名古屋工業大学

２０２４年度（令和６年度）

編入学者・転入学者選抜学力検査[問題]

― 専 門 試 験 ―

（情報工学科）

試 験 日 時　２０２３年6月２３日（金）

１０：００～１２：００

●解答上の注意

(1) 解答の際，解答用紙のホチキス止めを外してください。
(2) 配布物は，問題冊子１冊，解答用紙３枚，計算用紙１枚です。
(3) 解答は各問題番号に対応する解答用紙に解答してください。
(4) 解答が解答用紙表面に書ききれない場合は，裏面に続いてもよいが，その場合は表面の下側が裏面の上側になるようにし，上側２／３のスペースに解答を収めてください。
(5) 電卓は使用できません。
(6) 試験終了後は問題用紙と計算用紙を持ち帰ってください。

問題 1　設問すべてについて解答すること。ただし，回路を図示する場合には，必要に応じて図 1-1 に示す論理記号を用いること。また，論理変数 V の否定を $\bar{V}$ で表すものとする。

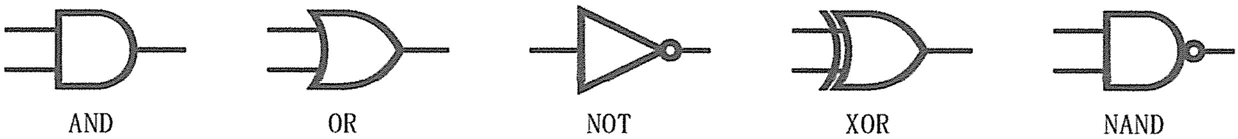

図 1-1

I　次の (1)～(3) の問いについて答えよ。

(1)　2 の補数で表現された 8 ビットの 2 進数 $A = 10010100$ を 10 進数に変換せよ。

(2)　4 桁の符号無し **4 進数** $B = 1302$ を 10 進数に変換せよ。

(3)　B に足すと演算結果が0となる 4 桁の符号無し **4 進数**を求めよ。ただし，演算結果が 5 桁になる場合，5 桁目は無視するものとする。

II　表 1-1 に示した論理関数 $X = F(A, B, C, D)$ の真理値表について，次の (1)～(3) の問いについて答えよ。なお，表 1-1 において＊はドントケアを表す。解答においてもドントケアは＊で表すこと。

(1)　X に関するカルノー図を図 1-2 の形式で完成せよ。なお，カルノー図については解答用紙に図 1-2 を転記して作成すること。カルノー図の各要素は0, 1 または＊とすること。

(2)　(1) で作成したカルノー図から，簡単化した論理関数 X を示せ。なお，簡単化の様子が分かるように，(1) で作成したカルノー図内に枠で囲み明示すること。

(3)　(2) で簡単化した論理関数 X の回路を，2 入力 AND と 2 入力 OR それぞれ一つずつを用いて図示せよ。

表 1-1

$ABCD$	X	$ABCD$	X
0000	0	1000	0
0001	0	1001	＊
0010	0	1010	0
0011	0	1011	0
0100	0	1100	1
0101	1	1101	1
0110	0	1110	1
0111	＊	1111	1

$AB \setminus CD$	00	01		
00				
01				

図 1-2

III　T-フリップフロップを用いて，入力$COUNT$の数を出力する同期式3ビット5進カウンタを作成する。このカウンタを作成するため図1-3のように3つのT-フリップフロップを用意した。この図において，カウンタ出力を上位ビットから Q_2, Q_1, Q_0 とする。表1-2にT-フリップフロップの真理値表を示す。表1-2は，クロックCKが入力されるたびに，フリップフロップの状態が T, Q の値に応じて Q から Q' に変化することを示す。このとき，次の(1)～(3)の問いについて答えよ。なお，ドントケアは＊で表し，Q_2, Q_1, Q_0 の初期値はいずれも0とする。

(1)　表1-3は3ビット5進カウンタの状態遷移表の一部である。この表は，カウンタの値が $Q_2Q_1Q_0$ のときに $COUNT$ が入力されると，カウンタの値が $Q_2'Q_1'Q_0'$ に変化することを表す。この状態遷移表を完成し，解答用紙に記述せよ。

