平成19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18年8月 22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) 5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 five 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).

十分な回数だけ連続微分可能な一変数関数 $f(x)$ に関する微分の差分近似について以下の問いに答えよ．ただし $h > 0$ を固定し，整数 i に対して

$$\begin{aligned} f_i &= f(x+ih) \\ f_i' &= f'(x+ih) \end{aligned}$$

と略記する．ここで $f'(x) = \mathrm{d}f/\mathrm{d}x$ である．

(1) $f'(x)$ の近似として定数係数 a_{-1}, a_0, a_1 により

$$f_0' \approx \frac{a_{-1}f_{-1} + a_0 f_0 + a_1 f_1}{h}$$

の形を考える．(左辺) − (右辺) のテーラー展開において，係数が 0 でない最初の項が h^p の項となる（h^{p-1} の項まで 0 となる）ようにするとき，次数 p が最大になるように定数係数 a_{-1}, a_0, a_1 を与え，そのときの p を求めよ．

(2) さらに精度の高い近似を求めるため

$$f_0' \approx \frac{b_{-2}f_{-2} + b_{-1}f_{-1} + b_0 f_0 + b_1 f_1 + b_2 f_2}{h} \qquad [1]$$

の形を考える．(左辺) − (右辺) のテーラー展開が h^{q-1} の項まで 0 となるようにするとき，次数 q が最大になるように定数係数 b_{-2}, b_{-1}, b_0, b_1, b_2 を与え，そのときの q を求めよ．

(3) 異なる手法として

$$c_{-1}f_{-1}' + c_0 f_0' + c_1 f_1' \approx \frac{d_{-1}f_{-1} + d_0 f_0 + d_1 f_1}{h} \qquad [2]$$

の形を考える（この場合，連立一次方程式を解いて f_i' を求めることになる）．(左辺) − (右辺) のテーラー展開が h^{r-1} の項まで 0 となるようにするとき，次数 r が最大になるように定数係数 c_{-1}, c_0, c_1, d_{-1}, d_0, d_1 を与え（$c_{-1} + c_0 + c_1 = 1$ となるように選べ），そのときの r を求めよ．

(4) 式 [1] により表される近似法と，式 [2] により表される近似法について，計算量・並列性・境界付近の点への適用のしやすさなどに着目し，それぞれの利点と欠点について論じよ．

Problem 1(100 points).

Answer the following questions about finite difference approximations of derivatives of (univariate) function $f(x)$ that is differentiable sufficiently many times. Let

$$\begin{aligned} f_i &= f(x+ih) \\ f_i' &= f'(x+ih) \end{aligned}$$

for fixed $h > 0$ and integer i. Here $f'(x) = \mathrm{d}f/\mathrm{d}x$.

(1) Consider an approximation of $f'(x)$ with constant coefficients a_{-1}, a_0 and a_1 as

$$f_0' \approx \frac{a_{-1}f_{-1} + a_0 f_0 + a_1 f_1}{h}.$$

Let the first non-zero term of the Taylor expansion of (left-hand side) − (right-hand side) be h^p (or the terms vanish up to h^{p-1}). Determine constant coefficients a_{-1}, a_0 and a_1 so as to maximize the order p, and give the resulting p.

(2) Consider an approximation

$$f_0' \approx \frac{b_{-2}f_{-2} + b_{-1}f_{-1} + b_0 f_0 + b_1 f_1 + b_2 f_2}{h} \qquad [1]$$

to improve precision. Let the terms of the Taylor expansion of (left-hand side)−(right-hand side) vanish up to h^{q-1}, and determine constant coefficients b_{-2}, b_{-1}, b_0, b_1 and b_2 so as to maximize the order q, and give the resulting q.

(3) Consider an approximation

$$c_{-1}f_{-1}' + c_0 f_0' + c_1 f_1' \approx \frac{d_{-1}f_{-1} + d_0 f_0 + d_1 f_1}{h} \qquad [2]$$

where f_i' are obtained by solving a system of linear equations. Let the terms of the Taylor expansion of (left-hand side) − (right-hand side) vanish up to h^{r-1}, and determine constant coefficients c_{-1}, c_0, c_1, d_{-1}, d_0 and d_1 so as to maximize the order r and satisfy $c_{-1}+c_0+c_1 = 1$, and give the resulting r.

(4) Discuss advantage(s) and disadvantage(s) of the approximations given by equations [1] and [2], especially computational costs, parallelism, and difficulties in applying to points near the boundaries.

