2A0F (hexadecimal) when the following page table is given.

Page number (decimal)	Frame number (binary)	Valid bit
0	111	1
1	000	0
2	110	1
3	000	0
4	101	1
5	000	0
6	000	0
7	000	0
8	000	0
9	001	1
10	100	1
11	000	1
12	011	1
13	000	0
14	000	0
15	010	1

- (3) Obtain the number of page faults caused by executing each of the following two pieces of program code written in C language.

Program Code 1:

```
for (j = 0; j < 1024; j++)
  for (i = 0; i < 1024; i++)
    sum += A[i * 1024 + j];
```

Program Code 2:

```
for (i = 0; i < 1024; i++)
  for (j = 0; j < 1024; j++)
    sum += A[i * 1024 + j];
```

Note that each program code is executed under the following assumptions.

- i , j , and sum are 32-bit integer variables. A is a 32-bit integer 1-dimensional array with 1024×1024 elements. The values of sum and each element of A are all set.
- Program code optimization by a compiler is disabled.
- In the initial state, all the pages that are allocated for A are not valid. Any data (including the program code and the page table) other than A are allocated to some valid page, and it is never paged out.
- Page allocation for the start address of A is aligned with a page boundary.
- Page replacement is based on the Least Recently Used (LRU) policy.

問題 3

以下では、決定性有限オートマトンを5つ組 $A = (Q, \Sigma, \delta, q_0, F)$ (Q は状態の有限集合, Σ は入力記号の有限集合, $\delta \in Q \times \Sigma \rightarrow Q$ は遷移関数, $q_0 \in Q$ は初期状態, $F \subseteq Q$ は受理状態の集合) で表し、文脈自由文法を4つ組 $G = (V, \Sigma, P, S)$ (V は非終端記号の有限集合, Σ は終端記号の有限集合, P は生成規則の有限集合, $S \in V$ は出発記号) で表す。文脈自由文法 $G = (V, \Sigma, P, S)$ がチョムスキー標準形であるとは、すべての生成規則が $B \rightarrow CD$, $B \rightarrow a$, $S \rightarrow \epsilon$ (B は非終端記号, C, D は S 以外の非終端記号, a は終端記号, ϵ は空列) のいずれかの形であることをいう。決定性有限オートマトンによって受理される言語を $\mathcal{L}(A)$, 文脈自由文法 G によって生成される言語を $\mathcal{L}(G)$ と書く。以下の問いに答えよ。

- (1) $A = (Q, \Sigma, \delta, q_0, F)$ を決定性有限オートマトンとする。 $\mathcal{L}(A)$ の補集合, すなわち $\Sigma^* \setminus \mathcal{L}(A)$ を受理する決定性有限オートマトンを与えよ。ただし $\delta \in Q \times \Sigma \rightarrow Q$ は全域関数であると仮定してよい。
- (2) 文脈自由文法 G を入力とし, $\mathcal{L}(G) = \emptyset$ であるか否かを決定するアルゴリズムを示せ。
- (3) チョムスキー標準形の文脈自由文法 $G = (V, \Sigma, P, S)$ および決定性有限オートマトン $A = (Q, \Sigma, \delta, q_0, F)$ に対し, 文脈自由文法 $G_A = (V_A, \Sigma, P_A, S')$ を以下によって定義する。

$$\begin{aligned} V_A &= \{S'\} \cup \{B_{q,q'} \mid B \in V \wedge (q, q' \in Q)\} \\ P_A &= \{B_{q,q'} \rightarrow C_{q,q''} D_{q'',q'} \mid (B \rightarrow CD \in P) \wedge (q, q', q'' \in Q)\} \\ &\quad \cup \{B_{q,q'} \rightarrow a \mid (B \rightarrow a \in P) \wedge (q, q' \in Q) \wedge (\delta(q, a) = q')\} \\ &\quad \cup \{S' \rightarrow S_{q_0, q} \mid q \in F\}. \end{aligned}$$

ただし $S' \notin \{B_{q,q'} \mid B \in V \wedge (q, q' \in Q)\}$ とする。 $\mathcal{L}(G_A) = (\mathcal{L}(G) \cap \mathcal{L}(A)) \setminus \{\epsilon\}$ が成り立つことを証明せよ。

- (4) チョムスキー標準形の文脈自由文法 G および決定性有限オートマトン A が与えられたとき, $\mathcal{L}(G) \subseteq \mathcal{L}(A)$ が成り立つか否かを決定する方法を与えよ。問い (1), (2) および (3) の結果を使ってもよい。

Problem 3

In the following, we represent a deterministic finite automaton as a quintuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ (where Q is a finite set of states, Σ is a finite set of input symbols, $\delta \in Q \times \Sigma \rightarrow Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states), and a context-free grammar as a quadruple $G = (V, \Sigma, P, S)$ (where V is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols, P is a finite set of production rules, and $S \in V$ is the start symbol). A context-free grammar $G = (V, \Sigma, P, S)$ is in *Chomsky normal form* if each production rule is of the form $B \rightarrow CD$, $B \rightarrow a$, or $S \rightarrow \epsilon$ (where B is a non-terminal symbol, C and D are non-terminal symbols other than S , a is a terminal symbol, and ϵ is an empty sequence). We write $\mathcal{L}(\mathcal{A})$ for the language accepted by a deterministic finite automaton \mathcal{A} , and $\mathcal{L}(G)$ for the language generated by a context-free grammar G . Answer the following questions.

- (1) Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Give a deterministic finite automaton that accepts the complement of $\mathcal{L}(\mathcal{A})$, i.e., $\Sigma^* \setminus \mathcal{L}(\mathcal{A})$. Here, you may assume that $\delta \in Q \times \Sigma \rightarrow Q$ is a total function.
- (2) Describe an algorithm that takes a context-free grammar G as an input, and decides whether or not $\mathcal{L}(G) = \emptyset$.
- (3) Given a context-free grammar $G = (V, \Sigma, P, S)$ in Chomsky normal form and a deterministic finite automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, we define a context-free grammar $G_{\mathcal{A}} = (V_{\mathcal{A}}, \Sigma, P_{\mathcal{A}}, S')$ by:

$$\begin{aligned} V_{\mathcal{A}} &= \{S'\} \cup \{B_{q,q'} \mid B \in V \wedge (q, q' \in Q)\} \\ P_{\mathcal{A}} &= \{B_{q,q'} \rightarrow C_{q,q''} D_{q'',q'} \mid (B \rightarrow CD \in P) \wedge (q, q', q'' \in Q)\} \\ &\quad \cup \{B_{q,q'} \rightarrow a \mid (B \rightarrow a \in P) \wedge (q, q' \in Q) \wedge (\delta(q, a) = q')\} \\ &\quad \cup \{S' \rightarrow S_{q_0, q} \mid q \in F\}. \end{aligned}$$

Here, assume $S' \notin \{B_{q,q'} \mid B \in V \wedge (q, q' \in Q)\}$. Prove that $\mathcal{L}(G_{\mathcal{A}}) = (\mathcal{L}(G) \cap \mathcal{L}(\mathcal{A})) \setminus \{\epsilon\}$ holds.

- (4) Give a method to decide, given a context-free grammar G in Chomsky normal form and a deterministic finite automaton \mathcal{A} as inputs, whether or not $\mathcal{L}(G) \subseteq \mathcal{L}(\mathcal{A})$ holds. You may use the results of questions (1), (2), and (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.

