Hint-based typing for STLC with Subtyping

Wojciech Kołowski

Types, contexts and judgements

Types:
$A, B ::= A \to B \mid A \times B \mid A + B \mid \top \mid \bot$

Typing contexts:
$\Gamma ::= \cdot \mid \Gamma, x : A$

Judgements:
$\Gamma \vdash e : A$

Subtyping relation

$$\frac{}{\bot <: A} \qquad \frac{}{A <: \top}$$

$$\frac{A' <: A \quad B <: B'}{A \to B <: A' \to B'}$$

$$\frac{A <: A' \quad B <: B'}{A \times B <: A' \times B'}$$

$$\frac{A <: A' \quad B <: B'}{A + B <: A' + B'}$$

The subtyping relation is the reflexive-transitive closure of the above rules.

Subtyping – properties

Subtyping is a partial order with top and bottom, congruent with type constructors.

- Reflexivity: $A <: A$
- Transitivity: $A <: B \implies B <: C \implies A <: C$
- Antisymmetry: $A <: B \implies B <: A \implies A = B$

Terms

Terms:

$$
\begin{aligned}
e ::= {} & \\
& x \mid \\
& \lambda x.\, e \mid e_1\ e_2 \mid \\
& (e_1, e_2) \mid \texttt{outl}\ e \mid \texttt{outr}\ e \mid \\
& \texttt{inl}\ e \mid \texttt{inr}\ e \mid \texttt{case}\ e\ \texttt{of}\ (e_1, e_2) \mid \\
& \texttt{unit} \mid \texttt{exfalso}\ e
\end{aligned}
$$

Note: `unit` is the term of the $\top$ type, whereas `exfalso` is the eliminator of the $\perp$ type.

Declarative typing – differences

$$\frac{\Gamma \vdash e : A \quad A <: B}{\Gamma \vdash e : B}\textsc{Sub}$$

Comments

The only difference between Extrinsic STLC with Subtyping and the original Extrinsic STLC is the addition of the subsumtpion rule. We invoke this rule every time we need subtyping to happen. However, this rule is stationary, and thus non-algorithmic, so we need to seek a better system.

Bidirectional typing – basics

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x \Rightarrow A}\textsc{Var}$$

$$\frac{\Gamma \vdash e \Leftarrow A}{\Gamma \vdash (e : A) \Rightarrow A}\textsc{Annot}$$

$$\frac{\Gamma \vdash e \Rightarrow A \quad A <: B}{\Gamma \vdash e \Leftarrow B}\textsc{Sub}$$

Metatheory

Hints

We no longer need holes – because of $\top$ and $\bot$, types suffice. We also don't need a separate order on hints, because subtyping suffices.

Least upper bound of two types

$$
\begin{aligned}
\bot \sqcup H &= H \\
H \sqcup \bot &= H \\
\top \sqcup H &= \top \\
H \sqcup \top &= \top \\
(H_1 \to H_2) \sqcup (H_1' \to H_2') &= (H_1 \sqcap H_1') \to (H_2 \sqcup H_2') \\
(H_1 \times H_2) \sqcup (H_1' \times H_2') &= (H_1 \sqcup H_1') \times (H_2 \sqcup H_2') \\
(H_1 + H_2) \sqcup (H_1' + H_2') &= (H_1 \sqcup H_1') + (H_2 \sqcup H_2')
\end{aligned}
$$

Greatest lower bound of two types

$$
\begin{aligned}
\bot \sqcap H &= \bot \\
H \sqcap \bot &= \bot \\
\top \sqcap H &= H \\
H \sqcap \top &= H \\
(H_1 \to H_2) \sqcap (H_1' \to H_2') &= (H_1 \sqcup H_1') \to (H_2 \sqcap H_2') \\
(H_1 \times H_2) \sqcap (H_1' \times H_2') &= (H_1 \sqcap H_1') \times (H_2 \sqcap H_2') \\
(H_1 + H_2) \sqcap (H_1' + H_2') &= (H_1 \sqcap H_1') + (H_2 \sqcap H_2')
\end{aligned}
$$

Hints for term constructors

$$
\begin{aligned}
\mathtt{hint}(\lambda x.\, e) &= \bot \to \top \\
\mathtt{hint}((e_1, e_2)) &= \top \times \top \\
\mathtt{hint}(\mathtt{inl}\ e) &= \top + \bot \\
\mathtt{hint}(\mathtt{inr}\ e) &= \bot + \top
\end{aligned}
$$

Terms and judgements

Terms:

$e ::=$

$x \mid (e : A) \mid$

$\lambda x.\, e \mid e_1 \; e_2 \mid$

$(e_1, e_2) \mid \texttt{outl}\ e \mid \texttt{outr}\ e \mid$

$\texttt{inl}\ e \mid \texttt{inr}\ e \mid \texttt{case}\ e\ \texttt{of}\ (e_1, e_2) \mid$

$\texttt{unit} \mid \texttt{exfalso}\ e$

Note: terms are the same as in Bidirectional STLC, i.e. annotations are types, not hints.

