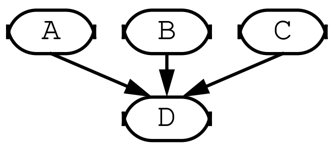
Procedure specialization

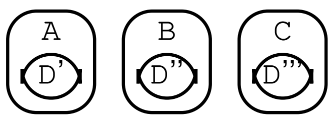
A.k.a. procedure cloning, customization

Halfway between inlining and interprocedural analysis

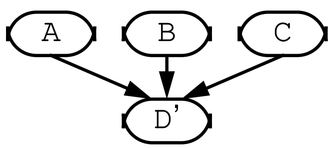
Source call graph:



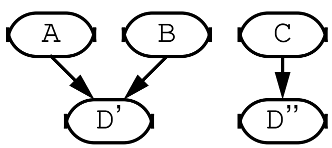
Inlining:



Interprocedural analysis:



Procedure specialization:



Abstract process

Given set of call sites of procedure P

e.g. $\{c_1, c_2, c_3, c_4, c_5\}$

Partition into equivalence classes of “similar” call sites

e.g. $\{\{c_1, c_2\}, \{c_3, c_4\}, \{c_5\}\}$

Copy P for each class, change calls accordingly

Do interprocedural analysis on changed call graph

Versus inlining:

+ less code explosion
+ works in presence of recursion

Versus interprocedural analysis:

+ better optimization

When to specialize?

A goal-directed algorithm for forward data flow problems,
e.g. constant propagation [Cooper, Hall, & Kennedy 92]

Step 1: propagate "cloning vectors" of available data flow information top-down through call graph

- each vector represents info available along some path through call graph
- can consider only "important" variables, to reduce cost

Step 2: merge equivalent cloning vectors, with bottom-up pass through call graph

- merge vectors that lead to same local optimizations and aren't helpful to callees

Step 3: perform cloning

Results?