1 Do you notice that each usage of local variable as an operand uses a

suffix that can be tracked to it's assignment? Why do some variables

have a suffix `(D)'?

Ans Local variables are suffixed with a unique number which represents

the assignment. For example, `d\_3' refers to the usage of variable

`d' and the only assignment which reaches this point is the

assignment `d\_3'. Variables that are likely to be used before any

assignment are suffixed with `(D)' to indicate that the assignment

was implicit in the declaration. This happens to both uninitialized

local variables and formal parameters. For example, variable `c' is

used as `c\_4(D)' in the `if' condition.

2 Find the point in the CFG in which the paths of the two assignments

to variable `a' and `b' merge. You will notice that PHI statements

have been introduced. Can you guess the syntax of the PHI

statements?

Ans A PHI node merges assignments from different paths and creates a new

assignment. For example:

# a\_1 = PHI <a\_5(3), a\_7(4)>

indicates that if the control reaches here from basic block 3 then

use the value of a\_5; otherwise if the control reaches here from

basic block 4 then use the value of a\_7.

3 Can you think of an example in which a PHI node merges more than two

versions of a variable?

Ans A switch statement can result in a merge of more than two paths (and

hence possibly more than two assignments). Consider:

int main()

{

int a, b;

switch (a)

{

case 1:

b = 2;

break;

case 2:

b = 3;

break;

case 3:

b = 4;

case 4:

b = 5;

break;

case 5:

case 6:

b = 6;

break;

default:

b = 7;

}

return b;

}