Typage
Master 2: Languages et Programmation

Delia Kesner, Giovanni Bernardi

2017 - 18

IRIF

université PARIS DIDEROT PARIS 7

Questions questions questions . . .

1. What is a complete lattice?

Questions questions questions ...

1. What is a complete lattice?
2. What is a complete partial order (CPO) ?

Questions questions questions . . .

1. What is a complete lattice?
2. What is a complete partial order (CPO) ?
3. What does the Knaster-Tarski theorem state ?

Questions questions questions . . .

1. What is a complete lattice?
2. What is a complete partial order (CPO) ?
3. What does the Knaster-Tarski theorem state ?
4. What does Kleene fixed point theorem state ?

Thus far . . .

Motivated by circularities, we discussed

Theory

1. Functions over partial orders F, $\langle P, \leq \rangle$
2. Fixed points $x = F(x)$

-	least	induction	Kleene fp theorem	μF
-	greatest	coinduction	Knaster-Tarski theorem	νF

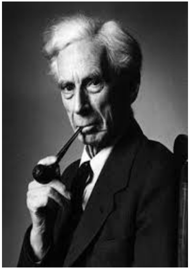
Applications

- Subtyping / equality for recursive types
- Equi-recursive type system how to type $\mathcal{Y}$

Plan

1. Historical remark
2. Recap material on coinductive subtyping
3. Proof theoretic approach
4. Examples examples examples

1908, Russell

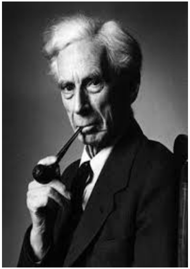


These fallacies [. . .] are to be avoided by what may be called the "vicious-circle principle;" *i.e.*, [. . .] whatever contains an apparent variable must be of a different type from the possible values of that variable [. . .] This is the guiding principle in what follows.

1968, Morris

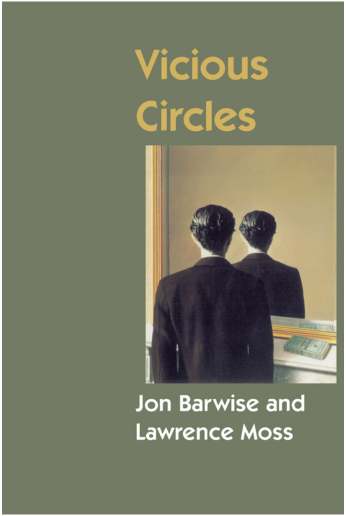
This construction is shown to be lacking [. . .] the type system makes the λ-calculus an uninteresting programming language; i.e. one without non-terminating computations.

1908, Russell



1968, Morris

1996



λ-calculus

typing rules from [Cardone and Coppo, 1991]

$$M, N ::= x \mid c \mid MN \mid \lambda x.M$$

An equi-recursive system

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{}{\Gamma, g : \mathit{typeof}(g) \vdash g : \mathit{typeof}(g)}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash M : A}\ A \approx B$$

Powerful type system, for instance we can type $\mathcal{Y}$

Type equivalence   syntactic approach

$$
\begin{aligned}
F &: \mathcal{P}(\mathrm{Types}^2_\mu) \to \mathcal{P}(\mathrm{Types}^2_\mu) \\
F(\mathcal{R}) &\triangleq \{\,(c,c) \mid c \in \mathcal{T}\,\} \\
&\cup \{\,(A_1 \times A_2, B_1 \times B_2) \mid \forall i \in \{1,2\}.\, A_i \mathrel{\mathcal{R}} B_i\,\} \\
&\cup \{\,(A_1 \to A_2, B_1 \to B_2) \mid B_1 \mathrel{\mathcal{R}} A_1, A_2 \mathrel{\mathcal{R}} B_2\,\} \\
&\cup \{\,(A, \mu x.B) \mid A \mathrel{\mathcal{R}} B\{x/\mu x.B\}\,\} \\
&\cup \{\,(\mu x.A, B) \mid A\{x/\mu x.A\} \mathrel{\mathcal{R}} B\,\}
\end{aligned}
$$

