Logic for Computer Science

Summer semester
2019-2020

LECTURE 7 : INTUITIONISTIC FIRST-ORDER LOGIC

Lectures : LORENZO CLEMENTE
Tutorials : DARIA WALUKIEWICZ, JACEK CHRZASZCZ, JĘDRZEJ KOŁODZIEJSKI
Labs : DARIA, JACEK + PIOTR HOFMAN

PLAN

- BHK interpretation for first-order logic.
- Examples of intuitionistic tautologies.
- Natural deduction for intuitionistic first-order logic.
- Examples of deductions.
- Curry-Howard correspondence and λP_1 calculus.
- Dependent types in programming.

Only mention:

- Kripke models for intuitionistic first-order logic & completeness.
- Negative translation: reduction of classical to intuitionistic logic.

BROWER - HEYTING - KOLMOGOROV interpretation

FORMULA	PROOF
$\bot$	(no such proof)
$\varphi \wedge \psi$	a pair $\langle p, q \rangle$ where p is a proof of φ and q a proof of ψ
$\varphi \vee \psi$	a pair $\langle i, p \rangle$ s.t. either $i = 0$ and p is a proof of φ, or $i = 1$ and p is a proof of ψ
$\varphi \to \psi$	a computable function mapping a proof of φ to one of ψ
$\exists x{:}A.\ \varphi$	a pair $\langle a, p \rangle$ where a is a value for x and p is a proof of $\varphi[x \mapsto a]$
$\forall x{:}A.\ \varphi$	a computable function mapping a to a proof of $\varphi[x \mapsto a]$

Syntactic sugaring:

$$\neg \varphi \equiv \varphi \to \bot$$

$$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$$

(NON)EXAMPLES of INTUITIONISTIC PROPOSITIONAL TAUTOLOGIES

A1: $\varphi \to \psi \to \varphi$

A2: $(\varphi \to \psi \to \theta) \to (\varphi \to \psi) \to \varphi \to \theta$

A3: $\neg\neg\varphi \to \varphi$

$\neg\neg\neg\varphi \to \neg\varphi$ triple negation elimination

$\neg\neg(\neg\neg\varphi \to \varphi)$ irrefutability of double negation elimination

$$\text{De Morgan}\left\{\begin{array}{l} \neg(\varphi \vee \psi) \leftrightarrow \neg\varphi \wedge \neg\psi \\ \neg\varphi \vee \neg\psi \to \neg(\varphi \wedge \psi) \\ \neg(\varphi \wedge \psi) \to \neg\varphi \vee \neg\psi \end{array}\right.$$

(NON) EXAMPLES of INTUITIONISTIC FIRST ORDER TAUTOLOGIES

A4: $(\forall x \cdot \varphi \to \psi) \to (\forall x \cdot \varphi) \to \forall x \cdot \psi$

A5: $\varphi \to \forall x \cdot \varphi$ if $x \notin fv(\varphi)$

A6: $(\forall x \cdot \varphi) \to \varphi[x \mapsto t]$ if t is free for x in φ

∞ De Morgan
$$\left\{\begin{array}{l} \neg \exists x \cdot \varphi \leftrightarrow \forall x \cdot \neg \varphi \\ \exists x \cdot \neg \varphi \to \neg \forall x \cdot \varphi \\ \neg \forall x \cdot \varphi \to \exists x \cdot \neg \varphi \end{array}\right.$$

$\exists x \cdot \varphi \vee \psi \leftrightarrow (\exists x \cdot \varphi) \vee (\exists x \cdot \psi)$

$\exists x \cdot \varphi \wedge \psi \to (\exists x \cdot \varphi) \wedge (\exists x \cdot \psi)$

$(\exists x \cdot \varphi) \wedge \psi \to \exists x \cdot \varphi \wedge \psi$ $\quad x \notin fv(\psi)$

$\forall x \cdot \varphi \wedge \psi \leftrightarrow (\forall x \cdot \varphi) \wedge (\forall x \cdot \psi)$

$(\forall x \cdot \varphi) \vee (\forall x \cdot \psi) \to \forall x \cdot \varphi \vee \psi$

$\forall x \cdot \varphi \vee \psi \to (\forall x \cdot \varphi) \vee \psi$ $\quad x \notin fv(\psi)$

no uniform choice

$\underbrace{\text{can look at } x}$

$\forall x \cdot p \vee q \to (\forall x\, p) \vee q$ fails already

NATURAL DEDUCTION for $\{\to, \forall\}$

Axiom: $$\frac{}{\Gamma, \varphi \vdash \varphi}\ (Ax)$$

Introduction rules

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ (\to I)$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x . \varphi}\ (\forall I) \quad x \notin fv(\Gamma)$$

