Algorithmic Subtyping for Higher Order Bounded Quantification

Dulma Rodriguez

Department of Computer Science
Ludwig-Maximilians-University Munich

14. Kolloquium Programmiersprachen und Grundlagen der Programmierung
Timmendorfer Strand, Germany
October 11, 2007

Introduction

- System: $F^{\omega}_{<:} = F^{\omega}$ + subtyping + bounded polymorphism.
- Models subtyping aspects of object oriented type systems.
- Metatheory: Algorithms for deciding typing and **subtyping**.
- Approach: Normalization of constructors and elimination of the transitivity rule.
- New results: Syntactical proofs of completeness, without model construction.

Example

- In Java Generics

```
class AbstractList<E> { public boolean add (E o) { } }
class Vector<E> extends AbstractList<E>
class LinkedList<E> extends AbstractList<E>
```

- Interpretation in $F^{\omega}_{<:}$

$$\begin{array}{lll} \mathsf{Vector}\!<\!E\!> & \rightsquigarrow & \lambda E.\mathsf{Vector}\,E \\ \mathsf{Vector}\!<\!E\!> \mathit{\ extends\ } \mathsf{AbstractList}\!<\!E\!> & \rightsquigarrow & \\ \multicolumn{3}{r}{(\lambda E.\mathsf{Vector}\,E) \leq (\lambda E.\ \mathsf{AbstractList}\,E) : * \to *} \end{array}$$

- We have then $\mathsf{Vector} \leq \mathsf{AbstractList}$, $\mathsf{LinkedList} \leq \mathsf{AbstractList}$ and

$$\mathsf{add} : \forall E \leq \top : *.\ \forall L \leq \mathsf{AbstractList} : * \to *.$$
$$E \to L\,E \to \mathit{boolean} \times L\,E$$

System $F^{\omega}_{<:}$

Type constructors

A, B, F, G	$::=$	X	type constructor variable
	$\mid$	$\lambda X F$	type-level abstraction
	$\mid$	$F\ G$	type-level application
	$\mid$	$A \to B$	function space
	$\mid$	$\forall X \leq G : \kappa.\ A$	bounded universal type
	$\mid$	$\top$	biggest type

Kinds

$$\kappa \quad ::= \quad * \mid \kappa_1 \to \kappa_2$$

Contexts and Judgements

Contexts

$$\Gamma \quad ::= \quad \diamond \mid X \leq G : \kappa$$

Notation $\Gamma, X : \kappa$ abbreviation for $\Gamma, X \leq \top_\kappa : \kappa$.

Judgements

$\Gamma \vdash F : \kappa$	Kinding
$\Gamma \vdash F = F' : \kappa$	Equality
$\Gamma \vdash F \leq F' : \kappa$	Subtyping

Equality

Example of type checking:

$$\mathsf{add}\,[E]\,[\lambda X.\mathit{LinkedList}\,X] : E \to \underbrace{(\lambda X.\mathsf{LinkedList}\,X)E}_{=_\beta \mathsf{LinkedList}\,E} \to \underbrace{(\lambda X.\mathsf{LinkedList}\,X)}_{=_\eta \mathsf{LinkedList}}E$$

Axioms

$$\frac{\Gamma, X:\kappa \vdash F : \kappa' \qquad \Gamma \vdash G : \kappa}{\Gamma \vdash (\lambda X F)\,G = [G/X]F : \kappa'} \ (\text{EQ-}\beta)$$

$$\frac{\Gamma \vdash F : \kappa \to \kappa'}{\Gamma \vdash (\lambda X.\,FX) = F : \kappa \to \kappa'}\ X \notin \mathsf{FV}(F)\ (\text{EQ-}\eta)$$

Motivation for subtyping

- $A \leq A'$ means where $t : A'$ needed $t : A$ can be used
- covariant = monotone
- contravariant = antitone
- The function space $(\rightarrow)$ is contravariant in the first argument and covariant in the second.
- Example

$$\frac{\mathsf{Int} \leq \mathsf{Real}}{\mathsf{Real} \rightarrow \mathsf{Int} \leq \mathsf{Int} \rightarrow \mathsf{Int} \leq \mathsf{Int} \rightarrow \mathsf{Real}}$$

Subtyping I

Rules for subtyping

$$\frac{\Gamma \vdash A : *}{\Gamma \vdash A \leq \top : *} \text{ (S-Top)} \qquad \frac{\Gamma \vdash A' \leq A : * \qquad \Gamma \vdash B \leq B' : *}{\Gamma \vdash A \to B \leq A' \to B' : *} \text{ (S-Arrow)}$$

$$\frac{\Gamma \vdash G = G' : \kappa \qquad \Gamma, X \leq G : \kappa \vdash A \leq A' : *}{\Gamma \vdash \forall X \leq G : \kappa . A \leq \forall X \leq G' : \kappa . A' : *} \text{ (S-All)}$$

$$\frac{\Gamma \vdash \qquad X \leq G : \kappa \in \Gamma}{\Gamma \vdash X \leq G : \kappa} \text{ (S-Var)} \qquad \frac{\Gamma, X \leq \top_{\kappa} : \kappa \vdash F \leq F' : \kappa'}{\Gamma \vdash \lambda X . F \leq \lambda X . F' : \kappa \to \kappa'} \text{ (S-Abs)}$$

$$\frac{\Gamma \vdash F \leq F' : \kappa \to \kappa' \qquad \Gamma \vdash H : \kappa}{\Gamma \vdash FH \leq F'H : \kappa'} \text{ (S-App)}$$

Subtyping II

Following rules make subtyping a partial order:

- Equality rule

$$\frac{\Gamma \vdash F = G : \kappa}{\Gamma \vdash F \leq G : \kappa} \text{ (S-Eq)}$$

- Transitivity rule

$$\frac{\Gamma \vdash F \leq G : \kappa \qquad \Gamma \vdash G \leq H : \kappa}{\Gamma \vdash F \leq H : \kappa} \text{ (S-Trans)}$$

Motivation for an algorithm

Problems of the declarative subtyping

- Declarative rules are not syntax directed.
- Declarative subtyping is not deterministic.
- Problematic rules: (S-Eq), (S-App) and (S-Trans).