We have

- $\langle \mathcal{P}(\mathrm{Types}^2_\mu), \subseteq \rangle$ complete lattice, F monotone
- $\nu F = \bigcup \{\, \mathcal{R} \in \mathcal{P}(\mathrm{Types}^2_\mu) \mid \mathcal{R} \subseteq F(\mathcal{R})\,\}$ by Knaster-Tarski
- Let $\leq_{sbt} \triangleq \nu F$, and $\approx \triangleq \leq_{sbt} \cap \leq_{sbt}^{-1}$

Let's change perspective

inference rule

$$\boxed{\dfrac{\text{premise}_1 \qquad \ldots \qquad \text{premise}_n}{\text{conclusion}} \ \text{side condition}}$$

example

$$\boxed{\dfrac{\Gamma \vdash M : B}{\Gamma \vdash M : A} \ A \approx B}$$

Back to induction (just to warm up)

$$A, B \quad ::= \quad int \;\mid\; real \;\mid\; A \to A$$

ground subtyping relation

$$int \leq_g int \qquad real \leq_g real \qquad int \leq_g real$$

How to define subtyping $\leq_{sbt}$ on types $A, B, \ldots$?

Back to induction (just to warm up)

$$A, B \quad ::= \quad int \;\mid\; real \;\mid\; A \to A$$

ground subtyping relation

$$int \leq_g int \qquad real \leq_g real \qquad int \leq_g real$$

How to define subtyping $\leq_{sbt}$ on types $A, B, \ldots$? Induction

inference rules

$$\frac{}{c_1 \leq_{sbt} c_2}\; c_1 \leq_g c_2 \qquad\qquad \frac{B_1 \leq_{sbt} A_1 \quad A_2 \leq_{sbt} B_2}{A_1 \to A_2 \leq_{sbt} B_1 \to B_2}$$

$\leq_{sbt}$ contains all pairs (A, B) s.t.

towards set theory

- we can **derive** $A \leq_{sbt} B$,
- with a **finite derivation tree**

Back to induction (just to warm up)

A derivation tree of depth 2 (i.e. finite)

$$\dfrac{\dfrac{}{int \leq_{sbt} real} \qquad \dfrac{}{int \leq_{sbt} real}}{real \to int \leq_{sbt} int \to real}$$

inference rules

$$\dfrac{}{c_1 \leq_{sbt} c_2}\; c_1 \leq_g c_2 \qquad\qquad \dfrac{B_1 \leq_{sbt} A_1 \quad A_2 \leq_{sbt} B_2}{A_1 \to A_2 \leq_{sbt} B_1 \to B_2}$$

$\leq_{sbt}$ contains all pairs (A, B) s.t.

towards set theory

- we can **derive** $A \leq_{sbt} B$,
- with a **finite derivation tree**

Back to induction (just to warm up)

A derivation tree of depth 2 (i.e. finite)

$$\dfrac{\dfrac{}{\mathit{int} \leq_{sbt} \mathit{real}} \qquad \dfrac{}{\mathit{int} \leq_{sbt} \mathit{real}}}{\mathit{real} \to \mathit{int} \leq_{sbt} \mathit{int} \to \mathit{real}}$$

inference rules

$$\dfrac{}{c_1 \leq_{sbt} c_2}\ c_1 \leq_g c_2 \qquad\qquad \dfrac{B_1 \leq_{sbt} A_1 \quad A_2 \leq_{sbt} B_2}{A_1 \to A_2 \leq_{sbt} B_1 \to B_2}$$

How to express this using sets/functions ?

From rules to functions

inference rules

$$\frac{}{c_1 \leq_{sbt} c_2}\; c_1 \leq_g c_2 \qquad\qquad \frac{B_1 \leq_{sbt} A_1 \quad A_2 \leq_{sbt} B_2}{A_1 \to A_2 \leq_{sbt} B_1 \to B_2}$$

What do the rules *mean*?