Elimination rules

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \varphi \to \psi}{\Gamma \vdash \psi}\ (\to E)$$

$$\frac{\Gamma \vdash \forall x . \varphi}{\Gamma \vdash \varphi[x \mapsto t]}\ (\forall E)$$

EXAMPLES

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{(Ax)}
\UnaryInfC{$\Gamma \vdash \forall x . \varphi \to \psi$}
\LeftLabel{($\forall E$)}
\UnaryInfC{$\Gamma \vdash \varphi \to \psi$}
\AxiomC{}
\RightLabel{(Ax)}
\UnaryInfC{$\Gamma \vdash \forall x . \varphi$}
\RightLabel{($\forall E$)}
\UnaryInfC{$\Gamma \vdash \varphi$}
\RightLabel{($\to E$)}
\BinaryInfC{$\Gamma \vdash \psi$}
\RightLabel{($\forall I$) $\forall \notin fv(\Gamma)$}
\UnaryInfC{$\Gamma := \{\forall x . \varphi \to \psi , \forall x . \varphi\} \vdash \forall x . \psi$}
\RightLabel{($\to I$)$\times 2$}
\UnaryInfC{$\vdash (\forall x . \varphi \to \psi) \to (\forall x . \varphi) \to \forall x . \psi$}
\end{prooftree}
$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(Ax)}
\UnaryInfC{$\forall x . \forall y . \varphi \vdash \forall x . \forall y . \varphi$}
\RightLabel{($\forall E$)}
\UnaryInfC{$\forall x . \forall y . \varphi \vdash \forall y . \varphi$}
\RightLabel{($\forall E$)}
\UnaryInfC{$\forall x . \forall y . \varphi \vdash \varphi$}
\RightLabel{($\forall I$)$\times 2$}
\UnaryInfC{$\forall x . \forall y . \varphi \vdash \forall y . \forall x . \varphi$}
\RightLabel{($\to I$)}
\UnaryInfC{$\vdash (\forall x . \forall y . \varphi) \to \forall y . \forall x . \varphi$}
\end{prooftree}
$$

HASKELL WILLIAM A.

CURRY-HOWARD CORRESPONDENCE

LOGIC		PROGRAMS
Formula φ	$\longleftrightarrow$	Type φ
Proof of φ	$\longleftrightarrow$	Program (λ-term) $\quad U : \varphi$
Proof normalization	$\longleftrightarrow$	Evaluation
Validity problem	$\longleftrightarrow$	Type inhabitation
Proof checking	$\longleftrightarrow$	Type reconstruction

CALCULUS λP_1 for $\{\to, \forall\}$

type constructor from Σ

$$\varphi, \psi ::= x \mid R(t_1, \ldots, t_n) \mid \varphi \to \psi \mid \forall x : A . \varphi$$

type (domain) variable

term over Σ

$$u, v ::= x \mid u\, v \mid \lambda x : \varphi . u \mid u\, t \mid \lambda x : A . u$$

proof variable

domain variable

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi}\ (Ax)$$

$$\frac{\Gamma, x : \varphi \vdash u : \psi}{\Gamma \vdash \lambda x : \varphi . u : \varphi \to \psi}\ (\to I)$$

$$\frac{\Gamma \vdash u : \varphi \to \psi \qquad \Gamma \vdash v : \varphi}{\Gamma \vdash u\, v : \psi}\ (\to E)$$

$$\frac{\Gamma \vdash u : \varphi}{\Gamma \vdash (\lambda x : A . u) : \forall x : A . \varphi}\ (\forall I) \quad x \notin fv(\Gamma)$$

$$\frac{\Gamma \vdash u : \forall x . \varphi}{\Gamma \vdash u\, t : \varphi[x \mapsto t]}\ (\forall E)$$

Two abstractions!

Two applications!

NATURAL DEDUCTION for $\{\to, \bot, \wedge, \vee, \neg, \forall, \exists\}$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ (\to I)$$

$$\frac{}{\Gamma, \varphi \vdash \varphi}\ (A_x)$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \varphi \to \psi}{\Gamma \vdash \psi}\ (\to E)$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}\ (\bot E)$$

miracle rule!