問題 2(100 点).
X 先生のクラスでテストを行った．このクラスには m 人の生徒がおり，そのうち i 番目の生徒のテストの得点は s_i であった．なお得点はすべて正の整数であったものとする．彼らの頑張りに対し，X 先生は n 個のキャンディーをご褒美として用意した．X 先生としては，とった得点に比例した個数のキャンディーをそれぞれの生徒に与えたいが，残念ながら $n \cdot s_i / \sum_i s_i$ が整数とならないため，公平にすべてのキャンディーを分配することができない．そのため，X 先生は以下のような方法でキャンディーを 1 つずつ配ることにした．

i 番目の生徒の持つキャンディーの数を c_i とする．最初はどの生徒もキャンディーは持っていない．この時，c_i/s_i の最小な生徒を一人選び，その生徒にキャンディーを 1 つ与える（すなわち，その生徒が k 番目の生徒だとすると，c_k を 1 増やす）．X 先生はこれを n 回繰り返すことで，キャンディーを配り終えた．以下の問いに答えよ．

(1) i 番目の生徒にキャンディーを d_i 個配った際の配り方の公平さの指標として

$$\min_i \{d_i/s_i\}$$

という数値を考え，この数値が大きければ大きいほど，配り方がよいものだと考えるものとする．この時，X 先生の配り方よりもよりよいキャンディーの配り方が存在しないことを証明せよ．

(2) それぞれ 70 点， 55 点， 25 点をとった 3 人の生徒に，35 個のキャンディーを X 先生の配り方を用いて配布した場合，各生徒に何個ずつキャンディーを配ることになるか？

(3) 最終的に X 先生が各生徒にキャンディーをいくつ配ったかを計算するアルゴリズムを示し，その計算量について考察せよ．

Problem 2(100 points).

Mr. X, a school teacher, did a test in his class. There are m students in the class and the ith student got s_i as a score in the test, where all the scores were positive integers. As the students worked very hard before the test, Mr. X brought n candies as awards for them. Mr. X wanted to fairly distribute the candies so that the number of the candies given to each student is proportional to his score. But, unfortunately, $n \cdot s_i / \sum_i s_i$ was not an integer for all the students, and Mr. X could not give out candies so fairly. Thus he decided to distribute the candies one by one in the following manner.

Let c_i be the number of candies that the ith student currently has. At first, all the students have no candies. Choose a student who has the smallest c_i/s_i value, and give him a candy (*i.e.* increase c_k by 1 where the chosen student is the kth student). Mr. X repeated this routine n times to finish the distribution of all the candies. Answer the following questions:

(1) Consider the value

$$\min_i \{d_i/s_i\}$$

as a fairness measure for a distribution of candies, where d_i is the number of candies that the ith student got. We consider that the larger the measure is, the better the distribution method is. Prove that there is no distribution method that is better than Mr. X's method.

(2) How many candies can each student get, if we distribute 35 candies by using Mr. X's method to three students who got 70, 55 and 25 as their scores?

(3) Describe an algorithm that computes how many candies Mr. X gave to each student in the end. Discuss the computational complexity of the algorithm.

問題 3(100 点).
c を定数記号，f を 1 引数の関数記号，P を 1 引数の述語記号とする．定数記号，関数記号，述語記号は，これら以外にはないものとする．以下の問いに答えよ．

(1) 次の論理式を充足する Herbrand 解釈を与えよ．

$$P(c) \land \neg P(f(c)) \land (\forall x.\ P(x) \supset P(f(f(x)))) \land (\forall x.\ P(f(f(x))) \supset P(x))$$

(2) 次の論理式を充足する Herbrand 解釈が存在しないことを説明せよ．

$$P(c) \land (\forall x.\ P(x) \supset P(f(x))) \land (\forall x.\ \exists y.\ P(x) \supset \neg P(y)) \qquad [1]$$

(3) Skolem 関数を導入して，論理式 [1] を充足する Herbrand 解釈を与えよ．

Problem 3(100 points).
Let c be a constant symbol, f a unary function symbol, and P a unary predicate symbol. There are no other constant symbols, function symbols, or predicate symbols. Answer the following questions:

(1) Give an Herbrand interpretation that satisfies the following formula.

$$P(c) \land \neg P(f(c)) \land (\forall x.\ P(x) \supset P(f(f(x)))) \land (\forall x.\ P(f(f(x))) \supset P(x))$$

(2) Explain that there exists no Herbrand interpretation that satisfies the following formula.

$$P(c) \land (\forall x.\ P(x) \supset P(f(x))) \land (\forall x.\ \exists y.\ P(x) \supset \neg P(y)) \qquad [1]$$

(3) Introduce a Skolem function, and obtain an Herbrand interpretation that satisfies the formula [1].

