Slide 1

A Taste of Intuitionistic Type Theory

Midland Graduate School, 18.10.00 - 15.11.00

`http://www.cs.nott.ac.uk/~txa/itt/`

Thorsten Altenkirch

October 18, 2000

1 Intuitionistic Logic

Slide 2

$\boxed{\textbf{A non-constructive proof}}$

Proposition: There are two irrational numbers a, b s.t. a^b is rational.

Slide 3

Proof:

We know that $\sqrt{2}$ is irrational.

There are two possibilities:

$\sqrt{2}^{\sqrt{2}}$ **is rational.** We are done with $a = b = \sqrt{2}$.

$\sqrt{2}^{\sqrt{2}}$ **is irrational.** Now consider $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$.

$$\begin{aligned} a^b &= (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} \\ &= \sqrt{2}^{(\sqrt{2}\sqrt{2})} \\ &= \sqrt{2}^2 \\ &= 2 \end{aligned}$$

Slide 4

Write down 2 irrational numbers a, b s.t. a^b is rational!

Slide 5

Tarski's semantics

A	B	$A \wedge B$
false	false	false
false	true	false
true	false	false
true	true	true

A	B	$A \vee B$
false	false	false
false	true	true
true	false	true
true	true	true

A	B	$A \implies B$
false	false	true
false	true	true
true	false	false
true	true	true

A	$\neg A$
false	true
true	false

Slide 6

$P(0)$	$P(1)$	$P(2)$	...	$\forall x.P(x)$
false	false	false	...	false
true	false	false	...	false
...	...	...	...	...
true	true	true	...	true

$P(0)$	$P(1)$	$P(2)$	...	$\exists x.P(x)$
false	false	false	...	false
true	false	false	...	true
...	...	...	...	...
true	true	true	...	true

Slide 7

Tautologies

Tarski's semantics implies that the following propostions are tautologies:

$$
\begin{aligned}
(A_1 \implies B) \wedge (A_2 \implies B) &\iff (A_1 \vee A_2) \implies B \\
\forall x. A(x) \implies B &\iff (\exists x. A(x)) \implies B \\
A \vee B &\iff \neg(\neg A \wedge \neg B) \\
\exists x. A(x) &\iff \neg \forall x. \neg A(x)
\end{aligned}
$$

Slide 8

BHK semantics

Brouwer

Heyting

Kolmogorov

Slide 9

Conjunction A proof of $A \wedge B$ is a proof of A and a proof of B.

Disjunction A proof of $A \vee B$ is a bit and

- a proof of A if the bit is 0,
- a proof of B if the bit is 1.

Implication A proof of $A \to B$ is a procedure which calculates a proof of B given a proof of A.

Absurdity There is no proof of false.

$\neg A$ Define $\neg A$ as $A \to$ false.

Slide 10

Universal quantification A proof of $\forall x.A(x)$ is a procedure which assigns to every individual i (i.e. number) a proof of $A(i)$.

Existential quantification A proof of $\exists x.A(x)$ is an individual i and a proof of $A(i)$.

Slide 11

BHK in Haskell

```
data And a b = Pair a b

data Or a b = In0 a | In1 b

type Impl a b = a -> b

type Iff a b = And (Impl a b) (Impl b a)
```

Slide 12

```
taut1 :: Iff (And (Impl a1 b) (Impl a2 b))
             (Impl (Or a1 a2) b)

taut1 = Pair
          (\ (Pair f g) ->
              \ a12 -> case a12 of In0 a1 -> f a1
                                   In1 a2 -> g a2)
          (\ h ->
              Pair (\ a1 -> h (In0 a1))
                   (\ a2 -> h (In1 a2)))
```

Slide 13

```
data False = Impossible (forall a.a)

type Not a = Impl a False
```

Slide 14

```
taut2 :: Impl (Or a b) (Not (And (Not a) (Not b)))
taut2 = (\ ab →
            (\ (Pair na nb) →
                          case ab of (In0 a) → na a
                                     (In1 b) → nb b))
```

Slide 15

$$\boxed{A \vee B \not\Leftarrow \neg(\neg A \wedge \neg B)}$$

$A \quad = \sqrt{2}^{\sqrt{2}}$ is irrational.

