令 和 6 年 度

名古屋大学大学院情報学研究科
情報システム学専攻
入学試験問題（専門）

令和５年８月２日

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．外国人留学生の志願者は、日本語と日本語以外の１言語間の辞書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．日本語または英語で解答すること。

５．問題冊子、解答用紙６枚、草稿用紙３枚が配布されていることを確認すること。

６．問題は問１～８の８問がある。このうち６問を選択して解答すること。なお，選択した問題番号を解答用紙の指定欄に記入すること。

７．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。

８．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、解答用紙表面右下に明記すること。

９．解答用紙は試験終了後に６枚とも提出すること。

１０．問題冊子、草稿用紙は試験終了後に持ち帰ること。

問1

(1) 1枚の10円玉と2枚の5円玉を使ってコイン投げを行い，表が出たコインの枚数を確率変数 X, 表が出たコインの金額の和を確率変数 Y で表す．たとえば，10円玉と1枚の5円玉について表，残り1枚の5円玉について裏が出た場合は $X = 2, Y = 15$ となる．

全てのコインは公正（表が出る確率，裏が出る確率はいずれも 1/2）と仮定し，以下の値を計算せよ．なお，計算結果には，分数や $\log_2 p$ の形の項（ただし p は奇数に限る）が含まれていても良い．

(a) X, Y のエントロピー $H(X), H(Y)$

(b) X と Y の結合エントロピー $H(X, Y)$

(c) 条件付きエントロピー $H(Y|X)$

(d) 相互情報量 $I(X; Y)$

(2) S は，右表に与えられる確率分布に従って各記号を生成する無記憶・定常な情報源である．この情報源 S から出力される記号を，瞬時復号可能な2元符号 C により符号化することを考える．

記号	確率	C の符号語
a	1/12	00
b	1/12	101
c	1/12	(*)
d	1/4	010
e	1/4	11
f	1/4	100

(e) 右表で示すように C の符号語を定めるとき，瞬時復号可能性を確保するためには，記号 c に対する符号語を何にすれば良いか答えよ．複数の解答が考えられる場合，もっとも短い符号語を示せ．

(f) 符号 C の平均符号語長を求めよ．解答は分数または実数（小数点以下2桁程度で四捨五入して良い）として示すこと．

(g) 情報源 S から出力される記号に対し，ハフマン符号を構成せよ．また，構成したハフマン符号の平均符号語長を示せ．

Translation of technical terms

コイン投げ	coin toss	無記憶	memoryless
確率変数	random variable	定常	stationary
公正	fair	情報源	information source
エントロピー	entropy	瞬時復号可能	immediately decodable
結合	joint	復号化	encode
条件付き	conditional	符号語	codeword
相互情報量	mutual information	平均符号語長	average codeword length
確率分布	probability distribution	ハフマン符号	Huffman code

問2

半順序集合$(X, \sqsubseteq_X)$と，その部分集合$S \subseteq X$について，以下のように定義する．

$$a \in X \text{ は } S \text{ の上限} \overset{\text{def}}{\iff} \forall x \in S.(x \sqsubseteq_X a) \text{ かつ } \forall b \in X.((\forall x \in S.(x \sqsubseteq_X b)) \text{ ならば } a \sqsubseteq_X b)$$

$$a \in X \text{ は } S \text{ の下限} \overset{\text{def}}{\iff} \forall x \in S.(a \sqsubseteq_X x) \text{ かつ } \forall b \in X.((\forall x \in S.(b \sqsubseteq_X x)) \text{ ならば } b \sqsubseteq_X a)$$

$$(X, \sqsubseteq_X) \text{ は束} \overset{\text{def}}{\iff} \forall a, b \in X.(\{a, b\} \text{ の上限と下限が存在する})$$

$$(X, \sqsubseteq_X) \text{ は完備束} \overset{\text{def}}{\iff} \forall S \subseteq X.(S \text{ の上限と下限が存在する})$$

以下の全ての問いに答えよ．

(1) 次ページの図1に挙げる半順序集合 A〜E それぞれについて、「a. 束でない」「b. 束であるが完備束でない」「c. 完備束である」のいずれであるか答えよ．