問題 4(100 点).
プログラミング言語処理系における，関数呼び出しの実装法に関して以下の問いに答えよ．

(1) ある言語では，一つのプログラムに対して，関数呼び出しが固定サイズのメモリ領域で実現可能という．この言語は関数呼び出しに関して，どのような仮定をおいていると考えられるか．

(2) 高階関数がなく，関数の入れ子があるような言語で，使用できる実装法にディスプレイ法がある．動作の詳細を説明せよ．

(3) 高階関数があり，かつ関数の入れ子がある言語で，(2) の実現法が破綻するプログラム例を挙げ，どう破綻するのか説明せよ．

(4) 高階関数があり，かつ関数の入れ子がある言語でも，(2) の実現法が使用できるようにするためには，言語が，関数に対してどのような制限を課せばよいか．なぜそれでうまくいくのかも説明せよ．ただし，内側の関数が，外側の関数で宣言された変数をアクセスできるものとする．

Problem 4(100 points).
Answer the following questions regarding to implementation methods for function calls in programming languages.

(1) Consider a language whose function calls can be implemented with a fixed amount of memory for a given program. What is an assumption of the language on function calls ?

(2) The “display” method is an implementation technique that can be used for a language that supports only first-order but possibly nested functions. Explain how it works in detail.

(3) Give a program example that is written in a language supporting both higher-order and nested functions and that defeats the implementation method in (2). Explain why.

(4) What restriction should we impose on functions in order to be able to use the implementation method in (2) even if the language supports both higher-order and nested functions? Explain why it works. Here, we assume that inner functions can access variables declared in outer functions.

問題 5(100 点).
パイプラインアーキテクチャは，CPU を高速化する重要な方法である．以下の問いに答えよ．

(1) 各パイプラインレジスタの遅延時間が T_p，パイプラインレジスタ間の組合せ論理回路遅延が，$T_1, T_2, \ldots, T_n$ であるとき，パイプライン化による速度向上比の上限値を示し，その根拠を簡潔に記述せよ．

(2) パイプラインアーキテクチャCPU において，構造ハザード，制御ハザード およびデータハザード の 3 種類がある．構造ハザード および制御ハザード について，その原因を示し，これらのハザード による速度低下を減少させる方法を述べよ．

(3) 命令発行および命令完了が Out of order である場合， 発生するデータハザード を述べよ．

(4) 問い (3) のデータハザード による速度低下を減少させる方法を述べよ．

(5) Out of order 実行モデルにおける割り込み処理の問題点および解決方法を述べよ．

Problem 5(100 points).
Pipeline architectures are an important method for improving the CPU performance. Answer the following questions.

(1) Let the delay time of each pipeline register be T_p, and the delay times of the combinatorial logic circuits between pipeline registers be $T_1, T_2, \ldots, T_n$. Show an upper limit of performance improvement ratio by employing the pipeline architecture, and explain it briefly.

(2) There are three types of hazards in a pipelined CPU: structural hazards, control hazards, and data hazards. Show sources of structural hazards and control hazards, and describe methods to reduce the performance degradation caused by those hazards.

(3) Describe data hazards in a pipeline architecture where the instruction issues and completions are out of order.

(4) Describe a way to reduce the performance degradation caused by the data hazards in the question (3).

(5) Explain an issue of interrupt handling in the out-of-oder execution model and describe a solution.

平成19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18年8月 22日
13:30〜16:00

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) 5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 five 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).
V を非終端記号の集合，Σ を終端記号の集合，S を開始記号，P を生成規則の集合とする文脈自由文法 $G = (V, \Sigma, S, P)$ は，その生成規則が次の形であるとき線形文法という．

$$\begin{aligned} A &\rightarrow uBv \\ A &\rightarrow w \end{aligned}$$

ここで A, B は非終端記号，すなわち $A, B \in V$ であり，u, v, w は終端記号列，すなわち $u, v, w \in \Sigma^*$ である．以下の問いに答えよ．

(1) 線形文法によって生成されるが正則でない言語 L を 1 つ与えよ．

(2) 言語 L を生成する線形文法を与えよ．

(3) 言語 L が正則でないことを証明せよ．

(4) 次の言語 L' は線形文法によっては生成されないことを証明せよ．

$$L' = \{a^m b^m c^n d^n \mid m, n \geq 1\}$$

Problem 1(100 points).
Let $G = (V, \Sigma, S, P)$ be a context-free grammar, where V is the set of nonterminal symbols, Σ is the set of terminal symbols, S is the start symbol, and P is the set of productions. We say that G is linear if the productions in P are in the following forms, where A and B are nonterminal symbols, i.e. $A, B \in V$, and u, v and w are strings in Σ^*:

$$\begin{aligned} A &\rightarrow uBv \\ A &\rightarrow w \end{aligned}$$

Answer the following questions:

(1) Give a language L which is generated by a linear context-free grammar but is not regular.

(2) Give a linear context-free grammar which generates L.

(3) Prove that L is not regular.

