One-and-a-halfth order Curry-Howard: Incomplete derivations

Murdoch J. Gabbay, www.gabbay.org.uk

Heriot-Watt University
Friday 18 April 2008

Joint work with Dominic Mulligan

What is Curry-Howard?

The Curry-Howard correspondence connects logic with typed λ-calculus:

- Predicates are types.
- Derivations are terms.
- Discharge is λ-abstraction.
- Modus ponens is application.
- β-reduction is proof-normalisation.

For example

$$\dfrac{\dfrac{\dfrac{[\mathsf{A}]^a \quad \mathsf{A}\Rightarrow\mathsf{B}}{\mathsf{B}} \qquad \dfrac{[\mathsf{A}]^a \quad \mathsf{A}\Rightarrow\mathsf{B}\Rightarrow\mathsf{C}}{\mathsf{B}\Rightarrow\mathsf{C}}}{\mathsf{C}}}{\mathsf{A}\Rightarrow\mathsf{C}}\,a \qquad \text{corresponds with} \quad \lambda a.((pa)qa)$$

Here a has type A, p has type $\mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C}$, and q has type $\mathsf{A} \Rightarrow \mathsf{B}$.

Top-down easy, bottom-up not-so-easy

This is fine for ‘top-down’ reasoning, where we assemble derivations. However, we may wish to reason ‘bottom-up’. We start from an incomplete derivation and work backwards to the assumptions. Then we may have ‘half a derivation’, like this:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\; X\\ \mathsf{B}\end{array} \qquad \dfrac{[\mathsf{A}]^a \quad \mathsf{A}\Rightarrow\mathsf{B}\Rightarrow\mathsf{C}}{\mathsf{B}\Rightarrow\mathsf{C}}}{\mathsf{C}}}{\mathsf{A}\Rightarrow\mathsf{C}}\,a$$

Here, the λ-calculus is less helpful. X corresponds with qa in the complete derivation, so (being straighforward about it) the incomplete derivation corresponds with ‘$\lambda a.((pa)X)$’. But X is under a λ-binder.

Top-down easy, bottom-up not-so-easy

X should be instantiated; substituted for without avoiding capture.

This is impossible within the λ-calculus.

There are reasons to care about this.

Philosophical: we are in the business of formalising reasoning and computation. Which logics and calculi underly incomplete derivations and their refinement to complete ones?

Practical: practical reasoning in theorem-provers is often bottom-up. We study Curry-Howard for incomplete derivations as the study of programming on and reasoning about intermediate proof-states.

Related work, our approach

Relevant work by Miller, McBride, Jojgov, Severi, Poll, Bloo, and many others.

We offer an approach based on nominal terms. Here's an example (definitions will follow). We refine an incomplete derivation to a complete one:

$$(\mathbf{1})\ \begin{array}{c}\vdots\ X\\ \bot\Rightarrow\mathsf{A}\end{array}\qquad (\mathbf{2})\ \begin{array}{c}[\bot]^a\\ \vdots\ X\\ \bot\Rightarrow\mathsf{A}\end{array}\qquad (\mathbf{3})\ \dfrac{\begin{array}{c}[\bot]^a\\ \vdots\ X'\\ \mathsf{A}\end{array}}{\bot\Rightarrow\mathsf{A}}(\Rightarrow\mathbf{I})^a\qquad (\mathbf{4})\ \dfrac{\dfrac{[\bot]^a}{\mathsf{A}}(\bot\mathbf{E})}{\bot\Rightarrow\mathsf{A}}(\Rightarrow\mathbf{I})^a$$

$(\mathbf{1})\ X{:}\bot\Rightarrow\mathsf{A}\vdash X{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{2})\ X{:}\bot\Rightarrow\mathsf{A},\ a{:}\bot;\ a\#X\vdash X{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{3})\ a{:}\bot,\ X'{:}\mathsf{A}\vdash\lambda a.X'{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{4})\ a{:}\bot\vdash\lambda a.\mathsf{xf}(a){:}\bot\Rightarrow\mathsf{A}$

