H.25年度過去問解答

$\boxed{1}$ アルゴリズムとプログラミング

プログラムはインサーションソート

(1)

(ア) `i--`
(イ) `data[MAX-i-1]`

(2-1)

(1,b)(2,d)(3,c)(4,h)(5,a)(6,e)(7,g)(8,f)

(2-2)

(a) 6 回
(b) 13 回
(c) 7 回
(d) 10 回
∗ 値が同じ要素が比較された場合も交換が起こることに注意

(2-3)

(エ) $a_i \leqq a_j$
(オ) k

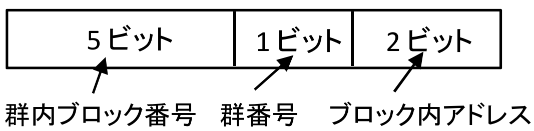
(3)

15 行目の条件が真ならば，要素の入れ替えが行われる．
ここで，同じ値を持つ要素が比較されたとき，15 行目の条件は真となり入れ替えが起こるため，順序関係が維持されなくなる．
したがって，このプログラムは安定ではない．

$\boxed{2}$ 計算機システムとシステムプログラム

(1)

- キャッシュサイズは 16byte，ブロックサイズは 4byte
  $16 \div 4 = 4$ より，
  キャッシュメモリのブロック枠は 4
- 2 ウェイなのでキャッシュメモリの各群のブロック枠は 2
  $4 \div 2 = 2$ より，群数は 2
- 群数は $2(=2^1)$ なので，
  群番号には 1 ビット必要
- ブロックサイズは $4(=2^2)$byte なので，
  ブロック内アドレスには 2 ビット必要
- 群内ブロック番号には 5 ビット必要 ($\because$ 8-1-2=5)
- メモリアドレスの構成は以下のようになる．



- 例：メモリアドレスが 01011011 の場合
  群内ブロック番号は 01011
  群番号は 0
  ブロック内アドレスは 11

(1-1-1)

問題中でアクセスされるブロックは 4 つ
(010110−−，010010−−，000101−−，011010−−)
したがって，初期参照によるキャッシュミス回数は 4

(1-1-2)

ブロックの置き換えを伴うキャッシュミスが起こるのは
5 つ目，9 つ目の 2 つ
したがって，ブロックの置き換えを伴うキャッシュミス回数は 2

(1-1-3)

全 10 回中，キャッシュミスが起こるのは
1 つ目，2 つ目，4 つ目，5 つ目，9 つ目の 5 つ
したがって，キャッシュミス率は
$\frac{5}{10} \times 100 = 50\,\%$

(1-2)

① 時間的局所性
ある領域のデータ転送が行われた後に，同一のデータ転送が再度，近い時間内に行われるとき，そのデータをキャッシュに保持していれば転送を速く行える．
② 空間的局所性
データ転送は，ある領域の連続，もしくは近傍の領域に発生することが多い．したがって，一度データ転送が行われた領域をキャッシュに保持することで，その後の転送の速度を高めることができる．

(1-3)

ブロックサイズが小さい：ブロックに保持できるデータ量が少ないので，置き換えが多く発生し，キャッシュのヒット率が低下するため
ブロックサイズが大きい：ブロックに保持されているデータは多いが，その中から探しているデータをヒットさせるのに時間がかかるため

(2-1)

(a) サ (b) ク (c) オ (d) キ (e) コ (f) イ (g) ス

(2-2-1)

① 到着順 (図 1)
平均 TAT=(20+52+52+76)/4=200/4=50
② 処理時間順 (図 2)
平均 TAT=(20+92+12+36)/4=160/4=40

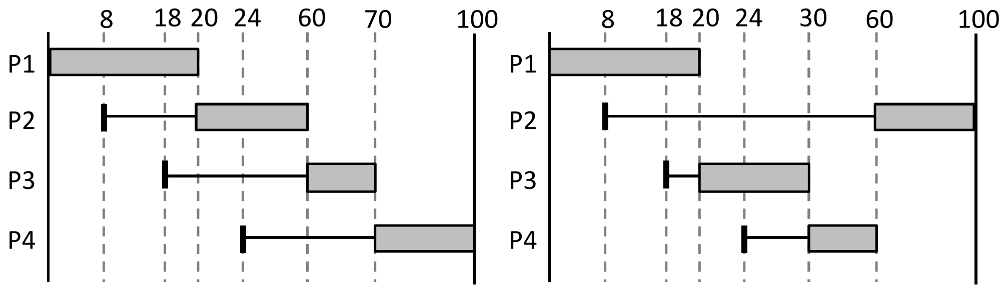


図 1: 到着順　　図 2: 処理時間順

(2-2-2)

