Typage

Recursive types

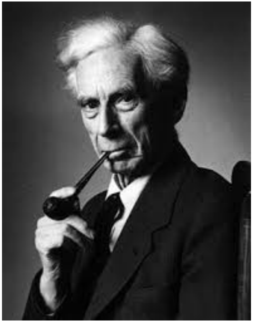
2018-2019
Giovanni Bernardi, gioXYZirif.fr
http://www.irif.fr/~gio/index.xhtml
Université Paris Diderot

Plan

1. Questions
2. Mini historical remarks
3. More fixed points
4. Deciding type equivalence
5. A type system with recursive types

Who conceived types ?

Who conceived types ?



Mathematical Logic
as Based on the Theory of Types
B. Russell
1908

Why?

$$A = \{ x \mid x \notin x \}$$

I'm being brief here...

Who brought types into PL?

Who brought types into PL?

A Formulation of the Simple Theory of Types

A. Church

1940

Why?

Think of properties of well-typed terms

Circularity

$$
\begin{aligned}
\mathit{fact} &\triangleq \lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\mathit{fact}(x-1)) \\
\texttt{List}\ 'a &\triangleq []\ |\ 'a : \texttt{List}\ 'a
\end{aligned}
$$

How to treat with circularity?

Circularity

$$\underline{\mathit{fact}} \triangleq \lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\underline{\mathit{fact}}(x-1))$$

$$\underline{\texttt{List}\ 'a} \triangleq [\,]\ |\ 'a : \underline{\texttt{List}\ 'a}$$

How to treat with circularity?

Circularity

$$\underline{fact} \quad \triangleq \quad \lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\underline{fact}(x-1))$$

$$\underline{\texttt{List}\ 'a} \quad \triangleq \quad [\,]\ |\ 'a : \underline{\texttt{List}\ 'a}$$

structure

$$\boxed{x \triangleq F(x)}$$

x fixed point of F

How to treat with circularity?

			notation
least fixed points	induction	recursion	μf
greatest fixed points	coinduction	corecursion	νf

Induction order-theoretic appoach[1]

Theorem (Kleene, 1936)

Let $\langle P, \leq \rangle$ be a CPO and $f : P \to P$ a continuous function. We have $\mu f = \bigcup_{n \geq 0} f^n(\bot)$. $\square$

[1] See Section 2.3 book by Sangiorgi.

Induction order-theoretic appoach[1]

A non-empty set D is a poset if equipped with a binary relation $\mathcal{R}$ reflexive, antisymmetric, and transitive. Notation $\langle D, \mathcal{R} \rangle$.
A poset $\langle D, \leq \rangle$ is

- *directed* if $D \neq \emptyset$ and $\forall a, b \in D. \exists c \in D. a \leq c$ and $b \leq c$.
- a *complete partial order* (CPO) if
  - D has a bottom $\perp$ element
  - $\bigsqcup D'$ exists for every directed subset of D' of D

Let $\langle P, \leq \rangle$, $\langle Q, \sqsubseteq \rangle$ be CPO. A function $f : P \to Q$ is *continuous* if for every directed subet D of P

- $f(D)$ is directed
- $f(\bigsqcup D) = \bigsqcup f(D)$

Theorem (Kleene, 1936)

Let $\langle P, \leq \rangle$ be a CPO and $f : P \to P$ a continuous function. We have $\mu f = \bigcup_{n \geq 0} f^n(\perp)$. □

[1] See Section 2.3 book by Sangiorgi.

A typical CPO

Let $\mathit{parts}(S) = \{S' \mid S' \subseteq S\}$.
For every non-empty set S the poset $\langle S, \subseteq \rangle$ is a CPO.

Example

Let $S = \{a, b, c\}$. The poset $\langle \mathit{parts}(S), \subseteq \rangle$ is

$$\begin{tikzcd}
& S \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
\{a, b\} \arrow[d, dash] \arrow[dr, dash] & \{a, c\} \arrow[dl, dash] \arrow[dr, dash] & \{b, c\} \arrow[dl, dash] \arrow[d, dash] \\
\{a\} \arrow[dr, dash] & \{b\} \arrow[d, dash] & \{c\} \arrow[dl, dash] \\
& \emptyset &
\end{tikzcd}$$

Factorial as least fixed point

$$\begin{array}{lcl} F(\underline{y}) & \triangleq & \lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\underline{y}(x-1)) \\ F & : & (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}) \end{array}$$

- $\langle \mathbb{N}^{\mathbb{N}}, \leq \rangle$ CPO with bottom $\emptyset$ and $F(y)$ continuous in y,

$$\mu y. F(y) = \bigcup_{n \geq 0} F^n(\emptyset)$$

- NB: $\mu y. F(y)$ is a function!