An example

$$(\mathbf{1})\ \begin{array}{c} \vdots\, X \\ \bot\Rightarrow\mathsf{A} \end{array} \qquad (\mathbf{2})\ \begin{array}{c} [\bot]^a \\ \vdots\, X \\ \bot\Rightarrow\mathsf{A} \end{array} \qquad (\mathbf{3})\ \dfrac{\begin{array}{c} [\bot]^a \\ \vdots\, X' \\ \mathsf{A} \end{array}}{\bot\Rightarrow\mathsf{A}}\,(\Rightarrow\mathbf{I})^a \qquad (\mathbf{4})\ \dfrac{\dfrac{[\bot]^a}{\mathsf{A}}\,(\bot\mathbf{E})}{\bot\Rightarrow\mathsf{A}}\,(\Rightarrow\mathbf{I})^a$$

$(\mathbf{1})\ X{:}\bot\Rightarrow\mathsf{A} \vdash X{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{2})\ X{:}\bot\Rightarrow\mathsf{A},\ a{:}\bot;\ a\#X \vdash X{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{3})\ a{:}\bot,\ X'{:}\mathsf{A} \vdash \lambda a.X'{:}\bot\Rightarrow\mathsf{A}$

$(\mathbf{4})\ a{:}\bot \vdash \lambda a.\mathsf{xf}(a){:}\bot\Rightarrow\mathsf{A}$

- Assumptions = atoms a. Types = predicates.
- Incomplete parts of the derivation = unknowns X. Types = predicates to be proved.
- Freshness $a\#X$, read 'a is fresh for X' means 'a must be discharged in whatever X is instantiated to'.

Has this been done before?

This paper is ‘just’ about a type system for nominal terms. Has this not been done before? Not in a way that helps us.

A sorting system for nominal terms from Nominal Unification is not suitable; it is designed to construct abstract syntax and atoms have sort ‘the sort of atoms’.

A typing system from Curry-Style types for nominal terms is not suitable; types corresponded to propositional logic with quantifiers whereas here, we want first-order logic and; we also want to represent $(\forall\mathbf{I})$ and $(\forall\mathbf{E})$ so terms (in a moment) may λ-abstract over and be applied to type variables and we require freshness for type variables.

Types, terms

Types: $\phi, \psi, \xi \ ::= \ \bot \mid \phi \Rightarrow \phi \mid \mathsf{P}(\overbrace{a, \ \ldots}^{\mathsf{ar}(\mathsf{P})\text{ var'bles}}) \mid \forall a.\phi.$

For example $\forall a.(\mathsf{P}(a, a) \Rightarrow \mathsf{P}(a, b))$ is a valid type if P has arity 2.

We equate types up to $\forall$-bound atoms. We write $\equiv$ for syntactic identity. We write $\phi[a := b]$ for the usual capture-avoiding substitution action of b for a.

Intuitively, types are first-order logic predicates.

Define free atoms of ϕ as standard:

$$\mathsf{fa}(\mathsf{P}(a, \ldots)) = \{a, \ldots\} \quad \mathsf{fa}(\phi \Rightarrow \psi) = \mathsf{fa}(\phi) \cup \mathsf{fa}(\psi)$$

$$\mathsf{fa}(\bot) = \emptyset \quad \mathsf{fa}(\forall a.\phi) = \mathsf{fa}(\phi) \setminus \{a\}$$

Types, terms

Terms: $r, s, t, \ldots \;::=\; a \mid X \mid \lambda a.r \mid r'r \mid \mathsf{xf}(r)$.

Write $\equiv$ for syntactic equivalence.

We may write $(\lambda a.r)t$ as $r[a{\mapsto}t]$, for example $(\lambda a.b)a \equiv b[a{\mapsto}a]$.

