

# Compiler Design

## Mid-Term 1

### Dataflow Analysis

March 3 2006, Time 90 minutes, Open book

all questions carry equal credit

---

1. The following three statements all pertain to partial-redundancy-elimination. For each statement say whether the statement is true or false. If it is true, explain carefully why it is true. If it is false give a counterexample to illustrate falsity.

1. PRE never increases program size, that is, it always removes at least as many occurrences of an expression as it adds.
2. Consider a basic block  $B$  with only one predecessor which is transparent wrt the expression  $e$ . If PRE inserts an evaluation of  $e$  in  $B$ 's sole predecessor, then an evaluation of  $e$  will never be inserted into  $B$ .

2. We say that an expression of the form  $x = a \text{ op } b$  in node  $s$  of the control graph *reaches* node  $n$  if there is a path from  $s$  to  $n$  that does not go through any assignment to  $a$  or  $b$  or through any computation of  $a \text{ op } b$ . Give the expressions for *gen* and *kill* for reaching expressions and formulate the dataflow equations.

3. In which of the three programs below can  $t = a+b$  be hoisted out of the loop. Give reasons.

```
L0: t = 0
L1: i = i+1
    t = a+b
    m[i] = t
    if i < N goto L1
L2: x = t
```

```
L0: t = 0
L1: if i geq N goto L2
    i = i+1
    t = a+b
    m[i] = t
    goto L1
L2: x = t
```

```
L0: t = 0
L1: i = i+1
    t = a+b
    m[i] = t
    t = 0
    if i < N goto L1
L2:
```

4. For the following flowgraph compute the dominator tree and identify the natural loops.