From rules to functions

inference rules

$$\boxed{\frac{}{(c_1, c_2)}\; c_1 \leq_g c_2 \qquad\qquad \frac{(B_1, A_1) \quad (A_2, B_2)}{(A_1 \to A_2, B_1 \to B_2)}}$$

What do the rules *mean*?

To define a binary relation $\leq_{sbt}$

From rules to functions

inference rules

$$\frac{}{(c_1, c_2)}\ c_1 \leq_g c_2 \qquad\qquad \frac{(B_1, A_1) \quad (A_2, B_2)}{(A_1 \to A_2, B_1 \to B_2)}$$

What do the rules *mean*?

To define a binary relation $\leq_{sbt}$, the rules define

$$\begin{array}{lcl} F & : & \mathit{parts}(\mathrm{Types}^2) \to \mathit{parts}(\mathrm{Types}^2) \\ F(\mathcal{R}) & \triangleq & \{\, (c_1, c_2) \mid c_1 \leq_g c_2 \,\} \\ & & \cup \{\, (A_1 \to A_2, B_1 \to B_2) \mid B_1 \mathrel{\mathcal{R}} A_1, A_2 \mathrel{\mathcal{R}} B_2 \,\} \end{array}$$

From rules to functions

inference rules

$$\frac{}{(c_1, c_2)}\ c_1 \leq_g c_2 \qquad\qquad \frac{(B_1, A_1) \quad (A_2, B_2)}{(A_1 \to A_2, B_1 \to B_2)}$$

What do the rules *mean*?

To define a binary relation $\leq_{sbt}$, the rules define

$$\begin{array}{lcl} F & : & parts(\text{Types}^2) \to parts(\text{Types}^2) \\ F(\mathcal{R}) & \triangleq & \{(int, int), (real, real), (int, real)\} \\ & & \cup \{ (A_1 \to A_2, B_1 \to B_2) \mid B_1 \mathrel{\mathcal{R}} A_1, A_2 \mathrel{\mathcal{R}} B_2 \} \end{array}$$

From derivation trees to function application

$$F(\mathcal{R}) \triangleq \{(int, int), (real, real), (int, real)\} \cup \{ (A_1 \to A_2, B_1 \to B_2) \mid B_1 \mathrel{\mathcal{R}} A_1, A_2 \mathrel{\mathcal{R}} B_2 \}$$

Let's use F,

$F^0(\emptyset) = \emptyset$ *by convention*

$F^1(\emptyset) =$

$F^2(\emptyset) =$

From derivation trees to function application

$$
\begin{aligned}
F(\mathcal{R}) \;&\triangleq\; \{(int, int), (real, real), (int, real)\} \\
&\quad \cup \{\, (A_1 \to A_2, B_1 \to B_2) \mid B_1 \;\mathcal{R}\; A_1, A_2 \;\mathcal{R}\; B_2 \,\}
\end{aligned}
$$

Let's use F,

$$
\begin{aligned}
F^0(\emptyset) &= \emptyset && \textit{by convention} \\
F^1(\emptyset) &= \{(int, int), (real, real), (int, real)\} = \;\leq_g \\
F^2(\emptyset) &=
\end{aligned}
$$

From derivation trees to function application

$$
\begin{aligned}
F(\mathcal{R}) \;\triangleq\;& \{(int, int), (real, real), (int, real)\} \\
& \cup \{\, (A_1 \to A_2, B_1 \to B_2) \mid B_1 \,\mathcal{R}\, A_1, A_2 \,\mathcal{R}\, B_2 \,\}
\end{aligned}
$$

Let's use F,

$$
\begin{aligned}
F^0(\emptyset) &= \emptyset && \textit{by convention} \\
F^1(\emptyset) &= \{(int, int), (real, real), (int, real)\} = \;\leq_g \\
F^2(\emptyset) &= \{(real \to int, int \to real), (int \to int, int \to int), \ldots\} \\
&\quad \cup \leq_g \\
\vdots \quad & \quad \vdots
\end{aligned}
$$