(2) (1) で「b. 束であるが完備束でない」と答えた半順序集合 $(X, \sqsubseteq_X)$ それぞれについて，上限または下限が存在しないような部分集合 $S \subseteq X$ を一つ挙げよ．

(3) 自然数の集合 $\mathbb{N}$ の冪集合 $\mathcal{P}(\mathbb{N})$ は，包含関係 $\subseteq$ によって完備束になる．集合 $S \subseteq \mathcal{P}(\mathbb{N})$ の上限は何か？説明せよ．

(4) (3) の半順序集合 $(\mathcal{P}(\mathbb{N}), \subseteq)$ について，次の条件 (i)，(ii) を満たす写像 $f : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ を一つ挙げ，その写像 f が条件 (i)，(ii) を満たすことを証明せよ．ここで，$S \subseteq \mathcal{P}(\mathbb{N})$ に対して，$\sqcup S$ は S の上限を表すとする．

条件 (i)　$\forall x, y \in \mathcal{P}(\mathbb{N}).(f(\sqcup\{x, y\}) = \sqcup\{f(x), f(y)\}$

条件 (ii)　$\exists S \subseteq \mathcal{P}(\mathbb{N}).f(\sqcup S) \neq \sqcup\{f(x) \mid x \in S\}$

Translation of technical terms

半順序集合	partially ordered set	元	element
部分集合	subset	記号	symbol
上限	least upper bound	有限長	finite length
下限	greatest lower bound	文字列	string
束	lattice	連接	concatenation
完備束	complete lattice	自然数	natural number
実数	real number	冪集合	power set
集合	set	包含関係	inclusion relation
絶対値	absolute value	写像	mapping
有理数	rational number		

A. $(\mathbb{R}, \leq)$： 実数の集合と，通常の大小関係.

B. $(\{r \in \mathbb{R} \mid |r| \leq 10\}, \leq)$：絶対値が 10 以下の実数の集合と，通常の大小関係.

C. $(\mathbb{Q} \cup \{+\infty, -\infty\}, \leq_\infty)$：有理数の集合に特別な元 $+\infty$ と $-\infty$ を追加した集合と，次で定義される半順序.

$$q \leq_\infty q' \overset{\text{def}}{\iff} q = -\infty \text{ または } q' = +\infty \text{ または } (q, q' \in \mathbb{Q} \text{ かつ } q \leq q')$$

D. $(\{0,1\}^*, \preceq_{\mathrm{P}})$：記号 0, 1 からなる有限長の文字列の集合と，次で定義される半順序．ただし，$w \cdot w''$ は，w と w'' の連接とする.

$$w \preceq_{\mathrm{P}} w' \overset{\text{def}}{\iff} \exists w'' \in \{0,1\}^*.(w' = w \cdot w'')$$

E. $(\{0,1\}^*, \preceq_{\mathrm{L}})$：記号 0, 1 からなる有限長の文字列の集合と，次で定義される半順序．ただし，$\preceq_{\mathrm{P}}$ は D で与えられた半順序とする.

$$w \preceq_{\mathrm{L}} w' \overset{\text{def}}{\iff} w \preceq_{\mathrm{P}} w' \text{ または } \exists w'' \in \{0,1\}^*.(w'' \cdot 0 \preceq_{\mathrm{P}} w \text{ かつ } w'' \cdot 1 \preceq_{\mathrm{P}} w')$$