(2)　(1)の状態遷移を実現するT-フリップフロップの入力 T_2, T_1, T_0 の真理値表を表1-4の形式で解答用紙に記述せよ。

(3)　各フリップフロップの出力 Q_2, Q_1, Q_0 を用いて，T_2, T_1, T_0 の論理関数を最も簡単化された形でそれぞれ答えよ。

表1-2

T	Q	Q'
0	0	0
0	1	1
1	0	1
1	1	0

表1-3

$Q_2Q_1Q_0$	$Q_2'Q_1'Q_0'$
000	001
001	
⋮	

表 1-4

$Q_2Q_1Q_0$	T_2	T_1	T_0
000			
001			
010			
⋮			

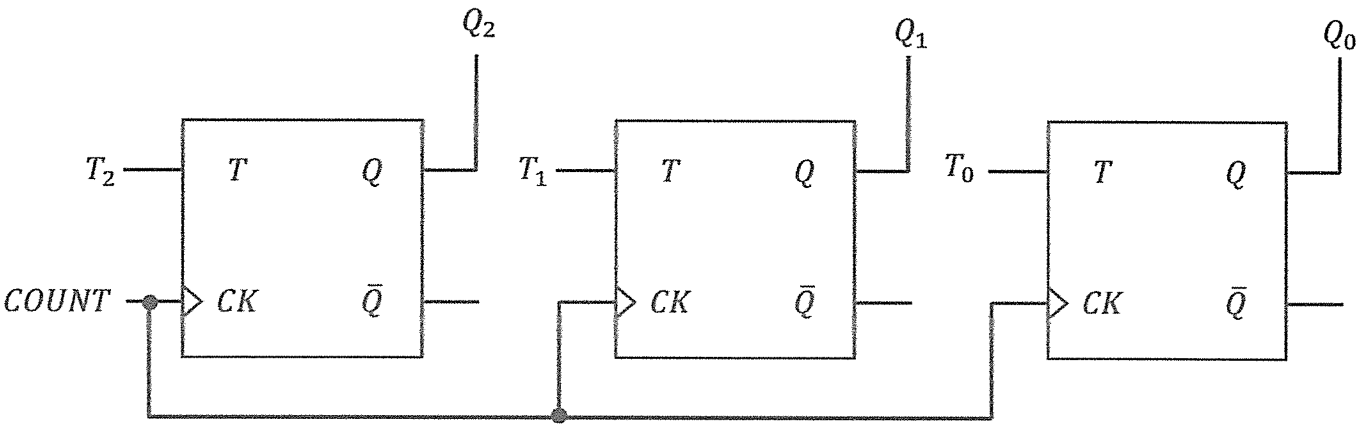

図 1-3

問題２　設問すべてについて解答すること。

Ⅰ　図 2-1 の C 言語で書かれたプログラムは，二次元配列で確保された二つの正方行列 `src1` および `src2` の行列積を計算するプログラムである。次の(1)～(4)の問いについて答えよ。なお，二次元配列で表された３×３の行列の要素アクセスの例を図 2-2 に示す。

(1)　図 2-1 の空欄（ア）～（オ）を適切に埋めることで完成させよ。

(2)　関数 `matmul1` に，引数 `src1 = {{2.0, 1.0}, {3.0, 0.0}}`，`src2 = {{0.0, 1.0}, {3.0, 2.0}}`，`N=2` を渡して呼び出した後の出力 `dst` の中身を答えよ。

