

平成 28 年 度

名古屋大学大学院情報科学研究科
社会システム情報学専攻
入 学 試 験 問 題

専 門

平成 27 年 8 月 6 日 (木)
12:30~15:30

注 意 事 項

1. 試験開始の合図があるまでは、この問題冊子を開いてはならない。
2. 試験終了まで退出できない。
3. 外国人留学生は、日本語から母語への辞書 1 冊に限り使用してよい。
電子辞書の持ち込みは認めない。
4. 問題冊子、解答用紙 4 枚、(小論文用和英 2 枚を含む) 草稿用紙 3 枚が
配布されていることを確認すること。
5. 問題は「小論文」、「確率・統計」、「プログラミング」、「データ工学」、
「知識工学」、「電子社会システム」、「情報デザイン」、「哲学の基礎」、
「論理的思考」、「情報学と社会」の 10 科目がある。小論文は必ず解答する
こと。また、残りの 9 科目から 2 科目を選択して 解答すること。
なお、選択した科目名を解答用紙の指定欄に記入すること。
6. 全ての解答用紙の所定の欄に受験番号を必ず記入すること。
解答用紙に受験者の氏名を記入してはならない。
7. 解答用紙に書ききれない場合は、裏面を使用してもよい。
ただし、裏面を使用した場合は、その旨を解答用紙表面右下に明記すること。
8. 解答用紙は試験終了後に 3 枚とも提出すること。
9. 問題冊子、草稿用紙、小論文で使用しなかった方の解答用紙は、
試験終了後に持ち帰ること。

小論文

問題

「情報技術と倫理 (IT and ethics)」について具体例を挙げながら知るところを述べ、それに関するあなたの考えを 800 字～1000 字で書きなさい。英語で解答する場合は 300～400 ワードで書きなさい。

*なお解答は、日本語 (和文)・英語 (英文)、どちらか一方の言語を選び、その言語用の解答用紙 (和文はマス目のあるもの、英文は罫線のみなもの) に記入し、その解答用紙のみを提出しなさい。

確率・統計

解の導出過程も書くこと。

[1] じゃんけん(ルールは下記 ※を参照)で3人の順位(1~3位)を決めたい。但し、じゃんけんの各回の試行は独立で、3人とも3種類の手を等確率で出すものとする。これについて、以下の問いに答えよ。

- (1) 1回目のじゃんけんでは、3人があいことなる確率を求めよ。
- (2) 2回目のじゃんけんでは、3人の順位が全て決まる確率を求めよ。
- (3) n 回目のじゃんけんでは、はじめて1位が決まる確率を求めよ。但し、 n は正整数とする。

※じゃんけんとは、複数の人が同時にグー・チョキ・パーの3種類の手いずれかを出すことで、勝敗を決める手段である。グーはチョキに勝ち、チョキはパーに勝ち、パーはグーに勝ち。全員が同じ手を出した場合や、3人が異なる手を出した場合は、あいことする。まず、3人のじゃんけんでは1人が勝った場合は、勝った人が1位となり、その後負けた2人でじゃんけんをして、その勝敗で2位と3位を決める。一方、3人のじゃんけんでは1人が負けた場合は、負けた人が3位となり、その後勝った2人でじゃんけんをして、その勝敗で1位と2位を決める。また、あいことなった場合は、勝敗が決まるまで同じメンバーでじゃんけんを繰り返す。

[2] 互いに独立な確率変数 X, Y について、以下の問いに答えよ。

- (1) X, Y の期待値と分散が、それぞれ $E(X) = 2, V(X) = 1, E(Y) = 5, V(Y) = 9$ で与えられるとき、確率変数 $W = (X - 2Y)^2$ の期待値 $E(W)$ を求めよ。
- (2) X, Y がともに区間 $[1, 2]$ における連続一様分布に従うとき、確率変数 $Z = \max\{X, Y\}$ の確率密度関数 $f_Z(z)$ を求めよ。

[3] 確率変数 X, Y の同時確率密度関数 $f_{X,Y}(x, y)$ が次式で与えられている。但し、 c は定数とする。これについて、以下の問いに答えよ。

$$f_{X,Y}(x, y) = \begin{cases} ce^{-x-y}, & 0 \leq x \leq y \\ 0, & \text{その他} \end{cases}$$

