CS101C Homework 2

Due: Wednesday, Apr 16, 2PM (firm)

Collaboration: For this homework, you can discuss the general principles and ideas of formal proofs (both in general and in MetaPRL specifically) and tactic proving, but you should work alone on the assigned problems.

Setup

Start this homework by updating MetaPRL to revision **3** (e.g. version "`0.8.1 (CS101 rev 3)`"). Upgrade instructions are available at `http://nogin.org/cs101c/mp-update.html`.

Next, in directory `theories/cs101` of your MetaPRL installation, create a file `cs101_hw2_`*name*`.ml`, where *name* is your login name (for example, if I was doing this homework, I would create `cs101_hw2_nogin.ml`). Also create the corresponding `.mli` file and add the file name (`cs101_hw2_`*name*) to the `MPFILES` variable in the `theories/cs101/Makefile`.

For this homework, you should be working in the `Cs101_int` formalization of the intuitionistic theory. You are not allowed to add any new `prim` rules to the system and you are not allowed to modify the system in any way, other than extending it with your new `Cs101_hw2_`*name* module.

Note: after you change the `MPFILES` variable in the `Makefile` or add a new `extends` or `open` directives to a MetaPRL file and before you run `make opt`, you might need to run `make depend` to update the cross-module dependencies.

Part I: Proofs

For each of the following formulas, define an `interactive` theorem in your `cs101_hw2_`*name*`.ml` file, and prove that theorem in the MetaPRL system. Feel free to use proof automation described in Lecture 4.

I.1 $(p \Rightarrow q) \Rightarrow \neg(p \wedge \neg q)$

I.2. $((p \vee q) \Rightarrow r) \Rightarrow ((p \Rightarrow r) \wedge (q \Rightarrow r))$

I.3. $\neg\neg(\neg\neg p \Rightarrow p)$

Part II: Tactics

In your `cs101_hw2_`*name*`.ml` file, program the following tactics:

II.1 `notandT` tactic that when applied to a goal of the form $\Gamma \vdash \neg(A \wedge B)$ (where Γ is an arbitrary sequence of formulas and A, B are arbitrary formulas) would produce a subgoal of the form Γ; A; $B \vdash \bot$.

II.2 `destructT` tactic that would apply the And-elim, Imp-intro and Not-intro rules repeatedly, until none of these rules can be applied anymore.

II.3 `completelyDestructT` tactic that would first apply the And-elim, Imp-intro and Not-intro rules repeatedly, until none of these rules can be applied anymore (use `destructT`). Next, it would apply And-intro and Or-elim repeatedly, until none of those rules apply. The whole process should be repeated until none of the 5 rules can be applied.

Note: in order to be able to access the new tactics from the MetaPRL proof editor, you need to declare them in the `cs101_hw2_`*name*`.mli` file as follows:

```
topval notandT: tactic
topval destructT: tactic
topval completelyDestructT: tactic
```

The `topval` declaration is very similar to OCaml's `val` declaration, except it also states that the value should be made available to the *toploop* (e.g. in the interactive dialog) of the MetaPRL proof editor.

Make sure you test each tactic on several examples.

Submission Instructions

First, `export` the proofs to `cs101_hw2_`*name*`.prla` file and submit the `cs101_hw2_`*name*`.ml`, `cs101_hw2_`*name*`.mli` and `cs101_hw2_`*name*`.prla` files. Send the files as text attachments in an email to `cs101-admin@metaprl.org`. Please include "`CS101 HW2`" in the message subject line.

Warning: the `.ml` file you submit **must** compile. Submissions that have syntax errors, or fail to compile for other reasons (for example, failing OCaml type-checker) are likely to only receive partial credit, or **no credit at all**.