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

\[ \text{fact} \; \triangleq \; \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (\text{fact}(x - 1)) \]

\[ \text{List } 'a \; \triangleq \; [] \mid 'a : \text{List } 'a \]

How to treat with circularity?
Circularity

\[ \text{fact} \triangleq \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (\text{fact}(x - 1)) \]

\[ \text{List}'a \triangleq [] \mid 'a : \text{List}'a \]

How to treat with circularity?
Circularity

\[
\text{fact} \triangleq \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (\text{fact}(x - 1))
\]

\[
\text{List}'a \triangleq [] | 'a : \text{List}'a
\]

How to treat with circularity?

least fixed points induction recursion \( \mu f \)
greatest fixed points coinduction corecursion \( \nu f \)
Theorem (Kleene, 1936)

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

\[ \]
Induction order-theoretic approach\textsuperscript{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 $\bot$ 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 subset $D$ 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 $\mu f = \bigcup_{n \geq 0} f^n(\bot)$.

\textsuperscript{1}See Section 2.3 book by Sangiorgi.
A typical CPO

Let \( \text{parts}(S) = \{ S' \mid S' \subseteq S \} \).

For every non-empty set \( S \) the poset \( \langle S, \subseteq \rangle \) is a CPO.

Example

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

\[
\begin{array}{c}
S \\
\{a, b\} & \{a, c\} & \{b, c\} \\
\{a\} & & \{b\} & \{c\} \\
\emptyset
\end{array}
\]
Factorial as least fixed point

\[ F(y) \triangleq \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (y(x - 1)) \]

\[ F : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow (\mathbb{N} \rightarrow \mathbb{N}) \]

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

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

\[ \text{NB: } \mu y. F(y) \text{ is a function!} \]
Factorial as least fixed point

\[ F(y) \triangleq \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (y(x - 1)) \]

\[ F : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}) \]

\[ \text{fact} \triangleq \mu y.F(y) \]

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

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

\( \text{NB: } \mu y.F(y) \) is a function!

from “definition” to property

\[ \text{fact}(x) = \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (\text{fact}(x - 1)) \]
Least fixed point \( \lambda \)-theoretic approach

\[
F \overset{\Delta}{=} \lambda y. \lambda x. \text{if } x = 0 \text{ then } 1 \text{ else } x \ast (y(x - 1))
\]

\[
\mathcal{Y} \overset{\Delta}{=} \lambda f. (\lambda x. f(xx))(\lambda x. f(xx))
\]

Theorem (Kleene, 1936)

For every \( \lambda \)-term \( M \) we have \( \mathcal{Y}M \overset{\beta}{=} M(\mathcal{Y}M) \).

Theorem (Morris, 1968)

For every \( \lambda \)-term \( M, A \) if \( A \overset{\beta}{=} MA \) then \( \mathcal{Y}M \leq A \).

We get:

- \( \mathcal{Y}F \) is a fixed point of \( F \)
- \( \mathcal{Y}F \) is the least fixed point of \( F \)

\( B \overset{\beta}{=} F(B) \)
Can $\mathcal{Y}$ be typed? intuitive argument

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

We need a type that satisfies $A = A \to A$
\( \mu \)-Types

\[
A ::= \mathcal{T} \mid x \mid \mu x.A \mid A \times A \mid A \rightarrow A
\]

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

When are two types equal?

\[
\begin{align*}
\mu y.y & \equiv \mu x.z \\
\mu y.y & \equiv \mu x.x \\
\mu x.(\text{int} \times x) & \equiv \text{int} \times \mu x.(\text{int} \times x) \\
\mu x.x \rightarrow x & \equiv (\mu x.x \rightarrow x) \rightarrow (\mu x.x \rightarrow x)
\end{align*}
\]
\( \mu \)-Types

\[
A ::= \mathcal{T} \mid x \mid \mu x. A \mid A \times A \mid A \to A
\]

