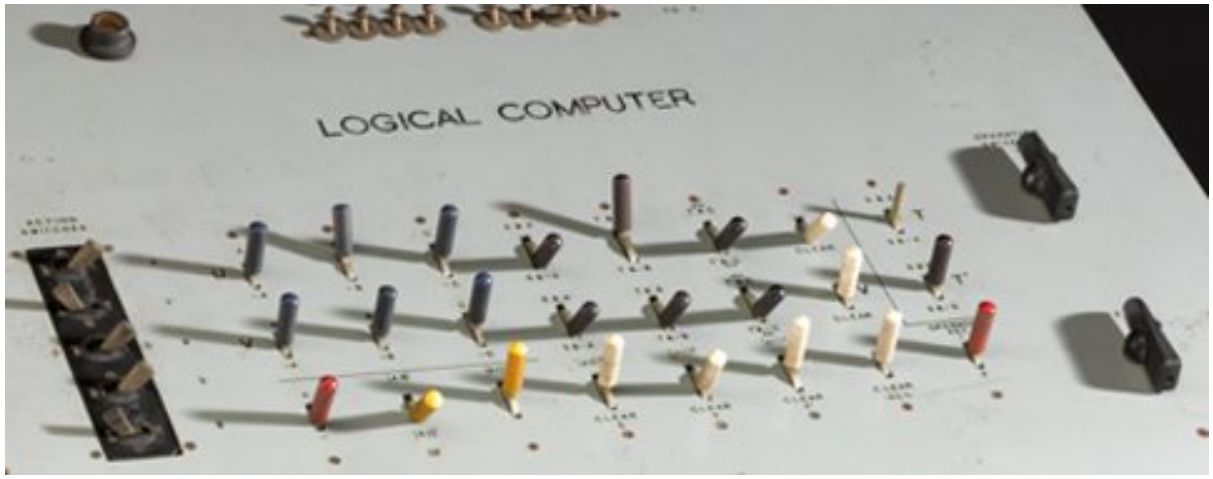
INF551

Computational Logic:

Artificial Intelligence in Mathematical Reasoning



Stéphane Graham-Lengrand

Stephane.Lengrand@Polytechnique.edu

Lecture IX

Intuitionistic logic(s)

& the proofs-as-programs paradigm

The drinker's theorem

"There is always someone such that, if he drinks, everybody drinks"

Let DF be the statement of the drinker's theorem:

$$\exists x(\text{DRINKS}(x) \Rightarrow \forall y\ \text{DRINKS}(y))$$

Contents

I. Excluded-middle vs Double negation

Two possible rules

$$\frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ \text{double negation} \qquad \frac{}{\Gamma \vdash A \vee \neg A}\ \text{excluded middle}$$

. . . allow to derive

$$\frac{\Gamma, \neg A \vdash A}{\Gamma \vdash A} \quad : \quad \frac{\dfrac{}{\Gamma, \neg A \vdash \neg A} \qquad \Gamma, \neg A \vdash A}{\dfrac{\Gamma, \neg A \vdash \bot}{\dfrac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}}} \qquad \frac{\Gamma \vdash A \vee \neg A \qquad \dfrac{}{\Gamma, A \vdash A} \qquad \Gamma, \neg A \vdash A}{\Gamma \vdash A}$$

. . . are interchangeable, in that one rule can be derived using the other:

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \neg(A \vee \neg A), A \vdash A}}{\Gamma, \neg(A \vee \neg A), A \vdash A \vee \neg A}}{\Gamma, \neg(A \vee \neg A), A \vdash \bot}}{\Gamma, \neg(A \vee \neg A) \vdash \neg A}}{\Gamma, \neg(A \vee \neg A) \vdash A \vee \neg A}}{\Gamma \vdash A \vee \neg A} \qquad \frac{\dfrac{\Gamma \vdash \neg\neg A}{\Gamma, \neg A \vdash \neg\neg A} \qquad \dfrac{}{\Gamma, \neg A \vdash \neg A}}{\dfrac{\Gamma, \neg A \vdash \bot}{\dfrac{\Gamma, \neg A \vdash A}{\Gamma \vdash A}}}$$

