PLC: Homework 1 [100 points, 120 max]

Deadline

If the first letter of your HawkId is between 'a' and 'k' inclusive, then your solution is due by 9pm Wednesday, February 5th. Otherwise, it is due by 9pm Friday, February 7th. Yes, this gives half the class a later deadline than the other half, but for the next homework, we will switch this (so 'l' through 'z' will have the earlier deadline). We are using two deadlines to help spread out the requests for help in office hours and in Piazza.

Late homework is **not** accepted. The class has a lot of extra-credit opportunities, and we are even experimenting with "second-chance" grading (see below). So even if you have to skip submitting a homework, it should not sink your grade for the class.

Must Turn In Using Subversion

Your solution to this homework must be turned in using via the personal subversion repository we have created for each person in the class. Also, the specific Agda code you need to look at is found in the course repository. See

`https://svn.divms.uiowa.edu/repos/clc/class/111-spring14/installing-and-using-svn.txt`

for instructions on how to download both repositories to your computer or to your directory on the CS lab machines, and how to install and/or use the TortoiseSVN client on Windows.

May Work with One Partner

You may work with one, and only one, partner on this assignment. Suppose Al and Ben are working together on this. Then Al (say) must submit the following using subversion:

- the actual solution (`bool-todo.agda`, explained below), and
- a file called `ack_partner.txt`, containing only Ben's HawkId.

If Al submits those files, then Ben must submit this file (using subversion):

- a file called `partner.txt` containing only your Al's HawkId.

You can create these `.txt` files use `emacs`, or another text editor like Notepad on Windows.

The point of the `partner.txt` file is so that we know to give Ben the same grade which we calculate when grading the submission in Al's directory. The point of the `ack_partner.txt` file is so that we know Al agrees ("acknowledges") that he and Ben worked together on this.

Important: if you work with a partner, you must both submit the required files by the earliest deadline that applies to one or the other of you. So if Al's HawkId would make his assignment due by the first deadline, then both Al and Ben must submit by then, even if Ben's HawkId would make his assignment due by the second deadline (if he were working by himself).

Second-Chance Grading

We will run our grading scripts and check in a grade-1.txt file to the hw1 subdirectory of your personal repository by noon on Monday, February 10th. To see your grade file, just do a subversion update on your hw1 directory (on Windows, right-click on the directory, then choose “TortoiseSVN” and then “Update”). You will then have 24 hours to make any further changes to your solution and commit it again with subversion. We will run our grading scripts again, to generate a grade-2.txt file in your personal repository. We will take the average of the grades you get for the first submission and the second-chance submission, as your final grade for the homework. To avoid confusion, please do not edit these grade files (we will store our own copies anyway).

While this allows you a chance to revise your submission (even solve problems you did not manage to solve by the original deadline), you do not have to make any changes if you do not wish to. In this case, we will just compute an identical grade-2.txt file, so the average grade will equal the grade you got on your original submission.

1 Reading

Read Chapters 1 and 2 of the book, which you can find as book/book.pdf in the class repository, or directly here:

https://svn.divms.uiowa.edu/repos/clc/class/111-spring14/book/book.pdf

2 Problems

Copy the file bool-to-do.agda from the hw/hw1/ subdirectory of the class repository to the hw1 subdirectory (already created) of your personal repository, on your computer or in your account on the CS lab machines. You will be filling in code in this file bool-to-do.agda, and then adding and committing it to your personal repository to turn it in (again, see the instructions linked at the top of this assignment for details).

Each hole in bool-to-do.agda is worth 10 points (the holes are the highlighted regions that look like “{ }0” after loading the file with Control-c Control-l in emacs; see Section 2.2.3 of the book). You must fill in hole 0, the definition of the imp function. Your definition should define imp to be boolean implication. There are several definitions for this which work, but however you do it, your imp function should satisfy the following truth table (you can test these possibilities by running test cases using Control-c Control-n, for example):

x	y	x imp y
tt	tt	tt
tt	ff	ff
ff	tt	tt
ff	ff	tt

Besides filling in hole 0 as just discussed, you are free to fill in any other holes you want. A perfect score is 100. We will not give any more points after 120, however. We make no guarantee that all problems are equally easy, or that problems earlier in `bool-to-do.agda` are easier than ones later in the file.

Very important: For any holes you choose not to fill in, you should remove those problems from the file before you submit. In the end, we are requiring the file you submit to check in Agda, and not to contain any holes. Your file may contain other Agda code or comments if you like. Do not change the names of any of the theorems in the file.

3 How We Will Grade

We will first confirm that your file can be checked in Agda, and that it does not contain any holes. To do this, we will run

```
agda.sh --safe --verbose=0 -i . -i PATH_TO_LIB bool-todo.agda
```

This should not produce any output. If it does produce output, that means there is either an error in your file, or an unfilled hole. If this occurs, grading cannot continue, and you will receive 0 points. If this minor disaster happens to you, you will have a chance to correct it according to the second-chance grading scheme mentioned above. And of course, if you have seen this work on your computer and think there is a problem with our grading script or something like that, you can let us know and we will investigate.

To emphasize: if we cannot type check your `bool-todo.agda` file as just described, you will receive 0 points for the first grade for this assignment (but can still revise in the second-chance period).

After confirming that your `bool-todo.agda` file type checks in Agda, we will determine how many problems you solved, in order to compute your grade. For hole 0, we will just confirm that it satisfies the truth table above. For the other holes, we will just check that what you proved indeed has the desired type. You get 10 points for each test that passes; in other words, each hole you correctly filled. You get 0 points for holes you did not fill.

4 How To Get Help

You can post questions in the `hw1` section on Piazza. You can also post in the `agda/emacs` section if your question is more about just using Agda and `emacs`. If you have a `hw1` question which is generic, not revealing too much of the solution you are attempting, you can post it for everyone to see on Piazza. Otherwise, just post it for the instructors only. We may subsequently make your question (and our answer) public, if we feel it would help other students. For truly private questions about the homework or class in general, please just email us.

You are also welcome to come to our office hours. See the "Course Staff" section of the Piazza page for times and locations of office hours.