Factorial as least fixed point

$$
\begin{array}{lcl}
F(\underline{y}) & \triangleq & \lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\underline{y}(x-1)) \\
F & : & (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}) \\
\mathit{fact} & \triangleq & \mu y.F(y)
\end{array}
$$

- $\langle \mathbb{N}^{\mathbb{N}}, \leq \rangle$ CPO with bottom $\emptyset$ and $F(y)$ continuous in y,

$$\mu y.F(y) = \bigcup_{n \geq 0} F^n(\emptyset)$$

- NB: $\mu y.F(y)$ is a function!

from "definition" to property

$$\boxed{\mathit{fact}(x) = \ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (\mathit{fact}(x-1))}$$

Least fixed point λ-theoretic approach

$$F \triangleq \lambda y.\lambda x.\ \texttt{if}\ x = 0\ \texttt{then}\ 1\ \texttt{else}\ x * (y(x-1))$$

$$\mathcal{Y} \triangleq \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

Theorem (Kleene, 1936)

For every λ-term M we have $\mathcal{Y}M \stackrel{\beta}{=} M(\mathcal{Y}M)$. □

Theorem (Morris, 1968)

For every λ-term M, A if $A \stackrel{\beta}{=} MA$ then $\mathcal{Y}M \leq A$. □

We get:

- $\mathcal{Y}F$ is a fixed point of F $\qquad B \stackrel{\beta}{=} F(B)$
- $\mathcal{Y}F$ is the least fixed point of F

Can $\mathcal{Y}$ be typed ? intuitive argument

Let $M = \lambda x.f(xx)$ and $\Gamma = \{x : A, x : A \to A, f : A \to B\}$.

type derivation of sub-term of $\mathcal{Y}$

$$\dfrac{\dfrac{}{\Gamma \vdash f : A \to B} \qquad \dfrac{\dfrac{}{\Gamma \vdash x : A \to A} \qquad \dfrac{}{\Gamma \vdash x : A}}{\Gamma \vdash xx : A}}{\dfrac{\Gamma \vdash f(xx) : B}{\Gamma \vdash \lambda x.f(xx) : A \to B}}$$

We need a type that satisfies

$$\boxed{A = A \to A}$$

μ-Types

$$A \;::=\; \mathcal{T} \mid \underline{x} \mid \underline{\mu x.A} \mid A \times A \mid A \to A$$

- $\mu x.T$ binds x in T, free and bound variables as expected
- μ-types are closed and <u>contractive</u> terms

when are two types equal ?

$$\boxed{\begin{array}{lcl}
\mu y.y & \stackrel{?}{=} & \mu x.z \\
\mu y.y & \stackrel{?}{=} & \mu x.x \\
\mu x.(int \times x) & \stackrel{?}{=} & int \times \mu x.(int \times x) \\
\mu x.x \to x & \stackrel{?}{=} & (\mu x.x \to x) \to (\mu x.x \to x)
\end{array}}$$

μ-Types

$$A \;::=\; \mathcal{T} \;\mid\; \underline{x} \;\mid\; \underline{\mu x.A} \;\mid\; A \times A \;\mid\; A \to A$$

- $\mu x.T$ binds x in T, free and bound variables as expected
- μ-types are closed and <u>contractive</u> terms

A *<u>contractive</u>* if for any subexpression of A of the form

$$\mu x.\mu x_1.\mu x_2.\ldots.\mu x_n.B$$

the term B is not x.

$$\mu y.y \;=\; \mu x.x$$

$$\mu x.(int \times x) \;\stackrel{?}{=}\; int \times \mu x.(int \times x)$$

$$\mu x.x \to x \;\stackrel{?}{=}\; (\mu x.x \to x) \to (\mu x.x \to x)$$

μ-Types

$$A \quad ::= \quad \mathcal{T} \;\mid\; \underline{x} \;\mid\; \underline{\mu x.A} \;\mid\; A \times A \;\mid\; A \to A$$

