Explorations of type systems

Semester Project Presentation

Samuel Grütter

EPFL

June 7, 2013

Explorations of Type Systems

The journey:

- A simple structurally typed language
- μ-recursive types
- EDOT: DOT as existential types
- Trying to generalize EDOT to DOT
- Infinite paths

Other topics:

- Types and mathematical sets
- Non-collapsed bounds in constructors
- Bugs detected

Existential types and DOT

DOT type $\top\{z \Rightarrow L : S..U \;,\; \overline{D}\}$

- $=$ the type inhabited by all objects o for which there exists a type T such that $S <: T$, $T <: U$, and o is of type $\top\left\{z \Rightarrow [T/z.L]\overline{D}\right\}$.
- Restriction: All type members of concrete (instantiable) types must have collapsed bounds

Intuition:

- L is a type that we don't know, but we know that $L : S..U$

Existential type $\exists\{L : S..U\}\{\overline{D}\}$

- $=$ the type inhabited by all objects o for which there exists a type T such that $S <: T$, $T <: U$, and o is of type $[T/L]\{\overline{D}\}$.

DOT/EDOT Example

$$\begin{aligned} T_1 = \top\{z \Rightarrow \\ &C : \bot..z.D \\ &D : \bot..z.C \\ &x : z.C \\ \} \end{aligned}$$

$$\begin{aligned} T_1 = \exists\{(X_C\ C \perp X_D)(X_D\ D \perp X_C)\}\{ \\ &x : X_C \\ \} \end{aligned}$$

$$\begin{aligned} T_2 = \top\{z \Rightarrow \\ &C : \bot..z.D \\ &D : \bot..z.C \\ &x : z.D \\ \} \end{aligned}$$

$$\begin{aligned} T_2 = \exists\{(X_C\ C \perp X_D)(X_D\ D \perp X_C)\}\{ \\ &x : X_D \\ \} \end{aligned}$$

Syntax differences DOT/EDOT

Constructor $\top\left\{z \Rightarrow \overline{D}\right\}\left\{\overline{d}\right\}$

- only $\top$ refinement
- all type declarations in $\overline{D}$ have collapsed bounds

Types T

- type variable X
- existential type $\exists\{\ \overline{(X\ L\ S\ U)}\ \}\{\overline{D}\}$
- $\top$ and $\bot$

Environment for subtype checking:

- $B ::= \{\overline{S <: U}\}$

Subtype rules (by example)

$$
\dfrac{
\dfrac{}{B_1 \vdash \bot <: X_C}
\quad
\dfrac{(X_C <: X_D) \in B_1}{B_1 \vdash X_C <: X_D}
\quad
\dfrac{}{B_1 \vdash \bot <: X_D}
\quad
\dfrac{(X_D <: X_C) \in B_1}{B_1 \vdash X_D <: X_C}
\quad
\dfrac{(X_C <: X_D) \in B_1}{B_1 \vdash X_C <: X_D}
}{
B_0 \vdash \exists\{(X_C \ C \perp X_D)(X_D \ D \perp X_C)\}\{x : X_C\} <: \exists\{(X_C \ C \perp X_D)(X_D \ D \perp X_C)\}\{x : X_D\}
}
$$

where

$$
\begin{aligned}
B_1 &= B_0 \cup \{\bot <: X_C \ , \ X_C <: X_D \ , \ \bot <: X_D \ , \ X_D <: X_C\} \\
B_0 &= \emptyset
\end{aligned}
$$

Subtype rules (by example)

$$\dfrac{\dfrac{}{B_1 \vdash \bot <: X_C} \quad \dfrac{(X_C <: X_D) \in B_1}{B_1 \vdash X_C <: X_D} \quad \dfrac{}{B_1 \vdash \bot <: X_D} \quad \dfrac{(X_D <: X_C) \in B_1}{B_1 \vdash X_D <: X_C} \quad \dfrac{(X_C <: X_D) \in B_1}{B_1 \vdash X_C <: X_D}}{B_0 \vdash \exists\{(X_C\ C \perp X_D)(X_D\ D \perp X_C)\}\{x : X_C\} <: \exists\{(X_C\ C \perp X_D)(X_D\ D \perp X_C)\}\{x : X_D\}}$$

where

$$\begin{aligned} B_1 &= B_0 \cup \{\bot <: X_C\ ,\ X_C <: X_D\ ,\ \bot <: X_D\ ,\ X_D <: X_C\} \\ B_0 &= \emptyset \end{aligned}$$

Subtype checking for two existential types:

1. "Synchronize" type variable names according to labels
2. Assume subtype relationships guaranteed in LHS
3. Prove subtype relationships needed in bounds of RHS
4. Prove subtype relationships needed for values and methods

Comparison to DOT

EDOT ("Trick 1"):

- use the subtype relationships guaranteed in LHS of $<:$ as assumptions to prove the subtype relationships in bounds of RHS

DOT:

- prove that LHS bounds are narrower than RHS bounds

$$\frac{\Gamma \vdash S' <: S\ ,\ T <: T'}{\Gamma \vdash (L : S..T) <: (L : S'..T')} \qquad (\textsc{TDecl-}{<:})$$

(DOT$<:$) $\Rightarrow$ (EDOT$<:$) because if we assume narrower bounds, we can prove the looser bounds (by transitivity)

Transitive closure

Trick 2: Transitive closure

$$\frac{(C <: E) \in \textbf{trcl}(\{C <: D,\ D <: E\})}{\{C <: D,\ D <: E\} \vdash\ C <: E}$$

Transitive closure

Trick 2: Transitive closure

$$\frac{(C <: E) \in \textbf{trcl}(\{C <: D,\ D <: E,\ E <: C\})}{\{C <: D,\ D <: E,\ E <: C\} \vdash\ C <: E}$$

DOT:

- C, E **wfe**?