Judgements:

$\Gamma \vdash e \Leftarrow A \Rightarrow B$ – in context Γ, term e checks with type A and infers type B

Declarative typing – basics

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A}\textsc{Var}$$

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash (e : A) : A}\textsc{Annot}$$

$$\frac{\Gamma \vdash e : A \quad A <: B}{\Gamma \vdash e : B}\textsc{Sub}$$

Algorithmic typing – basic rules

$$\frac{(x : B) \in \Gamma \quad B <: A}{\Gamma \vdash x \Leftarrow A \Rightarrow B}\textsc{Var}$$

$$\frac{\Gamma \vdash e \Leftarrow B \Rightarrow C \quad B <: A}{\Gamma \vdash (e : B) \Leftarrow A \Rightarrow C}\textsc{Annot}$$

$$\frac{\Gamma \vdash e \Leftarrow \mathtt{hint}(e) \Rightarrow A \quad e\ \mathtt{constructor}}{\Gamma \vdash e \Leftarrow \top \Rightarrow A}\textsc{Hole}$$

Note that the rule HOLE can only be applied once, because $\mathtt{hint}(e)$ can never be $\top$. After applying HOLE, the only applicable rules are the type-directed ones.

Algorithmic typing – type-directed rules

$$\frac{\Gamma, x : A \vdash e \Leftarrow B \Rightarrow B'}{\Gamma \vdash \lambda x.\, e \Leftarrow A \to B \Rightarrow A \to B'}$$

$$\frac{\Gamma \vdash f \Leftarrow \bot \to B \Rightarrow A \to B' \quad \Gamma \vdash a \Leftarrow A \Rightarrow A'}{\Gamma \vdash f\ a \Leftarrow B \Rightarrow B'}$$

$$\frac{\Gamma \vdash a \Leftarrow A \Rightarrow A' \quad \Gamma \vdash b \Leftarrow B \Rightarrow B'}{\Gamma \vdash (a, b) \Leftarrow A \times B \Rightarrow A' \times B'}$$

$$\frac{\Gamma \vdash e \Leftarrow A \times \top \Rightarrow A' \times B}{\Gamma \vdash \mathtt{outl}\ e \Leftarrow A \Rightarrow A'} \qquad \frac{\Gamma \vdash e \Leftarrow \top \times B \Rightarrow A \times B'}{\Gamma \vdash \mathtt{outr}\ e \Leftarrow B \Rightarrow B'}$$

Algorithmic typing – type-directed rules

$$\frac{\Gamma \vdash e \Leftarrow A \Rightarrow A'}{\Gamma \vdash \texttt{inl}\ e \Leftarrow A + B \Rightarrow A' + B}$$

$$\frac{\Gamma \vdash e \Leftarrow B \Rightarrow B'}{\Gamma \vdash \texttt{inr}\ e \Leftarrow A + B \Rightarrow A + B'}$$

$$\frac{\Gamma \vdash e \Leftarrow \top + \top \Rightarrow A + B \qquad \begin{array}{c}\Gamma \vdash f \Leftarrow A \to C_1 \Rightarrow A' \to C_2 \\ \Gamma \vdash g \Leftarrow B \to C_2 \Rightarrow B' \to C_3\end{array}}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ (f, g) \Leftarrow C_1 \Rightarrow C_3}$$

$$\frac{}{\Gamma \vdash \texttt{unit} \Leftarrow \top \Rightarrow \top} \qquad \frac{\Gamma \vdash e \Leftarrow \bot \Rightarrow \bot}{\Gamma \vdash \texttt{exfalso}\ e \Leftarrow A \Rightarrow A}$$

Algorithmic rules – alternative rules

$$\frac{\Gamma \vdash a \Leftarrow \top \Rightarrow A \quad \Gamma \vdash f \Leftarrow A \to B \Rightarrow A' \to B'}{\Gamma \vdash f\ a \Leftarrow B \Rightarrow B'}\ \textsc{AltApp}$$

Metatheory

- If $\Gamma \vdash e \Leftarrow A \Rightarrow A'$, then $A' <: A$