- $\mu x.T$ binds x in T, free and bound variables as expected
- μ-types are closed and contractive terms

when are two types equal ?

$$\begin{aligned}
\mu y.y &\stackrel{?}{=} \mu x.z \\
\mu y.y &\stackrel{?}{=} \mu x.x \\
\mu x.(int \times x) &\stackrel{?}{=} int \times \mu x.(int \times x) \\
\mu x.x \to x &\stackrel{?}{=} (\mu x.x \to x) \to (\mu x.x \to x)
\end{aligned}$$

Type equivalence semantic approach

Σ: set of symbols with an arity ranked alphabet

A <u>*tree*</u> over a ranked alphabet Σ is a partial function $t : \mathbb{N}_+^\star \to \Sigma$ such that

- *dom*(t) non-empty
- *dom*(t) prefix-closed
- for all $\pi \in dom(t)$
  - $i, j \in N_+^\star, 1 \leq i \leq j$ and $\pi j \in dom(t)$ imply $\pi i \in dom(t)$
  - $t(\pi) = A$ of arity $k \geq 0$ implies for $i \in \mathbb{N}_+, \pi i \in dom(t)$ iff $1 \leq i \leq k$

Extensional equivalence (naïve)

- f, g functions
- $f \stackrel{ext}{=} g$ if $dom(f) = dom(g)$ and $\forall x \in dom(f).\, f(x) = g(x)$

Type equivalence semantic approach

$$\Sigma = \mathcal{T} \cup \{\times, \rightarrow\}$$

$$\begin{array}{lcll}
\mathit{treeof}(c)(\varepsilon) & = & c & \text{where } c \in \mathcal{T} \\
\mathit{treeof}(A_1 \rightarrow A_2)(\varepsilon) & = & \rightarrow & \\
\mathit{treeof}(A_1 \rightarrow A_2)(i\pi) & = & \mathit{treeof}(A_i)(\pi) & \\
\vdots & & & \\
\mathit{treeof}(\mu x.A)(\pi) & = & \mathit{treeof}(A\{x/\mu x.A\})(\pi) &
\end{array}$$

Lemma

For every μ-type A the $\mathit{treeof}(A)$ is defined. **Why ?** $\square$

Let $A \stackrel{ext}{=} B$ whenever $\mathit{treeof}(A) \stackrel{ext}{=} \mathit{treeof}(B)$

Type equivalence semantic approach

$$\Sigma = \mathcal{T} \cup \{\times, \rightarrow\}$$

$$\begin{array}{lcll}
\mathit{treeof}(c)(\varepsilon) & = & c & \text{where } c \in \mathcal{T} \\
\mathit{treeof}(A_1 \rightarrow A_2)(\varepsilon) & = & \rightarrow & \\
\mathit{treeof}(A_1 \rightarrow A_2)(i\pi) & = & \mathit{treeof}(A_i)(\pi) & \\
\vdots & & & \\
\mathit{treeof}(\mu x.A)(\pi) & = & \mathit{treeof}(A\{x/\mu x.A\})(\pi) &
\end{array}$$

Lemma

For every μ-type A the $\mathit{treeof}(A)$ is defined. **Why ?** $\square$

Let $A \stackrel{ext}{=} B$ whenever $\mathit{treeof}(A) \stackrel{ext}{=} \mathit{treeof}(B)$

$$\boxed{\text{How to decide } \stackrel{ext}{=} \text{ ?}}$$

Type equivalence semantic approach

Let $A \overset{ext}{=} B$ whenever $\mathit{treeof}(A) \overset{ext}{=} \mathit{treeof}(B)$

- Fix two μ-types A, B
- to prove $A \overset{ext}{=} B$ we show

$$\boxed{\forall \pi} \in \{1,2\}^{\star}.\ \mathit{treeof}(A)(\pi) = \mathit{treeof}(B)(\pi)$$

general issue

$\boxed{\text{universal quantification}}$

not a problem if trees regular

real question: axiomatisation

$\boxed{\text{Can we characterise } \overset{ext}{=} \text{ syntactically?}}$

- Try typing $\mathcal{Y}$ in ocaml
- Find useful sections in Chapter 21 Pierce book
- Implement *treeof*

- Work on the project