Formal subtype rules

$$\frac{}{B \vdash T <: \top} \qquad (\mathrm{R}_1)$$

$$\frac{}{B \vdash \bot <: T} \qquad (\mathrm{R}_2)$$

$$\frac{}{B \vdash T <: T} \qquad (\mathrm{R}_3)$$

$$\frac{\mathbf{cond}(E_b, E_a) = \mathbf{false}}{B \nvdash E_b <: E_a} \qquad (\mathrm{R}_4)$$

$$\frac{\begin{array}{c} B \oplus \{\overline{(X_b\ L_b\ S_b\ U_b)} \lhd E_a\} \vdash \\ \mathbf{cond}(\exists\{\overline{(X_b\ L_b\ S_b\ U_b)}\}\{D_b\} \lhd E_a, E_a) \end{array}}{B \vdash \exists\{\overline{(X_b\ L_b\ S_b\ U_b)}\}\{D_b\} <: E_a} \qquad (\mathrm{R}_5)$$

$$\frac{\begin{array}{c} B \vdash U <: T \\ (S <: U) \in \mathbf{trcl}(\mathbf{rel}(B))\ ,\ S \neq U \end{array}}{B \vdash S <: T} \qquad (\mathrm{R}_6)$$

$$\frac{\begin{array}{c} B \vdash S <: U \\ (U <: T) \in \mathbf{trcl}(\mathbf{rel}(B))\ ,\ U \neq T \end{array}}{B \vdash S <: T} \qquad (\mathrm{R}_7)$$

EDOT

"Tricks":

1. use the subtype relationships guaranteed in LHS of $<:$ as assumptions to prove the subtype relationships in bounds of RHS
2. transitive closure of $<:$ relation over type variables

Win:

- no need for expansion
- can deal with cyclic subtype relation graphs

Trying to generalize EDOT to DOT

- We have an environment $\Gamma ::= \{\overline{x : T}\}; s$
- We need an environment $B ::= \{\overline{S <: U}\}$

Trying to generalize EDOT to DOT

- We have an environment $\Gamma ::= \{\overline{x : T}\}; s$
- We need an environment $B ::= \{\overline{S <: U}\}$

$$
\begin{aligned}
&\Gamma = \{(w : \top\{w \Rightarrow \\
&\qquad L : \bot..\top\{l \Rightarrow \\
&\qquad\qquad T : \bot..\top \\
&\qquad\qquad head : l.T \\
&\qquad\qquad tail : w.L\,\{w \Rightarrow T : \bot..l.T\} \\
&\qquad\} \\
&\}), \\
&(l : w.L)\}
\end{aligned}
$$

$$s = \emptyset$$

Trying to generalize EDOT to DOT

- We have an environment $\Gamma ::= \{\overline{x : T}\}; s$
- We need an environment $B ::= \{\overline{S <: U}\}$

$$
\begin{aligned}
\Gamma = \{(w &: \top\{w \Rightarrow \\
&L : \bot..\top\{l \Rightarrow \\
&\quad T : \bot..\top \\
&\quad \mathit{head} : l.T \\
&\quad \mathit{tail} : w.L\,\{w \Rightarrow T : \bot..l.T\} \\
&\} \\
\}&), \\
(l &: w.L)\}
\end{aligned}
$$

$$s = \emptyset$$

$$
\begin{aligned}
B = \{ \\
&\bot <: l.T,\ l.T <: \top, \\
&\bot <: l.\mathit{tail}.T,\ l.\mathit{tail}.T <: l.T, \\
&\bot <: l.\mathit{tail}.\mathit{tail}.T,\ l.\mathit{tail}.\mathit{tail}.T <: l.\mathit{tail}.T, \\
&\ldots \\
\}
\end{aligned}
$$

Infinite paths

$$
\begin{aligned}
T_1 = \top\{a \Rightarrow & \\
& L : \bot..\top\{z \Rightarrow \\
& \qquad M : z.f.M..z.f.M \\
& \qquad f : a.L \\
& \} \\
& m : \top \to a.L \\
\} &
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{val } a = \textbf{new } T_1\{ \\
&\qquad m(x) = \textbf{val } r = \textbf{new } a.L\{f = r\};\ r \\
&\};\ a
\end{aligned}
$$

- expand $r.M$ to $r.f.M$ to $r.f.f.M$ to ...
- infinite loop (even with fixed point approach)

Arbitrary depth limit

Arbitrary depth limit

- Scala: When calculating greatest lower bounds
- DOT: When calculating expansion

Explorations of Type Systems

The journey:

- A simple structurally typed language
- μ-recursive types
- EDOT: DOT as existential types
- Trying to generalize EDOT to DOT
- Infinite paths

Other topics:

- Types and mathematical sets
- Non-collapsed bounds in constructors
- Bugs detected

Questions?

Thank you ☺