(4) Prove that the following language L' is not generated by any linear context-free grammar.

$$L' = \{a^m b^m c^n d^n \mid m, n \geq 1\}$$

問題 2(100 点).
DNA 分子は A, T, C, G で表される 4 種類のヌクレオチドで鎖状に構成され，これらの 4 種類の文字の列として表現することが可能である．DNA が長大な場合，実験的にこの文字列を決定するために以下の手法がとられる．まず対象とする DNA 分子の複数の複製に対して，超音波を用いて複数の短い断片に切断し，それらの断片の文字列を決定する．この時，DNA 分子の切断はランダムに行われるため，2 つの分子が全く同じ場所で切断されることはほとんどない．したがって，切断された短い文字列には重なりがある（ことが多い）．この重なりを用いてもとの断片を復元する．

これは実験エラー等がない理想的な場合には，与えられた文字列集合 F に含まれるすべての文字列を部分配列として含む文字列（これを超文字列とよぶ）のうち，最短のものを計算する問題ととらえることができる．これを最短超文字列とよび $SSS(F)$ と表すものとする．たとえば，文字列集合 $F = \{$ACT,CTA,AGT$\}$ に対して，$SSS(F) =$ACTAGT である．なお，この問題において，与えられた文字列集合の中のどの文字列も，他の文字列を部分文字列として含むことはないものとする．

ここで，文字列 p の接尾辞（末尾にある部分文字列）と文字列 q の接頭辞（先頭にある部分文字列）で一致するもののうち最大のものを $overlap(p, q)$ と表す（一致するものがない場合には空文字列とし，$|overlap(p, q)| = 0$ とする）．このとき，文字列集合 F を点の集合と考え，任意の相異なる 2 点 $p, q \in F$ に対し，長さが $(|p| - |overlap(p, q)|)$ である有向枝 (p, q) が存在するような有向グラフ $G(F)$ を考える．（なお，$|s|$ は文字列 s の長さを表す．）たとえば，F 上の 2 つの文字列，$p =$ATCGG と $q =$CGGTTAG を考えると，$overlap(p, q) =$CGG であり，枝 (p, q) の長さは 2 である．以下の問いに答えよ．

(1) $F = \{$GCC,ATGC,TGCAT$\}$ としたとき，グラフ $G(F)$ を図示せよ．さらに $SSS(F)$ を求めよ．

(2) $G(F)$ 上でのすべての点を通る最短の閉路 $H(G(F))$ を考える．その長さ $w(H(G(F)))$ が $|SSS(F)|$ 以下であることを証明せよ．

(3) グラフ G 上において，G 上のどの点もいずれかの閉路に含まれるような閉路の集合を閉路カバーという．$G(F)$ において，閉路長の総和が最短である閉路カバーを $L(F) = \{C_1, C_2, \ldots, C_k\}$ とし，閉路 C_i の長さを n_i とする．この時，2 点 $p \in C_i, q \in C_j$ $(i \neq j)$ に対し，$|overlap(p, q)| \leq n_i + n_j$ が成立することを証明せよ．

(4) $G(F)$ 上の閉路 C に対して，その閉路上の点を適当な点から閉路に沿って点をたどるのと同じ順番で文字列を並べ，重なる部分をできるだけ重なるように作成した文字列を $str(C)$ と表すものとする．（これは開始点によって文字列が変わるため，一意とは限らない．）ここで，「できるだけ重ねる」とは，文字列 y が文字列 xy と yz の最大の重なりである場合に，xy と yz から新しい文字列 xyz を作成することをいう．たとえば $p =$ATGG, $q =$GGCT, $r =$TATG からなる閉路 D に対する $str(D)$ は，p からたどった場合には ATGGCTATG となる．ここで，$L(F)$ に含まれるすべての閉路 C_i に対し，1 本ずつ文字列 $str(C_i)$ を考え，これら k 本の文字列をすべてつなげてできる文字列 $u = str(C_1) + str(C_2) + \ldots + str(C_k)$ を考える．このとき，u が F の超文字列であり，さらに $|u|$ が $|SSS(F)|$ の定数倍以下となっていることを証明せよ．

Problem 2(100 points).

DNA is a chain molecule consisting of four kinds of nucleotides, denoted `A`, `T`, `C`, and `G`, and thus DNA structure can be represented as a string of these four characters. If the DNA chain is quite long, the following procedure is used to determine the string experimentally: First, more than one copy of the DNA molecule are fragmented by supersonication and the strings of these fragments are determined. Since the fragmentation occurs at random sites, the cutting points in each molecule are not the same in most cases. Thus, it is very likely that any fragment has another fragment(s) that overlaps partially. Using these overlaps, the entire string is (hopefully) reconstructed.

