An Introduction to Type Theory

Practical 1

Tallinn, September 2003

Thorsten Altenkirch

University of Nottingham

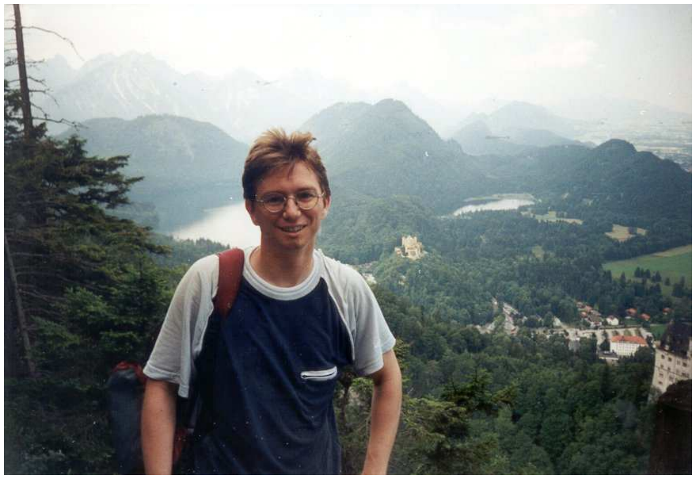
`tutch` - example of a proof checker

`tutch` - example of a proof checker

`tutch` ***tut**orial proof **ch**ecker*
is a proof checker for educational purposes implemented by Andreas Abel (Munich)
when he was working for Frank Pfenning at Carnegie Mellon University, USA

`tutch` - example of a proof checker

`tutch` ***tut****orial proof* ***ch****ecker*
is a proof checker for educational purposes implemented by Andreas Abel (Munich)
when he was working for Frank Pfenning at Carnegie Mellon University, USA



tutch syntax

<table>
<tr><th>Logic</th><th>tutch</th></tr>
<tr><td>$\wedge$</td><td>&</td></tr>
<tr><td>$\vee$</td><td>|</td></tr>
<tr><td>$\neg$</td><td>~</td></tr>
<tr><td>$\Rightarrow$</td><td>=></td></tr>
<tr><td>T</td><td>T</td></tr>
<tr><td>F</td><td>F</td></tr>
<tr><td>$\Leftrightarrow$</td><td><=></td></tr>
</table>

$$A \wedge B \Rightarrow B \wedge A$$

```
proof comAnd : A & B => B & A =
begin
[A & B;
  A;
  B;
  B & A];
A & B => B & A
end;
```

$A \wedge B \Rightarrow B \wedge A$

```
proof comAnd : A ∧ B ⇒ B ∧ A =
begin
[A ∧ B;
  A;
  B;
  B ∧ A];
A ∧ B ⇒ B ∧ A
end;
```

Running tutch

1. Create a text file `comAnd.tut`
2. Run tutch:

```
[txa@jacob mytutch]$ tutch comAnd.tut
TUTCH 0.51 beta, $Date: 2000/10/27 17:08:48 $
[Opening file comAnd.tut]
Proving C1a: A ∧ B ⇒ B ∧ A ...
QED
[Closing file comAnd.tut]
```

tutch verbose

```
[txa@jacob mytutch]$ tutch -v comAnd.tut
TUTCH 0.51 beta, $Date: 2000/10/27 17:08:48 $
[Opening file comAnd.tut]

Proving comAnd: A ∧ B ⇒ B ∧ A ...
  1  [ A ∧ B;
  2    A;                         by AndEL 1
  3    B;                         by AndER 1
  4    B ∧ A ];                   by AndI 3 2
  5  A ∧ B ⇒ B ∧ A                by ImpI 4
QED
```

An incomplete proof

```
proof comAnd : A ∧ B ⇒ B ∧ A =
begin
[A ∧ B;
 B ∧ A];
A ∧ B ⇒ B ∧ A
end;
```

tutch's error messages

```
[txa@jacob mytutch]$ tutch comAnd.tut
TUTCH 0.51 beta, $Date: 2000/10/27 17:08:48 $
[Opening file comAnd.tut]
Proving comAnd: A ∧ B ⇒ B ∧ A ...
comAnd.tut:4.2-4.7 Error:
Unjustified line A ∧ B  ⊢  B ∧ A
Assuming this line, checking remainder...
Proof incomplete
[Closing file comAnd.tut]
```

