

言語処理工学 A 期末テスト

2017年12月1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、教科書のCのサンプルコンパイラのif文の処理部分である。①～⑧を適当なものを補って完成させよ。なおif文の構文は `if (exp) st` もしくは `if (exp) st else st` で、`exp` は条件式、`st` は文(statement)を表す非終端記号である。

```
void if_st(blab, clab)
    int blab, clab;
    {
    int L1,L2;

    if(tok != SLPAREN) error(" '(' Expected");
    tok =scan();
    exp();
    L1 = get_inlabel();
    gen_code(" TST.W R0");
    gen_code(" BEQ L%d",①);
    if(tok != SRPAREN) error(" ')' Expected");
    tok = ②;
    st(blab, clab);
    if(③ == SELSE) {
        tok = scan();
        L2 = ④;
        gen_code(" BRA L%d",⑤);
        gen_code("L%d:",⑥);
            st(blab,clab);
        gen_code("L%d:",⑦);
    }
    else gen_code("L%d:",⑧);
    }
```

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)

```
1      i=1
2      t=1
3      s=0
4      if i=10 goto 8
5      s=s+i
6      i=i+t
7      goto 4
8      print s
9      print t
```

(注:print は引数を入力するマクロで、引数を参照するのみで変更しない)

- (1) このプログラムの基本ブロックに分け、フローグラフを書け。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) 得られたフローグラフの支配木 (dominator tree) を書け。
- (3) 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。
- (4) 各基本ブロックの Gen 集合、Kill 集合を求めよ。
- (5) 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] の結果から、最適化が適用できる行を列挙して、それぞれ、どのような最適化ができるか、また、なぜそれが可能か、[2] の結果を用いて説明せよ。(4 ページに書くこと)

解答

[1]

```
void if_st(blab, clab)
    int blab, clab;
    {
    int L1,L2;

    if(tok != SLPAREN) error(" '(' Expected");
    tok =scan();
    exp();
    L1 = get_inlabel();
    gen_code(" TST.W R0");
    gen_code(" BEQ L%d",L1);
    if(tok != SRPAREN) error(" ')' Expected");
    tok =scan();
    st(blab, clab);
    if(tok == SELSE) {
        tok = scan();
        L2 = get_inlabel();
        gen_code(" BRA L%d",L2);
        gen_code("L%d:",L1);
        st(blab,clab);
        gen_code("L%d:",L2);
    }
    else gen_code("L%d:",L1);
    }
```

[3]

(1) 6行目のtは、B2のINによると2行目の定義のみの参照であるので、1であることがわかる。したがって

```
6    i=i+1
```

と置き換えることができる。

(2) さらに置き換えられた6行目の番地コードは

```
6    inc(i)
```

という機械語に変換して効率よく実装が可能である。

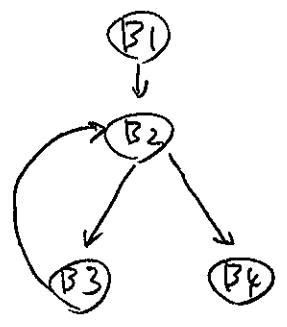
(3) 同様に9行目の `print t` はやはりtの値は2行目の定義が参照されるので1であり、`print 1`で置き換えることが可能である。

(4) さらに上記(1), (3)の結果、2行目の `t` の定義を参照してる文がなくなるので、2行目の削除が可能になる。

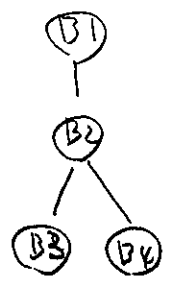
L2

(1)

行番号
 B1: 1~3
 B2: 4
 B3: 5~7
 B4: 8~9



(2)



(3) バックエッジ B3 → B2
 IL → {B2, B3}

(4)

	Gen	Kill
B1	1, 2, 3	5, 6
B2	∅	∅
B3	5, 6	1, 3
B4	∅	∅

(5)

Step	IN	OUT	Step	IN	OUT
Step 0	∅	1, 2, 3	Step 1	∅	1, 2, 3
B1	∅	1, 2, 3	B1	∅	1, 2, 3
B2	∅	∅	B2	1, 2, 3, 5, 6	1, 2, 3, 5, 6
B3	∅	5, 6	B3	∅	5, 6
B4	∅	∅	B4	∅	∅
Step 2	IN	OUT	Step 3	IN	OUT
B1	∅	1, 2, 3	B1	∅	1, 2, 3
B2	1, 2, 3, 5, 6	1, 2, 3, 5, 6	B2	1, 2, 3, 5, 6	1, 2, 3, 5, 6
B3	1, 2, 3, 5, 6	2, 5, 6	B3	1, 2, 3, 5, 6	2, 5, 6
B4	1, 2, 3, 5, 6	1, 2, 3, 5, 6	B4	1, 2, 3, 5, 6	1, 2, 3, 5, 6

IL 5, 2 step 4 停止