Type Checking Cubex

Ross Tate
CS 4120

Contexts in Cubex
• $\Psi$ – Class/Interface context
  – Specifies methods, inheritance, and classiness
• $\Theta$ – Kind context
  – Indicates what type variables are in scope
• $\Delta$ – Function context
  – Indicates the type schemes of functions in scope
• $\Gamma$ – Type context
  – Indicates the types of variables in scope

Subtyping

Method Lookup

Types

Expressions
Principal Types

- Expression $e$ has principal type $\tau$ in some context if
  - (context)$\vdash e : \tau$ holds
  - for all $\tau'$, if (context)$\vdash e : \tau'$ holds then (context)$\vdash \tau < \tau'$
- In other words, while $e$ may have many types, $\tau$ is the most precise one.
- Java, C#, and Scala do not have principal types
  - OCaml and Haskell have principal type schemes

Cubex has principal types!!!
Appending Iterables

- What is the principal type of $e_1 + e_2$ in Cubex?
  - Given each $e_i$ has principal type $\tau_i$
    
    $\tau_i : \text{Int} \times \text{Double}$

    $e_1 + e_2 : \text{Int} \times \text{Double} < \tau_1 \sqcup \tau_2$

Cubex has principal instantiations!!!

Join – Most precise common supertype

- $\tau_1 \sqcup \tau_2$ denotes the join of $\tau_1$ and $\tau_2$ (if it exists)
  - $\tau_1 \sqsubseteq \tau_1 \sqcup \tau_2$ and $\tau_2 \sqsubseteq \tau_1 \sqcup \tau_2$
  - For any $\tau$, $\tau_1 \sqsubseteq \tau$ and $\tau_2 \sqsubseteq \tau$ implies $\tau_1 \sqcup \tau_2 \sqsubseteq \tau$

  $\text{class} \ Foo \ 	ext{extends} \ A B \ B L D$

  $\text{class} \ Bar \ 	ext{extends} \ D C L A$

  $\text{fun} \ \text{L} \ Foo \ = \ A \& B \ D$

Cubex has joins!!!

Covariant Arrays (not in Cubex)

Contravariance (not in Cubex)