

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: $F_{<}^\omega = F^\omega + \text{subtyping} + \text{bounded polymorphism}$.
- 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 $F_{<}^\omega$:

$$\begin{aligned} \text{Vector}\langle E \rangle & \rightsquigarrow \lambda E. \text{Vector } E \\ \text{Vector}\langle E \rangle \text{ extends AbstractList}\langle E \rangle & \rightsquigarrow \\ & (\lambda E. \text{Vector } E) \leq (\lambda E. \text{AbstractList } E) : * \rightarrow * \end{aligned}$$

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

$$\text{add} : \forall E \leq \top : *. \forall L \leq \text{AbstractList} : * \rightarrow *$$

$$E \rightarrow L E \rightarrow \text{boolean} \times L E$$

System $F_{<}^\omega$ *Type constructors*

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

Kinds

$$\kappa ::= * \mid \kappa_1 \rightarrow \kappa_2$$

Contexts and Judgements

Contexts

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

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:

$$\text{add } [E] [\lambda X. \text{LinkedList } X] : E \rightarrow \underbrace{(\lambda X. \text{LinkedList } X) E}_{=_{\beta} \text{LinkedList } E} \rightarrow \underbrace{(\lambda X. \text{LinkedList } X) E}_{=_{\eta} \text{LinkedList}}$$

Axioms

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

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

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

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

Subtyping I

Rules for subtyping

$$\frac{\Gamma \vdash A : *}{\Gamma \vdash A \leq T : *} \quad (\text{S-Top})$$

$$\frac{\Gamma \vdash A' \leq A : * \quad \Gamma \vdash B \leq B' : *}{\Gamma \vdash A \rightarrow B \leq A' \rightarrow B' : *} \quad (\text{S-ARROW})$$

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

$$\frac{\Gamma \vdash X \leq G : \kappa \in \Gamma}{\Gamma \vdash X \leq G : \kappa} \quad (\text{S-VAR})$$

$$\frac{\Gamma, X \leq T_{\kappa} : \kappa \vdash F \leq F' : \kappa'}{\Gamma \vdash \lambda X. F \leq \lambda X. F' : \kappa \rightarrow \kappa'} \quad (\text{S-ABS})$$

$$\frac{\Gamma \vdash F \leq F' : \kappa \rightarrow \kappa' \quad \Gamma \vdash H : \kappa}{\Gamma \vdash FH \leq F'H : \kappa'} \quad (\text{S-APP})$$

Subtyping II

Following rules make subtyping a partial order:

- Equality rule

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

- Transitivity rule

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

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:

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

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

Normal forms

 β -normal forms

U, V, W	$::=$	$X\vec{V}$	neutral terms
		$V \rightarrow V'$	function space
		\top	biggest type
		$\forall X \leq U:\kappa. V$	bounded universal type
		λXV	type-level abstraction

Algorithmic subtyping rules

$$\frac{}{\Gamma \vdash_a V \leq T : *}$$

$$\text{(SA-TOP)} \quad \frac{\Gamma \vdash_a W \leq V : * \quad \Gamma \vdash_a V' \leq W' : *}{\Gamma \vdash_a V \rightarrow V' \leq W \rightarrow W' : *} \quad \text{(SA-}\rightarrow\text{)}$$

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

$$\frac{\Gamma, (X_i \leq T_{\kappa_i} : \kappa_i) \vdash_a V \leq V' : *}{\Gamma \vdash_a \lambda \vec{X}. V \leq \lambda \vec{X}. V' : \vec{\kappa} \rightarrow *}$$

$$\text{(SA-ABS)} \quad \frac{}{\Gamma \vdash_a X \vec{V} \leq X \vec{V} : *} \quad \text{(SA-REFL)}$$

$$\frac{(X \leq U : \kappa) \in \Gamma \quad \Gamma \vdash_a U @^{\vec{\kappa}} \vec{V} \leq W : *}{\Gamma \vdash_a X \vec{V} \leq W : *} \quad \text{(SA-BOUND)}$$

Proof obligations

Theorem (Soundness)

Let $\Gamma \vdash F, F' : \kappa$. If $\text{nf}(\Gamma) \vdash_a \text{nf}(F) \leq \text{nf}(F') : \kappa$, then $\Gamma \vdash F \leq F' : \kappa$.

The most difficult one is the completeness:

Theorem (Completeness)

If $\Gamma \vdash F \leq F' : \kappa$ then $\text{nf}(\Gamma) \vdash_a \text{nf}(F) \leq \text{nf}(F') : \kappa$.

Theorem (Termination)

If $\Gamma \vdash V \uparrow \kappa$ and $\Gamma \vdash V' \uparrow \kappa$ then the query $\Gamma \vdash_a V \leq V' : \kappa$ 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 $\Gamma, X:\kappa, \Gamma' \vdash_a V \leq V' : \kappa'$ then $\Gamma, [U^\kappa/X]\Gamma' \vdash_a [U^\kappa/X]V \leq [U^\kappa/X]V' : \kappa'$.

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)

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

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