Under ideal circumstances where experimental errors and other difficulties are ignorable, this method can be regarded as a problem of finding the shortest superstring of a given set F of strings, where a superstring of F means a string containing all elements of F as its substrings. We let $SSS(F)$ denote the shortest superstring of F. For example, $SSS(F)$ =`ACTAGT` for the string set $F = \{$`ACT`,`CTA`,`AGT`$\}$. In this problem, we assume that any string in the given string set does not contain any other string as its substring.

For two strings p and q, let $overlap(p, q)$ denote the longest string that is a suffix (substring that ends at the tail of the string) of p and is a prefix (substring that starts at the head of the string) of q at the same time. (If no such string exists, we let it be an empty string and let $|overlap(p, q)| = 0$.) We consider a directed graph $G(F)$ whose nodes are the elements of F and which has a directed edge between any two different nodes $p, q \in F$. (Note that $|s|$ denotes the length of the string s.) We let the edge length of (p, q) be $(|p| - |overlap(p, q)|)$. For p =`ATCGG` and q =`CGGTTAG` for example, $overlap(p, q)$ =`CGG` and the length of the edge (p, q) is 2. Answer the following questions:

(1) In case $F = \{$`GCC`,`ATGC`,`TGCAT`$\}$, draw the graph $G(F)$ and show the string $SSS(F)$.

(2) Let $H(G(F))$ be the shortest cycle that contains all the nodes in $G(F)$. Prove that the length of the cycle $w(H(G(F)))$ is smaller than or equal to $|SSS(F)|$.

(3) A cycle cover on graph G is a set of cycles where any node of G is contained in some cycle in the set. Let $L(F) = \{C_1, C_2, \ldots, C_k\}$ be a cycle cover in $G(F)$ such that the sum of the cycle lengths is the smallest, and let the length of C_i be n_i. Prove the inequation $|overlap(p, q)| \leq n_i + n_j$ for any two nodes $p \in C_i, q \in C_j$ such that $i \neq j$.

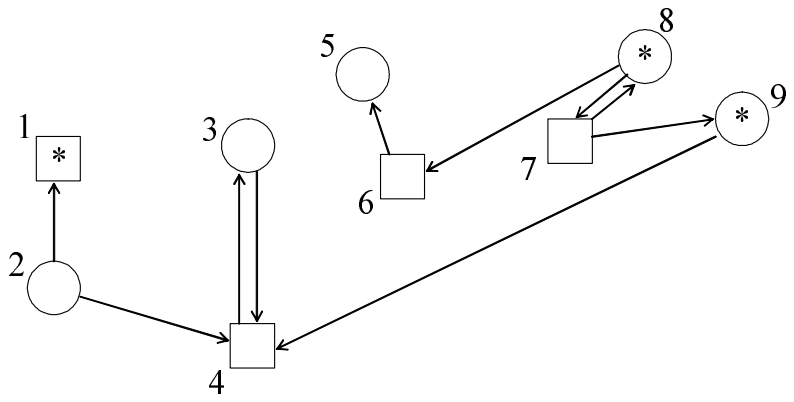
(4) For any cycle C in $G(F)$, consider a string $str(C)$ that can be obtained by the following method. (Note that $str(C)$ may not uniquely be determined.) Traverse all the nodes in the cycle from an arbitrary node, and construct a string by concatenating strings (that corresponds to the traversed nodes) in the traversal order as follows. When we concatenate two strings xy and yz, where y is the longest overlap between xy and yz, we make a new string xyz. For example, for a cycle D that consists of p =`ATGG`, q =`GGCT`, and r =`TATG`, $str(D)$ is `ATGGCTATG` if we traverse from p. Now consider the string $u = str(C_1) + str(C_2) + \ldots + str(C_k)$ that is constructed by naively concatenating k strings $str(C_i)$ $(1 \leq i \leq k)$ where $\{C_1, C_2, \ldots, C_k\} = L(F)$. Prove that u is a superstring of F and that $|u|$ is not larger than $|SSS(F)|$ multiplied by some constant.

問題 3(100 点).
以下のグラフは○と□の二人のプレーヤーの間のゲームの規則を表している．○ノードから出ている辺は○の手を表しており，その手を指すと辺の先のノードに至る．複数の辺が出ている場合は，そのどれかを選ぶことができる．□ノードも同様である．以下の問いに答えよ．

(1) プレーヤーは，次の手が打てなければ自分の負けとするとき，○が必勝戦略を持つノードをすべて求めよ．(これには，□が負けのノードも含まれる．) また，□が必勝戦略を持つノードをすべて求めよ．さらに，求めた各ノードについて簡単な説明を与えよ．

(2) (1) の条件に加えて，ある時点より*の付いていないノードのみが続く無限ゲームは○の勝ちとする．このとき，○が必勝戦略を持つノード，□が必勝戦略を持つノードをすべて求めよ．求めた各ノードについて簡単な説明を与えよ．