- (1) c の値を求めよ。
- (2) Y の周辺確率密度関数 $f_Y(y)$ を求めよ。
- (3) X と Y が独立であるか否かを、理由とともに答えよ。

[4] ある町で収穫されたリンゴの重さの母平均 μ [g] を推定したい。但し、母標準偏差は 50 [g] であるとわかっている。また、標準正規分布を $f(x)$ としたとき、

$$\int_{-1.96}^{1.96} f(x) dx = 0.95$$

とする。これについて、以下の問いに答えよ。

- (1) 無作為に抽出した 100 個の標本の重さの平均は 350 [g] であった。 μ の 95% の信頼区間を示せ。
- (2) 95% の信頼区間の幅が 7 [g] 以下になるように μ を推定するには、何個以上の標本を抽出する必要があるか答えよ。

Translation of technical terms

じゃんけん rock paper scissors, 試行 trial, 独立 independence, 手 hand gesture, 等確率 equal probability, あいこ draw, 確率 probability, 正整数 positive integer, グー rock, チョキ scissors, パー paper, 確率変数 random variable, 期待値 expectation, 分散 variance, 区間 interval, 連続一様分布 continuous uniform distribution, 確率密度関数 probability density function, 同時確率密度関数 joint probability density function, 定数 constant, 周辺確率密度関数 marginal probability density function, 母平均 population mean, 推定 estimation, 母標準偏差 population standard deviation, 標準正規分布 standard normal distribution, 無作為に randomly, 抽出 sampling, 標本 sample, 信頼区間 confidence interval

プログラミング

2分木は各頂点が高々2つの頂点を子に持つ根付き木である。2分木の各頂点はデータを保持し、その子は左右で区別される。以下では各頂点がデータとして整数を保持する2分木を考える。図1に2分木の例を示す。



図 1: 2分木の例

C言語プログラムでは以下に定義される構造体により2分木を表現することができる。

```
struct vertex {
    int value;
    struct vertex *left;
    struct vertex *right;
};
```

この構造体は1つの頂点を表現し、その頂点が保持する整数を格納するメンバ value, 左の子へのポインタを格納するメンバ left, 右の子へのポインタを格納するメンバ right から構成される。この構造体で表現された頂点の左の子, 右の子が存在しない場合はメンバ left, right にそれぞれ値 NULL を代入する。

ソースコード1は上述の構造体を利用して表現された2分木を取り扱うC言語プログラムである。大域変数 tree は2分木を参照するポインタ変数であり、宣言時の初期値を NULL とする。すなわち、頂点を持たない空の2分木を NULL で表す。insert 関数, eliminate 関数はポインタ tree が指す2分木に対して引数で与えられた整数を保持する頂点をそれぞれ挿入, 削除する関数である。create 関数は引数で与えられた整数を保持する頂点を作成する関数である。display 関数は引数で与えられたポインタが指す2分木(もしくは2分部分木)に対して頂点が保持する整数を先行順で出力する関数であり、再帰的に定義されている。

ソースコード1について以下の問いに答えよ。

[1] main 関数を18行目の末尾まで実行した時点, ならびに19行目の末尾まで実行した時点におけるポインタ tree が指す2分木を図1の記法に従いそれぞれ図示せよ。

[2] 21行目の末尾まで実行した時点において tree が指す2分木が図1(b)に示される2分木の根を参照するように21行目にデータの挿入・削除の操作を記述したい。21行目の (ア) に書くべき命令の列を1つ示せ。ただし、命令の列は以下の条件を満たすものとする。

- 命令は insert(x); または eliminate(y); のみである。ただし、x および y は整数である。
- insert 関数や eliminate 関数の引数に与えられる整数には同じものは高々1回しか現れない。

- [3] 三項演算子 $?:$ は「式1?式2:式3」のように記述されて式を構成する。「式1」を評価した結果が0でないときは「式2」を、0であるときは「式3」を評価した結果を式の値とする。例えば、式 $x \leq 0 ? x+1 : x+2$ を評価した値は x の値が -1 のときには 0 となり、 5 のときは 7 となる。この三項演算子を用いて、53行目から56行目のコード

```
if( x < p->value )
    p = p->left;
else
    p = p->right;
```

