# Logical Semantics and PER Models

CoMeta, Venezia, December 19, 2002

Ugo de'Liguoro
Dipartimento di Informatica
Università di Torino

# Subtyping polymorphism

Syntactic idea: A is a subtype of B if any term of type A can be safely used in a context expecting a term of type B.

$$(Sub)\quad \frac{\Gamma \mid\!- t : A \qquad \Sigma \mid\!- A <: B}{\Gamma \mid\!- t : B}$$

The judgment $\Sigma \mid\!- A <: B$ is deduced in a system axiomatizing the concept of "being a subtype" as "being a class of terms behaving in a compatible way with a (possibly) different class".

# Record and function subtyping

Type grammar

$$A ::= G \mid \{l_i{:}A_i\ ^{i\in I}\} \mid A \to A$$

Subtyping Rules

$$\frac{G <: G' \in \Sigma}{\Sigma \mid\!- G <: G'} \qquad \frac{\Sigma \mid\!- A_j <: B_j \quad \forall j \in J \subseteq I}{\Sigma \mid\!- \{l_i{:}A_i\ ^{i\in I}\} <: \{l_j{:}B_j\ ^{j\in J}\}}$$

$$\frac{\Sigma \mid\!- A' <: A \qquad \Sigma \mid\!- B <: B'}{\Sigma \mid\!- A \to B <: A' \to B'}$$

# Untyped interpretation and PERs

Polymorphic calculi are usually interpreted over some untyped λ-model $D$ via an erasing map: the type structure is then recovered via PERs (symmetric and transitive relations) over $D$:

erase: typed λ-calculus → type free λ-calculus

For any term $t$, $[[t]] = [[\text{erase}(t)]]^D$

For any type A, $R_A \in \text{PER}(D)$

For a judgment $t$:A, $[[t]] \in \text{Dom}(R_A) = \{d \mid d\ R_A\ d\}$

For an equation $t = s$ :A, $[[t]]\ R_A\ [[s]]$.

# Equations and subtyping

Equational theories of typed systems with subtyping satisfy:

$$(\mathit{SubEq}) \quad \frac{\Gamma \mid\!- t = s : \mathrm{A} \qquad \Sigma \mid\!- \mathrm{A} <: \mathrm{B}}{\Gamma \mid\!- t = s : \mathrm{B}}$$

The inverse implication does not hold (in general):

$$\{\mathrm{l}_1 : \mathit{int}, \mathrm{l}_2 : \mathit{int}\} <: \{\mathrm{l}_1 : \mathit{int}\}$$

$$\{\mathrm{l}_1 = 0, \mathrm{l}_2 = 1\} = \{\mathrm{l}_1 = 0, \mathrm{l}_2 = 0\} : \{\mathrm{l}_1 : \mathit{int}\}$$

but

$$\{\mathrm{l}_1 = 0, \mathrm{l}_2 = 1\} \neq \{\mathrm{l}_1 = 0, \mathrm{l}_2 = 0\} : \{\mathrm{l}_1 : \mathit{int}, \mathrm{l}_2 : \mathit{int}\}$$

# The inclusion semantics

If types are PERs, subtyping is interpreted as (set theoretic) inclusion of PERs:

$$A <: B \Rightarrow R_A \subseteq R_B$$

If R, S $\in$PER($D$) such that R $\subseteq$ S then Dom(R) $\subseteq$ Dom(S), and

$$\forall d \in D.\ [d]_R \subseteq [d]_S, \text{ where } [d]_R = \{e \in D \mid d\ R\ e\}$$

which gives a cohercion $[d]_R$ a $[d]_S$ from $D_R$ into $D_S$ where

$D_R = \{[d]_R \mid d \in \text{Dom}(R)\}$ (not injective, in general).

# The domain logic approach

Domains are isomorphic to certain filter spaces built out from "theories" $\langle L_A , \leq_A \rangle$ whose propositions denote compact opens:

**Stone Duality [and Abramsky application to domains]**

$$D_A \cong \mathrm{Filt}(L_A / \leq_A) \qquad \mathrm{K}(D_A) \cong L_A / \leq_A$$

Lattices, 2/3 SFP, Boolean algebras

Filters, Prime Filters, Ultrafilters

compact opens

Lindenbaum algebra

Completeness: $[[t]]^{\,L} = \{\varphi \in L_A \mid t \vdash \varphi\}$ (for $t : A$) and

$$t \vdash \varphi \;\Leftrightarrow\; t \models \varphi$$

# The basic domain equation

The untyped domain is the initial solution (in a given category) of the equation:

$$D = E + [L \to D] + [D \to D]$$

where records are finite functions over a set $L$ of labels.

