A Practical Framework for Curry-Style Languages

(Inspired by realizability semantics)

Rodolphe Lepigre

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Context: using realizability for programming languages

Last year's talk was about the PML language:

- A simple but powerful mechanism for program certification
- It is embedded in a (fairly standard) ML-style language
- Everything is backed by a (classical) realizability semantics
- Property: $v \in \phi^{\perp\perp} \Rightarrow v \in \phi$ for all ϕ closed under $(\equiv)$

Today's talk is about making Curry-style quantifiers practical:

- They are essential for PML (polymorphism, dependent types)
- But pose a practical issue due to non-syntax-directed rules
- Restricting quantifiers (prenex polymorphism) is not an option
- **Contribution:** a solution with subtyping inspired by semantics

In this talk we will stick to System F for simplicity

Quick reminder: Church-style versus Curry-style

Church-style System F:

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A . t : A \Rightarrow B}$$

$$\frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

$$\frac{\Gamma \vdash t : A \qquad X \notin \Gamma}{\Gamma \vdash \Lambda X. t : \forall X.A} \qquad \frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t\ B : A[X := B]}$$

Curry-style System F is obtained by removing the highlighted parts

A natural idea: using subtyping

We define a relation $(\subseteq)$ on types and use rule:

$$\frac{\Gamma \vdash t : A \qquad A \subseteq B}{\Gamma \vdash t : B}$$

This does help a bit already:

$$\frac{A \subseteq C}{\Gamma, x : A \vdash x : C} \qquad \frac{A \Rightarrow B \subseteq C \qquad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : C}$$

$$\frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

Ideally we would want quantifiers to be handled by subtyping

Containment system [Mitchell]

Is standard containment enough?

$$\frac{\{Y_1, \ldots, Y_m\} \cap FV(\forall X_1 \ldots \forall X_n.A) = \varnothing}{\forall X_1 \ldots \forall X_n.A \ \subseteq \ \forall Y_1 \ldots \forall Y_m.A[X_1 := B_1, \ldots, X_n := B_n]}$$

$$\frac{}{\forall X_1 \ldots \forall X_n.A \Rightarrow B \ \subseteq \ (\forall X_1 \ldots \forall X_n.A) \Rightarrow (\forall X_1 \ldots \forall X_n.B)}$$

$$\frac{A_2 \ \subseteq \ A_1 \qquad B_1 \ \subseteq \ B_2}{A_1 \Rightarrow B_1 \ \subseteq \ A_2 \Rightarrow B_2}$$

$$\frac{A \ \subseteq \ B \qquad B \ \subseteq \ C}{A \ \subseteq \ C} \qquad\qquad \frac{A \ \subseteq \ B}{\forall X.A \ \subseteq \ \forall X.B}$$

Can we derive the quantifier rules?

Yes we can derive the elimination rule:

$$\frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[X := B]} \quad \triangleq \quad \frac{\Gamma \vdash t : \forall X.A \qquad \dfrac{\varnothing \cap FV(\forall X.A) = \varnothing}{\forall X.A \subseteq A[X := B]}}{\Gamma \vdash t : A[X := B]}$$

No we cannot derive the introduction rule:

$$\frac{\Gamma \vdash t : A \qquad X \notin \Gamma}{\Gamma \vdash t : \forall X.A} \quad \triangleq \quad \frac{\Gamma \vdash t : A \qquad \dfrac{???}{A \subseteq \forall X.A}}{\Gamma \vdash t : \forall X.A}$$

Let us take a step back...

All we want is adequacy:

- If $\vdash t : A$ is derivable then $t \in [\![A]\!]$
- If $A \subseteq B$ then $[\![A]\!] \subseteq [\![B]\!]$

The subtyping part is not as fine-grained as it could be:

$$\frac{\vdash t : A \qquad A \subseteq B}{\vdash t : B} \quad \text{can be replaced by} \quad \frac{\vdash t : A \qquad \vdash t : A \subseteq B}{\vdash t : B}$$

Local subtyping is interpreted as an implication

Approach 1
(inspired by semantics)

Main idea of the approach

Based on a fine-grained semantic analysis we:

- Get rid of context and only work with closed terms
- To this aim terms are extended with choice operators
- The same kind of trick is used for quantifiers in types

Theorem (Adequacy)

- *If* $t : A$ *is derivable then* $[\![t]\!] \in [\![A]\!]$
- *If* $t : A \subseteq B$ *is derivable and* $[\![t]\!] \in [\![A]\!]$ *then* $[\![t]\!] \in [\![B]\!]$

Terms are interpreted using "pure terms"
(satisfying the intended semantic property)

Typing and subtyping rules

Syntax-directed typing rules:

$$\frac{\varepsilon_{x \in A}(t \notin B) : A \subseteq C}{\varepsilon_{x \in A}(t \notin B) : C} \qquad \frac{t : A \Rightarrow B \qquad u : A}{t\ u : B}$$

$$\frac{\lambda x.t : A \Rightarrow B \subseteq C \qquad t[x := \varepsilon_{x \in A}(t \notin B)] : B}{\lambda x.t : C}$$

Syntax-directed (local) subtyping rules:

$$\frac{}{t : A \subseteq A} \qquad \frac{t : A[X := C] \subseteq B}{t : \forall X.A \subseteq B} \qquad \frac{t : A \subseteq B[X := \varepsilon_X(t \notin B)]}{t : A \subseteq \forall X.B}$$

$$\frac{\varepsilon_{x \in A_2}(t\ x \notin B_2) : A_2 \subseteq A_1 \qquad t\ \varepsilon_{x \in A_2}(t\ x \notin B_2) : B_1 \subseteq B_2}{t : A_1 \Rightarrow B_1 \subseteq A_2 \Rightarrow B_2}$$