Typing and freshness contexts

A type assignment is a pair $a : \phi$, or $X : \phi$, or $a : *$. A typing context Γ is a finite set of type assignments, functional in that:

- If $a : \phi \in \Gamma$ then $a : * \notin \Gamma$. If $a : * \in \Gamma$ then $a : \phi \notin \Gamma$.
- If $a : \phi \in \Gamma$ and $a : \phi' \in \Gamma$ then $\phi = \phi'$. Similarly for X.

$a : \phi$ means 'a has type ϕ'; $a : *$ means 'a is a type variable'; $X : \phi$ means 'X has type ϕ'.

Typing and freshness contexts

We use atoms for type variables and term variables.

$a : \phi \in \Gamma$ means 'a behaves like a term variable'.

$a : * \in \Gamma$ means 'a behaves like a type variable'.

We could make a syntactic separation between atoms that can have types ($a : \phi \in \Gamma$), and atoms that can appear in types ($a : * \in \Gamma$), but this would duplicate the treatments of λ-abstraction, application, and freshness.

Typing and freshness contexts

Call a pair $a\#r$ of an atom and a term a freshness.

Call a freshness of the form $a\#X$ primitive.

Call a finite set of primitive freshnesses a freshness context.

Δ will range over freshness contexts.

Call $\Gamma; \Delta \vdash r$ a term-in-context.

Call $\Gamma; \Delta \vdash r : \phi$ a typing sequent.

Call $\Gamma; \Delta \vdash a\#r$ a freshness sequent.

Typing and freshness rules

$$\frac{(A \in \{a:\phi,\ a:*,\ X:\phi\})}{\Gamma,\ A;\ \Delta \vdash A}\,(\mathbf{Tax}) \qquad \frac{\Gamma;\Delta \vdash r:\bot}{\Gamma;\Delta \vdash \mathsf{xf}(r):\phi}\,(\mathbf{T\bot E})$$

$$\frac{\Gamma,\ a:\phi;\ \Delta \vdash r:\psi}{\Gamma,\ a:\phi;\ \Delta \vdash \lambda a.r:\phi \Rightarrow \psi}\,(\mathbf{T{\Rightarrow}I}) \qquad \frac{\Gamma;\ \Delta \vdash r':\phi \Rightarrow \psi \quad \Gamma;\ \Delta \vdash r:\phi}{\Gamma;\ \Delta \vdash r'r:\psi}\,(\mathbf{T{\Rightarrow}E})$$

$$\frac{\Gamma,\ a:*;\ \Delta \vdash r:\phi \quad a \notin \mathsf{fa}(\mathsf{important}(\Gamma,\ a:*;\Delta \vdash r))}{\Gamma,\ a:*;\ \Delta \vdash \lambda a.r:\forall a.\phi}\,(\mathbf{T\forall I})$$

$$\frac{\Gamma,\ b:*;\ \Delta \vdash r:\forall a.\phi}{\Gamma,\ b:*;\ \Delta \vdash rb:\phi[a:=b]}\,(\mathbf{T\forall E})$$

Typing and freshness rules

$$\frac{\Gamma,\ A;\ \Delta, b\#\mathcal{X} \vdash r : \phi \quad \Gamma,\ A;\ \Delta, b\#\mathcal{X} \vdash b\#r \quad (A \in \{b : \psi,\ b : *\},\ b \notin \Delta)}{\Gamma; \Delta \vdash r : \phi}\,(\mathbf{Tfr})$$

$$\frac{}{\Gamma,\ a : \phi,\ b : \phi; \Delta \vdash a\#b}\,(\mathbf{a\#b}) \qquad \frac{}{\Gamma,\ a : *,\ b : *; \Delta \vdash a\#b}\,(\mathbf{a\#b''})$$

$$\frac{}{\Gamma; \Delta \vdash a\#\lambda a.r}\,(\mathbf{a\#\lambda a}) \qquad \frac{\Gamma; \Delta \vdash a\#r}{\Gamma; \Delta \vdash a\#\lambda b.r}\,(\mathbf{a\#\lambda b}) \qquad \frac{}{\Gamma; \Delta,\ a\#X \vdash a\#X}\,(\mathbf{a\#X})$$

