Assignment 2:
Proof Terms

15-317: Constructive Logic

Out: Thursday, September 11, 2008
Due: Thursday, September 18, 2008, before class

1 Tutch Proofs (12 pts)

For each of the following propositions:

1. Determine whether it is true (constructively).
2. If the proposition is true, give a Tutch proof.
3. If the proposition is not true, put a comment in your Tutch file explaining why your attempted proofs get stuck. Comments are lines starting with `%`.

```
proof demorgan3 : ~(A | B) => ~A & ~B
proof demorgan4 : ~(A & B) => ~A | ~B
proof imp1 : ((A => B) => C) => (A | C) & (B => C)
proof imp2 : ((A | C) & (B => C)) => ((A => B) => C)
```

2 Proof Terms (13 pts)

Tutch can be used to check proof terms.

Here is a proof of `A => (A & A)`:

```
proof dup : A => (A & A) =
begin
[A;
 A & A];
A => A & A;
end;
```

Here is an *annotated* proof:

```
annotated proof dup : A => (A & A) =
begin
[u : A;
 (u,u) : A & A];
fn u => (u , u) : A => A & A;
end;
```

An annotated proof has the same structure as an ordinary Tutch proof, except each line has the form `proof : prop`, where the `proof` term justifies the `prop`osition. An assumptions is justified by a variable (`u` in the example above).

Annotated proofs connect proof terms to the Tutch proofs you already understand. But they're somewhat verbose, because you have to write the parts of the proof term multiple times. Tutch also allows you to write a complete proof term directly, which is more convenient once you've gotten used to proving as programming. For example:

```
term dup : A => (A & A) = fn u => (u , u);
```

For reference, Tutch uses the following syntax for proof terms:

Rule	Proof Term
$\supset I$	`fn x => M`
$\supset E$	`M N`
$\wedge I$	`(M , N)`
$\wedge E_L$	`fst M`
$\wedge E_R$	`snd M`
$\top I$	`()`
$\vee I_L$	`inl M`
$\vee I_R$	`inr M`
$\vee E$	`case M of inl u => M1 | inr v => M2 end`
$\bot E$	`abort M`

Task 1 (13 pts). Use Tutch to give annotated proofs and proof terms for the following:

```
annotated proof orcomm : (A | B) => B | A
term orcomm : (A | B) => B | A
annotated proof curry : ((A & B) => C) => (A => (B => C))
term curry : ((A & B) => C) => (A => (B => C))
term uncurry : (A => B => C) => ((A & B) => C)
```

3 Biconditional (15 pts)

In this problem, you will give a direct definition of "A iff B", which means "A implies B and B implies A".

Here is the intro rule:

$$\dfrac{\begin{array}{cc} \dfrac{}{A\ \mathsf{true}}u & \dfrac{}{B\ \mathsf{true}}v \\ \vdots & \vdots \\ B\ \mathsf{true} & A\ \mathsf{true} \end{array}}{A \equiv B\ \mathsf{true}} \equiv I^{u,v}$$

Task 1 (3 pts). Give the elimination rule(s).

Task 2 (4 pts). Annotate the introduction and elimination rules with proof terms. (See Lecture 4 for examples.)

Task 3 (4 pts). Using these proof terms, give the local reduction $\Rightarrow_R$ and local expansion $\Rightarrow_E$ rules. (See Lecture 5 for examples.)

Task 4 (4 pts). Show the cases of the subject reduction/expansion theorems for these rules. (See Lecture 5 for examples.)

Recall that these theorems are stated as follows:

- Reduction: If $\Gamma \vdash M : A$ and $M \Rightarrow_R M'$ then $\Gamma \vdash M' : A$.
- Expansion: If $\Gamma \vdash M : A$ and $M \Rightarrow_E M'$ then $\Gamma \vdash M' : A$.

You may use the following substitution lemma:

If $\Gamma, x : A \vdash M : B$ and $\Gamma \vdash N : A$ then $\Gamma \vdash [N/x]M : B$

4 Handin Instructions

- To run Tutch with the requirements files, run

  ```
  /afs/andrew/course/15/317/bin/tutch -r hw02.req <your file>
  ```

  This uses the requirements file `/afs/andrew/course/15/317/req/hw02.req`.

- To submit your Tutch proofs, run

  ```
  /afs/andrew/course/15/317/bin/submit -r hw02.req <your file>
  ```

  To check the status of your submission, run `/afs/andrew/course/15/317/bin/status hw02`.

  It is expected for Tutch to report that you have unsolved problems corresponding to those propositions in Problem 1 that you think are false.

- Submit your written work at the beginning of class, or, if you wish to do an electronic handin, copy a PDF to

  ```
  /afs/andrew/course/15/317/submit/<yourid>/hw02.pdf
  ```