Constructive Logic (15-317), Fall 2012
Assignment 3: Quantifiers and Metatheorems

Joe Tassarotti (`jtassaro@andrew`), Evan Cavallo (`ecavallo@andrew`)

Out: Thursday, September 18, 2012
Due: Thursday, September 25, 2012 (before class)

In this assignment, you will prove propositions with quantifiers ($\forall$ and $\exists$), and prove metatheorems about sequent calculus.

The Tutch portion of your work (Section 1) should be submitted electronically using the command

```
$ /afs/andrew/course/15/317/bin/submit -r hw03 <files...>
```

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

```
$ /afs/andrew/course/15/317/bin/status hw03
```

If you have trouble running either of these commands, email Joe or Evan.

The written portion of your work (Sections 2 and 3) should be submitted electronically before the beginning of class. Please convert your homework to a PDF file titled `hw03.pdf`, and put the file in:

```
/afs/andrew/course/15/317/submit/<your andrew id>
```

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 and scan them.

1 Tutch Proofs (15 points)

For quantifiers, you'll need a little bit of new syntax:

- $\forall x : \tau.A$ is written `!x:t.A`
- $\exists x : \tau.A$ is written `?x:t.A`

- The existential elimination rule requires a hypothetical with two assumptions. These are written by separating the assumptions with a comma. For example:

```
proof exp : (?x:t.A(x)) => ?x:t.A(x) =
begin
[(?x:t.A(x));
 [a:t , A(a);
  ?x:t.A(x)];
 ?x:t.A(x)];
(?x:t.A(x)) => ?x:t.A(x);
end;
```

Task 1 (15 points). Prove the following theorems using Tutch.

```
proof instance : (!x:t.P(x)) => (?y:t.T) => (?z:t.P(z))
proof dm : (!x:t.~P(x)) => ~(?x:t.P(x));
proof eximp : (?x:t. P(x) => Q(x)) => (!x:t.P(x)) => (?x:t.Q(x))
proof allor : ((!x:t. P(x)) | (!x:t. Q(x))) => !x:t. P(x) | Q(x)
proof spread : (?x:t.P(x)) => (!x:t.!y:t.P(x) => P(y)) => !x:t.P(x)
```

On Andrew machines, you can check your progress against the requirements file `/afs/andrew/course/15/317/req/hw03.req` by running the command

```
$ /afs/andrew/course/15/317/bin/tutch -r hw03 <files...>
```

2 Impossible! (10 points)

We have seen that it is much easier to demonstrate unprovability (and related results) in sequent calculus than natural deduction.

Task 2 (5 points). Assuming A is an atomic proposition, show that there is *no* proof of:

$$\cdot \Longrightarrow \neg(A \vee \neg A)$$

Task 3 (5 points). Assuming A and B are (distinct) atomic propositions, show there is *no* proof of:

$$\cdot \Longrightarrow ((A \supset B) \supset A) \supset A$$

3 The Case of the Missing Cases (12 points)

We claimed in lecture that any derivation in sequent calculus can be transformed into a verification; that is,

Theorem. If $\Gamma \Longrightarrow C$, then $\Gamma{\downarrow} \vdash C{\uparrow}$.

where for $\Gamma = A_1, \ldots, A_n$ we define $\Gamma{\downarrow} = A_1{\downarrow}, \ldots, A_n{\downarrow}$. We proved this by induction on the derivation $\mathcal{D}$ of $\Gamma \Longrightarrow C$. Prove the following cases we skipped in lecture:

Task 4 (3 points)**.** The last rule applied in $\mathcal{D}$ is $\supset R$.

Task 5 (3 points)**.** The last rule applied in $\mathcal{D}$ is $\supset L$.

In lecture, we proved

Theorem (Cut Elimination). If $\Gamma \Longrightarrow C$ and $\Gamma, C \Longrightarrow H$, then $\Gamma \Longrightarrow H$.

by lexicographic induction first on C, then on the derivation $\mathcal{D}$ of $\Gamma \Longrightarrow C$, and then on the derivation $\mathcal{E}$ of $\Gamma, C \Longrightarrow H$. Give proofs of the following cases which we skipped in lecture:

Task 6 (3 points)**.** The last rule applied in $\mathcal{E}$ is $\vee R_1$.

Task 7 (3 points)**.** The last rule applied in $\mathcal{E}$ is $\vee L$ and the principal formula in that application is not C.