と同様の動作をする代入文を以下の を埋めて完成させよ。

```
p =  ;
```

- [4] ソースコード1の `eliminate` 関数の定義にはメモリへの不正アクセスを発生させる可能性がある。例えば、19行目と次の行の間で `eliminate(10);` を実行した場合に不正アクセスが発生する。`eliminate` 関数の定義の1箇所命令を追加してその問題を解決せよ。なお、解答は「〇〇行目と次の行の間に△△を挿入」という形式で記すこと。
- [5] ソースコード1の `eliminate` 関数では不要になったメモリを解放していない。このままで挿入・削除の実行を繰り返した場合にどのような問題が起こりうるかを説明せよ。さらに、不要になったメモリを解放するように、`eliminate` 関数の定義の中の2箇所にそれぞれ命令を追加せよ。解答は「〇〇行目と次の行の間に△△を挿入」という形式で記せ。また、メモリの解放には標準ライブラリ関数である `free` 関数を使用すること。
- [6] 21行目の末尾まで実行した時点でポインタ `tree` が指す2分木は図1(b)になっている。このあと、22行目を実行した際に出力される文字列を示せ。
- [7] `display` 関数を実行した際に数が大きい順に出力されるようその定義を変更したい。以下の関数定義の中の ~ を埋めて変更せよ。頂点に保持される整数を出力する場合には `printf("%d,", p->value)` を記述すること。

```
void display(struct vertex *p){
    if( p == NULL ) return;
     ;
     ;
     ;
    return;
}
```

- [8] 2分木に指定した数が存在しているかどうかを探索する `member` 関数を以下の要件を満たすように作成したい。
- 引数として整数 x を受け取る。
 - ポインタ `tree` が指す2分木に x を保持する頂点が存在する場合には 1 を、そうでない場合には 0 を返す。

これらの条件を満たすように以下の関数定義の中の (カ) ~ (ケ) を埋めて member 関数の定義を完成させよ.

```

int member(int x){
    struct vertex *p;
    p = tree;
    while( p != NULL ){
        if( [ (カ) ] ) return 1;
        if( [ (キ) ] )
            [ (ク) ];
        else
            [ (ケ) ];
    }
    return 0;
}

```

Translation of technical terms

2分木	binary tree	関数	function
頂点	vertex	部分木	subtree
子	child	先行順	preorder
根付き木	rooted tree	出力する	print out
C言語	C programming language	再帰的に	recursively
構造体	structure	操作	operation
メンバ	member	命令	statement
ソースコード	source code	列	sequence
プログラム	program	三項演算子	ternary operator
大域変数	global variable	式	expression
格納する	store	代入文	assignment statement
参照する	refer	メモリ	memory
ポインタ	pointer	不正アクセス	illegal access
変数	variable	発生させる	raise
宣言	declaration	実行する	execute
初期値	initial value	解放する	free
空	empty	標準ライブラリ関数	standard library function
引数	argument	文字列	string
挿入する	insert	関数定義	function definition
削除する	delete	探索する	search

ソースコード 1: 2分木を処理する C 言語プログラム

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct vertex {
5     int value;
6     struct vertex *left;
7     struct vertex *right;
8 };
9
10 struct vertex *tree = NULL;
11
12 struct vertex *create(int x);
13 void insert(int x);
14 void eliminate(int x);
15 void display(struct vertex *p);
16
17 int main(){
18     insert(3); insert(12); insert(18); insert(11); insert(14);
19     eliminate(12); insert(2); insert(3);
20     insert(6); eliminate(3); insert(5);
21     (ア)
22     display(tree);
23     return 0;
24 }
25
26 struct vertex *create(int x){
27     struct vertex *p;
28     p = (struct vertex *) malloc(sizeof(struct vertex));
29     p->value = x;
30     p->left = NULL;
31     p->right = NULL;
32     return p;
33 }
34
35 void insert(int x){
36     struct vertex *p;
37
38     if( tree == NULL ){
39         tree = create(x);
40         return;
41     }
42
43     p = tree;
44     do{
45         if( p->value == x ) break;
46         else if( x < p->value && p->left == NULL ){
47             p->left = create(x);
48             break;
49         }else if( p->value < x && p->right == NULL ){
50             p->right = create(x);
51             break;
52         }
53         if( x < p->value )
54             p = p->left;
55         else

```