$$\frac{(a \notin \mathsf{fa}(\phi))}{\Gamma,\ a : *,\ b : \phi; \Delta \vdash a\#b}\,(\mathbf{a\#b'}) \qquad \frac{\Gamma; \Delta \vdash a\#r' \quad \Gamma; \Delta \vdash a\#r}{\Gamma; \Delta \vdash a\#r'r}\,(\mathbf{a\#app})$$

For comparison: natural deduction derivation rules

$$\frac{\bot}{\phi}\,(\bot\mathbf{E}) \qquad \frac{\begin{matrix}[\phi] \quad \Phi \\ \vdots \\ \psi\end{matrix}}{\phi \Rightarrow \psi}\,(\Rightarrow\mathbf{I}) \qquad \frac{\phi \Rightarrow \psi \quad \phi}{\psi}\,(\Rightarrow\mathbf{E})$$

$$\frac{\begin{matrix}\Phi \\ \vdots \\ \phi \quad (a \notin \mathsf{fa}(\Phi))\end{matrix}}{\forall a.\phi}\,(\forall\mathbf{I}) \qquad \frac{\forall a.\phi}{\phi[a := b]}\,(\forall\mathbf{E})$$

Important! $(\mathbf{T}\forall\mathbf{I})$

We write $\mathsf{important}(\Gamma;\Delta\vdash r)$ for $\{\phi \mid a:\phi\in\Gamma,\ \Gamma;\Delta\nvdash a\#r\}$.

$$\dfrac{\begin{array}{c}\Phi\\ \vdots\, r\\ \phi\end{array}\quad (a\notin \mathsf{fa}(\Phi))}{\forall a.\phi}\,(\forall\mathbf{I})$$

We don't know exactly what atoms were used to deduce ϕ, because r may contain unknowns.

$a\notin\mathsf{fa}(\mathsf{important}(\Gamma, a{:}*;\Delta\vdash r))$ in $(\mathbf{T}\forall\mathbf{I})$ generalises this to take account of unknowns and freshness assumptions on them.

Example typings

Write Γ for $a : \mathsf{A},\ p : \mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C},\ q : \mathsf{A} \Rightarrow \mathsf{B}$:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{}{\Gamma \vdash a : \mathsf{A}}\,(\mathbf{Tax})
      \qquad
      \dfrac{}{\Gamma \vdash q : \mathsf{A} \Rightarrow \mathsf{B}}\,(\mathbf{Tax})
    }{\Gamma \vdash qa : \mathsf{B}}\,(\mathbf{Ty{\Rightarrow}E})
    \qquad
    \dfrac{
      \dfrac{}{\Gamma \vdash a : \mathsf{A}}\,(\mathbf{Tax})
      \qquad
      \dfrac{}{\Gamma \vdash p : \mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C}}\,(\mathbf{Tax})
    }{\Gamma \vdash pa : \mathsf{B} \Rightarrow \mathsf{C}}\,(\mathbf{T{\Rightarrow}E})
  }{
    \dfrac{\Gamma \vdash (pa)qa : \mathsf{C}}{\Gamma \vdash \lambda a.((pa)qa) : \mathsf{A} \Rightarrow \mathsf{B}}\,(\mathbf{T{\Rightarrow}I})
  }\,(\mathbf{Ty{\Rightarrow}E})
}{p : \mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C},\ q : \mathsf{A} \Rightarrow \mathsf{B} \vdash \lambda a.((pa)qa) : \mathsf{A} \Rightarrow \mathsf{B}}\,(\mathbf{Tfr})
$$

