15-317: Constructive Logic, Fall 2020
Assignment 2: Tutch, Constructivity & Harmony!

Instructor: Karl Crary
TAs: Avery Cowan, Matthew McQuaid, Long Pham, Ariel Uy

Due: Friday, September 18, 2020, 11:59 pm

The assignments in this course must be submitted electronically through Autolab and Gradescope. Written homework PDFs will go to Gradescope, and code[1] will go to Autolab. Links to the course's Autolab page and Gradescope can be found on the course homepage. For this homework, submit three files:

- `hw1.pdf` (your written solutions) on Gradescope

Your tutch files must be submitted electronically to Autolab. Autolab submissions are sometimes rejected due to ill-formed tar files. It is the student's responsibility to review the Autolab feedback to ensure that the submission was accepted. Submit your homework as a `tar` archive containing the following files:

- `hw2_1a.tut`, ..., `hw2_1e.tut` (your Tutch solutions for task 1); and
- `hw2_3a.tut` and `hw2_3b.tut` (your Tutch solutions for task 3).

[1] either a single code file or a `tar` of multiple files, depending on assignment

1 Say Hi to Tutch!

Task 1 (10 points)**.** Prove the following theorems using Tutch. Place the proof for part a in `hw2_1a.tut`, part b in `hw2_1b.tut`, . . . , and part e in `hw2_1e.tut`.

a. `proof absurdity : A & ~A => B;`

b. `proof sCombinator : (A => B) => (A => B => C) => (A => C);`

c. `proof deMorgin : ~(A | B) => ~A & ~B;`

d. `proof deMorgout : ~A & ~B => ~(A | B);`

e. `proof covariance : (A => B) => (X => (Y | (A & Z))) => (X => (Y | (B & Z)));`

Recall that in Tutch, the constant `F` means $\bot$ and the notation `~A` is a shorthand for `A => F`, in the same way as $\neg A$ is a notation for $A \supset \bot$; `A | B` is the notation for $A \vee B$.

We have provided you with requirements files to check your progress against. For example, you can check your progress for part a by running

```
$ tutch -r ./hw2_1a.req hw2_1a.tut
```

2 The Wheat and the Chaff

Task 2 (10 points)**.** The skill of detecting bogus arguments is critical in mathematics. The fallacy of *denying the antecedent* occurs occasionally in everyday bogus arguments. It looks like this:

$$(A \supset B) \supset (\neg A \supset \neg B) \; \mathit{true} \qquad (*)$$

Show that this is bogus in the case where $\neg A \wedge B$ *true* by proving:

$$(\neg A \wedge B) \supset ((A \supset B) \supset (\neg A \supset \neg B)) \supset \bot \; \mathit{true}$$

Once again, recall that $\neg B$ is shorthand for $B \supset \bot$. Be sure to label each inference rule in your proof.

3 Constructive and Classical Reasoning

By default, proofs in Tutch must be intuitionistic. However, it is possible to use Tutch to check a classical proof by using the `classical proof` declaration form; this form adds the facility to reason by contradiction.

Proof by contradiction is when you prove A by assuming $\neg A$ and deriving a contradiction. The paradigmatic example of proof by contradiction is captured in the following Tutch code:

```
classical proof byContradiction : ~~A => A =
begin
  [~~A;
   [~A;
    F];
   A
  ];
  ~~A => A
end;
```

Tip: do not confuse *proof by contradiction* with *reductio ad absurdum*; the latter refers to concluding $\neg A$ from $A \supset \bot$, and is completely constructive.

Task 3 (20 points)**.** Which directions of the following equivalence can you prove using the rules of intuitionistic/constructive logic? If a constructive proof is not possible, is there a classical proof?

$$(A \supset B) \supset C \Leftrightarrow (A \vee C) \wedge (B \supset C) \;\; \mathit{true}$$

To answer this question, try to prove the following theorems in Tutch. Place the proof for part a in `hw2_3a.tut` and part b in `hw2_3b.tut`.

a. `proof right : ((A => B) => C) => (A | C) & (B => C)`

b. `proof left : ((A | C) & (B => C)) => ((A => B) => C)`

If the proof cannot be carried out, replace the `proof` declaration with `classical proof` and try again. Full points will not be awarded for a classical proof when a constructive one is possible.

We have provided you with requirements files to check your progress against. For example, you can check your progress for part a by running

```
$ tutch -r ./hw2_3a.req hw2_3a.tut
```

Please note that until the submission deadline, Autolab will only check for the existence of valid Tutch proofs and will assign the same number of points to both constructive and classical proofs. We will adjust the points awarded for classical versus constructive proofs after the submission deadline.

4 Harmony

Task 4 (10 points)**.** Consider a connective $\ltimes$ defined by the following rules:

$$\dfrac{A\ \textit{true} \qquad \begin{array}{c}\overline{A\ \textit{true}}^{\,u}\\ \vdots \\ B\ \textit{true}\end{array}}{A \ltimes B\ \textit{true}}\ \ltimes I^u \qquad\qquad \dfrac{A \ltimes B\ \textit{true}}{B\ \textit{true}}\ \ltimes E$$

1. Is this connective locally sound? If so, show the reduction; if not, explain (informally) why no such reduction exists.

2. Is this connective locally complete? If so, give an appropriate local expansion; otherwise, explain (informally) why no such expansion exists.

Task 5 (10 points)**.** Consider a connective $\odot$ with the following elimination rules:

$$\dfrac{A \odot B\ \textit{true} \qquad \begin{array}{c}\overline{A\ \textit{true}}^{\,u} \quad \overline{B\ \textit{true}}^{\,v}\\ \vdots \\ C\ \textit{true}\end{array}}{C\ \textit{true}}\ \odot E^{u,v}$$

(Normally we take the verificationist perspective that introduction rules come first, but this time we'll go in the opposite direction.)

1. Come up with a set of zero or more introduction rules for this connective.

2. Show that the connective is locally sound and complete for your choice of introduction rules.

3. Is it possible to come up with a notational definition $A \odot B \triangleq$ ______________ so that both your defined introduction rule(s) as well as the elimination rule given above are merely derived rules? You needn't prove that this fact, merely state yes or no. However, partial credit may be awarded for partially correct arguments.