(3) (1) の条件に加えて，無限ゲームの勝ち負けは以下のように決まるとする．*の付いたノードが無限回現れるならば□の勝ち．そうでなければ○の勝ち．このとき，○が必勝戦略を持つノード，□が必勝戦略を持つノードをすべて求めよ．(この場合，任意のノードにおいて，○または□が必勝戦略を持つ．) 求めた各ノードについて簡単な説明を与えよ．

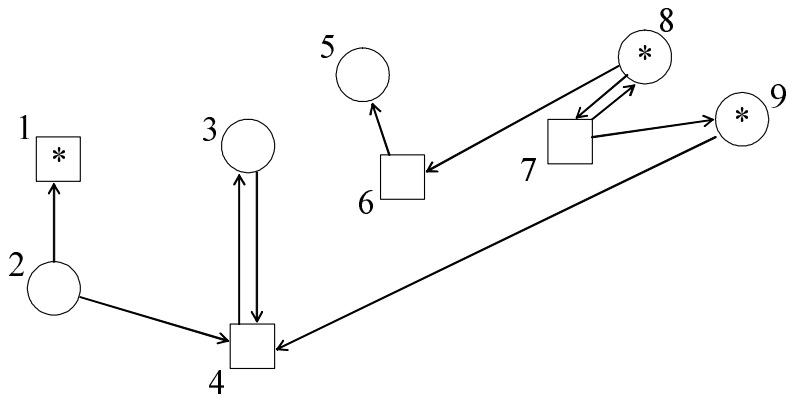


Problem 3(100 points).
The following graph expresses the rules of a game between two players ○ and □. An edge emanating from a ○ node denotes a move by the player ○, which leads to the node pointed to by the edge. If there are more than one edge from a node, the player can make a choice. Similarly, the player □ makes a move at a □ node. Answer the following questions:

(1) Assume the condition that a player loses a game if the player cannot make any move. Obtain all the nodes at which the player ○ has a winning strategy. (They include those nodes at which □ loses.) And obtain all the nodes at which □ has a winning strategy. Give also a brief explanation on each obtained node.

(2) In addition to the condition in (1), assume that ○ wins an infinite game in which only nodes that are not marked with * appear after a certain point. Obtain all the nodes at which the player ○ has a winning strategy, and all the nodes at which □ has a winning strategy. Give a brief explanation on each obtained node.

(3) In addition to the condition in (1), assume that □ wins an infinite game if an infinite number of nodes that are marked with * appear, and ○ wins otherwise. Obtain all the nodes at which the player ○ has a winning strategy, and all the nodes at which □ has a winning strategy. (In this situation, either ○ or □ has a winning strategy at any node.) Give a brief explanation on each obtained node.



問題 4(100 点).
与えられた初期多面体の各面を，ある規則に従い再帰的に分割した極限として得られる曲面を，細分割曲面という．下図の例では，図 1 で網掛けになっている 3 角形が図 2 では 4 つの細かい 3 角形に分割され，さらによりなめらかな形の多面体となるように各頂点座標が変更されている．

分割レベル $i\,(i = 0, 1, \ldots)$ の多面体上において，3 つの 3 角形面が隣接する頂点 $\boldsymbol{P}_i$ に着目し，それに隣接する 3 つの隣接点の 3 次元座標を $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$, $\boldsymbol{S}_i$ と書くことにする ($\boldsymbol{P}_i$, $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$, $\boldsymbol{S}_i$ は縦ベクトル)．ひとつ分割を進めたレベル $i+1$ の多面体において，着目している頂点とそれに隣接する 3 つの頂点の 3 次元座標を $\boldsymbol{P}_{i+1}$, $\boldsymbol{Q}_{i+1}$, $\boldsymbol{R}_{i+1}$, $\boldsymbol{S}_{i+1}$ と表すと，式 [1] が成り立つ．ただし，ここで ${\boldsymbol{P}_i}^\top$ は $\boldsymbol{P}_i$ の転置，つまり $\boldsymbol{P}_i$ を横ベクトルにしたものを表す．以下の問いに答えよ．

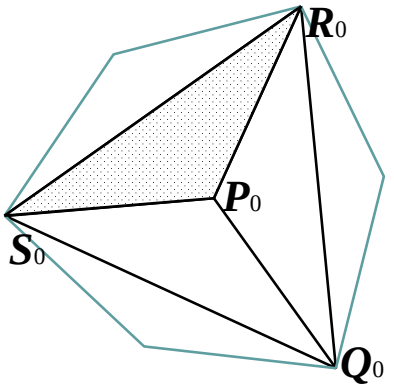


図 1: 初期多面体
(レベル 0)