図 1: 半順序集合 A〜E

問 3

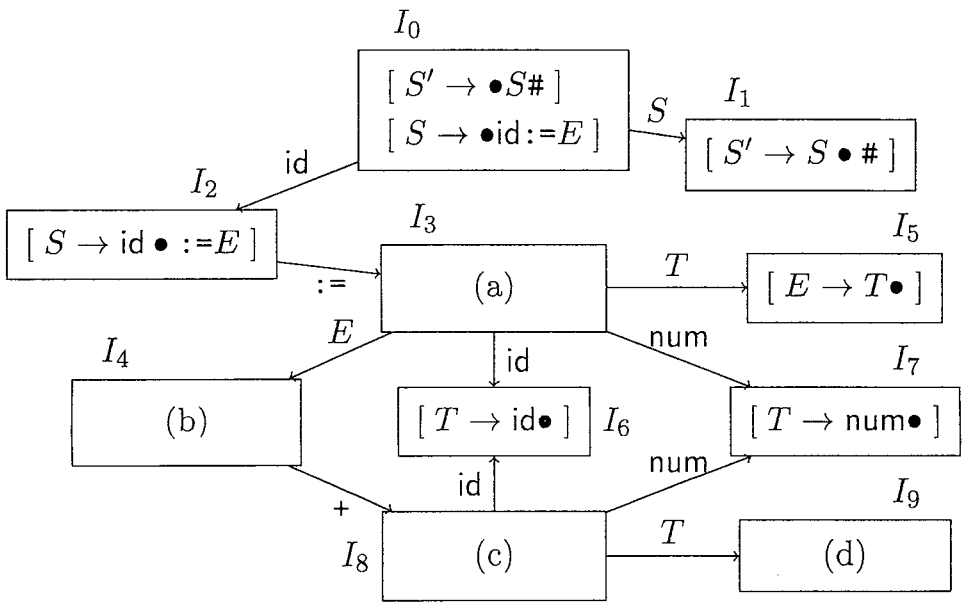
以下の文脈自由文法$G = \langle N, \Sigma, P, S' \rangle$について答えよ．ここで，非終端記号の集合$N = \{S', S, E, T\}$，終端記号の集合$\Sigma = \{+, \mathsf{id}, \mathsf{num}, :=\}$とし，生成規則の集合 P を以下に示す．

$$P = \left\{ \begin{array}{llll} 0: & S' \to S\#, & 3: & E \to T, \\ 1: & S \to \mathsf{id}\ :=E, & 4: & T \to \mathsf{id}, \\ 2: & E \to E + T, & 5: & T \to \mathsf{num} \end{array} \right\}$$

id と num は，入力を字句解析した結果得られるトークンであり，# は入力の最後を表す記号である．また，各規則の前の数字 n: は規則につけられた番号である．

(1) $\mathrm{Follow}_1(S)$, $\mathrm{Follow}_1(E)$, $\mathrm{Follow}_1(T)$ を求めよ．

(2) G を SLR(1)構文解析をするための LR(0) オートマトンの (a)~(d) にあてはまる LR(0)項集合を求めよ．



(3) 以下の形で G に対する SLR(1)構文解析表を作成せよ．action 表については状態 I_k へ遷移するシフト動作は sk, 番号 ℓ の生成規則による還元動作は rℓ, 受理は acc を記入せよ．GOTO 表については状態 I_m の番号 m を記入すること．

	action					*GOTO*		
	:=	+	id	num	#	S	E	T
I_0								
⋮			⋮				⋮	
I_9								

(4) SLR(1) 構文解析表に従ってプッシュダウン変換機械を動作させて構文解析する．プッシュダウン変換機械の時点表示は (スタック，入力ポインタ以降の入力テープ，出力テープ) の 3 項組で表す．スタックは $s_0X_1s_1X_2\cdots X_ns_n$ で表し (ここで，$s_i \in \{I_0,\cdots,I_9\}$, $X_j \in N \cup \Sigma$)，出力テープには還元に使用した生成規則の番号を順に書き込むこととする．構文解析表に基づくプッシュダウン変換機械の 1 ステップの動作を $\Rightarrow$ で表す．

id := id + num + id #

上記の入力によってプッシュダウン変換機械を動作させるときの初期時点表示から受理に至る遷移系列のうち (A) の部分を完成し，(B) の部分の生成規則の番号列を示せ．

$$(I_0, \mathsf{id:=id+num+id\#}, \varepsilon) \Rightarrow \cdots \quad (\mathrm{A}) \quad \cdots \Rightarrow (I_0SI_1, \#, \underbrace{r_1\cdots r_m}_{(B)}) \Rightarrow (acc, \varepsilon, r_1\cdots r_m 0)$$

((A) には複数の遷移が含まれるので $C_1 \Rightarrow C_2 \Rightarrow \cdots \Rightarrow C_n$ のように解答すること．C_j はプッシュダウン変換機械の時点表示である．)

Translation of techical terms:

