Dominance frontiers and SSA optimizations

bogdan.pasca@ens-lyon.fr

23 Nov 2010

1 Shreedar-Gao's Dominance Frontier Algorithm

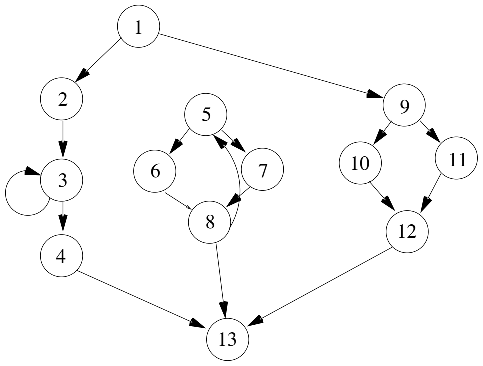

FIGURE 1 – CFG

Q 1.1 Compute the dominator tree starting with the control flow graph presented in figure 1.

Q 1.2 Apply the *Shreedar-Gao's Dominance Frontier Algorithm*, briefly sketched below, to compute the dominance frontier for node 5 in figure 1.

```
DF[x] = empty set
for each y in SubTree(x) do
     if(( arrow(y,z) == J-edge) and
                 (z.level <= x.level))
       then add z to DF[x]
```

Q 1.3 Think about the algorithm's implementation, knowing that it uses a work list of nodes hashed by their level in the dominator tree and a visited flag to avoid visiting the same node more than once.

2 SSA optimizations

Now that we know where to place the ϕ functions, let's apply the code optimization techniques under SSA on the following example :

```
i=1;
j=1;
k=0;
while(k<100)
     {
       if(j<20)
          {
            j=i;
            k=k+1;
          }
       else
          {
            j=k;
            k=k+2;
          }
     }
return j;
}
```

Q 2.1 Build the CFG

Q 2.2 Compute the dominance tree

Q 2.3 Find the dominance frontiers for placing the ϕ functions

Q 2.4 Apply constant propagation

Q 2.5 Use the additional reading on Conditional Constant Propagation to eliminate dead code.

Q 2.6 Once this is done, eliminate single-argument ϕ functions, and continue until ...