

言語処理工学 A 期末テスト

2019年11月29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。間違った場所に書いた場合は減点する。

[1] 次のコードは、教科書のCのサンプルコンパイラのwhile文の処理部分である。

構文は、`"while" "(" Expression ")" Statement`

(1)~(7)に入る行を、下記の(A)~(J)の候補の中から最適な1つを選んでプログラムを完成させよ。(各3点、最大20点)

```
void while_st(){
    int blab, clab;
    if(tok != SLPAREN) error(" '(' Expected");
    tok = scan(); /* get next token */
    (1)      ;
    (2)      ;
    (3)      ;
    gen_code(" TST.W R0); /* Expect R0 holds the result */
    (4)      ;
    gen_code(" BEQ L%d",blab); /* Zero means false */
    if(tok != SRPAREN) error(" ')' Expected");
    tok = scan();
    (5)      ;
    (6)      ;
    (7)      ;
}
```

- (A) `st(blab,clab) /* Statement */`
- (B) `exp() /* Expression */`
- (C) `blab = get_inlabel() /* Call label generator */`
- (D) `clab = get_inlabel() /* Call label generator */`
- (E) `gen_code(" BNE L%d", clab) /* branch not equal */`
- (F) `gen_code(" BNE L%d", blab)`
- (G) `gen_code(" BRA L%d", clab) /* branch always */`
- (H) `gen_code(" BRA L%d", blab)`
- (I) `gen_code("L%d:", blab)`
- (J) `gen_code("L%d:", clab)`

問題は裏面に続く

[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      a = 0
2      c = 0
3  L0:  b = 0
4  L1:  a = 1
5      b = b+a
6      if b<=10 goto L1
7      c = c+b
8      c = c+b
9      if c<=100 goto L0
10     exit
```

- (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。データフロー方程式を立てて、それを漸近的に解く途中経過も示せ。

[3] 上記 [2] のプログラムで、最適化が適用できる例を 2 つ示し、どのような最適化か、また、なぜそれが可能か、それぞれ説明せよ。[2] の結果を使っても使わなくてもよい。
(4 ページに書くこと) (10 点 × 2)

解答例

言語処理工学 A 期末テスト

2019年11月29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。間違った場所に書いた場合は減点する。

[1] 次のコードは、教科書のCのサンプルコンパイラのwhile文の処理部分である。

構文は、`"while" "(" Expression ")" Statement`

(1)~(7)に入る行を、下記の(A)~(J)の候補の中から最適な1つを選んでプログラムを完成させよ。(各3点、最大20点)

```
void while_st(){
    int blab, clab;
    if(tok != SLPAREN) error(" '(' Expected");
    tok = scan(); /* get next token */
    (1) ; (D)
    (2) ; (J)
    (3) ; (B)
    gen_code(" TST.W R0); /* Expect R0 holds the result */
    (4) ; (C)
    gen_code(" BEQ L%d",blab); /* Zero means false */
    if(tok != SRPAREN) error(" ')' Expected");
    tok = scan();
    (5) ; (A)
    (6) ; (G)
    (7) ; (I)
}
```

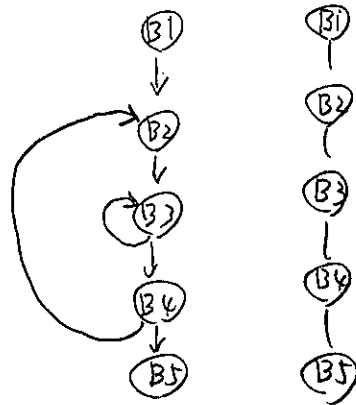
```
(A) st(blab,clab) /* Statement */
(B) exp() /* Expression */
(C) blab = get_inlabel() /* Call label generator */
(D) clab = get_inlabel() /* Call label generator */
(E) gen_code(" BNE L%d", clab) /* branch not equal */
(F) gen_code(" BNE L%d", blab)
(G) gen_code(" BRA L%d", clab) /* branch always */
(H) gen_code(" BRA L%d", blab)
(I) gen_code("L%d:", blab)
(J) gen_code("L%d:", clab)
```

問題は裏面に続く

[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)

```

Leader
B1 1 [ a = 0 *
    2 [ c = 0
B2 3 [ L0: b = 0 *
    4 [ L1: a = 1 *
B3 5 [ b = b+a
    6 [ if b<=10 goto L1
    7 [ c = c+b *
B4 8 [ c = c+b
    9 [ if c<=100 goto L0
B5 10 [ exit *
  
```



Back edge loop.
 $B3 \rightarrow B3 : \{ B3 \}$
 $B4 \rightarrow B2 : \{ B2, B3, B4 \}$

- (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。データフロー方程式を立てて、それを漸近的に解く途中経過も示せ。

次のページ

[3] 上記 [2] のプログラムで、最適化が適用できる例を 2 つ示し、どのような最適化か、また、なぜそれが可能か、それぞれ説明せよ。(4 ページに書くこと) (10 点 × 2)

- ① 4 行目 $a := 1$ の定義を 5 行目に伝搬させ、5 行目を $a = a + 1$ に書き換える。この 2 行目同一基本ブロック内で、他方の a の定義は 4 行目の a の定義により殺されるために可能。
- ② 1 行目の a の定義は不要。 a の唯一の参照は 5 行目。だが 4 行目で 1 は殺されるので、参照されるので不要。
($B3 \rightarrow B3$ のループ外に出す、のも OK)

[2] (6)

Gen Kill

B1	{1, 2}	{4, 7, 8}
B2	{3}	{5}
B3	{4, 5}	{1, 3}
B4	{8}	{2, 7}
B5	\emptyset	\emptyset

	step 0		step 1		step 2	
	IN	OUT	IN	OUT	IN	OUT
B1	\emptyset	{1, 2}	\emptyset	{1, 2}	\emptyset	{1, 2}
B2	\emptyset	{3}	{1, 2, 8}	{1, 3, 8}	{1, 2, 4, 5, 8}	{1, 2, 3, 4, 8}
B3	\emptyset	{4, 5}	{3, 4, 5}	{4, 5}	{1, 2, 3, 4, 5, 8}	{2, 4, 5, 8}
B4	\emptyset	{8}	{4, 5}	{4, 5, 8}	{4, 5}	{4, 5, 8}
B5	\emptyset	\emptyset	{8}	{8}	{4, 5, 8}	{4, 5, 8}

	step 3		step 4	
	IN	OUT	IN	OUT
B1	\emptyset	{1, 2}	\emptyset	{1, 2}
B2	{1, 2, 4, 5, 8}	{1, 2, 3, 4, 8}	{1, 2, 4, 5, 8}	{1, 2, 3, 4, 8}
B3	{1, 2, 3, 4, 5, 8}	{2, 4, 5, 8}	{1, 2, 3, 4, 5, 8}	{2, 4, 5, 8}
B4	{2, 4, 5, 8}	{4, 5, 8}	{2, 4, 5, 8}	{4, 5, 8}
B5	{4, 5, 8}	{4, 5, 8}	{4, 5, 8}	{4, 5, 8}

ENC → step 4 停止