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ (\wedge I)$$

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ (\wedge E_L)$$

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}\ (\wedge E_R)$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ (\vee I_L)$$

$$\frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}\ (\vee I_R)$$

$$\frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\ (\vee E)$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x . \varphi}\ (\forall I) \quad x \notin fv(\Gamma)$$

$$\frac{\Gamma \vdash \forall x . \varphi}{\Gamma \vdash \varphi[x \mapsto t]}\ (\forall E)$$

$$\frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash \exists x . \varphi}\ (\exists I)$$

$$\frac{\Gamma \vdash \exists x . \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}\ (\exists E)$$

EXAMPLES

$$
\cfrac{\cfrac{\cfrac{\cfrac{}{\forall x.\,\varphi \vdash \forall x.\,\varphi}\,(Ax)}{\forall x.\,\varphi \vdash \varphi[x \mapsto x]}\,(\forall E)}{\forall x.\,\varphi \vdash \exists x.\,\varphi}\,(\exists I)}{\vdash \forall x.\,\varphi \to \exists x.\,\varphi}\,(\to I)
$$

$$
\cfrac{\cfrac{\cfrac{(Ax)\,\cfrac{}{\Gamma \vdash \exists x.\,\varphi \to \bot} \qquad \cfrac{\cfrac{}{\Gamma \vdash \varphi}\,(Ax)}{\Gamma \vdash \exists x.\,\varphi}\,(\exists I)}{\Gamma := \{\neg \exists x.\,\varphi, \varphi\} \vdash \bot}\,(\to E)}{\cfrac{\neg \exists x.\,\varphi \vdash \neg \varphi}{\neg \exists x.\,\varphi \vdash \forall x.\,\neg \varphi}\,(\forall I)}\,(\forall I)}{\vdash \neg \exists x.\,\varphi \to \forall x.\,\neg \varphi}\,(\to I)
$$

MORE EXAMPLES

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{\Gamma \vdash (\exists x \cdot \varphi) \wedge \psi}\,(Ax)
      }{\Gamma \vdash \exists x \cdot \varphi}\,(\wedge E_L)
      \qquad
      \dfrac{}{\Gamma, \varphi \vdash \varphi}\,(Ax)
    }{\Gamma \vdash \varphi}\,(\exists E)
    \qquad
    \dfrac{
      \dfrac{}{\Gamma \vdash (\exists x \cdot \varphi) \wedge \psi}\,(Ax)
    }{\Gamma \vdash \psi}\,(\wedge E_R)
  }{\Gamma \vdash (\varphi \wedge \psi)[x \mapsto x]}\,(\wedge I)
}{
  \dfrac{\Gamma := \{(\exists x \cdot \varphi) \wedge \psi\} \vdash \exists x \cdot \varphi \wedge \psi}{\vdash (\exists x \cdot \varphi) \wedge \psi \to \exists x \cdot \varphi \wedge \psi}\,(\to I)
}\,(\exists I)
$$

UNIVERSAL DEPENDENT TYPES in programming

$A = \mathbb{N}$, $\Sigma = \{\underset{\text{(relation)}}{\text{Vector}} : \underbrace{1, 0, 1, \ldots}_{\text{(consts)}}\}$

Vector not a type, but a type constructor.

For every $n \in \mathbb{N}$, Vector n is a type (formula).

Suppose Vector n is the type of vectors of length n.

↑ depends on the choice of n (types depend on terms)

PRODUCTION

zeroes n : Vector n

$$\text{zeroes } n = (\underbrace{0, \ldots, 0}_{n})$$

what is the type of zeroes itself?

$$\text{zeroes} : \forall n : \mathbb{N} . \text{Vector } n$$

$$\text{zeroes} = \lambda n . (\underbrace{0, \ldots, 0}_{n})$$

CONSUMPTION

sequence of vectors of each len

$$f : (\forall n . \text{Vector } n) \to \mathbb{N}$$

$$f = \lambda g : (\forall n . \text{Vector } n) . \text{head}(g\, 5)$$

EXISTENTIAL DEPENDENT TYPES in programming

PRODUCTION

$$\text{repeat} : \mathbb{N} \to \mathbb{N} \to \overbrace{(\exists n . \text{Vector } n)}^{\text{dependent pair}}$$

$$\text{repeat} = \lambda x . \lambda n . (n, (\underbrace{x, \ldots, x}_{n \text{ times}}))$$