```

56     p = p->right;
57 }while(1);
58 return;
59 }
60
61 void eliminate(int x){
62     struct vertex *f, *p, *q;
63
64     p = tree;
65     if( p == NULL ) return;
66     do {
67         f = p;
68         if( x < p->value )
69             p = p->left;
70         else if( p->value < x )
71             p = p->right;
72     }while( x != p->value );
73
74     if( p->left == NULL || p->right == NULL ){
75         if( p->right == NULL )
76             q = p->left;
77         else
78             q = p->right;
79         if( p == tree )
80             tree = q;
81         else{
82             if( f->left == p )
83                 f->left = q;
84             else
85                 f->right = q;
86         }
87     }else{
88         q = p->right;
89         f = q;
90         while( q->left != NULL ){
91             f = q;
92             q = q->left;
93         }
94         p->value = q->value;
95         if( q == f )
96             p->right = q->right;
97         else
98             f->left = q->right;
99     }
100     return;
101 }
102
103 void display(struct vertex *p){
104     if( p == NULL ) return;
105     printf("%d,", p->value );
106     display(p->left);
107     display(p->right);
108     return;
109 }

```


データ工学

以下の設問にすべて答えなさい。

問 1 以下のデータベーススキーマ (database schema) を考える。左側に各リレーションスキーマ (relation schema) を、右側にそれぞれのタプル (tuple) の例を示す。下線は主キー (primary key) を示す。

映画 (<u>映画 ID</u> , 映画名, 年, 監督 ID)	(M53, ローマの休日, 1953, D12)
監督 (<u>監督 ID</u> , 監督名)	(D12, ウィリアム・ワイラー)
俳優 (<u>俳優 ID</u> , 俳優名, 性別, 国籍)	(A25, オードリー・ヘプバーン, 女, イギリス)
出演 (<u>俳優 ID</u> , <u>映画 ID</u> , 役割)	(A25, M53, アン王女)
上映 (<u>映画 ID</u> , <u>映画館名</u> , 入場料)	(M53, 名大シアター, 1000)

このデータベースに対し、以下の問合せをリレーショナル代数 (relational algebra) で記述しなさい。

- (1) ウィリアム・ワイラー (William Wyler) が監督している映画に出演している女優の名を求める問合せ
- (2) 監督した映画がすべて 1960 年より前の作品であるような監督の監督名を求める問合せ

問 2 問 1 のデータベースに対し、以下の問合せを SQL で記述しなさい。

- (1) オードリー・ヘプバーン (Audrey Hepburn) と共演した (同じ映画に出演した) 俳優の俳優名を求める問合せ (注: オードリー・ヘプバーン自身の名も出力されてよいとする)
- (2) オードリー・ヘプバーンが出演している映画を最も安い入場料で見ることができる映画館の映画館名を求める問合せ

問 3 リレーション $R(A, B, C, D, E, F)$ について、

$$A, B \rightarrow E$$

$$B, C \rightarrow F$$

$$C \rightarrow D$$

$$D \rightarrow A$$

$$F \rightarrow C$$

という関数従属性 (functional dependency) が成立するとする。

- (1) リレーション R の候補キー (candidate key) をすべて求めなさい。
- (2) リレーション R を $R_1(A, B, C, E)$ と $R_2(B, C, D, F)$ に分解したとする。この分解は無損失結合分解 (lossless join decomposition) であるか、理由をつけて答えなさい。