II. Annoying things in logic so far

Formal proof of the Drinker's theorem

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\ldots \vdash \neg\textsc{drinks}(y)} \quad \dfrac{}{\ldots \vdash \textsc{drinks}(y)}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}), \neg\textsc{drinks}(y), \textsc{drinks}(y) \vdash \bot}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}), \neg\textsc{drinks}(y), \textsc{drinks}(y) \vdash \forall z\ \textsc{drinks}(z)}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}), \neg\textsc{drinks}(y) \vdash \textsc{drinks}(y) \Rightarrow \forall z\ \textsc{drinks}(z)}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}), \neg\textsc{drinks}(y) \vdash \exists x(\textsc{drinks}(x) \Rightarrow \forall z\ \textsc{drinks}(z))}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}), \neg\textsc{drinks}(y) \vdash \bot}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}) \vdash \neg\neg\textsc{drinks}(y)}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}) \vdash \textsc{drinks}(y)}}{\neg\textsc{df}, \textsc{drinks}(\textsc{Bob}) \vdash \forall y\ \textsc{drinks}(y)}}{\neg\textsc{df} \vdash \textsc{drinks}(\textsc{Bob}) \Rightarrow \forall y\ \textsc{drinks}(y)}}{\neg\textsc{df} \vdash \exists x(\textsc{drinks}(x) \Rightarrow \forall y\ \textsc{drinks}(y))}}{\vdash \textsc{df}}$$

Problem with this

We have proved the theorem

. . . but we are still incapable of identifying the person satisfying the property

(or rather, our choice depends on the context)

In other words, we fail to provide a witness of existence

In other words, the logic we use does not have the witness property

The logic we use lacks a certain dose of constructivism

Lack of witness, another example

Predicate P: assuming $P(0), \neg P(2)$

Can we prove that there is an integer x such that $P(x) \wedge \neg P(S(x))$?

$$P(0), \neg P(2) \vdash \exists x \, (P(x) \wedge \neg P(S(x)))$$

Is there an integer n such that we can prove $P(n) \wedge \neg P(S(n))$?

$$P(0), \neg P(2) \vdash P(n) \wedge \neg P(S(n))$$

Concrete example:

Let $u_0 := \sqrt{2}$, $u_{x+1} := u_x^{\sqrt{2}}$, and $P(x)$ be "u_x irrational"

$P(0)$, $P(2)$?

Applying the above: There is x such that $P(x)$ and $\neg P(x+1)$

Therefore: There is an irrational r ($:= u_x$) such that $r^{\sqrt{2}}$ rational

r is either $\sqrt{2}$ or $\sqrt{2}^{\sqrt{2}}$, depending on whether $\sqrt{2}^{\sqrt{2}}$ is rational or not

The annoying mismatch

Remark:

$\vdash A \wedge B$ if and only if both $\vdash A$ and $\vdash B$

The object-level $\wedge$ matches the meta-level "and"

$\vdash \forall x\, A[x]$ if and only if for all terms t we have $\vdash A[t]$ (t not necessarily closed)

The object-level $\forall$ matches the meta-level "for all"

Clearly: If either $\vdash A$ or $\vdash B$ then $\vdash A \vee B$

If there is a term t such that $\vdash A[t]$ then $\vdash \exists x\, A[x]$

But	If you have...	...you don't necessarily have
	$\vdash \exists x\, A[x]$	an n such that $\vdash A[n]$
Example	$\vdash \exists x\, (P(x) \vee \neg P(S(x)))$	an n such that $\vdash P(n) \vee \neg P(S(n))$
	$\vdash A \vee B$	either $\vdash A$ or $\vdash B$
Example	$\vdash A \vee \neg A$	either $\vdash A$ or $\vdash \neg A$ (e.g., A Goedel formula)

For $\vee$ and $\exists$, there is a mismatch between the object-level and the meta-level

The culprit and how to fix the mismatch

In all our examples, mismatch entirely due to:

- the Law of Excluded Middle
- or, equivalently, the Elimination of Double Negation.

The fix is easy: **Disallow** those laws

You get what is called **Intuitionistic Logic**(s) -as opposed to Classical logic(s)

Distinction can be done for propositional logic, predicate logic, higher-order logic, etc.

The claim: we recover a full match between object-level and meta-level

- $\vdash A \wedge B$ if and only if both $\vdash A$ and $\vdash B$
- $\vdash \forall x\ A[x]$ if and only if for all terms t we have $\vdash A[t]$
- $\vdash A \vee B$ if and only if either $\vdash A$ or $\vdash B$
- $\vdash \exists x\ A[x]$ if and only if there is a term t such that $\vdash A[t]$

The above match works in the empty theory, not in any theory!

(imagine the theory $A \vee B$ or the theory $\exists x\ A$)

III. The proofs-as-programs paradigm

Now that we got rid of Excluded Middle/Elimination of Double Negation...