CONSUMPTION

$$\text{length} : (\exists n . \text{Vector } n) \to \mathbb{N}$$

$$\text{length} = \lambda x . \text{let } (n, xs) = x \text{ in } n$$

(alternative writing:

$\text{length} = \lambda (n, xs) . n$)

CURRY

UNCURRY

Alternative type

$$\text{length}' : \forall n . (\text{Vector } n \to \mathbb{N})$$

$$\text{length}' = \lambda n . \lambda xs . n$$

CALCULUS λP_1 for $\{\to, \bot, \wedge, \vee, \forall, \exists\}$, signature Σ

$$
\begin{aligned}
\varphi, \psi ::= \;& x \mid \\
& R(t_1, \dots, t_m) \mid \\
& \bot \mid \\
& \varphi \to \psi \mid \\
& \varphi \wedge \psi \mid \\
& \varphi \vee \psi \mid \\
& \forall x{:}A.\, \varphi \mid \\
& \exists x{:}A.\, \varphi
\end{aligned}
$$

$$
\begin{aligned}
U, V ::= \;& x \mid \\
& \alpha \mid \\
& \varepsilon(U) \mid \\
& \lambda \alpha{:}\varphi.\, U \mid U\,V \mid \\
& (U, V) \mid \pi_1 U \mid \pi_2 V \mid \\
& in_1 U \mid in_2 V \mid \mathrm{case}\ U\ \mathrm{of}\ V_1[\alpha_1]\ \mathrm{or}\ V_2[\alpha_2] \mid \\
& \lambda x{:}A.\, U \mid U\,t \mid \\
& (t, U) \mid \mathrm{let}\ (x{:}A, \alpha{:}\varphi) = U\ \mathrm{in}\ V
\end{aligned}
$$

ranges over the domain A

term built from constant & function symbols in Σ

NORMALISATION of λP_1 EXPRESSIONS (β-reduction)

$$(\lambda \alpha : \varphi \cdot u)\, v \twoheadrightarrow_\beta u[\alpha \mapsto v]$$

$$(\lambda x : A \cdot u)\, t \twoheadrightarrow_\beta u[x \mapsto t]$$

$$\pi_i (u_1, u_2) \twoheadrightarrow_\beta u_i$$

$$\text{case } in_i\, u \text{ of } v_1[\alpha_1] \text{ or } v_2[\alpha_2] \twoheadrightarrow_\beta v_i[\alpha_i \mapsto u]$$

$$\text{let } (x : A, \alpha : \varphi) = (t, u) \text{ in } v \twoheadrightarrow_\beta v[x \mapsto t, \alpha \mapsto u]$$

$\twoheadrightarrow_\beta$ is strongly normalising on λP_1 expression

TYPING RULES for λP_1

$$\frac{}{\Gamma, \alpha : \varphi \vdash \alpha : \varphi}\,(Ax) \qquad \frac{}{\Gamma, x : A \vdash x : A}\,(Ax) \qquad \frac{\Gamma \vdash u : \bot}{\Gamma \vdash \varepsilon(u) : \alpha}\,(\bot E)$$

$$\frac{\Gamma, \alpha : \varphi \vdash u : \psi}{\Gamma \vdash (\lambda \alpha : \varphi \cdot u) : \varphi \to \psi}\,(\to I) \qquad \frac{\Gamma \vdash u : \varphi \to \psi \quad \Gamma \vdash v : \varphi}{\Gamma \vdash u\; v : \psi}\,(\to E)$$

$$\frac{\Gamma \vdash u : \varphi \quad \Gamma \vdash v : \psi}{\Gamma \vdash (u, v) : \varphi \wedge \psi}\,(\wedge I) \qquad \frac{\Gamma \vdash u : \varphi \wedge \psi}{\Gamma \vdash \pi_1 u : \varphi}\,(\wedge E_L) \qquad \frac{\Gamma \vdash u : \varphi \wedge \psi}{\Gamma \vdash \pi_2 u : \psi}\,(\wedge E_R)$$

$$\frac{\Gamma \vdash u : \varphi}{\Gamma \vdash in_1(u) : \varphi \vee \psi}\,(\vee I_1) \qquad \frac{\Gamma \vdash u : \psi}{\Gamma \vdash in_2(u) : \varphi \vee \psi}\,(\vee I_2)$$

$$\frac{\Gamma \vdash u : \varphi \vee \psi \quad \Gamma, x_1 : \varphi \vdash v_1 : \gamma \quad \Gamma, x_2 : \psi \vdash v_2 : \gamma}{\Gamma \vdash \mathrm{case}\; u \;\mathrm{of}\; v_1[x_1] \;\mathrm{or}\; v_2[x_2] : \gamma}\,(\vee E)$$