- (3) リレーション R を第三正規形 (third normal form) に分解しなさい。
- (4) (3) の結果において、第三正規形であるがボイス・コッド正規形 (Boyce-Codd normal form) ではないリレーションをボイス・コッド正規形に分解しなさい。

問4 B+ 木は、DBMS におけるファイル編成において広く用いられている手法である。B+ 木の (a) ハッシュファイルに対する利点、および、(b) 索引付きファイルに対する利点をそれぞれ述べなさい。

問5 同時実行制御に関して、以下のスケジュールを考える。 $R_i(X)$ はトランザクション T_i による項目 X の read, $W_i(X)$ はトランザクション T_i による項目 X の write, C_i はトランザクション T_i の commit を表す。

$R_1(A) R_1(B) W_1(B) C_1 R_3(C) R_3(B) W_3(B) C_3 W_2(A) W_2(C) C_2$

このスケジュールについて、以下の各性質が成り立つか否かを理由を付して答えなさい。

- (1) 競合直列可能性 (conflict serializability)
- (2) ビュー直列可能性 (view serializability)

知識工学

ある高速鉄道沿いのある地点において、平日の 8 時から 12 時までの時間帯に特急(super express) ω_1 , 急行(express) ω_2 , 快速(rapid) ω_3 の 3 種類の列車が通過する本数はそれぞれ 40, 22, 10 であり、休日の同時間帯に同地点を通過する本数はそれぞれ 36, 23, 13 であるとする。また、この地点で特急、急行、快速の通過速度 (V) を計測して、表 1 のような分布 (distribution) を得たとする。通過速度①～⑦を観測するという事象 (event) を x_1, x_2, \dots, x_7 とするとき、以下の問いに答えよ。なお、求める確率は分数で表してもよい。

表 1 特急, 急行, 快速の通過速度(km/h)の観測データ

通過速度(V)の範囲	特急	急行	快速
① $V \geq 265$	24%	0%	0%
② $255 \leq V < 265$	32%	5%	0%
③ $245 \leq V < 255$	31%	50%	4%
④ $235 \leq V < 245$	12%	33%	37%
⑤ $225 \leq V < 235$	1%	11%	33%
⑥ $215 \leq V < 225$	0%	1%	18%
⑦ $V < 215$	0%	0%	8%

- 問1 ベイズの定理 (Bayes' theorem) にしたがって、事前確率 (prior probability) $P(\omega_i)$, 周辺確率 (marginal probability) $p(x_k)$, 尤度 (likelihood) $p(x_k|\omega_i)$ を用いて事後確率 (posterior probability) $P(\omega_i|x_k)$ を表せ(ただし, $i=1, 2, 3; k=1, 2, \dots, 7$)。
- 問2 表1に示す観測データから、特急 ω_1 , 急行 ω_2 , 快速 ω_3 が x_3, x_4, x_5 の速度で通過する確率 $p(x_k|\omega_i)$ ($i=1, 2, 3; k=3, 4, 5$) をそれぞれ求めよ。
- 問3 以上の結果にもとづいて、平日の 8 時から 12 時までの時間帯において、 x_3, x_4, x_5 の各速度で通過する電車は、特急、急行、快速のいずれと考えるのが最も誤り (error) が少ないかをそれぞれ答えよ。計算過程を示せ。
- 問4 同様に、休日の 8 時から 12 時までの時間帯において、 x_3 の速度で通過する電車は、特急、急行、快速のいずれと考えるのが最も誤りが少ないかを答えよ。なお、そのときの誤り確率 (error probability) を求めよ。計算過程を示せ。

電子社会システム

以下の問題を2問とも解答しなさい。

- (1) 以下の、情報通信技術 (ICT: Information and Communication Technology) 及びCG (Computer Graphics) に関連する用語について、すべてを簡潔に説明しなさい。
1. データサイエンティスト (data scientist)
 2. WebGL
 3. 共通語彙基盤 (IMI: Infrastructure for Multilayer Interoperability)
 4. マルチレゾリューションシェーディング (multi-resolution shading)
 5. 公共クラウドシステム (Public Cloud System)
- (2) オープンイノベーション (open innovation) は、様々な組織が持つ技術やアイデアを組み合わせ、革新的な研究成果やビジネスモデルにつなげるイノベーションの方法論である。この方法論は昨今の自治体における情報化推進の現場においても注目されている。