Rules for $\wedge$

AndI

```
A;
B;
A ∧ B;
```

AndEL

```
A ∧ B;
A;
```

AndER

```
A ∧ B;
B;
```

Rules for $\Rightarrow$

ImpI

```
[ A;
  ⋮
  B];
A ⇒ B;
```

ImpE

```
A ⇒ B;
A;
B;
```

The simplest proof?

```
proof I: A ⇒ A =
begin
[ A;
  A];
A ⇒ A
end;
```

tutch output

```
  Proving I: A ⇒ A ...
  1  [ A;
  2     A ];     by Hyp 1
  3  A ⇒ A      by ImpI 2
QED
```

Rules for $\vee$

OrIL

$$\texttt{A;}$$
$$\texttt{A} \vee \texttt{B;}$$

OrIR

$$\texttt{B;}$$
$$\texttt{A} \vee \texttt{B;}$$

Rules for $\vee$ (cont)

OrE

```
[ A;
   ⋮
   C];
[ B;
   ⋮
   C];
A ∨ B;
C;
```

$A \vee B \Rightarrow B \vee A$

```
proof comOr : A ∨ B ⇒ B ∨ A =
begin
[ A ∨ B;
  [ A;
    B ∨ A];
  [ B;
    B ∨ A];
  B ∨ A];
A ∨ B ⇒ B ∨ A
end;
```

tutch output

```
Proving comOr: A ∨ B ⇒ B ∨ A ...
  1  [ A ∨ B;
  2    [ A;
  3      B ∨ A ];                          by OrIR 2
  4    [ B;
  5      B ∨ A ];                          by OrIL 4
  6    B ∨ A ];                            by OrE 1 3 5
  7  A ∨ B ⇒ B ∨ A                        by ImpI 6
QED
```

T and F

TrueI

$$\frac{}{\mathsf{T};}$$

FalseE

$$\frac{\mathsf{F};}{\mathsf{A};}$$

$\neg$ and $\Leftrightarrow$

$\neg A$	is defined as	$A \Rightarrow \mathsf{F}$
$A \Leftrightarrow B$	is defined as	$(A \Rightarrow B) \wedge (B \Rightarrow A)$

$\neg(A \wedge \neg A)$

```
proof incons : ¬(A ∧ ¬ A) =
begin
[ A ∧ ¬ A;
  A;
  ¬ A;
  F];
¬(A ∧ ¬ A)
end;
```

Propositional logic

Prove the following in intuitionistic propositional logic using tutch (`prop.req`):

1. $A \Rightarrow A \wedge A$
2. $A \vee A \Rightarrow A$
3. $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow (A \Rightarrow C)$
4. $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
5. $\neg A \wedge \neg B \Leftrightarrow \neg(A \vee B)$

Using `prop.req`

You can check your proofs by running tutch under unix by typing

```
tutch -r ./prop.req -v prop.tut
```

in the shell window and edit your proofs until you get the message

```
Congratulations! All problems solved!
```

Predicate logic: tutch syntax

Predicate logic: tutch syntax

Logic	tutch
$\forall$	!
$\exists$	?

Predicate logic: tutch syntax

Logic	tutch
$\forall$	!
$\exists$	?

Tutch is *typed*, we write

$$\forall x : T.P(x)$$

$$\exists x : T.P(x)$$

where T is a type (e.g. `nat`).

Predicate logic: tutch syntax

Logic	tutch
$\forall$	!
$\exists$	?

Tutch is *typed*, we write

$$\forall x : T.P(x)$$

$$\exists x : T.P(x)$$

where T is a type (e.g. `nat`).

In the moment we just use a type variable (e.g. t).

Rules for ∀

ForallI

```
[x:t;
 ⋮
 P];
∀ x:t.P;
```

ForallE

```
∀ x:t.P;
u : t;
P[x:=u]
```

Example of a proof using ∀

```
proof allCom :
   (∀ x:t.∀ y:t. P(x,y))⇒(∀ y:t.∀ x:t.P(x,y)) =
begin
[∀ x:t . ∀ y:t. P(x,y);
 [ y:t;
   [ x:t;
     ∀ y:t. P(x,y);
     P(x,y)];
   ∀ x:t.P(x,y)];
 ∀ y:t.∀ x:t.P(x,y)];
(∀ x:t.∀ y:t. P(x,y))⇒(∀ y:t.∀ x:t.P(x,y));
end;
```

Predicate logic

Prove the following in intuitionistic propositional logic using tutch (`pred.req`)

1. $(\forall x : t.P(x)) => (\forall y : t.P(y))$
2. $(\forall x : t.P(x) \wedge Q(x)) <=> (\forall x : t.P(x)) \wedge (\forall x : t.Q(x))$
3. $(\exists x : t.P(x) \wedge Q) => (\exists x : t.P(x)) \wedge Q$
4. $(\exists x : t.P(x) \vee Q(x)) <=> (\exists x : t.P(x)) \vee (\exists x : t.Q(x))$