$$\frac{\Gamma, x : A \vdash u : \varphi}{\Gamma \vdash (\lambda x : A \cdot u) : \forall x : A \cdot \varphi}\,(\forall I) \quad x \notin fv(\Gamma) \qquad \frac{\Gamma \vdash u : \forall x \cdot \varphi}{\Gamma \vdash (u\; t) : \varphi[x \mapsto t]}\,(\forall E)$$

$$\frac{\Gamma \vdash u : \varphi[x \mapsto t]}{\Gamma \vdash (t, u) : \exists x \cdot \varphi}\,(\exists I) \qquad \frac{\Gamma \vdash u : \exists x : A \cdot \varphi \qquad \Gamma, \alpha : \varphi \vdash v : \psi}{\Gamma \vdash (\mathrm{let}\; (x : A, \alpha : \varphi) = u \;\mathrm{in}\; v) : \psi}\,(\exists E)$$

DEPENDENT TYPES in PROGRAMMING

- Popular programming languages / interactive theorem provers based on dependent types: Coq, Agda, Idris, Lean.
- Based on the more expressive λP calculus:
  - No distinction between proofs $\alpha : \varphi$ and values $x : A$ (like in λP_1).
  - Advantages: only one abstraction $\lambda x . \varphi$ and application $u\,v$.
  - Disadvantages: typing is more complicated (details omitted).
- Include also inductive types ($\mathbb{N}$, lists, vectors, ...).

KRIPKE MODELS for INTUITIONISTIC FIRST ORDER LOGIC

Kripke model $K = (W, \leq, [\![\cdot]\!])$, $\leq \subseteq W \times W$ partial order

Domain: $[\![w]\!] \subseteq A$ s.t. $w \leq w'$ implies $[\![w]\!] \subseteq [\![w']\!]$

Interpretation of $R : n \in \Sigma$: $[\![R]\!]_w \subseteq A^n$ s.t. $w \leq w'$ implies $[\![R]\!]_w \subseteq [\![R]\!]_{w'}$

Satisfaction relation:

$w \models R(a_1, \dots, a_n)$ if $(a_1, \dots, a_n) \in [\![R]\!]_w$

$w \models \varphi \wedge \psi$ if $w \models \varphi$ and $w \models \psi$

$w \models \varphi \vee \psi$ if $w \models \varphi$ or $w \models \psi$

$w \models \varphi \to \psi$ if for every $w' \geq w$, $w' \models \varphi$ implies $w \models \psi$

$w \models \forall x \cdot \varphi$ if for every $w' \geq w$ and $a \in [\![w]\!]$, $w \models \varphi[x \mapsto a]$

$w \models \exists x \cdot \varphi$ if there is $a \in [\![w]\!]$ s.t. $w \models \varphi[x \mapsto a]$

SOUNDNESS & COMPLETENESS: $\models \varphi$ iff $\vdash_{ND} \varphi$ satisfied in any w of any K's model

NEGATIVE TRANSLATION of CLASSICAL into INTUITIONISTIC FOL

by structural induction on formulas: (Gentzen, Gödel)

$$\widetilde{R(t_1,\dots,t_n)} \equiv \neg\neg R(t_1,\dots,t_n)$$

$$\widetilde{\varphi \wedge \psi} \equiv \tilde{\varphi} \wedge \tilde{\psi}$$

$$\widetilde{\varphi \vee \psi} \equiv \neg(\neg\tilde{\varphi} \wedge \neg\tilde{\psi})$$

$$\widetilde{\varphi \to \psi} \equiv \tilde{\varphi} \to \tilde{\psi}$$

$$\widetilde{\neg\varphi} \equiv \neg\tilde{\varphi}$$

$$\widetilde{\forall x.\varphi} \equiv \forall x.\tilde{\varphi}$$

$$\widetilde{\exists x.\varphi} \equiv \neg\forall x.\neg\tilde{\varphi}$$

Facts:

1. $\models \varphi \leftrightarrow \tilde{\varphi}$ classically
2. $\tilde{\varphi}$ contains no $\vee, \exists$
3. $\models \varphi$ classically

   iff

   $\models \tilde{\varphi}$ intuitionistically

(details omitted)

Consequences: 1. FOL, IFOL equiconsistent.

2. FOL validity reduces to IFOL validity.