- 自治体が行うオープンイノベーションについて、独創的かつ具体的な提案 (an original and concrete proposal) を一つ考案し説明しなさい。

情報デザイン

次の二つの問 A か B のどちらか一方に答えなさい。A, B にまたがって解答した答案は採点の対象としません。

- 問 A
- (1) ウェブにおけるユーザビリティ (usability) とユーザ・エクスペリエンス (user experience) を簡潔に定義し、両者の関係について論じなさい。
 - (2) リチャード・ソール・ワーマン (Richard Saul Wurman) が、情報の組織化の基準として提案している技術について解説しなさい。

- 問 B
- (1) ヴィジュアル・リテラシー (visual literacy) について、ポール・メッサリス (Paul Messaris) が挙げる、広く支持されている見解について説明しなさい。
 - (2) あなた自身の研究領域における、ヴィジュアル・リテラシーの必要性について、具体的に論じなさい。

哲学の基礎

次の事項の内から5つを選んで説明しなさい。6つ以上解答した場合は、採点対象としません。

[美学]

1. 「影像 (eidolon)」としての芸術
2. 詩と歴史の違い
3. 触覚 (tactile, haptic) の美学

[社会哲学]

4. 社会契約説における「自然状態」 (“State of nature” in social contract theory)
5. トクヴィルの自発結社観 (Tocqueville’s view of association)
6. コントの「三段階の法則」 (Comte’s “Law of three stages”)

[科学哲学]

7. 斉一性原理 (The Uniformity Principle)
8. 利用可能性バイアス (availability bias)
9. 反証主義 (falsificationism)

[倫理学]

10. 倫理的相対主義 (ethical relativism)
11. 正義 (justice)
12. 徳倫理 (virtue ethics)

[哲学史]

13. アリストテレス哲学における「第一質料」 (“Prime matter” in Aristotelian philosophy)
14. モンテーニュの『随想録』 (*Essays of Michel de Montaigne*)
15. カント哲学における「物自体」 (“Thing-in-itself” in Kantian philosophy)

論理的思考

次の問題群 A、B いずれかを選び、そこに含まれるすべての問題に解答しなさい。A、B にまたがって解答した答案は、採点の対象としません。

問題群 A : Formal Logic

問 1 次の概念のそれぞれについて、100 字程度で解説せよ。

- (1) シンタクスとセマンティクス (syntax and semantics)
- (2) 推移的な関係 (transitive relation)
- (3) 第 2 階の述語論理 (second-order predicate logic)
- (4) ファジーな述語 (fuzzy predicate)

問 2 古典命題論理の自然演繹 (natural deduction) の体系に # という新しい結合子 (connective) を加え、それについての次の規則を加える。

- 構文規則：任意の論理式 A, B に対して $(A\#B)$ は論理式。
- 推論規則：
 - 左導入則： A を仮定して矛盾が導かれたら $A\#B$ を推論してよい。このとき仮定 A は落とされる。
 - 右導入則： B を仮定して矛盾が導かれたら $A\#B$ を推論してよい。このとき仮定 B は落とされる。
 - 除去則： $A\#B, A, B$ から矛盾を推論してよい

この体系を $N_{\#}$ と呼ぶ。この時、以下の問いに答えよ。

- (1) $A\#B \vdash_{N_{\#}} \neg A \vee \neg B$ および $\neg A \vee \neg B \vdash_{N_{\#}} A\#B$ が成り立つことを、自然演繹の証明図を書いて示せ。証明図の表記は Fitch スタイルまたは Gentzen スタイルのどちらかとする。
- (2) (1) を踏まえて $A\#B$ がどのような真理条件を持つかを説明せよ。ただし # を加えて拡張した体系が古典命題論理の保存的拡大 (conservative extension) であること、自然演繹が古典命題論理に対して健全 (sound) かつ完全 (complete) であること、および通常の結合子の真理条件については前提してよい。

問 3 H と R はともに 2 項述語であるとする。「 H が R のハブを持つ」のは Rxy を満たすすべての対象 x, y に対して、ある対象 z が存在して Hzx と Hzy が成り立つ時であると

する。また「 H が R のグローバルなハブを持つ」のはある対象 z が存在して、 Rxy を満たすすべての対象 x, y に対して Hzx と Hzy が成り立つ時であるとする。

この時、以下の問いに答えよ。

- (1) 「 H が R のハブを持つ」、「 H が R のグローバルなハブを持つ」をそれぞれ第1階の述語論理の適切な論理式に翻訳しなさい。