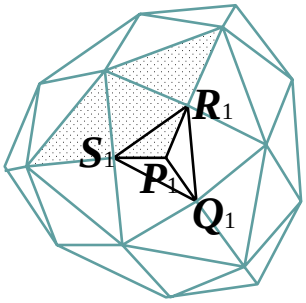


図 2: 一回分割を施した多面体
(レベル 1)

$$\begin{pmatrix} {\boldsymbol{P}_{i+1}}^\top \\ {\boldsymbol{Q}_{i+1}}^\top \\ {\boldsymbol{R}_{i+1}}^\top \\ {\boldsymbol{S}_{i+1}}^\top \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} {\boldsymbol{P}_i}^\top \\ {\boldsymbol{Q}_i}^\top \\ {\boldsymbol{R}_i}^\top \\ {\boldsymbol{S}_i}^\top \end{pmatrix} \text{ ただし } \boldsymbol{A} = \frac{1}{16} \begin{pmatrix} 7 & 3 & 3 & 3 \\ 6 & 6 & 2 & 2 \\ 6 & 2 & 6 & 2 \\ 6 & 2 & 2 & 6 \end{pmatrix} \ (i = 0, 1, \ldots) \qquad [1]$$

(1) $\boldsymbol{P}_2$ を，$\boldsymbol{P}_0$, $\boldsymbol{Q}_0$, $\boldsymbol{R}_0$, $\boldsymbol{S}_0$ を用いて表せ．

(2) 行列 $\boldsymbol{A}$ の固有値は，1, $\frac{1}{4}$, $\frac{1}{16}$ で，重複固有値を含む．このことを示せ．

(3) 行列 $\boldsymbol{A}$ は一次独立な 4 本の固有ベクトルを持つ．行列 $\boldsymbol{A}$ の固有値を λ_k $(k = 1, 2, 3, 4)$，対応する固有ベクトルを $\boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$ とすると，$\boldsymbol{A}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$ を満たす．
ベクトル $\boldsymbol{p}$ を，スカラ値 a, b, c, d を用いて，$\boldsymbol{p} = a\boldsymbol{v}_1 + b\boldsymbol{v}_2 + c\boldsymbol{v}_3 + d\boldsymbol{v}_4$ のように表現したとき，$\boldsymbol{A}^n\boldsymbol{p}$ を固有値 λ_k $(k = 1, 2, 3, 4)$ と固有ベクトル $\boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$ を用いて表せ．

(4) $\boldsymbol{P}_i$ は，分割を繰り返すとある座標に収束する．その理由を記せ．

(5) 分割を進めた際に，$\boldsymbol{P}_i$ が最終的に収束する点 $\lim_{i\to\infty} \boldsymbol{P}_i$ を，$\boldsymbol{P}_0$, $\boldsymbol{Q}_0$, $\boldsymbol{R}_0$, $\boldsymbol{S}_0$ を用いて表せ．

Problem 4(100 points).
A subdivision surface is a limit obtained by recursively refining each face of an initial polygon. In the example of the following figures, a shaded triangle of Fig. 1 is split into four small triangles in Fig. 2, where the corresponding vertices are smoothed out to curve the associated surface shape.

Suppose, at the refinement level $i\,(i = 0, 1, \ldots)$, we have a vertex $\boldsymbol{P}_i$ that has three adjacent vertices $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$, and $\boldsymbol{S}_i$, where $\boldsymbol{P}_i$, $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$, and $\boldsymbol{S}_i$ are column vectors. Let the vertex corresponding to $\boldsymbol{P}_i$ at the level $i+1$ be $\boldsymbol{P}_{i+1}$, and three adjacent vertices be $\boldsymbol{Q}_{i+1}$, $\boldsymbol{R}_{i+1}$, and $\boldsymbol{S}_{i+1}$, then they satisfy the equation [1]. Here ${\boldsymbol{P}_i}^\top$ represents the transpose of the column vector $\boldsymbol{P}_i$, i.e. the row vector for the coordinate of the target vertex. Answer the following questions.

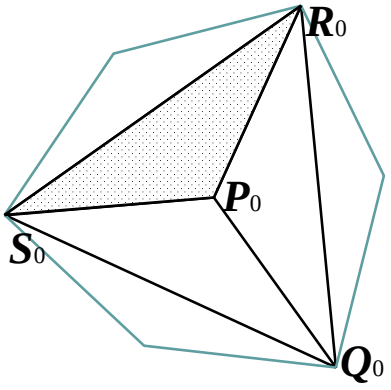


Fig. 1: Initial polygon (Level 0)

