The Curry-Howard isomorphism

Julio Mariño

Theory of Programming Languages
MASTER IN FORMAL METHODS FOR SOFTWARE ENGINEERING
Universidad (Politécnica | Complutense | Autónoma) de Madrid

December 1, 2020

the trailer...

meet Harold:

- Harold works as a software developer in the City of London
- He has been hired by Schönfinkel, Ltd. to develop a proof compression scheme to be integrated in a top-secret project that uses the *proof carrying code* approach.

the trailer...

meet Harold:

- Harold works as a software developer in the City of London
- He has been hired by Schönfinkel, Ltd. to develop a proof compression scheme to be integrated in a top-secret project that uses the *proof carrying code* approach.

Harold Potter

motivation

- Imagine you are assigned the task of providing a compact, textual representation for *proofs in natural deduction*, e.g. proofs in NJ.

$$\dfrac{\dfrac{\dfrac{[(p \to q) \wedge (q \to r)]^a}{q \to r}\wedge e_2 \qquad \dfrac{\dfrac{[(p \to q) \wedge (q \to r)]^a}{p \to q}\wedge e_1 \qquad [p]^b}{q}\to e}{r}\to e}{p \to r}\to i^b}{((p \to q) \wedge (q \to r)) \to (p \to r)}\to i^a$$

- Basically, proofs are trees, and can be represented as terms, but care has to be taken to keep track of the references to assumptions, here shown in different colours and identified by labels.

IIL (intuitionistic implicative logic)

- In order to simplify the task, we can start by considering just the *implicative fragment* of NJ, that is, the system that only has axioms, $\to e$ and $\to i$.
- This fragment is called IIL (intuitionistic implicative logic).

$$\dfrac{\dfrac{[p \to q]^b \qquad [p]^a}{q}\to e}{\dfrac{p \to q}{(p \to q) \to (p \to q)}\to i^a}\to i^b$$

- We can represent an axiom by a pair, made of a label and a formula:

$$ax(b, p \to q)$$

- A proof ending in an application of the modus ponens rule ($\to e$) has two proofs for the premises and a conclusion:

$$mp(left, right, conclusion)$$

  but the conclusion can be reconstructed from the premises, so we will represent it just as

$$mp(left, right)$$

- Analogously, an application of the ($\to i$) rule will comprise the premise, the label for the assumption being cancelled – we drop the conclusion as before, and, optionally, a copy of the assumption for clarity:

$$imp(label, proof) \quad \text{or} \quad imp(label, assumption, proof)$$

from IIL proofs to terms

- Our example

$$\dfrac{\dfrac{\dfrac{[p \to q]^b \qquad [p]^a}{q} \to e}{p \to q} \to i^a}{(p \to q) \to (p \to q)} \to i^b$$

  can be represented as:

$$imp(b, p \to q, imp(a, p, mp(ax(b, p \to q), ax(a, p))))$$

- If we make some *cosmetic* changes

$$ax(b, p \to q) \rightsquigarrow b_{p \to q}$$

$$mp(left, right) \rightsquigarrow (left\ right)$$

$$imp(x, a, t) \rightsquigarrow \lambda x : a.t$$

  we obtain the following:

$$\lambda b_{p \to q}.\lambda a_p.b_{p \to q} a_p$$

  or, for short:

$$\lambda b : p \to q.\lambda a : p.\ b\ a$$

PROOFS ARE TERMS

- We have just seen that the proof for our example in IIL is (isomorphic to) a Church-style simply typed lambda term (in this case, the APPLY λ-term).
- This is no coincidence, as the three proof elements in IIL (axioms, $\to e$ and $\to i$) are in a one-to-one correspondence to the elements in TA_λ: axioms, APL and ABS.
- Exercise: Prove the isomorphism. (Hint: Use induction on λ-terms.)
- Exercise: Provide a λ-term for the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{[q \to r]^b \qquad \dfrac{[p \to q]^a \qquad [p]^c}{q} \to e}{r} \to e}{p \to r} \to i^c}{(q \to r) \to (p \to r)} \to i^b}{(p \to q) \to ((q \to r) \to (p \to r))} \to i^a$$

What does that λ-term represent?

