CS101C Homework 6

Due: Wednesday, May 14, 11:59PM (firm)

Collaboration: For this homework, you can discuss the general principles and ideas as well as the material presented in class, but you should work alone on the assigned problems.

Setup

Start this homework by running "`make clean`" in your MetaPRL directory and updating MetaPRL to revision **9** (e.g. version "`0.8.3 (CS101 rev 9)`"). Upgrade instructions are available at `http://nogin.org/cs101c/mp-update.html`. Note that we are now going to be working with the `itt` theory, so make sure the `THEORIES` variable in `mk/config` file is set to "`base itt`" (or to "`base itt cs101`" if you want to still be able to refer to the `cs101` theory).

Next, in directory `theories/itt` of your MetaPRL installation, create a file `cs101_hw6_`*name*`.ml`, where *name* is your login name (for example, if I was doing this homework, I would create `cs101_hw6_nogin.ml`). Also create the corresponding `.mli` file and add the file name (`cs101_hw6_`*name*) to the end of the `MPFILES` variable in the `theories/itt/Makefile`.

For this homework, you should be working in the `Itt_theory` formalization of the type theory. You are not allowed to add any new `prim` rules or rewrites to the system and you are not allowed to modify the system in any way, other than extending it with your new `Cs101_hw6_`*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.

In this homework you might need to use the ITT analogs of the `Cs101_lc` operators. ITT has almost the same operators (with the same syntax for entering them) with similar rules. To find out more about ITT implementation of those constructs, see the corresponding theory files in MetaPRL (`itt_fun`, `itt_dfun`, `itt_prod`, `itt_dprod`, `itt_void`, `itt_unit` and `itt_union`), the `theories.pdf` file and the `itt_quickref.txt` file [1].

[1] The `itt_quickref.txt` file could be somewhat outdated. Please let `cs101-admin@metaprl.org` know if you find any omissions or inaccuracies in `theories.pdf` or `itt_quickref.txt`

Problems

1. Prove $(\lambda T.(T = T \in \mathbb{U}_3)) \in (\mathbb{U}_1 \to ???)$ for an appropriate ???. Note: to enter $\mathbb{U}_1$ into MetaPRL, type `univ[1:l]`.

2. Come up with a term ??? such that $??? \in (\mathbb{U}_0 \to \mathbb{U}_1)$, but ??? is not a member of $\mathbb{U}_0 \to \mathbb{U}_0$. Add to `cs101_hw6_`*name* and prove an `interactive` theorem $??? \in (\mathbb{U}_0 \to \mathbb{U}_1)$.

3. Prove $??? \in (x : \mathbb{U}_1 \times x)$ for an appropriate ???.

4. Prove $??? \in (x : \mathbb{U}_3 \to (y : x \to (y = y \in x)))$ for an appropriate ???.

5. Prove the rule

$$\frac{\Gamma \vdash t_1 = t_2 \in B \qquad \Gamma \vdash ???}{\Gamma \vdash (\lambda x.t_1) = (\lambda x.t_2) \in (A \to B)}$$

   for an appropriate ??? that does *not* mention t_1, t_2.

6. Prove

$$\frac{\Gamma; x : A \vdash t_1[x] \in C \qquad \Gamma; x : ??? \vdash t_2[x] \in C}{\begin{array}{l}\Gamma; x : (A + ???) \vdash \\ \quad \texttt{decide}\{x; v.t_1[v]; u.t_1[u]\} = \texttt{decide}\{x; v.t_1[v]; u.t_2[u]\} \in C\end{array}}$$

   for an appropriate ???.

Submission Instructions

First, `export` the proofs to `cs101_hw6_`*name*`.prla` file and submit the `cs101_hw6_`*name*`.ml`, `cs101_hw6_`*name*`.mli` and `cs101_hw6_`*name*`.prla` files. Send the files as text attachments in an email to `cs101-admin@metaprl.org`. Please include "`CS101 HW6`" 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**.