This is achieved, in the category of algebraic lattices, by taking the space of filters $^{o}$ (ordered by inclusion) formed out of properties:

$$\sigma, \tau ::= \alpha \mid \omega \mid \sigma \to \tau \mid \sigma \wedge \tau \mid \langle l_i : \sigma_i \ {}^{i \in I} \rangle$$

which are preordered by a relation $\leq$ such that (e.g.):

$$\langle l_i : \sigma_i \ {}^{i \in I} \rangle \leq \langle l_i : \sigma_i \ {}^{i \in J} \rangle \ (J \subseteq I), \langle l_i : \sigma_i \ {}^{i \in I} \rangle = \Lambda_{i \in I} \langle l_i : \sigma_i \rangle$$

# Typed languages of properties

The properties are classified by languages $L_A$ indexed over types; then $\langle L_A , \leq_A \rangle$ is a propositional theory where

$$\leq_A = \leq \cap\, L_A \times L_A.$$

This allows for an interpretation of types as subtheories of the the propositional theory giving º:

$\omega \in L_A$ (for all A); $\alpha \in L_G$ (for G ground)

$\sigma, \tau \in L_A \Rightarrow \sigma \wedge \tau \in L_A$

$\sigma \in L_A , \tau \in \ ,_B \Rightarrow \sigma \rightarrow \tau \in L_{A \rightarrow B}$

$\sigma_i \in L_{Bi} \Rightarrow \langle l_i : \sigma_i \ \ ^{i \in I} \rangle \in L_{\{li:\, Bi \ \ i \in I\}}$

# The logical interpretation of types

Let $^{o}_{A}$ be the set of filters out of the theory $\langle L_A , \leq_A \rangle$: this is the A view of $^{o}$ which interprets A.

On the other hand let $R_A$ be the relation over $^{o}$ defined by:

$$f \; R_A \; g \Leftrightarrow f = g = \omega\uparrow \; \vee \; f \cap L_A = g \cap L_A \neq \omega\uparrow$$

**The Isomorphism Theorem**

$$\Phi: L_A \cong A/R_A$$

where $\Phi(p) = [\{\sigma \mid \exists\tau \in L_A . \tau \leq \sigma\}]_{RA}$

with inverse $\Phi^{-1}(f) = f \cap L_A$

# Sub-bases and CUPERs

The construction of $\mathrm{R}_A$ can be read topologically as follows:

**Definition**. A *sub-basis* $X \subseteq \mathrm{K}(D)$ is a set including $\bot$ and closed under finite sups of (compatible) elements. Then

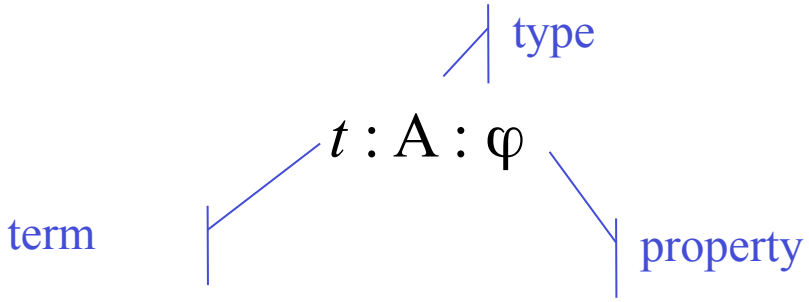
$$d \,\mathrm{R}_X\, e \Leftrightarrow d = e = \bot \vee \mathrm{K}(d) \cap X \setminus \{\bot\} = \mathrm{K}(e) \cap X \setminus \{\bot\} \neq \varnothing$$

**Theorem.** Any $\mathrm{R}_X$ is a *pointed complete* PER, namely includes the pair $\langle \bot, \bot \rangle$ and is closed under directed sups (of pairs). Moreover, if $D$ is equipped with a *notion of approximation*, (a family of continuous maps $(.)_{[n]} : D \to D$ such that $\sup_n (.)_{[n]} = \mathrm{Id}_D$), then $\mathrm{R}_X$ is *uniform* ($d \,\mathrm{R}_X\, e \Rightarrow \forall n.\ d_{[n]} \,\mathrm{R}_X\, e_{[n]}$) iff

$$d \in \mathrm{Dom}(\mathrm{R}_X) \wedge d_{[n]} \neq \bot \Rightarrow \mathrm{K}(d_{[n]}) \cap X \neq \{\bot\}.$$