PROPOSITIONS AS TYPES

- Indeed, the λ-term corresponding to the proof of $(p \to q) \to (p \to q)$ can be assigned the simple type $(p \to q) \to (p \to q)$ in TA_λ, and the one for $(p \to q) \to (q \to r) \to (p \to r)$ can be assigned the type $(p \to q) \to (q \to r) \to (p \to r)$, etc.
- Exercise: Prove it. (Hint: Reuse your previous proof.)
- Exercise: Prove that no λ-term can be assigned the type $(p \to q) \to (q \to p)$.
- Exercise: Give a "convincing argument" that $(p \to (q \to r)) \to (q \to (p \to r))$ is provable in IIL.

implications

(philosophical)

- Taking the isomorphism a bit loosely, every program can be seen as a *constructive proof* of something, maybe its type(s) or specifications. For example, a sorting method is a constructive proof that for every list, a sorted permutation of it exists.
- Is there an interpretation for untypeable λ-terms?
- If proofs in IIL correspond to simply-typed λ-terms, what do proofs in NJ correspond to? (More on this in the following slides.)
- Is there a correspondence for proofs in classical logic (e.g. NK)?
- Is there a correspondence for first-order predicate logic?
- Is there a correspondence for proofs in other proof systems, e.g. Gentzen's sequent calculi (LJ, etc.).
- Did Church really *invent* the λ-calculus? Or was it just waiting to be *discovered*, like the sculpture inside the stone? This may sound bizarre, but the correspondence with natural deduction provides a sense of "necessity" to its very existence. One may argue that while syntax and types arise from the isomorphism, the operational semantics of the λ-calculus (β-reduction) is one original contribution by Church, but as we will see...

REDUCTION IS PROOF SIMPLIFICATION

- Indeed, it was a well known fact that applying the rules of IIL in a certain sequence, for example, one application of ($\to$e) right after ($\to$i) may lead to unnecessarily long proofs:

$$\dfrac{\dfrac{[p]^a}{p \to p} \to \mathrm{i}^a \qquad p^b}{p} \to \mathrm{e}$$

  This is called a *detour*, being

$$p^b$$

  an equivalent proof.
- When β-reduction is applied to the λ-term obtained from a proof in IIL, one detour is eliminated. Terms in β-normal form correspond to proofs without that kind of detours.

completing the isomorphism

unleash your powers!

IIL

simply-typed λ-calculus

NJ

adding conjunction

- A proof ending in an application of the ($\wedge$ i) rule is just a pair of proofs:

$$\frac{\text{proof of } \phi \qquad \text{proof of } \psi}{\text{proof of } \phi \wedge \psi} \wedge \text{i}$$

- A proof ending in an application of the *wedge*-elimination rules ($\wedge$ e_1, $\wedge$ e_2), only has to record which part of the previous proof was used: the *first* or the *second*.
- In summary, if s is a proof for ϕ and t is a proof for ψ, (s, t) is a proof for $\phi \wedge \psi$. If t is a proof for $\phi \wedge \psi$, $fst(t)$ is a proof for ϕ and $snd(t)$ a proof of ψ.
- Exercise: Provide an extended λ-term for the following proof:

$$\dfrac{\dfrac{\dfrac{[(p \to q) \wedge (q \to r)]^a}{q \to r} \wedge e_2 \qquad \dfrac{\dfrac{[(p \to q) \wedge (q \to r)]^a}{p \to q} \wedge e_1 \qquad [p]^b}{q} \to e}{r} \to e}{\dfrac{r}{p \to r} \to i^b}{((p \to q) \wedge (q \to r)) \to (p \to r)} \to i^a$$

- Exercise: Extend β-reduction to λ-terms with pairs and projections.

adding disjunction

- This is an exercise in your second sheet...