$B \quad = \sqrt{2}^{\sqrt{2}}$ is rational

Or even better:

$A \quad =$ There are infinitely many twin primes.

$B \quad =$ There are only finitely many twin primes.

Slide 16

Judgements of propositional logic

- P is a proposition

$$P : \mathbf{Prop}$$

- p is a proof of P:

$$p : P$$

Slide 17

Implication

Formation

$$\frac{A : \mathbf{Prop} \qquad B : \mathbf{Prop}}{A \to B : \mathbf{Prop}} \to \mathcal{F}$$

Introduction

$$\frac{\begin{array}{c}[x : A] \\ \vdots \\ b : B\end{array}}{[x : A]b : A \to B} \to \mathcal{I}$$

Elimination (Modus Ponens)

$$\frac{f : A \to B \qquad a : A}{fa : B} \to \mathcal{E}$$

Slide 18

Example $(A, B, C : \mathbf{Prop})$

$$\frac{\dfrac{[x : A]}{x : A}}{[x : A]x : A \to A} \to \mathcal{I}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[f : A \to B \to C] \qquad [a : A]}{fa : B \to C} \to \mathcal{E} \qquad [b : B]}{fab : C} \to \mathcal{E}}{[a : A]fab : A \to C} \to \mathcal{I}}{[b : B][a : A]fab : B \to A \to C} \to \mathcal{I}}{[f : A \to B \to C][b : B][a : A]fab : (A \to B \to C) \to (B \to A \to C)} \to \mathcal{I}$$

Slide 19

Truth, conjunction

Formation

$$\frac{}{\text{True} : \mathbf{Prop}}\ \text{True}\mathcal{F} \qquad \frac{A : \mathbf{Prop} \qquad B : \mathbf{Prop}}{A \wedge B : \mathbf{Prop}}\ \wedge\mathcal{F}$$

Introduction

$$\frac{}{\text{true} : \text{True}}\ \text{True}\mathcal{I} \qquad \frac{a : A \qquad b : B}{\text{pair}\ a\ b : A \wedge B}\ \wedge\mathcal{I}$$

Elimination

$$\frac{p : A \wedge B}{\text{fst}\ p : A}\ \wedge 1\mathcal{E} \qquad \frac{p : A \wedge B}{\text{snd}\ p : B}\ \wedge 2\mathcal{E}$$

Slide 20

Example

$$\cfrac{\cfrac{\cfrac{[p : A \wedge B]}{\text{snd}\ p : B}\ \wedge 2\mathcal{E} \qquad \cfrac{[p : A \wedge B]}{\text{fst}\ p : A}\ \wedge 1\mathcal{E}}{\text{pair}\ (\text{snd}\ p)\ (\text{fst}\ p) : B \wedge A}\ \wedge\mathcal{I}}{[p : A \wedge B]\text{pair}\ (\text{snd}\ p)\ (\text{fst}\ p) : (A \wedge B) \to (B \wedge A)}\ \to \mathcal{I}$$

Slide 21

Absurdity and Disjunction

Formation

$$\frac{}{\text{False} : \textbf{Prop}}\ \text{False}\mathcal{F} \qquad \frac{A : \textbf{Prop} \qquad B : \textbf{Prop}}{A \vee B : \textbf{Prop}}\ \vee\mathcal{F}$$

Introduction

$$\frac{a : A}{\text{inl }\ a : A \vee B}\ \vee 1\mathcal{I} \qquad \frac{b : B}{\text{inr }\ b : A \vee B}\ \vee 2\mathcal{I}$$

Elimination

$$\frac{C : \textbf{Prop} \qquad f : \text{False}}{\text{falseE }\ f : C}\ \text{False}\mathcal{E} \qquad \frac{\begin{array}{c}[a : A] \\ \vdots \\ p : C\end{array} \qquad \begin{array}{c}[b : B] \\ \vdots \\ q : C\end{array} \qquad r : A \vee B}{\text{case }\ ([a : A]p)\ ([b : A]q)\ r : C}\ \vee\mathcal{E}$$