(a) タイムスライス 4
平均 TAT=(40+90+36+76)/4=242/4=60.5
(b) タイムスライス 8
平均 TAT=(52+86+44+76)/4=258/4=64.5

(2-2-3)

タイムスライスの値が小さくなるほど，ディスパッチ回数が増えるため，ディスパッチに要する時間の分だけ処理時間も長くなる．したがって，プロセス切り替えの処理時間が 0 のときと比べて平均 TAT は大きくなる．

$\boxed{3}$ 離散構造

(1)

(a) 恒真

(b) 恒真ではないが充足可能

・真とする解釈

例：$\exists x p(x)$ が偽となるような解釈

$p(x)$：$x=c$ など

・偽とする解釈

例：$\exists x p(x)$ が真，$\forall x p(x)$ が偽となるような解釈

$p(x)$：$x=a$ など

(c) 充足不能

この論理式を変形すると

$\forall y \neg p(y) \Leftrightarrow \forall x p(x)$

となるので充足不能

(d) 恒真ではないが充足可能

・真とする解釈

$q(x,y)$：$x=a$ or $x=c$ など

・偽とする解釈

$q(x)$：$x=y$ など

(2-1)

$$\begin{aligned}\neg E &= A \wedge B \wedge C \wedge \neg D \\ &= p(f(g(f(g(g(a)))))) \wedge \forall x(p(f(g(x))) \Rightarrow p(x)) \\ &\qquad \wedge \forall x(p(g(f(x))) \Rightarrow p(x)) \wedge \neg \exists x p(g(x)) \\ &= p(f(g(f(g(g(a)))))) \wedge \forall x(\neg p(f(g(x))) \vee p(x)) \\ &\qquad \wedge \forall x(\neg p(g(f(x))) \vee p(x)) \wedge \forall x \neg p(g(x)) \\ &= \forall x(p(f(g(f(g(g(a)))))) \wedge (\neg p(f(g(x))) \vee p(x)) \\ &\qquad \wedge (\neg p(g(f(x))) \vee p(x)) \wedge \neg p(g(x))) \\ &= E'\end{aligned}$$

(2-2)

$$\begin{cases} p(f(g(f(g(g(a)))))) & \text{—— ①} \\ \neg p(f(g(x))) \vee p(x) & \text{—— ②} \\ \neg p(g(f(x))) \vee p(x) & \text{—— ③} \\ \neg p(g(x)) & \text{—— ④} \end{cases}$$

② の x に $g(a)$ を代入

$\neg p(f(g(g(a)))) \vee p(g(a))$ —— ⑤

④の x に a を代入

$\neg p(g(a))$ —— ⑥

⑤, ⑥ より

$\neg p(f(g(g(a))))$ —— ⑦

② の x に $f(g(g(a)))$ を代入

$\neg p(f(g(f(g(g(a)))))) \vee p(f(g(g(a))))$ —— ⑧

⑦, ⑧ より

$\neg p(f(g(f(g(g(a))))))$ —— ⑨

①, ⑨ より

0

以上より，空節が導き出されたので E' は充足不能

(3)

- 反射律

  条件より，$\forall\, x \in X \cup Y$ について (x, x) が成り立つので，

  反射律を満たす．

- 反対称律

  $(x, y) \in R \wedge (y, x) \in R$ が成り立つとき，条件より，

  ①$x, y \in X$ かつ $x, y \in Y$　もしくは

  ②$x = y$ かつ $x, y \in X \cup Y$

  が成立する可能性が考えられる．

  しかし，$X \cap Y = \Phi$ より，① は成立しえないことがわかるので，

  $(x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$ となる．

  したがって，反対称律を満たす．

- 推移律

  $(x, y) \in R \wedge (y, z) \in R$ が成り立つとき，条件より，

  ①$x, y \in X$ かつ $y, z \in Y$　もしくは

  ②$x = y$ かつ $x, y \in X$ かつ $z \in Y$　もしくは

  ③$x \in X$ かつ $y = z$ かつ $y, z \in Y$　もしくは

  ④$x = y = z$

  が成立する可能性が考えられる．

  しかし，$X \cap Y = \Phi$ より，① は成立しえないことがわかる．

  また，②または③ が成立するときは，

  $x \in X$ かつ $z \in Y$ より，$(x, z) \in R$ が成立することがわかる．

  ④ が成立するときは，

  R が反射性を持つことから，$(x, z) \in R$ が成り立つことがわかる．

  したがって，$(x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$ となるので，

  推移律を満たす．

以上より，
反射律，反対称律，推移律を満たすので，R は順序関係である．

(4-1)

①$L_n(a)$ について

$L_n(a)$ は，L_{n-1} の最後に a を付け加えたものである.