Solutions

- $\beta\eta$-Normalizing the constructors:

$$\Gamma \vdash F = G : \kappa \Rightarrow \mathsf{nf}(F) \equiv \mathsf{nf}(G)$$

- Eliminating the transitivity rule.
  Idea: transitivity only for variables.

Normal forms

β-normal forms

$$
\begin{array}{rcll}
U, V, W & ::= & X\vec{V} & \text{neutral terms} \\
& | & V \to V' & \text{function space} \\
& | & \top & \text{biggest type} \\
& | & \forall X \leq U : \kappa.\, V & \text{bounded universal type} \\
& | & \lambda X V & \text{type-level abstraction}
\end{array}
$$

Algorithmic subtyping rules

$$\frac{}{\Gamma \vdash_{\mathsf{a}} V \leq \top : *} \text{ (SA-Top)} \qquad \frac{\Gamma \vdash_{\mathsf{a}} W \leq V : * \qquad \Gamma \vdash_{\mathsf{a}} V' \leq W' : *}{\Gamma \vdash_{\mathsf{a}} V \to V' \leq W \to W' : *} \text{ (SA-}\to\text{)}$$

$$\frac{\Gamma, X \leq V : \kappa \vdash_{\mathsf{a}} W \leq W' : *}{\Gamma \vdash_{\mathsf{a}} \forall X \leq V : \kappa . W \leq \forall X \leq V : \kappa . W' : *} \text{ (SA-}\forall\text{)}$$

$$\frac{\Gamma, (X_i \leq \top_{\kappa_i} : \kappa_i) \vdash_{\mathsf{a}} V \leq V' : *}{\Gamma \vdash_{\mathsf{a}} \lambda \vec{X} . V \leq \lambda \vec{X} . V' : \vec{\kappa} \to *} \text{ (SA-Abs)} \qquad \frac{}{\Gamma \vdash_{\mathsf{a}} X \vec{V} \leq X \vec{V} : *} \text{ (SA-Refl)}$$

$$\frac{(X \leq U : \kappa) \in \Gamma \qquad \Gamma \vdash_{\mathsf{a}} U @^{\vec{\kappa}} \vec{V} \leq W : *}{\Gamma \vdash_{\mathsf{a}} X \vec{V} \leq W : *} \text{ (SA-Bound)}$$

Proof obligations

Theorem (Soundness)

Let $\Gamma \vdash F, F' : \kappa$. *If* $\mathsf{nf}(\Gamma) \vdash_{\mathsf{a}} \mathsf{nf}(F) \leq \mathsf{nf}(F') : \kappa$, *then* $\Gamma \vdash F \leq F' : \kappa$.

The most difficult one is the completeness:

Theorem (Completeness)

If $\Gamma \vdash F \leq F' : \kappa$ *then* $\mathsf{nf}(\Gamma) \vdash_{\mathsf{a}} \mathsf{nf}(F) \leq \mathsf{nf}(F') : \kappa$.

Theorem (Termination)

If $\Gamma \vdash V \Uparrow \kappa$ *and* $\Gamma \vdash V' \Uparrow \kappa$ *then the query* $\Gamma \vdash_{\mathsf{a}} V \leq V' : \kappa$ *terminates.*

Lemmas needed for proving completeness

- Reflexivity: The algorithmic subtyping relation is reflexive.
- Transitivity: The algorithmic subtyping relation is transitive.
- Application: The application rule is admissible, for it we need:

Lemma (Substitution)

Let $\Gamma, X:\kappa, \Gamma' \vdash V, V' \Uparrow \kappa'$ *and* $\Gamma \vdash U \Uparrow \kappa$.
If $\Gamma, X:\kappa, \Gamma' \vdash_{\mathsf{a}} V \leq V' : \kappa'$ *then* $\Gamma, [U^{\kappa}/X]\Gamma' \vdash_{\mathsf{a}} [U^{\kappa}/X]V \leq [U^{\kappa}/X]V' : \kappa'$.

Conclusions

We provide:

- Algorithm for deciding subtyping of system $F^{\omega}_{<:}$.
- Syntactical proofs of its soundness, completeness and termination.

Further work

- Extending system $F^{\omega}_{<:}$ with polarities. (in progress)

$$\frac{\Gamma \vdash F : +\kappa \to \kappa' \qquad \Gamma \vdash G \leq G' : \kappa}{\Gamma \vdash F\ G \leq F\ G' : \kappa'} \ (\text{LEQ-ARG}+)$$

$$\frac{\Gamma \vdash F : -\kappa \to \kappa' \qquad -\Gamma \vdash G' \leq G : \kappa}{\Gamma \vdash F\ G \leq F\ G' : \kappa'} \ (\text{LEQ-ARG}-)$$