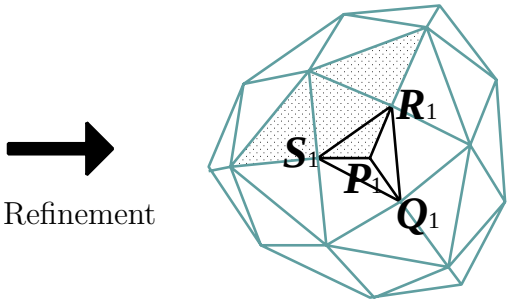


Fig. 2: The refined polygon (Level 1)

$$\begin{pmatrix} {\boldsymbol{P}_{i+1}}^\top \\ {\boldsymbol{Q}_{i+1}}^\top \\ {\boldsymbol{R}_{i+1}}^\top \\ {\boldsymbol{S}_{i+1}}^\top \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} {\boldsymbol{P}_i}^\top \\ {\boldsymbol{Q}_i}^\top \\ {\boldsymbol{R}_i}^\top \\ {\boldsymbol{S}_i}^\top \end{pmatrix} \text{ where } \boldsymbol{A} = \frac{1}{16} \begin{pmatrix} 7 & 3 & 3 & 3 \\ 6 & 6 & 2 & 2 \\ 6 & 2 & 6 & 2 \\ 6 & 2 & 2 & 6 \end{pmatrix} (i = 0, 1, \ldots) \qquad [1]$$

(1) Write $\boldsymbol{P}_2$ in terms of $\boldsymbol{P}_0$, $\boldsymbol{Q}_0$, $\boldsymbol{R}_0$, and $\boldsymbol{S}_0$.

(2) The eigenvalues of the matrix $\boldsymbol{A}$ are 1, $\frac{1}{4}$, and $\frac{1}{16}$ with duplicated eigenvalues. Verify this.

(3) The matrix $\boldsymbol{A}$ has four linearly independent eigenvectors. Letting the eigenvalues of the matrix $\boldsymbol{A}$ be λ_k $(k = 1, 2, 3, 4)$ and their corresponding eigenvectors be $\boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$, we have $\boldsymbol{A}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$.

Suppose that the vector $\boldsymbol{p}$ is represented by $\boldsymbol{p} = a\boldsymbol{v}_1 + b\boldsymbol{v}_2 + c\boldsymbol{v}_3 + d\boldsymbol{v}_4$ using some scalar values a, b, c, and d. Express $\boldsymbol{A}^n\boldsymbol{p}$ in terms of the eigenvalues λ_k $(k = 1, 2, 3, 4)$ and the eigenvectors $\boldsymbol{v}_k$ $(k = 1, 2, 3, 4)$.

(4) The vertex $\boldsymbol{P}_i$ converges to some point if we infinitely refine the initial polygon. Explain why.

(5) Consider the point $\lim_{i\to\infty} \boldsymbol{P}_i$ to which the vertex $\boldsymbol{P}_i$ converges if we infinitely refine the initial polygon. Express $\lim_{i\to\infty} \boldsymbol{P}_i$ in terms of $\boldsymbol{P}_0$, $\boldsymbol{Q}_0$, $\boldsymbol{R}_0$, and $\boldsymbol{S}_0$.

問題 5(100 点).
主記憶中のページフレーム $m(\geq 1)$ 個を使ってデマンドページングするシステムを考える．以下の問いに答えよ．

(1) 最初，主記憶中にページは存在せず，以下のページ番号参照列を処理すると仮定する．

```
0 2 3 1 0 2 4 0 2 3 1 4
```

ページ置き換えアルゴリズムとして LRU(Least Recently Used) を用いた場合，ページフレーム数 m に対するページフォールト回数 $F(m)$ の値を求めよ．

(2) 問い (1) と同じ条件下において，FIFO 置き換えアルゴリズムが，あるページフレーム数 m に対し，$F(m) < F(m+1)$ となることを示せ．

(3) オペレーティングシステムによるデマンドページングおよび様々なページ置き換えアルゴリズム実装のためのアーキテクチャ支援機構を述べよ．

(4) オペレーティングシステムは，問い (3) で答えた機構をどのように使って，疑似的に LRU アルゴリズムを実装しているか述べよ．

Problem 5(100 points).
Consider a demand paging system which has $m(\geq 1)$ page frames in the main memory. Answer the following questions.

(1) Assume that no pages are allocated at the initialization, and the following page reference sequence is processed:

```
0 2 3 1 0 2 4 0 2 3 1 4
```

In case of using the LRU (Least Recently Used) page replacement algorithm, give the number of page faults $F(m)$ for m page frames.

(2) Show that for some m the expression $F(m) < F(m+1)$ is satisfied under the same condition in the question (1) except that the FIFO replacement algorithm is used.

(3) Describe an architectural support mechanism for implementing demand paging and various page replacement algorithms in an operating system.

(4) Describe how an operating system approximately implements the LRU algorithm using the mechanism answered in the question (3).