Slide 22

Example

Define

$$\text{not }\ A \equiv A \to \text{False}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[f : \text{not }\ A \vee B] \qquad [a : A]}{\text{inl }\ a : A \vee B}\ \vee 1\mathcal{I}}{f(\text{inl }\ a) : \text{False}}\ \to\mathcal{E}}{[a : A]f(\text{inl }\ a) : \text{not }\ A}\ \to\mathcal{I} \qquad \dfrac{\dfrac{\dfrac{[f : \text{not }\ A \vee B] \qquad [b : B]}{\text{inr }\ b : A \vee B}\ \vee 2\mathcal{I}}{f(\text{inr }\ b) : \text{False}}\ \to\mathcal{E}}{[b : B]f(\text{inr }\ b) : \text{not }\ B}\ \to\mathcal{I}}{\dfrac{\text{pair }\ [a : A]f(\text{inl }\ a)\ [b : B]f(\text{inr }\ b) : (\text{not }\ A) \wedge (\text{not }\ B)}{[f : \text{not }\ A \vee B] \cdots : \text{not }\ A \vee B \to (\text{not }\ A) \wedge (\text{not }\ B)}\ \to\mathcal{I}}\ \wedge\mathcal{I}$$

Slide 23

Judgements of predicate logic

- A is a *set.* (sort,type, ...)

$$A : \mathbf{Set}$$

- P is a predicate over $A_1, \ldots, A_n$ (where $A_i : \mathbf{Set}$):

$$P : A_1 \to \cdots \to A_n \to \mathbf{Prop}$$

  special case $n = 0$: P is a proposition ($P : \mathbf{Prop}$).

- f is a function $A_1 \to \cdots \to A_n \to B$ ($A_i, B : \mathbf{Set}$):

$$f : A_1 \to \cdots \to A_n \to B$$

  special case $n = 0$: f is a constant.

Slide 24

Universal quantification

Formation

$$\frac{A : \mathbf{Set} \qquad \begin{array}{c}[x : A] \\ \vdots \\ P : \mathbf{Prop}\end{array}}{\{x : A\}P : \mathbf{Prop}} \forall\mathcal{F}$$

Introduction

$$\frac{A : \mathbf{Set} \qquad \begin{array}{c}[x : A] \\ \vdots \\ p : P\end{array}}{[x : A]p : \{x : A\}P} \forall\mathcal{I}$$

Slide 25

Elimination

$$\frac{f : \{x : A\}P \qquad a : A}{fa : P[x := a]}\,\forall\mathcal{E}$$

Slide 26

Example

$$S : \mathbf{Set}$$

$$P, Q : S \to \mathbf{Prop}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[f : \{x : S\}(Px) \to Qx] \qquad [x : S]}{fx : (Px) \to Qx}\,\forall\mathcal{E} \qquad \dfrac{[g : \{x : S\}Px] \qquad [x : S]}{gx : Px}\,\forall\mathcal{E}}{fx(gx) : Qx}\to\mathcal{E}}{[x : S]fx(gx) : \{x : S\}Qx}\,\forall\mathcal{I}}{[g : \ldots][x : S]fx(gx) : (\{x : S\}Px) \to \{x : S\}Qx}\to\mathcal{I}}{[f : \ldots][g : \ldots][x : S]fx(gx) : (\{x : S\}(Px) \to Qx) \to (\{x : S\}Px) \to \{x : S\}Qx}\to\mathcal{I}$$

Slide 27

Existential quantification

Formation

$$\frac{A : \mathbf{Set} \qquad \begin{array}{c}[x : A]\\ \vdots \\ P : \mathbf{Prop}\end{array}}{\text{Exists }([x : A]P) : \mathbf{Prop}}\ \exists\mathcal{F}$$

Introduction

$$\frac{\text{Exists }([x : A]P) : \mathbf{Prop} \qquad a : A \qquad p : P[x := a]}{\text{existsI } a\ p : \text{Exists }([x : A]P)}\ \exists\mathcal{I}$$

Slide 28

Elimination

$$\frac{\begin{array}{c}[x : A] \quad [y : P]\\ \vdots \\ q : Q\end{array} \qquad p : \text{Exists }([x : A]P)}{\text{existsE }([x : A][y : P]q)\ p : Q}\ \exists\mathcal{E}$$