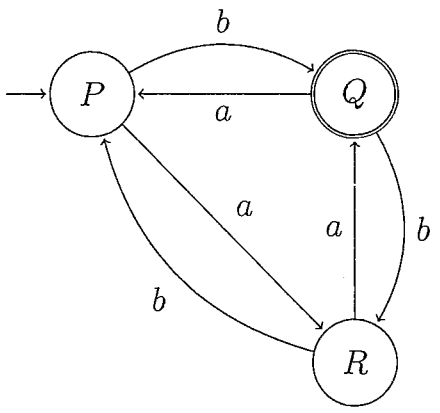
文脈自由文法	context-free grammar	シフト動作	shift operation
非終端記号	non-terminal symbol	還元動作	reduce operation
終端記号	terminal symbol	受理	accept
生成規則	production rule	プッシュダウン変換機械	pushdown transducer
字句解析	lexical analysis	時点表示	configuration
トークン	token	スタック	stack
構文解析	parsing	入力テープ	input tape
LR(0) 項	LR(0) term	出力テープ	output tape
構文解析表	parsing table	初期時点表示	initial configuration
状態	state	遷移系列	transition sequence

問4

本問では，アルファベットを $\Sigma = \{a, b\}$ とする．

(1) ab あるいは ba で終わる文字列全体を認識する非決定性有限オートマトンのうち，その状態数が5以下のものをひとつ示せ．ただし，決定性有限オートマトンを答えても構わない．

(2) 以下に示す決定性有限オートマトン M_1 が認識する言語 L_1 がどのような文字列からなるか説明せよ．



決定性有限オートマトン M_1

(3) (2) で与えた言語 L_1 に対して以下に示す反復補題を適用したとき，反復補題が $m = 3$ と定めたとする．このとき各問に答えよ．

反復補題

正規言語 L に対して以下を満たす正整数 m が存在する．

$|w| \geq m$ を満たす任意の文字列 $w \in L$ に対して，i),ii),iii) と $w = xyz$ のすべてを満たす文字列 x, y, z が存在する．

i) 任意の $i \geq 0$ について $xy^iz \in L$　　ii) $|y| \geq 1$　　iii) $|xy| \leq m$

(a) $w = bbabbaa \in L_1$ に対して条件 i),ii),iii) と $w = xyz$ のすべてを満たす x, y, z を示せ．

(b) ある文字列 $w \in L_1$ に対する反復補題が示す w の分割 xyz の x が空列 ε であった．このときの w のうちで $3 \leq |w| \leq 4$ を満たすすべての文字列を示せ．

(4) 上記の反復補題を証明せよ．

Translation of technical terms

アルファベット	alphabet	言語	language
文字列	string	反復補題	pumping lemma
非決定性有限オートマトン	non-deterministic finite automaton	正規言語	regular language
認識する	recognize	正整数	positive integer
状態数	number of states	分割	split
決定性有限オートマトン	deterministic finite automaton	空列	empty string

問5

(1) k 種類の命令（I_1, I_2, …, I_k）を実行可能なプロセッサを考える．このプロセッサでは，命令を逐次的に実行する（一つの命令が完了するまで次の命令を実行できない）ものとする．また，同時に複数の命令を実行することはなく，一つの命令実行完了後，次のクロックサイクルから次の命令の実行を開始するものとする．このプロセッサ上で実行されるコード系列が与えられており，そのコード系列では各命令（I_1, I_2, …, I_k）の実行命令数が以下の通りであるものとする．

- ✓ 命令 I_j $(1 \leq j \leq k)$ の実行命令数　：n_j

上記のコード系列をプロセッサ上で実行する間，クロックサイクル時間は T [秒]に固定されているものとする．また，各命令の CPI (clock cycles per instruction) は以下の通りであるものとする．

- ✓ 命令 I_j $(1 \leq j \leq k)$ の CPI　：CPI_j

このとき，与えられたコード系列の実行時間 S [秒] を n_j, CPI_j, T の式で示せ．

(2) (1)の具体的な例として，3 種類の命令（I_1, I_2, I_3）を実行可能なプロセッサを考える．このプロセッサ上で，コード系列 1 あるいはコード系列 2 のいずれかを実行する．表 1 に，コード系列 1 とコード系列 2 における，各命令（I_1, I_2, I_3）の実行命令数をそれぞれ示す．また，表 2 に，命令（I_1, I_2, I_3）の CPI を示す．命令 I_1, I_2, または I_3 実行時のクロックサイクル時間 Tは，全て 3.0×10^{-10} 秒であるものとする．