Semantics for $\lambda^{\to}$

Definition

A TAS or typed applicative structure $\langle \mathsf{D}, \mathsf{App}, \mathsf{Const} \rangle$ consists of an indexed family $\mathsf{D} = \{\mathsf{D}_\alpha\}$ of domains (sets) D_α for each type α, an indexed family App of functions $\mathsf{App}_{\alpha,\beta} : \mathsf{D}_{\beta\alpha} \times \mathsf{D}_\alpha \to \mathsf{D}_\beta$ for each pair (α, β) of types, and an indexed intepretation function $\mathsf{Const} = \{\mathsf{Const}_\alpha\}$ taking constants of each type α to elements of D_α.

A typed applicative structure is functionally *extensional* if for every f, g in $\mathsf{D}_{\beta\alpha}$, whenever $\mathsf{App}(f, d) = \mathsf{App}(g, d)$ for all $d \in \mathsf{D}_\alpha$ then $f = g$.

Environments

Given a TAS $\mathtt{D}$, an *environment* for $\mathtt{D}$is a function from the set of types variables to $\mathtt{D}$, i.e. from variables of each type α to the carrier $\mathtt{D}_\alpha$.

Environmental Models

Given a typed applicative structure D, an *environmental model* consists of a **total** function $[\![\ ]\!]_\varphi$ from the open terms of the language into D for each assignment φ respecting types, for which the following equalities hold:

$$[\![c]\!]_\varphi = \mathsf{Const}(c) \qquad \text{for constants } c$$

$$[\![x]\!]_\varphi = \varphi(x) \qquad \text{for variables } x$$

$$[\![(MN)]\!]_\varphi = \mathsf{App}([\![M]\!]_\varphi, [\![N]\!]_\varphi)$$

$$\mathsf{App}([\![\lambda x_\alpha.M_\beta]\!]_\varphi, d) = [\![M]\!]_{\varphi[x:=d]} \qquad \text{and it is the unique such function.}$$

The Substitution Lemma

Lemma

For any terms M, N and any environment φ

$$[\![M[N/x]]\!]_\varphi = [\![M]\!]_{\varphi[x:=[\![N]\!]_\varphi]}$$

That is to say: *syntactic and semantic* notions of substitution coincide.
Another useful basic result:
If M is a term and x (of type α) is not free in M then for any $d \in \mathsf{D}_\alpha$ and any environment φ, $[\![M]\!]_{\varphi[x:=d]} = [\![M]\!]_\varphi$

Soundness

Theorem (Soundness)

Suppose M, N are closed terms in $\lambda^{\to}$.
If $M \equiv_{\beta\eta} N$ then in any model $[\![M]\!] = [\![N]\!]$.

Completeness

The converse of the preceding theorem holds:

Theorem (Completeness)

Suppose M, N are closed terms in $\lambda^{\to}$.
If in any model $[\![M]\!] = [\![N]\!]$ then $M \equiv_{\beta\eta} N$.

It turns out that one can pick a single model with this property. In fact two very important, and quite different ones have the property that if they identify interpretations of two closed terms M, N, then the terms must be $\beta\eta$-equivalent.

The Open Term Model

Let D_α be the set of all open terms of type α in normal form, and $\mathsf{App}(M,N) = \mathbf{nf}(MN)$. Then $\mathsf{D} = \langle \mathsf{D}_\alpha, \mathsf{App} \rangle$ with the identity environment yields an environmental model in which for any term M, we have $[\![M]\!]_= \mathbf{nf}(M)$. This gives the completeness theorem as a corollary.

The Full Type Hierarchy

Let D_α be given as follows:

- any infinite set if α is atomic (and these should be pairwise disjoint)
- $\mathsf{D}_{\alpha\to\beta} = \mathsf{D}_\beta^{\mathsf{D}_\alpha}$

and let application be ordinary function application. Then $\mathsf{D} = \langle \mathsf{D}_\alpha, \mathsf{App}\rangle$ (with any) environment yields an environmental model in which for any closed terms M, N, we have $[\![M]\!] = [\![N]\!] \Rightarrow M \equiv_{\beta\eta} N$. This gives the completeness theorem as a corollary.