Example typings

$$
\dfrac{
  \dfrac{
    \dfrac{}{\Gamma,\ X:\mathsf{B} \vdash X:\mathsf{B}}\,(\mathbf{Tax})
    \qquad
    \dfrac{
      \dfrac{}{\Gamma,\ X:\mathsf{B} \vdash a:\mathsf{A}}\,(\mathbf{Tax})
      \qquad
      \dfrac{}{\Gamma,\ X:\mathsf{B} \vdash p:\mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C}}\,(\mathbf{Tax})
    }{\Gamma,\ X:\mathsf{B} \vdash pa:\mathsf{B} \Rightarrow \mathsf{C}}\,(\mathbf{T{\Rightarrow}E})
  }{
    \dfrac{\Gamma,\ X:\mathsf{B} \vdash (pa)X:\mathsf{C}}{\Gamma,\ X:\mathsf{B} \vdash \lambda a.((pa)X):\mathsf{A} \Rightarrow \mathsf{B}}\,(\mathbf{T{\Rightarrow}I})
  }\,(\mathbf{Ty{\Rightarrow}E})
}{p:\mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C},\ q:\mathsf{A} \Rightarrow \mathsf{B},\ X:\mathsf{B} \vdash \lambda a.((pa)X):\mathsf{A} \Rightarrow \mathsf{B}}\,(\mathbf{Tfr})
$$

Derivations of $\Gamma \vdash a\#\lambda a.((pa)qa)$ and $\Gamma,\ X:\mathsf{B} \vdash a\#\lambda a.((pa)X)$ are elided.

Example typings

Another example illustrates the side-condition on $(\mathbf{T}\forall\mathbf{I})$:

$$\dfrac{\dfrac{\dfrac{\mathsf{A} \qquad \dfrac{\forall c.(\mathsf{A} \Rightarrow \mathsf{P}(c))}{\mathsf{A} \Rightarrow \mathsf{P}(c)}\,(\forall\mathbf{E})}{\mathsf{P}(c)}\,(\Rightarrow\mathbf{E})}{\forall c.\mathsf{P}(c)}\,(\forall\mathbf{I}) \qquad (\forall c.\mathsf{P}(c)) \Rightarrow \mathsf{B}}{\mathsf{B}}\,(\Rightarrow\mathbf{E})
\qquad\qquad
\dfrac{\begin{array}{c}\vdots\; X\\ \forall c.\mathsf{P}(c)\end{array} \qquad (\forall c.\mathsf{P}(c)) \Rightarrow \mathsf{B}}{\mathsf{B}}\,(\Rightarrow\mathbf{E})$$

are represented, writing Γ for

$a : \mathsf{A},\ p : \forall c.(\mathsf{A} \Rightarrow \mathsf{P}(c)),\ q : (\forall c.\mathsf{P}(c)) \Rightarrow \mathsf{B},\ c : *$, by:

Example typings

$$
\dfrac{
  \dfrac{
    \dfrac{}{\Gamma \vdash a{:}\mathsf{A}}\,(\mathbf{Tax})
    \qquad
    \dfrac{
      \dfrac{}{\Gamma \vdash p : \forall c.(\mathsf{A}\Rightarrow\mathsf{P}(c))}\,(\mathbf{Tax})
    }{\Gamma \vdash pc : \mathsf{A}\Rightarrow\mathsf{P}(c)}\,(\mathbf{T}\forall\mathbf{E})
  }{\Gamma \vdash pca : \mathsf{P}(c)}\,(\mathbf{T}{\Rightarrow}\mathbf{E})
  \quad
  \begin{pmatrix} c\notin\mathsf{fa}(\mathsf{A}), \\ c\notin\mathsf{fa}(\forall c.(\mathsf{A}\Rightarrow\mathsf{P}(c))) \end{pmatrix}
}{\Gamma \vdash \lambda c.(pca) : \forall c.\mathsf{P}(c)}\,(\mathbf{T}\forall\mathbf{I})
$$

