Homework 1: Finding Dominators
Paul Vines and Eric Mullen
CSE 501

We implemented the simplest algorithm for finding dominators within our code, in the interest of it being simple to stare at and verify that it was correct. As per our discussions in class, the calculation of the dominator is not a large part of compilation time, and thus is not something to be particularly practically concerned with how fast it runs, more that it is implemented correctly.

We implemented our basic compiler framework and calculation of dominators in Scala, which means you'll need an up to date Scala compiler to run it.

Usage instructions:

1. Go into the `code` directory included in the tarball, and compile everything with `scalac *.scala`
2. Included in the `bin` directory is a script called `startc`. This takes a dart program, translates it to SIR (Start Intermediate Representation), and then feeds that SIR to our optimizer. At this stage of the project our optimizer only computes dominators, but in the future it will perform more interesting optimizations.
3. Be sure to edit the `startc` script to update the path variables, as the paths on your machine are of course different than those on ours.

For an analysis we took the included `loop` dart program, and wrote a script to duplicate it some number of times. We then ran the duplicated program through our dominator calculator, and included is `benchmark.ods`, which shows number of duplications versus runtime. We realize this is not a particularly good test of runtime of complicated, crazy graphs, but it does give us a little bit of an idea of how the program scales. As expected in this sense, the runtime scales linearly with the number of duplications.