From derivation trees to function application

A derivation tree of depth 2

$$\dfrac{\dfrac{}{(\mathit{int}, \mathit{real})} \qquad \dfrac{}{(\mathit{int}, \mathit{real})}}{(\mathit{real} \to \mathit{int}, \mathit{int} \to \mathit{real})}$$

$$\begin{aligned}
F^0(\emptyset) &= \emptyset \qquad \textit{by convention} \\
F^1(\emptyset) &= \{(\mathit{int}, \mathit{int}), (\mathit{real}, \mathit{real}), (\mathit{int}, \mathit{real})\} = \ \leq_g \\
F^2(\emptyset) &= \{(\mathit{real} \to \mathit{int}, \mathit{int} \to \mathit{real}), (\mathit{int} \to \mathit{int}, \mathit{int} \to \mathit{int}), \ldots\} \\
&\quad \cup \leq_g \\
\vdots & \qquad \vdots
\end{aligned}$$

From derivation trees to function application

Definition

$\leq_{sbt}$ contains all pairs (A, B) s.t.

- we can **derive** $A \leq_{sbt} B$,
- using a **finite derivation tree**

Lemma

A derivation tree $\dfrac{\vdots}{(A, B)}$ *has depth* $\color{red}{n}$ *iff* $(A, B) \in F^{n}(\emptyset)$. $\square$

but then ...

From derivation trees to function application

Definition

$\leq_{sbt}$ contains all pairs (A, B) s.t.

- we can **derive** $A \leq_{sbt} B$,
- using a **finite derivation tree**

Lemma

A derivation tree $\dfrac{\vdots}{(A, B)}$ *has depth* n *iff* $(A, B) \in F^n(\emptyset)$. $\square$

but then ...

Corollary

$\leq_{sbt} = \bigcup_{n=0} F^n(\emptyset)$, thus by Kleene fixed point theorem

$$\boxed{\leq_{sbt} = \mu F}$$

What about coinduction ?

$$A \;::=\; int \mid real \mid x \mid \mu x.A \mid A \to A$$

$$\begin{array}{lcl} F & : & parts(\mathrm{Types}_\mu^2) \to parts(\mathrm{Types}_\mu^2) \\ F(\mathcal{R}) & \triangleq & \leq_g \\ & & \cup \{ (A_1 \to A_2, B_1 \to B_2) \mid B_1 \mathrel{\mathcal{R}} A_1, A_2 \mathrel{\mathcal{R}} B_2 \} \\ & & \cup \{ (A, \mu x.B) \mid A \mathrel{\mathcal{R}} B\{x/\mu x.B\} \} \\ & & \cup \{ (\mu x.A, B) \mid A\{x/\mu x.A\} \mathrel{\mathcal{R}} B \} \end{array}$$

- $\langle parts(\mathrm{Types}_\mu^2), \subseteq \rangle$ complete lattice, F monotone
- νF exists by Knaster-Tarski
- Let $\leq_{sbt} \triangleq \nu F$, $\qquad \approx \triangleq \leq_{sbt} \cap \leq_{sbt}^{-1}$

What about coincidence ?

$$A \ ::= \ int \mid real \mid x \mid \mu x.A \mid A \to A$$

inference rules

$$\frac{}{c_1 \leq'_{sbt} c_2} \; c_1 \leq_g c_2 \qquad \frac{B_1 \leq'_{sbt} A_1 \quad A_2 \leq'_{sbt} B_2}{A_1 \to A_2 \leq'_{sbt} B_1 \to B_2}$$

$$\frac{A \leq'_{sbt} B\{x/\mu x.B\}}{A \leq'_{sbt} \mu x.B} \qquad \frac{A\{x/\mu x.A\} \leq'_{sbt} B}{\mu x.A \leq'_{sbt} B}$$

Definition (alternative)

$\leq'_{sbt}$ contains all pairs (A, B) s.t.

