

PLC: Workout 8 [80 points]

Due date: Friday, April 8th, by midnight

About This Homework

This assignment is about theorem proving in Agda for the booleans and also using simple equational reasoning. You will want to try installing Agda early in case of any difficulties. Instructions for this are below.

How to Turn In Your Solution

Follow this link to create your repo on Github Classrooms:

<https://classroom.github.com/a/gE27eMpz>

If you work with a partner, please add to your repo a file called `partners.txt`, and list your name and your partner's name there. Both of you should submit your solution: each partner pushes it to their github repo.

How To Get Help

You can post questions in the Discussions section of ICON.

You are also welcome to come to our office hours. See the course's Google Calendar, linked from the github page for the class, for times and Zoom links for office hours. You can also find these on the "Zoom links, office hours" page under "Pages" on ICON:

<https://uiowa.instructure.com/courses/179776/pages/office-hours>

1 Reading

Read Chapters 2 and 3 of Verified Functional Programming in Agda, available for free (on campus or VPN) here:

<https://dl-acm-org.proxy.lib.uiowa.edu/doi/book/10.1145/2841316>

2 Installing Agda

Agda is installed on the CS Windows computers. To remote in, see

<https://clas.uiowa.edu/linux/services/vmwareview>

But you will probably find it worth the effort to install Agda on your own computer. For Windows, the easiest thing is to use our installer:

<https://homepage.divms.uiowa.edu/~astump/agda/Agda2.6.0.1.v1.msi>

Otherwise, try following the directions here:

<https://agda.readthedocs.io/en/latest/getting-started/installation.html>

On that page, there is an explanation of what to do for Mac, under the heading “OS X”.

To install more directly, first make sure you have a graphical version of emacs installed (not just one running within your terminal; we have seen problems with that on Mac). There are versions of emacs for Windows and Mac, as well as Linux. Once you have emacs, you can try to install from your Haskell installation, by first doing `cabal install Agda` and then `agda-mode setup`.

2.1 If you have problems installing on Mac

If you are having trouble getting Agda working on Mac, it might be because for some reason, on at least some version of MacOS, it seems that emacs does not get the path that your terminal gets. So emacs cannot find the program `agda-mode`, and it also cannot find the program `agda`. These issues have to be fixed separately, in two steps:

1. in your terminal, run `"agda-mode locate"` (no quotes). If this fails, you have a different problem. If it succeeds, it will print out a rather long file path. Copy this carefully (from `"/Users/"` all the way to `".el"`) and then open your `.emacs` file in emacs. Instead of the code that you may already have in your `.emacs` file that looks like

```
(load-file (let ((coding-system-for-read 'utf-8))
  (shell-command-to-string "agda-mode locate")))
```

you want to put instead

```
(load-file (let ((coding-system-for-read 'utf-8))
  "THAT_BIG_PATH"))
```

where `THAT_BIG_PATH` should be replaced with the path that got printed out from running `agda-mode locate`.

After making this change to your `.emacs` file, quit emacs and restart it.

2. Open up `bool.agda` in the IAL. You should see (Agda) at the bottom, but probably will get some error message from emacs. This is because emacs cannot find the `"agda"` program. We can tell emacs where that program is as follows. Type `"Alt-x"` and then `"customize-group"` (no quotes). Hit enter. Then type `"agda2"` (again no quotes) and hit enter. You will see a

list of options. Near the bottom is one for "Agda Program Name". You want to change this one. You click on the little triangle, and you should see a text box that will say "agda". You want to change that to the full path for the agda program on your computer. You figure out what that path is by opening a terminal and typing "which agda" (no quotes), then enter. Copy that path to the little text box (replacing "agda"). Then click the button right under that line that says "State" and select "Save for future sessions". Close emacs and open it up again. If everything goes well, now typing Control-c Control-l in bool.agda should work with syntax highlighting.

3 Installing the Iowa Agda Library (IAL)

You clone the repo here from github:

<https://github.com/cedille/ial>

4 Configuring and testing Agda and the IAL [10 points]

Finally, you need to tell Agda how to find the Iowa Agda Library. If you are using a CS Windows machine, then open the file `h:/.emacs`. Otherwise, open `~/.emacs` in emacs (you type "Control-x Control-f ~/.emacs"). Add the following text, where instead of the word `PATH`, you should have the path to your copy of the IAL (wherever you put it):

```
(custom-set-variables
 '(agda2-program-args (quote ("--include-path=PATH"))))
```

That should be a single forward tick mark on the second line of that code (might render incorrectly in this PDF). On Windows, I found I could put backslashes if I escaped them (double backslash), like this (where `Myself` is, of course, your actual Windows username):

```
C:\\Users\\Myself\\Documents\\ial
```

To prove that all this is working for you, open `bool.agda` in the IAL and type Control-c Control-l to load the file with Agda. If this succeeds you should get syntax highlighting for the file. Now take a screenshot called `screenshot.png`, capturing your Emacs window with `bool.agda` highlighted. (I found that for some reason, Agda often says "Another command is currently in progress" when I do this, and I must first type Control-c Control-x Control-r to restart Agda, and then do Control-c Control-l.)

5 Boolean theorems [30 points]

In files `bools1.agda` through `bools6.agda`, you will find six lemmas to prove. I did one like this in class on March 31st, and will do a couple more on April 5th. When you load the file with Control-c Control-l, you will see holes on the right-hand sides of the definitions of those lemmas. Remove those holes (Control-k with your cursor right before the hole will cut it out), and fill the definitions

in with proofs. Just as for Haskell, you can (and generally must) write multiple equations to define the function (i.e., proof, under the Curry-Howard isomorphism) for different patterns of the inputs. [5 points each]

6 Simple equational theorems [40 points]

In files `eq1.agda` through `eq8.agda`, you will find eight problems similar to some I did March 31st in class. Prove these using equational reasoning (mostly `rewrite`). I will do some more April 5th. [5 points each]