

# CS126: Introduction to Compilers

## Assignment 9: Graph Algorithms

Out: 11/14/03

Due: 11/21/03

### 1.0 Introduction

A good way of learning exactly what the various optimization algorithms do is to work them out by hand on a relatively simple (although non-trivial) example.

Consider the following Decaf method:

```
public static int[][] makePascalTable(int n1, int n2)
{
    int[][] tab = new int[n1][n2];

    int i = 0;
    while (i < n1) {
        int j = 0;
        while (j < n2) {
            if (i == 0 || j == 0)
                tab[i][j] = 1;
            else
                tab[i][j] = tab[i-1][j] + tab[i][j-1];
            j = j + 1;
        }
        i = i + 1;
    }
    return tab;
}
```

This method generates the following intermediate code:

```
Code for routine Pascal$makePascalTable (4) :
Block_1:
    ENTER        #2
    PARAM       #0,T2
    PARAM       #1,T3

Block_2:
    LDC         #2,T6
    LDC         #2,T6
    CALLR      #"_ArrayAllocate",T6,T6,T2,T3,T5
    MOVE       T5,T4
    LDC         #0,T8
    MOVE       T8,T7

Block_3:
    CMP        T7,T2,T9
    BCOND      T9,Block_5,Block_4

Block_4:
    MOVE       T4,T1
    RETVAL    T1
    BRANCH    Block_13
```

```

Block_5:
    LDC        #0,T8
    MOVE      T8,T10

Block_6:
    CMP T10,T3,T11
    BCOND     T11,Block_8,Block_7

Block_7:
    LDC        #1,T22
    iADD      T7,T22,T37
    MOVE      T37,T7
    BRANCH   Block_3

Block_8:
    LDC        #0,T8
    CMP T7,T8,T12
    BCOND     T12,Block_10,Block_9

Block_9:
    LDC        #0,T8
    CMP T10,T8,T13
    BCOND     T13,Block_10,Block_12

Block_10:
    LDC        #4,T14
    LDC        #4,T14
    aADD      T4,T14,T15
    iMUL      T7,T14,T16
    aADD      T15,T16,T17
    LD        [T17],T18
    LDC        #4,T14
    LDC        #4,T14
    aADD      T18,T14,T19
    iMUL      T10,T14,T20
    aADD      T19,T20,T21
    LDC        #1,T22
    MOVE      T22,T23
    ST        T23,[T21]

Block_11:
    LDC        #1,T22
    iADD      T10,T22,T36
    MOVE      T36,T10
    BRANCH   Block_6

Block_12:
    LDC        #4,T14
    LDC        #4,T14
    aADD      T4,T14,T15
    iMUL      T7,T14,T16
    aADD      T15,T16,T17
    LD        [T17],T18
    LDC        #4,T14
    LDC        #4,T14
    aADD      T18,T14,T19
    iMUL      T10,T14,T20
    aADD      T19,T20,T21
    LDC        #1,T22
    iSUB      T7,T22,T24
    LDC        #4,T14
    LDC        #4,T14
    aADD      T4,T14,T15
    iMUL      T24,T14,T25

```

```

aADD      T15, T25, T26
LD        [T26], T27
LDC       #4, T14
LDC       #4, T14
aADD      T27, T14, T28
iMUL     T10, T14, T20
aADD      T28, T20, T29
LDC       #4, T14
LDC       #4, T14
aADD      T4, T14, T15
iMUL     T7, T14, T16
aADD      T15, T16, T17
LDC       #1, T22
iSUB     T10, T22, T30
LD        [T17], T18
LDC       #4, T14
LDC       #4, T14
aADD      T18, T14, T19
iMUL     T30, T14, T31
aADD      T19, T31, T32
LD        [T32], T33
LD        [T29], T34
iADD     T34, T33, T35
MOVE     T35, T23
ST       T23, [T21]
BRANCH   Block_11

Block_13:
EXIT

Block_14:
BRANCH   Block_13

```

Given this:

1. Draw the corresponding graph of basic blocks.
2. Identify any critical edges in the flow graph and remove them appropriately; draw the resultant graph.
3. Compute and draw the dominator tree of the resultant graph.
4. For each temporary  $T$  compute  $LIVEUSE(T)$ , the set of blocks containing a use of the temporary  $T$  not preceded by an evaluation of  $T$ .
5. For each block  $B$  compute  $LIVEKILL(B)$ , the set of temporaries  $T$  for which there is an evaluation of  $T$  in  $B$ .
6. For each temporary  $T$  compute  $LIVEIN(T)$ , the set of blocks  $B$  where the temporary  $T$  is live at the start of  $B$ .
7. For each basic block  $B$  compute  $LIVEOUT(B)$ , the set of temporaries live at the end of block  $B$ .
8. Compute the set  $GLOBAL$  of temporaries that are live at the start of some block.