$$
\dfrac{
  \dfrac{
    \Gamma \vdash \lambda c.(pca) : \forall c.\mathsf{P}(c)
    \qquad
    \dfrac{}{\Gamma \vdash q : (\forall c.\mathsf{P}(c))\Rightarrow\mathsf{B}}\,(\mathbf{Tax})
  }{\Gamma \vdash q(\lambda c.(pca)) : \mathsf{B}}\,(\mathbf{T}{\Rightarrow}\mathbf{E})
}{a : \mathsf{A},\ p : \forall c.(\mathsf{A} \Rightarrow \mathsf{P}(c)),\ q : (\forall c.\mathsf{P}(c)) \Rightarrow \mathsf{B} \vdash q(\lambda c.(pca)) : \mathsf{B}}\,(\mathbf{Tfr})
$$

Example typings

$$
\dfrac{
  \dfrac{
    \dfrac{}{\Gamma,\ X : \mathsf{P}(c) \vdash X : \mathsf{P}(c)}\,(\mathbf{Tax})
    \quad
    \begin{pmatrix}
      c \notin \mathsf{fa}(\mathsf{A}) \\
      c \notin \mathsf{fa}(\forall c.(\mathsf{A}{\Rightarrow}\mathsf{P}(c))) \\
      c \notin \mathsf{fa}((\forall c.\mathsf{P}(c)){\Rightarrow}\mathsf{B}))
    \end{pmatrix}
  }{\Gamma,\ X : \mathsf{P}(c) \vdash \lambda c.X : \forall c.\mathsf{P}(c)}\,(\mathbf{T}\forall\mathbf{I})
  \quad
  \dfrac{}{\Gamma,\ X : \mathsf{P}(c) \vdash q : (\forall c.\mathsf{P}(c)){\Rightarrow}\mathsf{B}}\,(\mathbf{Tax})
}{
  \dfrac{\Gamma,\ X : \mathsf{P}(c) \vdash q(\lambda c.X) : \mathsf{B}}{a : \mathsf{A},\ p : \forall c.(\mathsf{A} \Rightarrow \mathsf{P}(c)),\ q : (\forall c.\mathsf{P}(c)) \Rightarrow \mathsf{B},\ X : \mathsf{P}(c) \vdash q(\lambda c.X) : \mathsf{B}}\,(\mathbf{Tfr})
}\,(\mathbf{T{\Rightarrow}E})
$$

Derivations of freshnesses are elided.

Admissible rules

$$\frac{\Gamma;\Delta \vdash r : \phi \quad (X \notin \Gamma)}{\Gamma,\ X : \psi;\Delta \vdash r : \phi}\,(\mathbf{WeakX})$$

$$\frac{\Gamma;\Delta \vdash r : \phi \quad (B \in \{b : \psi,\ b : *\},\ a \notin \Gamma)}{\Gamma,\ B;\Delta, b\#\mathsf{unkn}(\Gamma) \vdash r : \phi}\,(\mathbf{Weaka})$$

$$\frac{\Gamma, X{:}\psi;\Delta, a_1\#X, \ldots, a_n\#X \vdash r : \phi \quad \Gamma;\Delta \vdash t{:}\psi \quad \Gamma;\Delta \vdash a_i\#t\ (1{\leq}i{\leq}n) \quad (X{\notin}\Delta)}{\Gamma;\Delta \vdash r[X := t] : \phi}\,(\mathbf{Cut})$$

Conclusions

‘Holes’ commonly turn up where there are meta-variables. Freshness side-conditions on those holes are also common:

- λ-calculus: $(\lambda x.r)[y{\mapsto}t] = \lambda x.(r[y{\mapsto}t])$ if x is fresh for t
- π-calculus: $\nu x.(P \mid Q) = P \mid \nu x.Q$ if x is fresh for P
- First-order logic: $\forall x.(\phi \Rightarrow \psi) = \phi \Rightarrow \forall x.\psi$ if x is fresh for ϕ

…and the Curry-Howard correspondence, where X is an ‘unknown derivation’ and $a\#X$ means ‘a is discharged in X’.