CS 444 - Compiler Construction **Winter 2020**

Lecture 13: February 24th , 2020

Lecturer: Ondřej Lhoták *Notes By: Harsh Mistry*

13.1 Context-sensitive analysis continued

- In Java : A2 and A3
  1. Build global environment
  2. Resolve type names
  3. Build/check class hierarchy (methods/files)
  4. Disambiguate ambiguous namespace
     - In Java you can't simply determine the namespace based on location of usage
  5. Resolve "expressions" (Variables, static fields)
  6. Type checking
  7. Resolve methods instance (non-static) fields
- (6) Type Checking Continued: Type system for Joos
  - C, L, $\sigma \vdash$ E : τ
    In class C with local environment L, if method returns σ, expression E has type τ
  - C, L, $\sigma \vdash$ S
    In class C, ..., statement S is statically type correct
  - Literals
    - $$\frac{}{C, L, \sigma \vdash 43 : int}$$
    - $$\frac{}{C, L, \sigma \vdash true : boolean}$$
    - $$\frac{}{C, L, \sigma \vdash "ABC" : string}$$
    - $$\frac{}{C, L, \sigma \vdash' a' : char}$$
    - $$\frac{}{C, L, \sigma \vdash null : null}$$
    - $$\frac{L(x) = \gamma}{C, L, \sigma \vdash x : \tau}$$
    - $$\frac{}{C, L, \sigma \vdash this : C}$$

- Operators (Refer to JLS 15)
  - $$\frac{C, L, \sigma \vdash E : boolean}{C, L, \sigma \vdash !E : boolean}$$
  - $$\frac{C, L, \sigma \vdash E_1 : \tau_1 \quad C, L, \sigma \vdash E_2 : \tau_2 \quad num(\tau_1) \quad num(\tau_2)}{C, L, \sigma \vdash E_1 + E_2 : int}$$
  - $$\frac{C, L, \sigma \vdash E_1 : string \quad C, L, \sigma \vdash E_2 : \tau_2 \quad \tau_2 \neq void}{C, L, \sigma \vdash E_1 + E_2 : string}$$
  - $$\frac{C, L, \sigma \vdash E : \tau_1 \quad C, L, \sigma \vdash E_2 : string \quad \tau_1 \neq void}{C, L, \sigma \vdash E_1 + E_2 : string}$$
- Assignment
  - $$\frac{C, L, \sigma \vdash E : \tau_2 \quad L(x) = \tau_2 \quad \tau_1 := \tau_2}{C, L, \sigma \vdash x = E : \tau_1}$$
- Assign-ability (Refer to JLS 5)
  - $$\frac{D \leq C}{C := D}$$
  - $$\frac{}{\sigma := \sigma}$$
  - $$\frac{}{int := short}$$
  - $$\frac{}{int := char}$$
  - $$\frac{}{C := null}$$
  - $$\frac{}{short := byte}$$
  - $$\frac{\sigma := \tau \quad \tau := P}{\sigma := P}$$
- Statements
  - $$\frac{C, L, \sigma \vdash E : \tau}{C, L, \sigma \vdash E}$$
  - $$\frac{\forall i : C, L, \sigma \vdash S_i}{C, L, \sigma \vdash \{S_1, \ldots, S_n\}}$$
  - $$\frac{C, L[x \to \tau], \sigma \vdash S}{C, L, \sigma \vdash \{\tau x; S\}}$$
  - $$\frac{C, L, \sigma \vdash E : boolean \quad C, L, \sigma \vdash S}{C, L, \sigma \vdash if(E)S}$$
- Fields

    - $$\frac{static\ \tau\ f \in contain(D)}{C, L, \sigma \vdash D.f : \tau}$$
    - $$\frac{C, L, \sigma \vdash E : D \quad \tau f \in contain(D)}{C, L, \sigma \vdash E.f : \tau}$$
    - $$\frac{static\ \tau_1\ f \in contain(D) \quad C, L, \sigma \vdash E : \tau_2 \quad \tau_1 := \tau_2}{C, L, \sigma \vdash D.f = E}$$
    - $$\frac{C, L, \sigma \vdash E : D \quad \tau_1 f \in contain(D) \quad C, L, \sigma \vdash E_2 : \tau_2 \quad \tau_1 := \tau_2}{C, L, \sigma \vdash E_1.f = E_2 : \tau_1}$$
  - Comparisons
    - $$\frac{C, L, \sigma \vdash E_1 : \tau_1 \quad C, L, \sigma \vdash E_2 : \tau_2 \quad num(\tau_1) \quad num(\tau_2)}{C, L, \sigma \vdash E_1 == E_2 : boolean}$$
      - Works for other comparison operators (i.e $! =, \leq$, etc)
    - $$\frac{C, L, \sigma \vdash E_1 : \tau_1 \quad C, L, \sigma \vdash E_2 : \tau_2 \quad \tau_1 := \tau_2 \vee \tau_2 := \tau_2}{C, L, \sigma \vdash E_1 == E_2 : boolean}$$
      - Can be used for $! =$, but not for less then operators
  - Casts
    - $$\frac{C, L, \sigma \vdash E_1 : \tau_1 \quad num(\tau_1) \quad num(\tau_2)}{C, L, \sigma \vdash (\tau_2)E : \tau_2}$$
    - $$\frac{C, L, \sigma \vdash E_1 : \tau_1 \quad \tau_1 := \tau_2 \vee \tau_2 := \tau_1}{C, L, \sigma \vdash (\tau_2)E : \tau_2}$$