表 1: 各コード系列における実行命令数

	I_1	I_2	I_3
コード系列 1	7.0×10^9	2.0×10^9	1.0×10^9
コード系列 2	1.0×10^9	1.0×10^9	4.0×10^9

表 2: 各命令の CPI

I_1	I_2	I_3
1	2	3

(a) コード系列 1 を実行した場合の総実行命令数，平均 CPI，実行時間をそれぞれ有効数字 2 桁で示せ．

(b) コード系列 2 を実行した場合の総実行命令数，平均 CPI，実行時間をそれぞれ有効数字 2 桁で示せ．

(3) (2)において，すべての I_3 命令に対して，「I_3 1 命令」を「I_1 2 命令」に変換できるものとする．

(a) 上記の変換を行った場合，コード系列 1 の実行時間は(2)(a)と比較して何秒短縮されるか算出せよ．

(b) 上記の変換を行った場合，コード系列 2 の実行時間は(2)(b)と比較して何秒短縮されるか Amdahl の法則を用いて算出せよ．Amdahl の法則を用いて算出したことが分かるように算出過程を示すこと．

(4) 要求された処理をプロセッサ上で実現する際に，プロセッサの性能をどのように評価するべきか考える．プロセッサの性能を評価する尺度（以下，性能評価尺度）の中でも，「要求された処理を実現するコード系列を，プロセッサ上で実行するのに要する時間」は最も公平なもののひとつである．この点に注意して，以下の問いに答えよ．

(a) 「要求された処理を実現するコード系列の総実行命令数」を性能評価尺度として用いることの問題点を 150 文字以内（英語の場合は 100 語以内）で説明せよ．

(b) 「プロセッサのクロック周波数」を性能評価尺度として用いることの問題点を 150 文字以内（英語の場合は 100 語以内）で説明せよ．

(c) 「要求された処理を実現するコード系列をプロセッサ上で実行した際の MIPS (million instructions per second) 値」を性能評価尺度として用いることの問題点について，150 文字以内（英語の場合は 100 語以内）で説明せよ．

Translation of technical terms

命令	instruction
実行可能	executable
プロセッサ	processor
逐次的	sequential
複数	multiple
コード系列	code sequence
実行命令数	number of executed instructions
クロックサイクル時間	clock cycle time
実行時間	execution time
総実行命令数	total number of executed instructions
変換	conversion
Amdahl の法則	Amdahl's law
性能	performance
評価	evaluation
クロック周波数	clock frequency

問6

ソースコード 1, 2 はどちらもフィボナッチ数を計算する C 言語プログラムである．ソースコード 1 の fib1 はフィボナッチ数を再帰的に計算し，ソースコード 2 の fib2 にはメモ化が導入されている．本問では 32 ビットの符号付き整数でオーバーフローが発生しない範囲として，40 番目までのフィボナッチ数のみを扱うこととする．以下の全ての問いに答えよ．

(1) ソースコード 1 の [A] に 7 を埋めて main 関数を実行したときに標準出力に印字される実行結果を書け．

(2) ソースコード 2 の [B] に 10 を埋めて main 関数を実行したときに標準出力に印字される実行結果を書け．

(3) ソースコード 2 の [B] に整数 m（$1 < m \leq 40$）を埋めて main 関数を実行したときに 44 行目の printf 文を実行するときに cntr が保持する値を m の式で表せ．

(4) ソースコード 1 の [A] とソースコード 2 の [B] に同じ整数を埋めたとき，一般にソースコード 2 の main 関数の実行はソースコード 1 の main 関数の実行よりも効率的である．その理由を 150 字以内（英語の場合，100 words 以内）で説明せよ．

(5) ソースコード 2 の 43 行目の配列 memo の初期化は 44 行目の fib2(n) の計算において参照されることがない要素も含めて初期化している．[B] に与えた整数 m について適切にフィボナッチ数を計算するために必要な要素のみを-1 で初期化するように 43 行目を変更せよ．解答では 43 行目に記述するコードを記すこと．

