

322 Compilers Assignment: graph-test

Due Thursday, April 23rd, noon

Your job: Design (at least) 15 test cases for graph coloring and graph construction as a pair of files, an L2 input function and the expected output. This is the shape of the coloring function:

```
graph-color : (i ...) -> ((var var ...) ...)
                ((var reg) ...) or #f
```

The `graph-color` function accepts an L2 function (as a list of instructions), and returns two values: a graph to color and either a coloring for that graph or `#f`, indicating the algorithm failed to color the graph.

The graph should be printed out as an adjacency table, mapping each variable to its neighbors. Sort each sequence of neighbors alphabetically by the name of the variable and sort the entire graph by the names of the variables for each node. The graph should include all of the registers except `ebp` and `esp`.

The coloring should be printed out as a sequence of pairs, mapping each non-register variable to a register (color), sorted by the name of the variable. If all of the vertices in the graph printed as the first result have fewer than 5 neighbors, then the result must not be `#f`. Otherwise, it should only be `#f` when the graph cannot be colored (but the checker doesn't check this aspect).

The `graph-color` function should be wrapped up into a script that accepts a filename naming a file that contains the arguments in the file. The script should write their answers to `stdout`.

For example, if the file `f.L2f` contains:

```
(:f (x <- 1) (eax += x) (return))
```

Then this transcript shows how your script might behave (the `ebx` in the last line might be a different register):

```
% graph-color f.L2f
((eax ebx ecx edi edx esi x)
 (ebx eax ecx edi edx esi)
 (ecx eax ebx edi edx esi)
 (edi eax ebx ecx edx esi x)
 (edx eax ebx ecx edi esi)
 (esi eax ebx ecx edi edx x)
 (x eax edi esi))
((x ebx))
```

For example, if the file `g.L2f` contains:

```
(:g
 (rx <- eax)
 (rx += ebx) (rx += ecx) (rx += edx)
 (rx += edi) (rx += esi) (rx += eax))
```

then this is how your script must behave:

```
% graph-color g.L2f
((eax ebx ecx edi edx esi rx)
 (ebx eax ecx edi edx esi rx)
 (ecx eax ebx edi edx esi rx)
 (edi eax ebx ecx edx esi rx)
 (edx eax ebx ecx edi esi rx)
 (esi eax ebx ecx edi edx rx)
 (rx eax ebx ecx edi edx esi))
#f
```

Use `check-coloring` to make sure your colorings are sensible.

Hand in your assignment by uploading it to the server at <http://penghu.eecs.northwestern.edu:8123/>. The uploaded file should be a gzipped tar file named `name.graph-test.tar.gz`. The `name` should be your family name in all lowercase letters (except for the names Liu, Zhou, or Wang, see below) unless you are pair programming, in which case it should be both family names in alphabetical order, separated by `+`. If your name has any non-alphabetic characters, remove them first. For example, if Conan O'Brien and Shawn Knowles-Carter were pair programming and handing in this assignment, they'd send in a tarfile named `obrien+kowlescarter.graph-test.tar.gz`. If your family name is Liu, Zhou, or Wang, then include your first name as well, but also without any spaces. For example, if your name

is Liu Bolin, then use `bolinliu` as your name. And if Bolin and Shawn team up, they'd submit `bolinliu+kowlescarter.graph-test.tar.gz`.

The file must contain a single directory named `graph-test` containing the test cases.

The input files should use the suffix `.L2f` and the correct answers should use the suffix `.gres`. Your scripts must run on the t-lab machines (under linux).