(3)　このアルゴリズムのオーダーをビックオー記法で答えよ。

(4)　図 2-3 のプログラムは，関数の引数を一次元配列として受け付けるように書き直し，配列添え字のアクセス順序を変更したものである。空欄（カ）～（ク）を適切に埋めることで完成させよ。なお，一次元配列は各行の値がシーケンシャルに格納されているものとする。例えば，配列における要素の格納順が，2 行 2 列の行列であれば，{(1 行目，1 列目)，(1 行目，2 列目)，(2 行目，1 列目)，(2 行目，2 列目)} の順に格納されるものとする。

```
void matmul1(double** src1, double** src2, double** dst, int N){
        for (int k = 0; k < N; k++){
                for (int j = 0; j < N; j++){
                        double sum = 0.0;
                        for (int i = 0; i < N; i++){
                                sum += src1[（ア）][（イ）] * src2[（ウ）][（エ）];
                        }
                        dst[（オ）][k] = sum;
                }
        }
}
```

図 2-1：行列積を計算するプログラム 1

a[0][0]; //1行1列目の要素

$$A=\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

a[1][2]; //2行3列目の要素

図 2-2：2 次元配列の要素アクセス例

```
void matmul2(double* src1, double* src2, double* dst, int N){
        for (int k = 0; k < N; k++){
                double* sptr1 = src1 + k * N;
                double* dptr1 = dst + k * N;
                for (int j = 0; j < N; j++){
                        double sum = 0.0;
                        for (int i = 0; i < N; i++){
                                sum += sptr1[（カ）] * （キ）;
                        }
                        （ク） = sum;
                }
        }
}
```

図 2-3：行列積を計算するプログラム 2

II　数値積分に関する次の問い(1)～(4)について答えよ。

図 2-4 の C 言語で書かれたプログラムは数値積分の関数であり，関数名 integral_daikei は台形積分法，関数名 integral_kubun は区分求積法による数値積分である。(1)および(2)の問いについて答えよ。

(1)　図 2-4 の空欄（ア）～（エ）を適切に埋めることで完成させよ。ただし，同じ記号には同じ文字列が入る。

(2)　積分対象関数が偶関数であることを利用して図 2-4 の数値積分を高速化したい。main 関数では，区間 a=-10 から b=10 まで分割数 n=10000 で区分求積法を呼び出している。これを半分の分割数にして同等精度に計算したい場合，main 関数の空欄（オ）～（キ）を適切に埋めよ。

また，一般的な数値積分に関する問い(3)および(4)について答えよ。

(3)　台形積分法と区分求積法のどちらが数値計算精度の高いアルゴリズムか答えよ。

(4)　数値積分において，積分区間が正負で対称の時，（図 2-4 のプログラムを例にした場合，引数が a=-10, b=10 の場合），(a)～(g)のうち正しいものをすべて選択せよ。

(a) 積分対象の関数が奇関数の場合，台形積分法の数値計算精度が高い。
(b) 積分対象の関数が奇関数の場合，区分求積法の数値計算精度が高い。
(c) 積分対象の関数が奇関数の場合，どちらの数値計算精度も同じになる。
(d) 積分対象の関数が偶関数の場合，台形積分法の数値計算精度が高い。
(e) 積分対象の関数が偶関数の場合，区分求積法の数値計算精度が高い。
(f) 積分対象の関数が偶関数の場合，どちらの数値計算精度も同じになる。
(g) 解は未定になる。

```
double func(double x){
	return  x * x;
}
double integral_daikei(double a, double b, int n){
	double h = (b - a) / n;
	double sum = 0.0;
	sum = (ア) * func(a);
	for (int i = 1; i < n; i++){
		double x = （イ）;
		sum += func(x);
	}
	sum += （ウ） * func(b);
	return （エ）;
}
double integral_kubun(double a, double b, int n){
	double h = (b - a) / n;
	double sum = 0.0;
	for (int i = 0; i < n; i++){
		double x = （イ）;
		sum += func(x);
	}
	return （エ）;
}
int main(){
	int n = 10000;
	printf("daikei %f¥n", integral_daikei(-10, 10, n));
	printf("kubun1 %f¥n", integral_kubun(-10, 10, n));
	printf("kubun2 %f¥n", （オ） * integral_kubun(（カ）, （キ）, n/2));
	return 0;
}
```