ソースコード 1: フィボナッチ数を再帰的に計算する C 言語プログラム

```
1  #include <stdio.h>
2
3  int cntr=0;
4
5  int fib1(int n) {
6    int z;
7    cntr++;
8    if( n<=0 )
9      z=0;
10   else if( n==1 )
11     z=1;
12   else
13     z=fib1(n-1)+fib1(n-2);
14   return z;
15 }
16
17 int main() {
18   int n=[ A ];
19   printf("%d,%d",fib1(n),cntr);
20   return 0;
21 }
```

ソースコード 2: メモ化を用いてフィボナッチ数を計算する C 言語プログラム

```
22 #include <stdio.h>
23
24 int cntr=0;
25 int memo[41];
26
27 int fib2(int n) {
28   int z;
29   if( memo[n]>=0 ) return memo[n];
30   cntr++;
31   if( n<=0 )
32     z=0;
33   else if( n==1 )
34     z=1;
35   else
36     z=fib2(n-1)+fib2(n-2);
37   memo[n]=z;
38   return z;
39 }
40
41 int main() {
42   int i, n=[ B ];
43   for( i=0 ; i<41 ; i++ ) memo[i]=-1;
44   printf("%d,%d",fib2(n),cntr);
45   return 0;
46 }
```

(6) 関数呼び出しを用いずにフィボナッチ数を計算する関数 `fib3` を，下記の関数定義の中の［（ア）］，［（イ）］，［（ウ）］を埋めて完成せよ．ただし，任意の整数 m（$0 \leq m \leq 40$）について，`fib3`(m) の返り値が `fib1`(m) の返り値と一致すること．

```
int fib3(int n) {
  int i, p=0, q=1, tmp;
  for( i=0 ; i<n ; i++ ) {
    tmp=［（ア）］;
    p=［（イ）］;
    q=［（ウ）］;
  }
  return p;
}
```

Translation of technical terms

ソースコード	source code	標準出力	standard output
フィボナッチ数	Fibonacci number	印字する	print
C 言語	C language	配列	array
再帰的	recursively	初期化	initialization
メモ化	memoization	参照する	refer to
符号付き整数	signed integer	要素	element
オーバーフロー	overflow	関数呼び出し	function call
関数	function	返り値	return value
実行する	execute		

問 7

次の文を読んで，UML(Unified Modeling Language)に関する下の(1)~(4)に答えよ．

図 1 はウェブブラウザを使って情報 R を参照するときの振る舞いを表したシーケンス図S である．情報 R は ID, password を事前に登録したユーザのみが参照できる．クラス Web server はクラス Web browser に情報 R を表示するための HTML(Hyper Text Markup Language)ページを送信する．クラス Authenticator はユーザが事前に設定した ID, password を記録しておき，クラス Web server 経由で送られてくる ID, password と一致するか確かめる．一致した場合にはクラス Session manager に ID を送信し，sessionID の生成を要求する．クラス Web server は情報 R を表示するための HTML ページにその sessionID を含め，ログイン結果として送信する．クラス Web server はログイン結果ページ以外の HTML ページ~~以外~~には sessionID を含めない．クラス Web browser から ID, sessionID を含めて情報 R の要求があった場合，Web server は ID と sessionID が正当なものか Session manager に確認し，その結果を identificationResult として受け取る．

(1) HTML ページを表すクラス Web page をクラス図で定義せよ．個々の HTML ページには他のページと重複しない URL(Uniform Resource Locator)が割り当てられる．クラス Web server は複数の HTML ページを保持する．解答には Web server クラスを含め，Web page クラスとの関連，関連名，多重度が明確であるものは，もれなく記述すること．操作は省略してもよい．

(2) 認証が必要でない(sessionID を使わない)情報 P をユーザが参照する場合のシーケンス図 S'を書け．なお，情報 P を表示するための HTML ページを表すメッセージのラベルは resource P page とすること．また，S で定義されているクラスやメッセージを使う場合には同一の名称とすること．

(3) S に含まれる各クラスのクラス図を書け．クラス名はシーケンス図のクラス名と同一とし，属性名はシーケンス図のメッセージと同一とすること．操作は省略し，シーケンス図にない情報に対応する属性名は適切な名称を設定すること．S からわかる関連や多重度をもれなく記入すること．

(4) クラス Authenticator，クラス Session manager が~~備えておくべき情報~~(Sに示されていないものを含む)をクラス図として定義せよ．必要な属性，操作を定義すること．クラス名，属性名，操作名のうち S で定義されているものは S と同一の名称とすること．