Slide 29

Example

$$S : \mathbf{Set}$$

$$P : S \to \mathbf{Prop}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[f : \{x : S\}\text{not } (Px)][x : S]}{fx : \text{not } Px}\,\forall\mathcal{E} \qquad [q : Px]}{fxq : \text{False}}\to\mathcal{E} \qquad [p : \text{Exists } ([x : S]Px)]}{\text{existsE } ([x : S][q : Px]fxq)\ p : \text{False}}\,\exists\mathcal{E}}{[f]\text{existsE } ([x : S][q : Px]fxq)\ p : \text{not } (\{x : S\}\text{not } (Px))}\to\mathcal{I}}{[p][f]\text{existsE } ([x : S][q : Px]fxq)\ p : (\text{Exists } ([x : S]Px)) \to \text{not } (\{x : S\}\text{not } (Px))}\to\mathcal{I}$$

Slide 30

(Weak) Normalisation

A proof is **weakly normal** if it starts with an introduction rule.

Theorem: If there is a proof of A : **Prop** then there is a weakly normal proof of A.

Slide 31

Corollary: (Consistency)
There is no proof of False.

Corollary: (Disjunctive property)
If there is a proof of $A \vee B$ then there is a proof of A or B.

Corollary: (Existential property)
If there is a proof of $\exists x.A(x)$ (with $A : S \to \mathbf{Prop}$) then there is a term $t : S$ and a proof of $A(t)$.

Slide 32

Corollary: (Program Extraction)
If there is a proof of $\forall x.\exists y.B(x, y)$ with $B : S \to T \to \mathbf{Prop}$ then there is an algorithm $f : S \to T$ and for any $s : S$ a proof of $B(s, f(s))$.

Slide 33

Using LEGO

The LEGO homepage:

`http://www.dcs.ed.ac.uk/home/lego/`

Download the latest version and appropriate documentation. In the moment we do not use the library.

From the course homepage

`http://www.cs.nott.ac.uk/~txa/itt/`

the LEGO files `prop.l` and `pred.l` can be downloaded.

Slide 34

Example session

```
[txa@jacob lego]$ lego
LEGOPATH=.:/usr/lib/lego/lib_Type:/usr/lib/lego

Standard ML with LEGO
Generated  using Standard ML of New Jersey, Version 0.93, February 15, 1993
LEGO 1.3.1
 ...
Lego> [A,B:Prop];
decl  A B : Prop
Lego> [l1 = [ab:Or A B][h:And (not A) (not B)]case (fst h) (snd h) ab
    : (Or A B)->not (And (not A) (not B))];
Lego> defn  l1 = [ab:Or A B][h:And (not A) (not B)]case (fst h) (snd h) ab
      l1 : (Or A B)->not (And (not A) (not B))
```

Slide 35

```
Lego> Goal (Or A B) -> (not (And (not A) (not B)));
Goal
   ?0 : (Or A B)->not (And (not A) (not B))
Lego> Intros ab h;
Intros (2) ab h
  ab : Or A B
  h : And (not A) (not B)
   ?1 : False
Lego> Refine case ? ? ab;
Refine by  case ?5 ?6 ab
   ?5 : A->False
   ?6 : B->False
Lego> Refine fst h;
Refine by  fst h
   ?6 : B->False
Lego> Refine snd h;
Refine by  snd h
Discharge..  h ab
*** QED ***   (* time= 0.0  gc= 0.0  sys= 0.0 *)
Lego> Save l2;
"l2"  saved as global, unfrozen
Lego> l2;
value of l2 = [ab:Or A B][h:And (not A) (not B)]case (fst h) (snd h) ab
type  of l2 = (Or A B)->not (And (not A) (not B))
```

Slide 36

Exercises

Introduce propositional variables:

`[A,B,C:Prop];`

and construct LEGO derivations for

1. $(A \to B \to C) \to (A \wedge B) \to C$
2. $((A \wedge B) \to C) \to (A \to B \to C)$
3. not $(A \wedge (\text{not } A))$
4. not (not $(A \vee (\text{not } A))$) (*not so easy*).

Introduce a set and a predicate

`[S:Set][P:S→ Prop];`

1. not (Exists $(([x:S]Px)) \to \{x:S\}$not (Px)
2. $\{x:S\}$not $(Px) \to$ not (Exists $(([x:S]Px)))$