

# 言語処理工学 A 期末テスト

2020年11月27日

井上克郎

ノート教科書持ち込みなし。[1]は解答用紙表紙、[2]は折りたたんだ内側2枚に、[3]は裏表紙に解答を書くこと。間違った場所を書いた場合は減点する。

[1] 次のコードは、ある簡単な PASCAL プログラムを CASL にコンパイルしたものである。元の PASCAL プログラムを書け。ユーザー定義識別子名（プログラム、関数、変数名など）は適当に決めて良い。また、このプログラムの実行結果（出力）を書け。（20点）

```

CASL  START  BEGIN
BEGIN  LAD    GR6, 0 ; reset write buffer
      LAD    GR7, LIBBUF ; set output buffer address
      LAD    GR5, 0 ; set frame pointer FP(GR5) null
      JUMP   LLMAIN ; go to the main body
; beginning of subprogram
LLfx  NOP    ; entry of subprogram
      LD     GR1,1,GR5 ; get a formal parameter
      LD     GR2,GR1
      CALL   WRTINT ; write an integer in GR2
      CALL   WRTLN ; write return
      LD     GR1,2,GR5 ; get a formal parameter
      PUSH   0,GR1 ; start +
      LD     GR1,1,GR5 ; get a formal parameter
      LD     GR2,GR1
      POP    GR1
      ADDA   GR1,GR2 ; End of +
      ST     GR1,ZZi ; complete assignment
      RET
; end of subprogram
LLMAIN NOP ; entry of main body
      LAD    GR1,1 ; set integer constant
      PUSH   0,GR1 ; set an actual parameter to stack top
      LAD    GR1,2 ; set integer constant
      PUSH   0,GR1 ; set an actual parameter to stack top
      PUSH   0,GR5 ; save FP (GR5) to stack
      LD     GR5,GR8 ; set new FP (GR5)
      CALL   LLfx
      POP    GR5 ; restore FP(GR5)
      LAD    GR8,2,GR8 ; discard parameters
      LD     GR1,ZZi ; load global variable
      LD     GR2,GR1
      CALL   WRTINT ; write an integer in GR2
      CALL   WRTLN ; write return
      RET
; start of global variable area
ZZi   DS    1
; end of global variable area
LIBBUF DS    256
      END
; lib.cas
...

```

問題は裏面に続く

- r, r1, r2 は、いずれも GR を表す。指定できる GR は GR0~GR8.
- adr は、アドレスを示す。指定できる値の範囲は 0~65535.
- x は、指標レジスタとして用いる GR を示す。指定できる GR は GR1~GR8.

表 1: ロード、ストア、ロードアドレス命令

命令	書き方		命令の説明	FR の設定
	命令コード	オペランド		
ロード	LD	r1, r2	r1 ← (r2)	○+1
Load		r, adr [, x]	r ← (実効アドレス)	
ストア	ST	r, adr [, x]	実効アドレス ← (r)	-
ロードアドレス	LAD	r, adr [, x]	r ← 実効アドレス	

表 2: 算術、論理演算命令

命令	書き方		命令の説明	FR の設定
	命令コード	オペランド		
算術加算	ADDA	r1, r2	r1 ← (r1) + (r2)	-
ADD Arithmetic		r, adr [, x]	r ← (r) + (実効アドレス)	
プッシュ	PUSH	adr [, x]	SP ← (SP) -L 1, (SP) ← 実効アドレス	-
ポップ	POP	r	r ← ((SP)) (SP) ← (SP) +L 1	
コール	CALL, subroutine	CALL adr [, x]	SP ← (SP) -L 1, (SP) ← (PR), PR ← 実効アドレス	-
リターン	RET		PR ← ((SP)) (SP) ← (SP) +L 1	

[2] 次の 3 番地コードに関して答えよ。(解答用紙 2 - 3 ページに答えを書くこと)  
(各 10 点)

- ①  $s = 0$
- ②  $a = 1$
- ③  $b = 2$
- ④ L1:  $s = s + b$
- ⑤ `if b<10 goto L3`
- ⑥ L2:  $a = a + 1$
- ⑦ `goto L1`
- ⑧ L3:  $a = b + 3$
- ⑨ `if s<=20 goto L2`
- ⑩  $s = s + 1$

