Constructive Logic (15-317), Fall 2015
Assignment 4: Classical Logic and Computation

Contact: Vincent Huang (`vincenth@andrew.cmu.edu`)
Recitation FILL ME IN

Due Tuesday, October 6, 2015

Classical logic is cool! Standard ML is cool!

. . . oh, no. . .

The Tutch portion of your work should be submitted electronically using the command

```
$ /afs/andrew/course/15/317/bin/submit -r hw4 hw4.tut
```

from any Andrew server. You may check the status of your submission by running the command

```
$ /afs/andrew/course/15/317/bin/status hw4
```

If you have trouble running either of these commands, email Anna, Michael, or Vincent.

The SML portion of your work should be submitted electronically by placing `hw04.sml` in your handin directory (`/afs/andrew/course/15/317/submit/andrewid/hw4`).

The written portion of your work should be submitted at the beginning of class. If you are familiar with LaTeX, you are encouraged to use this document as a template for typesetting your solutions, but you may alternatively write your solutions *neatly* by hand.

1 An instant classic (6 points)

Tutch allows you to do classical proofs by employing the following rule:

$$\cfrac{\cfrac{\overline{\neg A \; true}^{\; u}}{\vdots}}{\cfrac{\bot \; true}{A \; true}} \; \mathrm{PBC}^u_{\mathrm{Tutch}}$$

Task 1. Derive this rule using only the PBC^k and contra rules for classical logic, plus the usual natural deduction rules.

2   Can Tutch this (6 points)

Prove the following theorems in Tutch, using classical reasoning *if necessary* (i.e. don't sprinkle contradictions all over the place just because you can).

```
classical proof greaterThanDNE : ~~~A => ~A;
classical proof notAllProps : ~(!x:t. ~A(x)) => ?x:t. A(x);
```

3   The best programming language in the world (6 points)

In the code handout for this assignment, you will find. . . gasp. . . some actual code this week! It may be helpful for you to look at `http://www.smlnj.org/doc/SMLofNJ/pages/cont.html`[1] Then write the following:

Task 2. `val magic :  unit -> ('a -> 'b, 'b -> 'a) either`

Task 3. `val more_magic :  ('a -> 'b) -> ('a -> 'c, 'b) either`

In case the `SMLofNJ` in the code handout wasn't enough of a hint, please use SML/NJ for this assignment. We have provided a signature file and a `sources.cm` file for you to load your code into the REPL.

Also, please don't use exceptions, things from the `Unsafe` structure, or whatever other twisted ways you can think of to do this. You get that we want you to practise using continuations. Be good. Do that.

4   Konsidering komputation (2 points)

Task 4. Explain the computational content of the SML code you wrote. That is, what do the results of `magic` and `more_magic` do if you run them?

---

[1]You might also want to note that the person who set this assignment has no idea what `isolate`, `'a control_cont`, `capture`, and `escape` are, so you probably don't need to know either.