図 2-4：数値積分を計算するプログラム

問題3　設問すべてについて解答すること。導出過程も簡潔に示すこと。ただし，解答においては最も簡約化した形で示すこと。ここで簡約化とは，分数に関しては既約形，対数に関しては分数を含まない最も簡単な形（例：$\log_2 6 = 1 + \log_2 3$）に変形することを指す。また，$0\log_2 0 = 0$とする。

I　qを$0 \le q \le 1$を満たす実数とし，確率変数$X, Y \in \{0,1\}$についての確率が

$$P(X=1)=q, \quad P(Y=1|X=1)=\frac{1}{4}, \quad P(Y=0|X=0)=\frac{1}{4}$$

のように与えられるとき，次の(1)～(5)の問いについて答えよ。また，解答においては可能な限り，2元エントロピー関数$h(a) = -a\log_2 a - (1-a)\log_2(1-a)$を用いること。

(1) エントロピー$H(X)$をqを含む式で表せ。

(2) 結合（同時）エントロピー$H(X,Y)$をqを含む式で表せ。

(3) 条件付きエントロピー$H(Y|X)$を求めよ。

(4) 相互情報量$I(Y;X)$をqを含む式で表せ。

(5) 相互情報量$I(Y;X)$を最大にするqの値およびその最大値を求めよ。

II　$X_t \in \{A,B,C,D\}$ $(t = 0,1,2,\ldots)$を，定常マルコフ情報源Sの出力シンボル系列とする。ここで出力シンボルの生起確率$P(X_t|X_{t-1})$は以下のとおりとする。

$$P(X_t=A|X_{t-1}=A)=\frac{3}{4}, \quad P(X_t=B|X_{t-1}=A)=\frac{1}{4}$$

$$P(X_t=B|X_{t-1}=B)=\frac{1}{2}, \quad P(X_t=C|X_{t-1}=B)=\frac{1}{2}$$

$$P(X_t=C|X_{t-1}=C)=\frac{1}{4}, \quad P(X_t=D|X_{t-1}=C)=\frac{3}{4}$$

$$P(X_t=D|X_{t-1}=D)=\frac{1}{4}, \quad P(X_t=A|X_{t-1}=D)=\frac{3}{4}$$

このとき，次の(1)～(4)の問いについて答えよ。

(1) マルコフ情報源Sのシャノン線図（状態遷移図）を下に示す図 3-1 の例にならって示せ。この例では，A（またはB）を囲む丸は，シンボルA（またはB）を出力した後の状態を表し，状態を結ぶ矢印の上に書かれた数値は出力シンボルの生起確率（状態の遷移確率）を表している。

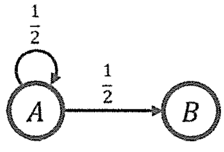

図 3-1：シャノン線図の例

(2) マルコフ情報源Sの定常確率分布$P(X_t)$を求めよ。

(3) 情報源Sを(2)で得られた定常確率分布$P(X_t)$に従う無記憶情報源と見なしたとき，情報源アルファベット$\{\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D}\}$を符号アルファベット$\{0,1\}$を用いてハフマン符号化せよ。導出に用いたハフマン木，得られた符号語および平均符号語長を答えよ。

(4) 情報源Sがマルコフ情報源であることを利用して，情報源Sをハフマン符号化することを考える。以下の文章の空欄①～⑥に入る適切な語句，数式または数値を答えよ。

> 情報源Sがマルコフ情報源であることを利用して，情報源Sをハフマン符号化する場合，平均符号語長は(3)の場合と比べて$\boxed{①}$くなる。
>
> 情報源Sはマルコフ情報源であるため，$t \geq 1$において時刻tにおける出力シンボルX_tの生起確率は$\boxed{②}$に依存して変化する。よって，Sがマルコフ情報源であることを利用する場合，$t \geq 1$では$\boxed{②}$毎に$\boxed{③}$をしてハフマン符号化を行えばよい。このとき生起確率が0となるシンボルを無視すると，$t \geq 1$における平均符号語長は$\boxed{④}$ビットとなる。$t = 0$における平均符号語長は$\boxed{⑤}$ビットとなるが，全体としての（$t \geq 0$における）平均符号語長は$\boxed{⑥}$ビットとなる。