- (1) このプログラムの基本ブロックに分け、プログラムの上から順に、各ブロックに番号 B1 ~ をつけ、各ブロックに入るコードを行番号で明示せよ。
- (2) 各基本ブロックを頂点としたフローグラフを書け。
- (3) 得られたフローグラフの支配木 (dominator tree) を書け。
- (4) フローグラフ中の各バックエッジを挙げ、それぞれが構成するループの各頂点を示せ。
- (5) 各基本ブロックの Gen 集合、Kill 集合を求めよ。
- (6) 各基本ブロックの入り口 (IN)、出口 (OUT) で出現しうる変数定義の行番号の集合 (データフロー方程式の解) を示せ。(途中結果も示せ)

[3] 上記 [2] のプログラムにおいて、最適化が適用できる 2 箇所を探し、それぞれについて、どのような書き換えが可能か、また、なぜそれが安全にできるか、説明せよ。[2] の結果を使っても使わなくてもよい。(4 ページめに書くこと) (10 点 × 2)

解答例

[1]

以下のようなプログラム。

```
program test(input, output);
```

```
var    i : integer;
```

```
procedure fx(m,n : integer);
```

```
begin
```

```
    writeln(n);
```

```
    i:= m+n
```

```
end;
```

```
begin
```

```
    fx(1,2);
```

```
    writeln(i)
```

```
end.
```

-----

実行結果

2

3

[2] 下に添付

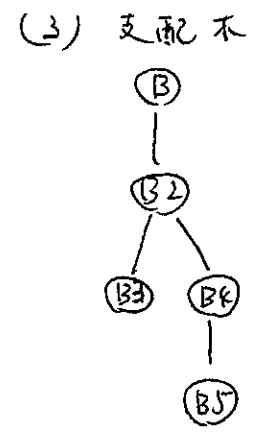
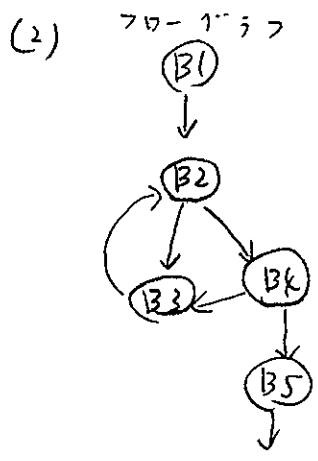
[3] 例えば以下のような解答

\* B2 の入り口で b に関する定義は③のみなので④の b を 2 と定数に書き換えることが可能。

\* 同様に⑧は  $a=2+3$  と書き換えられ、その計算を実行して  $a=5$  と書き換えられる。

\* 同様に⑤は  $2<5$  となり無条件分岐の L3 へと書き換え可能

- (1)
- B1: ① ② ③
  - B2: ④ ⑤
  - B3: ⑥ ⑦
  - B4: ⑧ ⑨
  - B5: ⑩



- (4) B → I, J: 10-70  
B3 → B2: {B2, B3, B4}

(5)

Blk#	Gen	Kill
1	1 2 3	4 6 8 10
2	4	1 10
3	6	2 8
4	8	2 6
5	10	1 4

(6)

step 0 (initial)			step 1			step 2		
Blk#	IN	OUT	Blk	IN	OUT	Blk	IN	OUT
1	-	1 2 3	1	-	1 2 3	1	-	1 2 3
2	-	4	2	1 2 3 6	2 3 4 6	2	1 2 3 4 6	2 3 4 6
3	-	6	3	4 8	4 6	3	2 3 4 6 8	3 4 6
4	-	8	4	4	4 8	4	2 3 4 6	3 4 8
5	-	10	5	8	8 10	5	4 8	8 10

step 3			step 4		
Blk	IN	OUT	Blk	IN	OUT
1	-	1 2 3	1	-	1 2 3
2	1 2 3 4 6	2 3 4 6	2	1 2 3 4 6	2 3 4 6
3	2 3 4 6 8	3 4 6	3	2 3 4 6 8	3 4 6
4	2 3 4 6	3 4 8	4	2 3 4 6	3 4 8
5	3 4 8	3 8 10	5	3 4 8	3 8 10

同じよ、2 停止