15-317: Constructive Logic
Recitation 2: Tutch

Tutch (**Tut**orial **Ch**ecker) is a proof-checker for constructive logic proofs. Tutch uses a "forward" notation for proofs, where you say what's true and Tutch builds a natural deduction proof behind the scenes.

Here's the notation Tutch uses for the logical connectives:

$A \wedge B$	`A & B`
$\top$	`T`
$A \vee B$	`A | B`
$\bot$	`F`
$A \supset B$	`A => B`

1 Tutch Basics

Here's a simple attempt at a Tutch proof:

```
proof example : A & B =
begin
A;
B;
A & B;
end;
```

The first line says that we're trying to prove `A & B`, and that the proof is named `example`; the actual proof is surrounded by `begin` and `end`. Tutch attempts to justify each line in the proof by applying a single inference rule to facts proved in previous lines. In this case, Tutch reports the following:

```
1  A;              UNJUSTIFIED
2  B;              UNJUSTIFIED
3  A & B           by AndI 1 2
```

The first two lines are unjustified: we have no way of knowing `A` and `B`. But from the first two lines, it is possible to conclude `A & B` using the `AndI` rule. Tutch notates this derivation by `AndI 1 2`, which means "Apply the `AndI` rule to the derivation justifying line 1 and the derivation justifying line 2". (Of course, in this example, there are no such derivations—we'll fix that in a bit).

If we want to prove `(A & B) & (A & B)`, we can write:

```
proof example2 : (A & B) & (A & B) =
begin
```

```
A;
B;
A & B;
(A & B) & (A & B);
end;
```

and Tutch will respond:

```
  1  A;                          UNJUSTIFIED
  2  B;                          UNJUSTIFIED
  3  A & B;                      by AndI 1 2
  4  (A & B) & A & B             by AndI 3 3
```

Note that we could not leave out line (3), concluding `A & B`, because each line must be justified using *one* inference rule and the preceding facts. Also, note that preceding facts can be used more than once.

Because there are unjustified lines, these proofs are incomplete, and Tutch does not accept them.

2 Hypotheticals

At this point, you may be wondering how we ever get a complete proof. The answer: by making assumptions. For example, say we want to prove `A => (A & A)`. In natural deduction, the proof looks like this:

$$\frac{\dfrac{\overline{A}^{\,u} \quad \overline{A}^{\,u}}{A \,\&\, A}\,\&I}{A \Rightarrow (A \,\&\, A)} \Rightarrow I^u$$

The implication introduction is justified by a *hypothetical derivation* with assumption u.

How do we write a hypothetical in Tutch? Like so:

```
[A;
 A & A];
```

The proposition on the first line, following the square-bracket, is the new assumption. Each subsequent line up to the matching close-bracket is justified with that assumption in scope. Here, we make one conclusion, `A & A`, which is justified by applying and-introduction to two uses of the hypothesis, as in the natural deduction proof above. Tutch responds:

```
  1  [ A;
  2    A & A ];          by AndI 1 1
```

In general, you will make more than one conclusion in a hypothetical; e.g.,

```
[A;
 A & A;
 (A & A) & (A & A)];
```

A hypothetical

```
[A;
 .
 .
 .
 B];
```

justifies an implication `A => B`.

For example, the full proof of `A => (A & A)` looks like this:

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

3 Bigger Examples

In recitation, we did the following proofs:

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

proof distribimpand1 : (A => (B & C)) => (A => B) & (A => C) =
begin
[A => (B & C);
 [A;
  B & C;
  B];
 A => B;
 [A;
  B & C;
  C];
 A => C;
 (A => B) & (A => C)];
(A => (B & C)) => (A => B) & (A => C);
end;

proof distriborimp1 : ((A | B) => C) => (A => C) & (B => C) =
begin
[(A | B) => C;
 [A;
```

```
  A | B;
  C];
 A => C;
 [B;
  A | B;
  C];
 B => C;
 (A => C) & (B => C)];
((A | B) => C) => (A => C) & (B => C);
end;
```

On Andrew, you can run Tutch on a file with:

```
/afs/andrew/course/15/317/bin/tutch <file>
```

To see the natural deduction proof that Tutch finds, add the `-v` flag.