したがって，

$|L_n(a)| = |L_{n-1}|$

②$L_n(b)$ について

$L_n(b)$ は，L_{n-1} の最後に b を付け加えたもののうち，bb を含まないものである.

ここで，L_{n-1} が bb を含まないことは自明なので，

”L_{n-1} の最後に b を付け加えたもののうち bb を含まないもの”は，

”$L_{n-1}(a)$ の最後に b を付け加えたもの”と一致する.

以上より，

$$\begin{aligned}|L_n(b)| &= |L_{n-1}(a)| \\ &= |L_{n-2}| \quad (\because |L_n(a)| = |L_{n-1}|)\end{aligned}$$

(4-2)

L_n は，$L_n(a)$ と $L_n(b)$ を足し合わせたもの，

すなわち，$L_n = L_n(a) \cup L_n(b)$ である.

また，$L_n(a) \cap L_n(b) = \Phi$ は自明.

以上より，

$|L_n| = |L_n(a)| + |L_n(b)| = |L_{n-1}| + |L_{n-2}|$

したがって，漸化式は

$|L_n| - |L_{n-1}| - |L_{n-2}| = 0$

$\boxed{4}$ 計算理論

(1)

プッシュダウンオートマトンを作って受理できるかどうか考える

L_1：×　L_2：○　L_3：×　L_4：×　L_5：○　L_6：×

(2)

$$\begin{cases} E \to T \mid E + T \\ T \to F \mid T * F \\ F \to I \mid (E) \\ I \to a \mid b \mid Ia \mid Ib \end{cases}$$

- F に対する単記号規則を除去
  単記号規則 $F \to I$ を置き換える
  $F \to a \mid b \mid Ia \mid Ib \mid (E)$
- T に対する単記号規則を除去
  単記号規則 $T \to F$ を置き換える
  $T \to a \mid b \mid Ia \mid Ib \mid (E) \mid T * F$
- E に対する単記号規則を除去
  単記号規則 $E \to T$ を置き換える
  $E \to a \mid b \mid Ia \mid Ib \mid (E) \mid T * F \mid E + T$　(答)

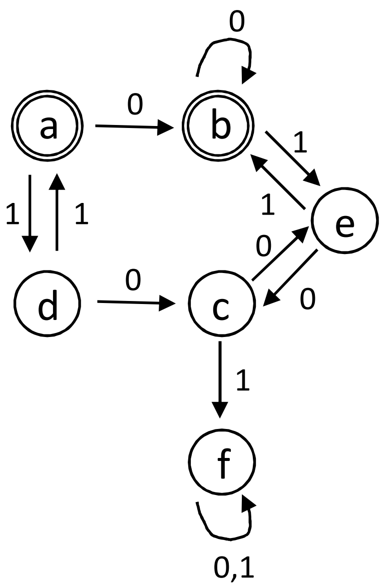
(3)

正規文法と文脈自由文法の比較：構文解析アルゴリズムにおいて，正規文法は自然言語を取り扱うことができないが，文脈自由文法を用いれば，それらを取り扱うことができるようになる．

文脈依存文法と文脈自由文法の比較：文脈依存文法は文字列が文法によって生成できるかどうかの判定に膨大な計算時間を要する．一方，文脈自由文法はそうはならない．

(4-1)

2 進数として見たとき，3 の倍数となるような記号列である．
これは，正規表現で表すと $(0^*(1(00)^*1)^*0^*)^*$ となる．
これを基にオートマトンを作成すると下図のようになる．
(ア) b (イ) e (ウ) e (エ) f (オ) c (カ) b (キ) f (ク) f



(4-2)

$k = 1$ のとき 010^*
$k = 2$ のとき 00110^*
$k = 3$ のとき 0001110^*
$\cdots$となる．
したがって，これは，$0^k1^k0^*(k > 0)$ を受理することがわかる．
(サ) (0,A)/AA (シ) (1,A)/ε (ス) (0,ε)/ε