- we can derive $A \leq'_{sbt} B$
- with a finite **or a circular** derivation tree

What about coinduction ?

$$A \quad ::= \quad int \mid real \mid x \mid \mu x.A \mid A \to A$$

inference rules

$$\frac{}{c_1 \leq'_{sbt} c_2}\; c_1 \leq_g c_2 \qquad \frac{B_1 \leq'_{sbt} A_1 \quad A_2 \leq'_{sbt} B_2}{A_1 \to A_2 \leq'_{sbt} B_1 \to B_2}$$

$$\frac{A \leq'_{sbt} B\{x/\mu x.B\}}{A \leq'_{sbt} \mu x.B} \qquad \frac{A\{x/\mu x.A\} \leq'_{sbt} B}{\mu x.A \leq'_{sbt} B}$$

Definition (alternative)

$\leq'_{sbt}$ contains all pairs (A, B) s.t.

- we can derive $A \leq'_{sbt} B$
- with a finite **or a circular** derivation tree

What about coinduction ?

$$A \quad ::= \quad int \;\mid\; real \;\mid\; x \;\mid\; \mu x.A \;\mid\; A \to A$$

inference rules

$$\frac{}{c_1 \leq'_{sbt} c_2}\; c_1 \leq_g c_2 \qquad \frac{B_1 \leq'_{sbt} A_1 \quad A_2 \leq'_{sbt} B_2}{A_1 \to A_2 \leq'_{sbt} B_1 \to B_2}$$

$$\frac{A \leq'_{sbt} B\{x/\mu x.B\}}{A \leq'_{sbt} \mu x.B} \qquad \frac{A\{x/\mu x.A\} \leq'_{sbt} B}{\mu x.A \leq'_{sbt} B}$$

Definition (alternative)

$\leq'_{sbt}$ contains all pairs (A, B) s.t.

- we can derive $A \leq'_{sbt} B$
- with a finite **or a** **circular** derivation tree

A circular derivation tree

Let $A = \mu x.x \to int$, let's show that $A \leq'_{sbt} A \to int$.

$$
\cfrac{\cfrac{\cfrac{}{A \leq'_{sbt} A \to int}}{A \leq'_{sbt} A} \qquad \cfrac{}{int \leq'_{sbt} int}}{\cfrac{A \to int \leq'_{sbt} A \to int}{A \leq'_{sbt} A \to int}}
$$

A circular derivation tree

Let $A = \mu x.x \to int$, let's show that $A \leq'_{sbt} A \to int$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \leq'_{sbt} A \to int}}{A \leq'_{sbt} A} \qquad \dfrac{}{int \leq'_{sbt} int}}{A \to int \leq'_{sbt} A \to int}}{A \leq'_{sbt} A \to int}$$

What's the relation with νF ??

A circular derivation tree

Let $A = \mu x.x \to int$, let's show that $A \leq'_{sbt} A \to int$.

$$\dfrac{\dfrac{\dfrac{\overline{A \leq'_{sbt} A \to int}}{A \leq'_{sbt} A} \qquad \overline{int \leq'_{sbt} int}}{A \to int \leq'_{sbt} A \to int}}{A \leq'_{sbt} A \to int}$$

$\boxed{\text{What's the relation with } \nu F \text{ ??}}$

- $\mathcal{R} \triangleq \{(A, A \to int), (A \to int, A \to int), (A, A), (int, int)\}$
- $\mathcal{R} \subseteq F(\mathcal{R})$ post-fixed point
- $\mathcal{R} \subseteq \nu F = \leq_{sbt}$
- In fact we have $\leq_{sbt} = \leq'_{sbt}$

Summary

Induction

- least fixed points
- finite derivation trees

Coinduction

- greatest fixed points
- finite and circular derivation trees

Example

Subtyping relation

Other more abstract approaches exist category theory


That's all Folks!