Interpretation of terms and types

We interpret terms using “pure terms“ (without choice operators)

$$[\![x]\!] = x \qquad [\![\lambda x.t]\!] = \lambda x.[\![t]\!] \qquad [\![t\ u]\!] = [\![t]\!]\ [\![u]\!]$$

$$[\![\varepsilon_{x\in A}(t^* \notin B)]\!] = \begin{cases} u \in [\![A]\!] \text{ s.t. } [\![t[x := u]]\!] \notin [\![B]\!] \text{ if it exists} \\ \text{any } t \in \mathcal{N}_0 \text{ otherwise} \end{cases}$$

We interpret types as (saturated) sets of normalizing terms

$$[\![\Phi]\!] = \Phi \qquad [\![A \Rightarrow B]\!] = [\![A]\!] \Rightarrow [\![B]\!] \qquad [\![\forall X.A]\!] = \cap_{\Phi\in\mathcal{F}}[\![A[X := \Phi]]\!]$$

$$[\![\varepsilon_X(t \notin A)]\!] = \begin{cases} \Phi \in \mathcal{F} \text{ such that } [\![t]\!] \notin [\![A[X := \Phi]]\!] \text{ if it exists} \\ \mathcal{N}_0 \text{ otherwise} \end{cases}$$

$$\Phi \Rightarrow \Psi \ = \ \{t \mid \forall u \in \Phi, t\, u \in \Psi\}$$

Let us look at one case of the adequacy lemma

$$\frac{\lambda x.t : A \Rightarrow B \subseteq C \qquad t[x := \varepsilon_{x \in A}(t \notin B)] : B}{\lambda x.t : C}$$

$$[\![\varepsilon_{x \in A}(t^* \notin B)]\!] = \begin{cases} u \in [\![A]\!] \text{ s.t. } [\![t[x := u]]\!] \notin [\![B]\!] \text{ if it exists} \\ \text{any } t \in \mathcal{N}_0 \text{ otherwise} \end{cases}$$

Approach 2

(using syntactic translations)

A more standard type system

Syntax-directed typing rules:

$$\frac{\Gamma, x : A \vdash x : A \subseteq C}{\Gamma, x : A \vdash x : C}$$

$$\frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

$$\frac{\Gamma \vdash \lambda x.t : A \Rightarrow B \subseteq C \qquad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : C}$$

Syntax-directed (local) subtyping rules:

$$\frac{}{\Gamma \vdash t : A \subseteq A}$$

$$\frac{\Gamma \vdash t : A[X := C] \subseteq B}{\Gamma \vdash t : \forall X.A \subseteq B}$$

$$\frac{\Gamma \vdash t : A \subseteq B \qquad X \notin \Gamma}{\Gamma \vdash t : A \subseteq \forall X.B}$$

$$\frac{\Gamma, x : A_2 \vdash x : A_2 \subseteq A_1 \qquad \Gamma, x : A_2 \vdash t\ x : B_1 \subseteq B_2}{\Gamma \vdash t : A_1 \Rightarrow B_1 \subseteq A_2 \Rightarrow B_2}$$

Elimination of subtyping: translation to System F+η

System F+η is obtained by adding the rule:

$$\frac{\Gamma \vdash \lambda x.t\, x : A \Rightarrow B \qquad x \notin t}{\Gamma \vdash t : A \Rightarrow B}$$

Theorem (Translation to F+η)

- *If $\Gamma \vdash t : A$ is derivable then it is also derivable in System F+η*
- *If $\Gamma \vdash t : A \subseteq B$ is derivable then $\Gamma \vdash t : B$ is derivable in System F+η given a derivation of $\Gamma \vdash t : A$*

Translation of subtyping leads to a "piece of proof":

If $\Gamma \vdash t : A \subseteq B$ is derivable then we get
$$\begin{array}{c}\Gamma \vdash t : A \\ \vdots\ \Pi \\ \Gamma \vdash t : B\end{array}$$

The most interesting case (arrow subtyping rule)

$$\frac{\Gamma, x : A_2 \vdash x : A_2 \subseteq A_1 \qquad \Gamma, x : A_2 \vdash t\ x : B_1 \subseteq B_2}{\Gamma \vdash t : A_1 \Rightarrow B_1 \subseteq A_2 \Rightarrow B_2}$$

$$\dfrac{\dfrac{\begin{array}{c} \dfrac{\dfrac{\Gamma \vdash t : A_1 \Rightarrow B_1}{\Gamma, x : A_2 \vdash t : A_1 \Rightarrow B_1}\ x\ \text{fresh} \qquad \begin{array}{c}\overline{\Gamma, x : A_2 \vdash x : A_2}\\ \vdots\ \Pi_1 \\ \Gamma, x : A_2 \vdash x : A_1\end{array}}{\Gamma, x : A_2 \vdash t\ x : B_1} \\ \vdots\ \Pi_2 \\ \Gamma, x : A_2 \vdash t\ x : B_2 \end{array}}{\Gamma \vdash \lambda x.t\ x : A_2 \Rightarrow B_2} \qquad x \notin t}{\Gamma \vdash t : A_2 \Rightarrow B_2}$$

Translation from System F+η

Given the subsumption rule the translation is immediate

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash t : A \subseteq B}{\Gamma \vdash t : B}$$

A couple of remarks:

- We conjecture that subsumption is admissible
- The rule is useful anyway for ascription (rule below)
- (Remember that type-checking remains undecidable here)

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash t : A \subseteq B}{\Gamma \vdash (t : A) : B}$$

Thanks! Questions?

@ https://lepigre.fr
✉ lepigre@mpi-sws.org