Constructive Logic (15-317), Fall 2016
Assignment 2: Proofs as Programs, Classical Logic

Oliver Daids (ojd@andrew.cmu.edu)

Due Tuesday, September 20, 2016

You will be using Tutch for the majority of this assignment, but there is also a written portion. This assignment is due at the beginning of class on the above date and must be submitted electronically via autolab. Submit your homework as a **tar** archive containing two files: **hw2.pdf** (your written solutions) and **hw2.tut** (your Tutch solutions).

There is also an opportunity for bonus points. You do not need to submit anything for the Bonus section, but if you do, you can earn up to 5 bonus points on this assignment. It is highly recommended that you attempt the problems, but do not worry if you are stuck.

1 Tutch Proofs

Tutch allows you to annotate your proof with proof terms by declaring it with `annotated proof`. An annotated proof is just like a regular Tutch proof, but each line `A` is annotated with the term that justifies it `M : A`.

```
annotated proof andComm : A & B => B & A =
begin
[ u : A & B;
  snd u : B;
  fst u : A;
  (snd u, fst u) : B & A];
fn u => (snd u, fst u) : A & B => B & A
end;
```

It is also possible to simply give the proof term. To give a proof term in Tutch, declare it with `term` rather than `proof`:

```
term andComm : A & B => B & A =
  fn u => (snd u, fst u);
```

For more examples, see Chapter 4 of the Tutch User's Guide. The proof terms are very similar to the ones given in lecture and are summarized in Section A.2.1 of the Guide.

Task 1 (6 points)**.** Give annotated proofs for the following theorems using Tutch.

```
annotated proof implOr : (A => C) & (B | C) => (B => A) => C;
annotated proof loop : (A => B) => (C => A) => (C => B);
annotated proof contrapositive1 : (A => B) => (~B => ~A)
```

Task 2 (10 points)**.** Give proof terms for the following theorems using Tutch.

```
term smap : A => (A => B) => B;
term exception : (A | B) => ~B => A;
term curry : (A & B => C) => A => B => C;
term uncurry : (A => B => C) => (A & B) => C;
term split : (A | B => C) => (A => C) & (B => C);
```

On Andrew machines, you can check your progress against the requirements file by running the command

```
$ tutch -r ./hw2.req hw2.tut
```

2 All the things you can do with a $\diamond$

Consider the $\diamond$ connective.

$$\dfrac{\begin{array}{c}\overline{A\ \mathit{true}}^{\,u}\\ \vdots\\ B\ \mathit{true}\end{array}}{\diamond(A,B,C)\ \mathit{true}}\ \diamond I_1^u \qquad \dfrac{\begin{array}{c}\overline{A\ \mathit{true}}^{\,u}\\ \vdots\\ C\ \mathit{true}\end{array}}{\diamond(A,B,C)\ \mathit{true}}\ \diamond I_2^u \qquad \dfrac{A\ \mathit{true} \quad \begin{array}{c}\overline{A\ \mathit{true}}^{\,u}\\ \vdots\\ B\ \mathit{true}\end{array} \quad \begin{array}{c}\overline{A\ \mathit{true}}^{\,u}\\ \vdots\\ C\ \mathit{true}\end{array}}{\diamond(A,B,C)\ \mathit{true}}\ \diamond E^u$$

Task 3 (4 points)**.** Give a proof term assignment for the rules.

Task 4 (4 points)**.** Show all the local reduction(s) and expansion(s) for these rules (proving local soundness and completeness) in proof term notation. Be sure to indicate which are reductions and which are expansions.

3 Redex Redux

Task 5 (2 points)**.** Give *two* different natural deduction proofs of $A \wedge B \supset A \vee B$ *true*. How many natural deduction proofs of this judgement exist? Explain clearly.

Task 6 (3 points)**.** Give *two* different natural deduction proofs of $A \wedge B \supset A \vee B \uparrow$. How many natural deduction proofs of this judgement exist? Explain clearly.

Task 7 (2 points)**.** Give a proof of $(A \wedge \neg A) \supset A \vee A \uparrow$. How many proofs of this judgement exist? Explain clearly.

4 An instant classic

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

$$\dfrac{\begin{array}{c}\overline{\neg A\ \mathit{true}}\ u\\ \vdots \\ \bot\ \mathit{true}\end{array}}{A\ \mathit{true}}\ \mathrm{PBC}^{u}_{\mathrm{Tutch}}$$

While you can and should be able to prove that this is equivalent to DNE, LEM, and PBC, we will not require you to do so for this homework.

Task 8 (6 points)**.** Prove the following theorems using Tutch.

```
classical proof contrapositive2 : (~B => ~A) => (A => B)
classical proof greaterThanDNE : (~~~A => ~A)
```

Task 9 (3 points)**.** Prove the following theorem using Tutch. Note that this is not a classical theorem. Once you have completed it, please consider why (you do not need to write it down).

```
proof LEMImpliesDNE : (~A | A) => (~~A => A)
```

5 Bonus

Task 10 (2.5 points)**.** Apply reduction rules to the following term until it is not possible to apply any further reduction rules.

$$(\lambda F.\ (\lambda f.\ F\,(f\,f))\ (\lambda f.\ F\,(f\,f)))\ \ (\lambda x.x)$$

Task 11 (2.5 points)**.** Prove the following theorem using Tutch.

```
proof DNEImpliesLEM : (~~A => A) => (~A | A)
```