- (2) H が R のグローバルなハブを持つならば H が R のハブを持つことをタブロー (tableau) によって示しなさい。

問題群 B：クリティカル・シンキング

問1 クロード君は新米の文化人類学者である。彼はある集落にフィールドワークに出かけた。人口約 1000 人のこの集落は孤立しており、外部との交渉がまったくない。長い間自給自足の生活を営んできた。フィールドワークを続けるうちに、クロード君は、この集落にはある種の贈り物儀礼があることに気づいた。成人の間で、主として冬に行われるその儀礼は、現地の言葉で「セイボ」と呼ばれている。贈り物としてやりとりされる品物も「セイボ」と呼ばれ、種類は多種多様であるが、どれもごくささやかなものである。クロード君はこの発見に喜び、セイボがどのような規則のもとで営まれているかをさらに観察し、次のような規範を見いだしたとして、学会で報告した。

【セイボが従う規範】

- (1) すべての成人は必ず誰かにセイボを贈るべし。誰にもセイボを贈らない成人は礼儀を欠く者として集落から排除される。
- (2) ある者からおまえにセイボが贈られたとき、その者に直接セイボを贈り返してはならない。
- (3) おまえがおまえ自身にセイボを贈るという馬鹿なまねは決してしてはならない。
- (4) おまえがセイボを贈った相手がさらに幾人かの者にセイボを贈ったなら、おまえはそれらの者にもセイボを贈らねばならない。

クロード君の発表が終わるやいなや、学会の重鎮であるマルセル教授が立ち上がり、次のように発言した。

クロード君、君が見いだしたと主張する規範のどれかは絶対に間違いだ。その集落を訪れたことがない私にどうしてそんなことが断言できるのかと言いたいだろう。私の観察と君の観察のどちらが正確か、という問題ではないのだよ。その集落が、君の主張する4つの規範すべてに従うことは論理的に不可能なのだ。

クロード君は、何が批判されているのか分からず呆然としてしまった。さてそこで、マルセル教授に代わって、なぜ下線部が成り立つのかを分かりやすく解説しなさい。

問2 ある飲料メーカーが開発中の製品 A、B、C、D について、8 人のモニターにアンケート調査を行い、好きな製品と嫌いな製品を挙げてもらった。しかしその結果のデータからモニターの職業についての情報が誤って削除されてしまった。以下がアンケートの結果である。

	性別	年齢	好き	嫌い	職業
回答者 a	男性	20 代	A	C	?
回答者 b	男性	20 代	B, C	A	?
回答者 c	男性	30 代	A, C	B	?
回答者 d	男性	30 代	D	A, B	?
回答者 e	女性	20 代	A	C	?
回答者 f	女性	20 代	B, D	C	?
回答者 g	女性	30 代	B	A	?
回答者 h	女性	30 代	A, C	D	?

アンケート係は記憶を振り絞ってどうにか以下のことを思い出した。

1. 回答者はすべて会社員、自営業、医者、公務員のいずれかだった。これらカテゴリーの複数に属する回答者はおらず、またどのカテゴリーにも少なくとも一人の回答者が属していた。
2. 医者の過半数は C が嫌いと答えた。
3. 医者はみな A が好きと答えた。
4. 男性の会社員がいた。
5. D を嫌いと答えた会社員はいなかった。
6. 自営業は全員、男性だった。
7. 会社員はすべて女性であるか 30 代であるかのどちらかだった。
8. B が好きと答えた公務員も C が好きと答えた公務員もいなかった。

このとき回答者 c の職業は何かを答え、その結論に至った推論の過程を丁寧かつ分かりやすく説明しなさい。

問 3 以下の議論を読み、問いに応えなさい。

現在の日本の国会には衆議院と参議院の二つの議会があり、衆議院での多数党が与党になる。新しい法律が制定されるためには原則としてこの両院での可決が必要である。しかし私はこの国会の二院制に反対である。二院制があるために国家にとって重大な法案の審議に時間がかかってしまうことがよくあるからである。特に衆議院と参議院の多数派が異なっている「ねじれ国会」の状態になっているときにはなおさらである。このような遅滞を招いている二院制は廃止した方がよい。そうすれば国民の声の代表者である政権与党による法案の審議・可決がより円滑に速やかに進むだろう。

- (1) この議論において筆者が一番主張したいことは何か。
- (2) 筆者が (1) の主張の根拠としているのは何か。

- (3) この議論において (1) の主張を結論する際に暗黙に前提されていることは何か。最も重要と思われることを一つ答えよ。
- (4) (3) の答えに基づいて、この議論に対する反論を書きなさい。

情報学と社会

次の A, B, C の内の一つだけを選んで答えなさい。二つ以上解答した場合は、採点対象としません。

A. [情報の編集]

Wiki システムを情報の編集と関わらせて論文や書物の執筆に利用する手立てについて述べなさい。

B. [情報学と倫理]

次の二つの問の両方に答えなさい。

問 1 アジェンダセッティングについて説明し、その意義と問題点について例をあげて説明しなさい。

問 2 記号の消費について、例をあげて説明しなさい。

C. [情報学と芸術]

次の三つの事柄について、説明しなさい。

1. 身体感覚と「自分ごと」
2. 加算系造形方式と減算系造形方式
3. 時間軸のデザイン