...let's look at the fragment of propositional logic concerning implication only:

$$\frac{}{\Gamma, P \vdash P}$$

$$\frac{\Gamma \vdash P \Rightarrow Q \quad \Gamma \vdash P}{\Gamma \vdash Q}$$

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}$$

What can we say?

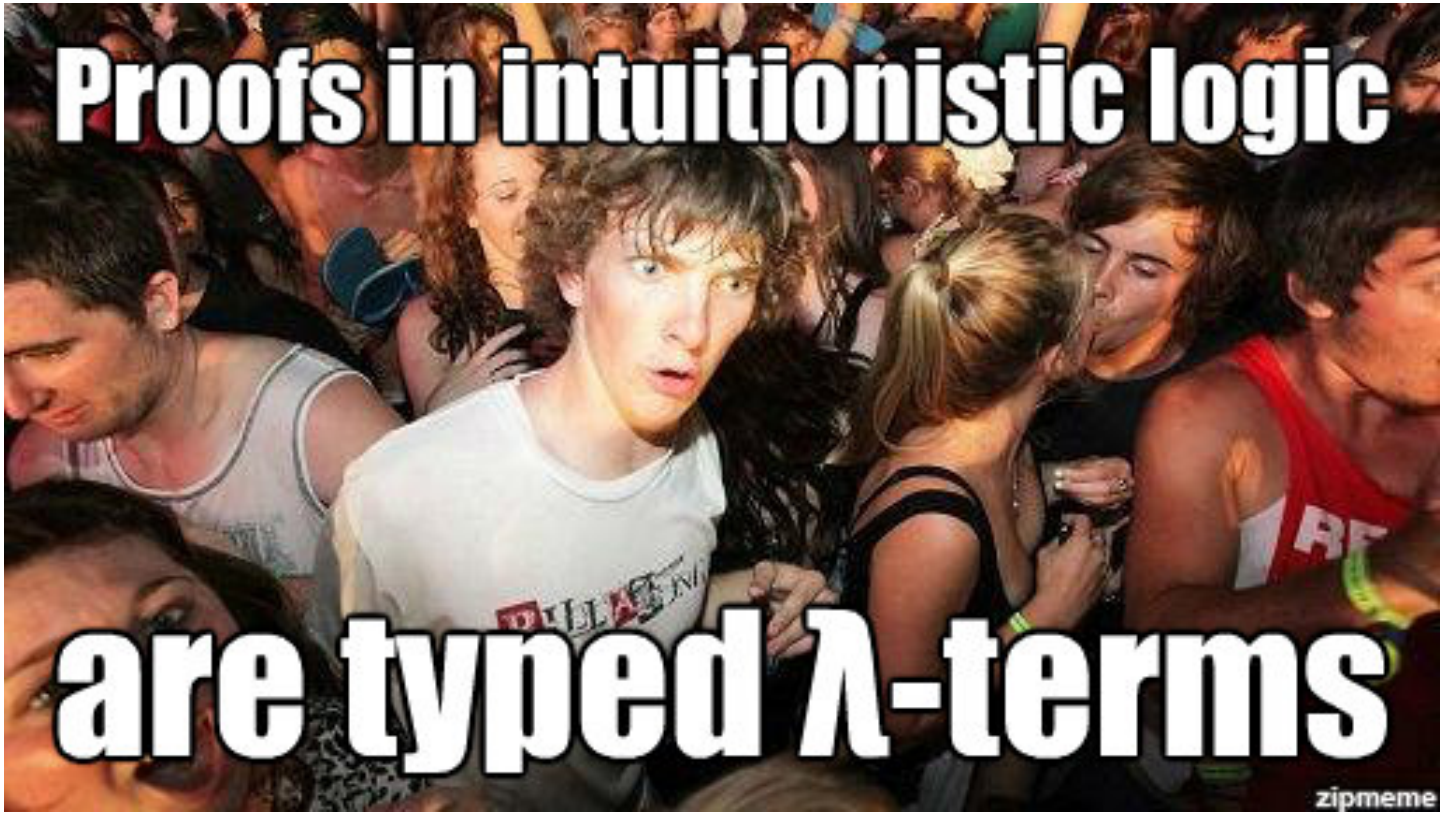
Now let's look again at the typing rules for the λ-calculus:

$$\frac{}{\Delta, x : A \Vdash x : A}$$

$$\frac{\Delta \Vdash t : A \to B \qquad \Delta \Vdash u : A}{\Delta \Vdash t\ u : B}$$

$$\frac{\Delta, x : A \Vdash t : B}{\Delta \Vdash \lambda x.t : A \to B}$$

$$
\begin{array}{ccc}
\Gamma & \longleftrightarrow & \Delta \\
P & \longleftrightarrow & A \\
Q & \longleftrightarrow & B \\
\Rightarrow & \longleftrightarrow & \rightarrow
\end{array}
$$