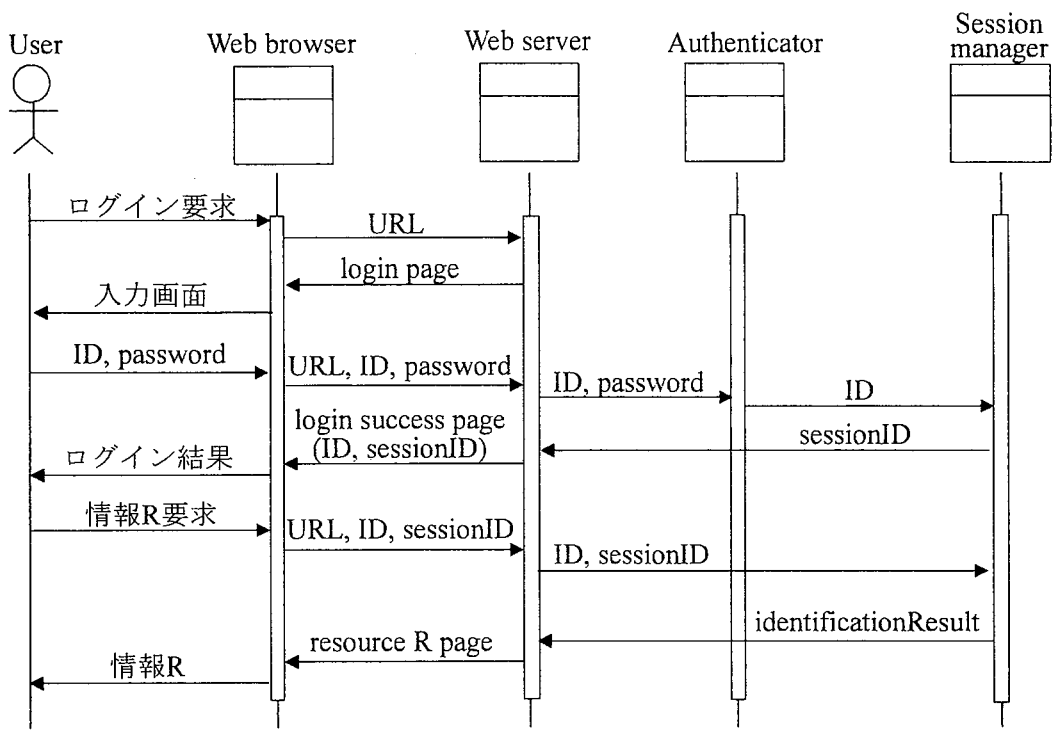


図 1 シーケンス図 S

問8

インターネットプロトコル（IP）と，その上で信頼性のあるストリーム通信を実現するためのTCPなどのトランスポートプロトコル（以下，単にトランスポートプロトコルという）に関する以下の問いに答えよ。

(1) IPアドレスは，ネットワーク部とホスト部の2つの部分で構成されている。例えば，133.6.204.129/16というIPv4アドレスでは，133.6がネットワーク部，204.129がホスト部である。ネットワーク部とホスト部について説明し，それらを分ける理由を述べよ。

(2) トランスポートプロトコルでは，受信側からの確認応答（ACK）を受け取る前に，送信側が複数のIPデータグラムを送信する設計となっているのが一般的である。このような設計にする利点を述べよ。

(3) IPデータグラムの到着順序が入れ替わる原因を1つ挙げよ。また，IPデータグラムの到着順序の入れ替わりを修正するために，トランスポートプロトコルでとれる対策について説明せよ。

(4) フロー制御（受信側の処理速度に合わせて，送信側が送信するデータ量を調整する機能）を，トランスポートプロトコルで実現するための方法を説明せよ。

(5) 輻輳制御（ネットワーク内で混雑が発生した場合に，送信するデータ量を調整する機能）を，トランスポートプロトコルで実現するための方法を説明せよ。

Translation of technical terms

インターネットプロトコル	Internet Protocol
トランスポートプロトコル	transport protocol
IPアドレス	IP address
ネットワーク部	network part
ホスト部	host part
IPデータグラム	IP datagram
確認応答	acknowledgement
フロー制御	flow control
輻輳制御	congestion control