# Program logic

This is a system deducing three places judgments:




$$(\rightarrow I)\quad \frac{\Delta, x : A : \varphi \ \vdash t : B : \psi}{\Delta \vdash \lambda x.t : A \rightarrow B : \varphi \rightarrow \psi}$$

$$(\wedge I)\quad \frac{\Delta \vdash t : A : \varphi \qquad \Delta \vdash t : A : \psi}{\Delta \vdash t : A : \varphi \wedge \psi}$$

$$(\leq)\quad \frac{\Delta \vdash t : A : \varphi \ \ \varphi \leq_A \psi}{\Delta \vdash t : A : \psi}$$

# Subtyping in program logic

$$(Sub)\quad \frac{\Delta \mid\!- t : A : \varphi \qquad A <: B \qquad \varphi \in \text{‚}_B}{\Delta \mid\!- t : B : \varphi}$$

It follows that, given

$$[[\Gamma \mid\!- t : A]]^L =^{\text{def}} \{\varphi \in L_A \mid \exists \Delta.\ \Gamma < \Delta \wedge \Delta \mid\!- t : A : \varphi\}$$

where $\Gamma < \Delta$ if $\forall\, y : B \in \Gamma\ \ \exists\, \varphi \in L_B\ .\ y : B : \varphi \in \Delta$, we have

$$A <: B \Rightarrow \forall \Gamma.\ [[\Gamma \mid\!- t : A]]^L \supseteq [[\Gamma \mid\!- t : B]]^L$$

$$\Gamma \mid\!- t = s : A \Rightarrow [[\Gamma \mid\!- t : A]]^L = [[\Gamma \mid\!- s : A]]^L$$

# The logical meaning of the isomorphism theorem

Let $t$ be any (closed) term and A be a type; let $[[.]]^F$ be the standard interpretation of the type free $\lambda$–calculus over F (which is a filter model), then it turns out

$$[[\text{erase}(t)]]^F \cap L_A = [[t : A]]^L$$

from which we can prove:

**Theorem**. $[[t : A]]^L = [[s : A]]^L \Leftrightarrow [[\text{erase}(t)]]^F \; R_A \; [[\text{erase}(s)]]^F$

This gives the logical meaning of subtyping which is

$$A <: B \Rightarrow \forall t. \; [[t : A]]^L \cap L_B = [[t : B]]^L$$

# More on sub-bases and PERs

Consider the following functors over the category of PER:

- $f \,\mathrm{R}\!\rightarrow \mathrm{S}\, g \Leftrightarrow \forall\, d, e.\, d\ \mathrm{R}\ e \Rightarrow f(d)\ \mathrm{S}\ g(e)$
- $d\ \{\mathrm{l}_i : \mathrm{R}_i\ ^{i\in \mathrm{I}}\}\ e \Leftrightarrow \forall i \in \mathrm{I}.\ d.\mathrm{l}_i\ \mathrm{R}_i\ e.\mathrm{l}_i$

These are (almost) mirrored by certain sub-bases:

- $X \rightarrow Y = \{\sup_{\ i\in \mathrm{I}} (d_i \Rightarrow e_i) \mid \text{it exists, } d_i \in X,\ e_i \in Y\}$
- $\{\mathrm{l}_i : X_i\ ^{i\in \mathrm{I}}\} = \{\sup_{\ i\in \mathrm{J}} (\mathrm{l}_i \Rightarrow d_i) \mid \mathrm{J} \subseteq \mathrm{I},\ \ d_i \in X_i\}$

**Proposition**

- $\mathrm{R}_{\{\mathrm{l}i\ :\ Xi\ i\in \mathrm{I}\}} = \{\mathrm{l}_i : \mathrm{R}_{Xi}\ ^{i\in \mathrm{I}}\}$
- $\mathrm{R}_X \rightarrow \mathrm{R}_Y \subseteq\ \mathrm{R}_{X \rightarrow Y}$

# Open problems and further work

Unfortunately $R_X \rightarrow R_Y \subseteq R_{X \rightarrow Y}$ is a proper inclusion, that is the logical interpretation, which is also based on PERs, is different from the standard interpretation of polymorphic calculi within this category.

Is there a logical interpretation which exactly matches the standard PER semantics of typed λ-calculi?

Is there a categorical reconstruction of the idea of logical interpretation of subtyping providing the guidelines to extend the program logic to more sophisticated type systems (system **F** at least)?