More precisely

Propositions are Types

Proofs are Programs

Every proof tree in (the implication fragment of) intuitionistic logic

can be annotated to be the typing tree of some λ-term

(λ-calculus variables annotate hypothese, a λ-term annotates the conclusion)

Conversely, every typing tree, for some λ-term t,

can be turned into a proof tree in (the implication fragment of) intuitionistic logic,

simply by hiding variables and λ-term annotations

It is the **Curry-Howard isomorphism**

Re-expressing the rules using annotations

Let's use

- α, β, etc. for the variables annotating hypotheses

  (not to confuse with the variables x, y, etc. in the terms of predicate logic)

- M, N, etc. for the λ-terms annotating proof-trees

  (not to confuse with the terms of predicate logic t, u,...)

$$\frac{}{\Gamma, \alpha : P \vdash \alpha : P}$$

$$\frac{\Gamma, \alpha : P \vdash M : Q}{\Gamma \vdash \lambda\alpha.M : P \Rightarrow Q} \qquad \frac{\Gamma \vdash M : P \Rightarrow Q \qquad \Gamma \vdash N : P}{\Gamma \vdash M\ N : Q}$$

What about the other connectives?

We can extend the λ-calculus

to account for the introduction and elimination rules of the other connectives

$\wedge$, $\vee$

$$\frac{\Gamma \vdash M : P_1 \quad \Gamma \vdash N : P_2}{\Gamma \vdash (M, N) : P_1 \wedge P_2} \qquad \frac{\Gamma \vdash M : P_1 \wedge P_2}{\Gamma \vdash \pi_i(M) : P_i}\, i \in \{1, 2\}$$

$P_1 \wedge P_2$ is a product type "$P_1 * P_2$" (e.g., in OCaml)

$$\frac{\Gamma \vdash M : P_i}{\Gamma \vdash \mathsf{inj}_i(M) : P_1 \vee P_2}\, i \in \{1, 2\}$$

$$\frac{\Gamma \vdash M : P_1 \vee P_2 \quad \Gamma, \alpha_1 : P_1 \vdash N_1 : Q \quad \Gamma, \alpha_2 : P_2 \vdash N_2 : Q}{\Gamma \vdash \mathsf{match}\ M\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto N_1, \mathsf{inj}_2(\alpha_2) \mapsto N_2 : Q}$$

$P_1 \vee P_2$ is a sum type "$P_1 + P_2$"

$\forall$, $\exists$, $\bot$

$$\frac{\Gamma \vdash M : P}{\Gamma \vdash \lambda x.M : \forall x P} \qquad \frac{\Gamma \vdash M : \forall x P}{\Gamma \vdash M\, t : \left\{{}^{t}/_{x}\right\} P}$$

$$\frac{\Gamma \vdash M : \left\{{}^{t}/_{x}\right\} P}{\Gamma \vdash \langle t, M \rangle : \exists x P} \qquad \frac{\Gamma \vdash M : \exists x P \quad \Gamma, \alpha : P \vdash N : Q}{\Gamma \vdash \text{let } \langle x, \alpha \rangle = M \text{ in } N : Q}$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \text{abort}(M) : P}$$

$\neg P$ defined as $P \Rightarrow \bot$, and $\top$ defined as $\neg\bot$ (i.e., $\bot \Rightarrow \bot$)

Summing up the syntax

	Intro-constructs	Elim-constructs	
$M, N, \ldots ::= \alpha$			axiom
$\mid$	$\lambda\alpha.M$	$\mid M\ N$	$\Rightarrow$
$\mid$	(M, N)	$\mid \pi_i(M)$	$\wedge$
$\mid$	$\mathsf{inj}_i(M)$	$\mid$ match M with $\mathsf{inj}_1(\alpha_1) \mapsto N_1, \mathsf{inj}_2(\alpha_2) \mapsto N_2$	$\vee$
$\mid$	$\lambda x.M$	$\mid M\ t$	$\forall$
$\mid$	$\langle t, M\rangle$	$\mid$ let $\langle x, \alpha\rangle = M$ in N	$\exists$
		$\mid \mathsf{abort}(M)$	$\bot$

