

# Lwnn Typing Rules

CS 260, Fall 2013

## 1 Typing Rules

### 1.1 Helpers

We use a `ClassTable` object to embody the necessary global set of classes. Our `ClassTable` is adapted from the one used in the `FetherweightJava` paper and is a map from class names to class declarations. Providing a mechanism for looking up field and method types for a given class. A program then is a pair  $(CT, e)$ , we also make the assumption that the `ClassTable` is fixed.

$$\begin{aligned} lookup &: ClassTable \times ClassT \rightarrow Class \\ lookup(ct, C) &= ct[C] \end{aligned}$$

$$\begin{aligned} method &: ClassT \times Var \rightarrow Type \\ method(C, m) &= classTable.lookup(C).method(m) \end{aligned}$$

$$\begin{aligned} field &: ClassT \times Var \rightarrow Type \\ field(C, f) &= classTable.lookup(C).field(f) \end{aligned}$$

### 1.2 Subtyping

$$A <: A \quad (\text{REFLEXIVITY})$$

$$\frac{C <: B \quad B <: A}{C <: A} \quad (\text{TRANSITIVITY})$$

$$\frac{class\ C\ extends\ D\ \{\dots\}}{C <: D} \quad (\text{INHERITANCE})$$

### 1.3 Classes

$$\frac{\Gamma \vdash}{todo} \quad (\text{T-CLASS})$$

## 1.4 Methods

$$\frac{\Gamma \vdash}{\text{todo}} \quad (\text{T-METHOD})$$

## 1.5 Statements

$$\frac{\Gamma \vdash x : \tau \quad \Gamma \vdash e : \tau}{x := e : \text{null}} \quad (\text{T-ASSIGN})$$

$$\frac{\Gamma \vdash e_1 : C \quad \text{field}(C, x) = \tau_f \quad \Gamma \vdash e_2 : \tau_f}{e_1.x := e_2 : \text{null}} \quad (\text{T-UPDATE})$$

$$\frac{\Gamma \vdash e : C \quad \text{method}(C, m) = \vec{\tau}_d \rightarrow \tau_r \quad \Gamma \vdash \vec{e}_i : \vec{\tau}_c \quad \vec{\tau}_c <: \vec{\tau}_d \quad \Gamma \vdash x : \tau_r}{\Gamma \vdash x := e.m(\vec{e}_i) : \text{null}} \quad (\text{T-METHOD-INVOCATION})$$

$$\frac{\Gamma \vdash x : D \quad \Gamma \vdash e : C \quad \text{method}(C, "C") = \vec{\tau}_i \rightarrow C \quad \vec{e}_i : \vec{\tau}_i \quad C <: D \quad \text{new } C(\vec{e}_i) : D}{\Gamma \vdash x := \text{new } C(\vec{e}_i) : \text{null}} \quad (\text{T-NEW})$$

$$\frac{\Gamma \vdash e : \text{bool}}{\text{if } (e) \vec{s}_1 \text{ else } \vec{s}_2 : \text{null}} \quad (\text{T-IF})$$

$$\frac{\Gamma \vdash e : \text{bool}}{\text{while } (e) \vec{s} : \text{null}} \quad (\text{T-WHILE})$$

## 1.6 Expressions

$$\Gamma \vdash i : \text{int} \quad (\text{T-INT})$$

$$\Gamma \vdash \text{str} : \text{string} \quad (\text{T-STRING})$$

$$\Gamma \vdash \text{true} : \text{bool} \quad (\text{T-TRUE})$$

$$\Gamma \vdash \text{false} : \text{bool} \quad (\text{T-FALSE})$$

$$\Gamma \vdash \text{null} : \text{null} \quad (\text{T-NULLS})$$

$$\Gamma \vdash x : \Gamma(x) \quad (\text{T-VAR})$$

$$\frac{\Gamma \vdash e : C \quad \text{field}(C, f) = \tau}{\Gamma \vdash e.f : \tau} \quad (\text{T-ACCESS})$$

$$\frac{\oplus \in \{+, -, *, \div, <, \leq\} \quad \Gamma \vdash e_1 : \mathbf{int} \quad \Gamma \vdash e_2 : \mathbf{int}}{e_1 \oplus e_2 : \mathbf{int}} \text{ (T-NUMOPS)}$$

$$\frac{\oplus \in \{\wedge, \vee\} \quad \Gamma \vdash e_1 : \mathbf{bool} \quad \Gamma \vdash e_2 : \mathbf{bool}}{e_1 \oplus e_2 : \mathbf{bool}} \text{ (T-BOOLOPS)}$$

$$\frac{\oplus \in \{=, \neq\} \quad \Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{e_1 \oplus e_2 : \mathbf{bool}} \text{ (T-EQOPS)}$$