- \( \mu x. T \) binds \( x \) in \( T \), free and bound variables as expected
- \( \mu \)-types are closed and \textit{contractive} terms

A \textit{contractive} 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 \( x \).

\[
\mu y. y = \mu x. x
\]

\[
\mu x. (\text{int} \times x) \equiv \text{int} \times \mu x. (\text{int} \times x)
\]

\[
\mu x. x \to x \equiv (\mu x. x \to x) \to (\mu x. x \to x)
\]
μ-Types

\[ A ::= \; \mathcal{T} \; | \; x \; | \; \mu x.A \; | \; A \times A \; | \; A \to A \]

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

| \( \mu y.y \) | \( = \) | \( \mu x.z \) |
| \( \mu y.y \) | \( = \) | \( \mu x.x \) |
| \( \mu x.(\text{int} \times x) \) | \( = \) | \( \text{int} \times \mu x.(\text{int} \times x) \) |
| \( \mu x.x \to x \) | \( = \) | \( (\mu x.x \to x) \to (\mu x.x \to x) \) |
Type equivalence semantic approach

Σ: set of symbols with an arity ranked alphabet

A tree over a ranked alphabet Σ is a partial function $t : \mathbb{N}^*_+ \rightarrow \Sigma$ such that

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

Extensional equivalence (naïve)

- $f, g$ functions
- $f \equiv^\text{ext} g$ if $\text{dom}(f) = \text{dom}(g)$ and $\forall x \in \text{dom}(f). f(x) = g(x)$
Type equivalence  semantic approach

\[ \Sigma = \mathcal{T} \cup \{\times, \rightarrow\} \]

\[
\begin{align*}
treeof(c)(\varepsilon) &= c \quad \text{where } c \in \mathcal{T} \\
treeof(A_1 \rightarrow A_2)(\varepsilon) &= \rightarrow \\
treeof(A_1 \rightarrow A_2)(i\pi) &= treeof(A_i)(\pi) \\
&
\vdots \\
treeof(\mu x. A)(\pi) &= treeof(A\{x/\mu x. A\})(\pi)
\end{align*}
\]

**Lemma**

*For every \( \mu \)-type \( A \) the \( \text{treeof}(A) \) is defined.*  \( \text{Why?} \)

Let \( A \overset{\text{ext}}{=} B \) whenever \( \text{treeof}(A) \overset{\text{ext}}{=} \text{treeof}(B) \)
Type equivalence semantic approach

\[ \Sigma = \mathcal{T} \cup \{\times, \to\} \]

\[ \text{treeof}(c)(\varepsilon) = c \quad \text{where} \ c \in \mathcal{T} \]
\[ \text{treeof}(A_1 \to A_2)(\varepsilon) = \to \]
\[ \text{treeof}(A_1 \to A_2)(i\pi) = \text{treeof}(A_i)(\pi) \]
\[ \vdots \]
\[ \text{treeof}(\mu x.A)(\pi) = \text{treeof}(A\{x/\mu x.A\})(\pi) \]

Lemma

For every \( \mu \)-type \( A \) the \( \text{treeof}(A) \) is defined. Why?

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

How to decide \( \overset{\text{ext}}{=} \)?
Type equivalence  

Let $A^{\text{ext}} = B$ whenever $\text{treeof}(A)^{\text{ext}} = \text{treeof}(B)$

- Fix two $\mu$-types $A, B$
- to prove $A^{\text{ext}} = B$ we show

$$\forall \pi \in \{1, 2\}^*. \text{treeof}(A)(\pi) = \text{treeof}(B)(\pi)$$

**general issue**

**universal quantification**

not a problem if trees regular

**real question: axiomatisation**

Can we characterise $\equiv^{\text{ext}}$ syntactically?
▶ Try typing ∀ in ocaml
▶ Find useful sections in Chapter 21 Pierce book
▶ Implement treeof
▶ Work on the project