Reductions

$$(\lambda\alpha.M)\ N \longrightarrow_{\mathsf{root}} \left\{{}^{N}\!/_{\alpha}\right\} M$$

$$\pi_i((M_1, M_2)) \longrightarrow_{\mathsf{root}} M_i$$

$$\mathsf{match}\ \mathsf{inj}_i(M)\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto N_1, \mathsf{inj}_2(\alpha_2) \mapsto N_2 \longrightarrow_{\mathsf{root}} \left\{{}^{M}\!/_{\alpha_i}\right\} N_i$$

$$(\lambda x.M)\ t \longrightarrow_{\mathsf{root}} \left\{{}^{t}\!/_{x}\right\} M$$

$$\mathsf{let}\ \langle x, \alpha\rangle = \langle t, M\rangle\ \mathsf{in}\ N \longrightarrow_{\mathsf{root}} \left\{{}^{t,M}\!/_{x,\alpha}\right\} N$$

+ some permutation rules such as

$$(\mathsf{match}\ M\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto N_1, \mathsf{inj}_2(\alpha_2) \mapsto N_2)\ N$$

$$\longrightarrow_{\mathsf{root}}\ \mathsf{match}\ M\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto (N_1\ N), \mathsf{inj}_2(\alpha_2) \mapsto (N_2\ N)$$

$$\pi_i(\mathsf{match}\ M\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto N_1, \mathsf{inj}_2(\alpha_2) \mapsto N_2)$$

$$\longrightarrow_{\mathsf{root}}\ \mathsf{match}\ M\ \mathsf{with}\ \mathsf{inj}_1(\alpha_1) \mapsto \pi_i(N_1), \mathsf{inj}_2(\alpha_2) \mapsto \pi_i(N_2)$$

$\ldots$

Results from Lecture 7 still hold

Remark: If $\Gamma \vdash M : P$ then $\mathrm{FV}(M)$ is included in the domain of Γ

Substitution: If $\Gamma, \alpha : P \vdash M : Q$ and $\Gamma \vdash N : P$, then $\Gamma \vdash \left\{ {}^{N}/_{\alpha} \right\} M : Q$

Subject Reduction: If $\Gamma \vdash M : P$ and $M \longrightarrow M'$ then $\Gamma \vdash M' : P$

Through the Curry-Howard isomorphism,

This is describing a proof transformation process

Strong Normalisation: If $\Gamma \vdash M : P$ then M is strongly normalising

(careful with the permutation rules, though)

The process of transforming proofs terminates, producing proofs of a particular shape:

the typing trees of irreducible λ-terms

Corollary:

Every theorem P that has a proof in a theory $\mathcal{T}$, also has a proof of that shape

Shape of those proofs in the empty theory

Theorem:

1. Any closed, irreducible and typed λ-term, is an intro-construct
2. There is no closed, irreducible λ-term of type $\bot$

Proof: by simultaneous induction on the size of λ-terms

(and 2. easy consequence of 1.)

Corollary (stilll in the empty theory)

Consistency: intuitionistic predicate logic without axioms is consistent

Proof: If $\bot$ has a proof, it also has a proof whose λ-term is an intro-construct. Impossible.

Witness property: if $\vdash \exists x\ P[x]$ then there is a term t such that $\vdash P[t]$

Proof: The proof can be transformed into a proof annotated by an intro-construct, necessarily $\langle t, M \rangle$, which provides t and the proof of $\vdash P[t]$

Disjunction property: if $\vdash P_1 \vee P_2$ then either $\vdash P_1$ or $\vdash P_2$

Proof: The proof can be transformed into a proof annotated by an intro-construct, necessarily $\mathrm{inj}_i(M)$, where M annotates a proof of $\vdash P_i$

Conclusion: In the empty theory, we recover a full match between object-level and meta-level (in the sense discussed before)

Remark: Law of Excluded Middle would break all of the above approach

In non-empty theories

Axioms labelled by variables α, β, etc.

without computational role

Theorem about shape of irreducible typed λ-terms no longer holds if not closed

In some theories (e.g., $\mathcal{PA}$), a computational role can be given to axioms

Theorem holds again, and its corollaries:

Consistency, Witness property, Disjunction property

Programming by proving

In arithmetic, does $\forall x\, \exists y\, (x = 2 \times y \vee x = 2 \times y + 1)$

have a proof in intuitionistic logic? by induction on x!

What about $\exists y\, (25 = 2 \times y \vee 25 = 2 \times y + 1)$?

What is the witness? How do you compute it?

An intuitionistic proof of

$$\forall x\, \exists y\, (x = 2 \times y \vee x = 2 \times y + 1)$$

is a program that computes the half here by recursion on x!

Its execution mechanism is the proof-transformation process described before

The program is *correct* with respect to the *specification*

$$x = 2 \times y \vee x = 2 \times y + 1$$

See other examples in the practical

Questions?