Types

A4M36TPJ, 2013/2014

Motivation

“Well-typed programs never go wrong”

What does “go wrong” mean?

- Program crash
- Execution error
- Divergence
- Unknown method call.....

The Purpose of Typing

- Well-typed programs are more likely to actually do what they are supposed to do.

Two Kinds of Typing

- Static typing.
- Dynamic checking.

Static Typing

- Formally specified by a Type System.

Simple Type System Specification

- Judgements.
- Type rules.
- Environment.

Judgements General Form

$$\Gamma \vdash \mathfrak{J}$$

Judgements Examples

$$\Gamma \vdash M : A$$

$$\emptyset \vdash true : Boolean$$

$$\emptyset \vdash 1 : Nat$$

Type Rules

- Type rules assert the validity of certain judgments on the basis of other judgments that are already known to be valid.
- The process gets off the ground by some intrinsically valid judgment (usually: empty environment is well formed).

Type Rules (II)

$$\frac{\Gamma_1 \vdash \mathfrak{J}_1 \cdots \Gamma_n \vdash \mathfrak{J}_n}{\Gamma \vdash \mathfrak{J}}$$

Example of EXPR Language

$$
\begin{aligned}
Expr ::= {} & Num \mid \\
& Bool \mid \\
& \triangle Expr \mid \\
& Expr \odot Expr \mid \\
& Expr \leq Expr \mid \\
& Expr \texttt{ nand } Expr \mid \\
& \texttt{if } Expr \texttt{ then } Expr \texttt{ else } Expr,
\end{aligned}
$$

where Num is a predefined set of integer numbers (a.k.a. Z) and $Bool$ is a predefined set of boolean values.

Rules for EXPR

Convention: $e, e', e'', \ldots \in Expr$, $b, b' \in Bool$ and $n, n' \in Num$.

$$\frac{}{n : Number} \quad \text{(Val N)}$$

$$\frac{}{b : Boolean} \quad \text{(Val B)}$$

Rules (II) for EXPR

$$\frac{e : Number}{\triangle e : Number} \quad \text{(Val triangle)}$$

$$\frac{e : Number \quad e' : Number}{e \odot e' : Number} \quad \text{(Val circle)}$$

$$\frac{e : Number \quad e' : Number}{e \leq e' : Boolean} \quad \text{(Val leq)}$$

Rules(III) for EXPR

$$\frac{e : Boolean \quad e' : Boolean}{e \ \texttt{nand} \ e' : Boolean} \text{ (Val nand)}$$

$$\frac{e : Boolean \quad e' : Number \quad e'' : Number}{\texttt{if} \ e \ \texttt{then} \ e' \ \texttt{else} \ e'' : Number} \text{ (Val ifNum)}$$

$$\frac{e : Boolean \quad e' : Boolean \quad e'' : Boolean}{\texttt{if} \ e \ \texttt{then} \ e' \ \texttt{else} \ e'' : Boolean} \text{ (Val ifBool)}$$

Judgement for EXPR

- Do we need an Environment in Type System of EXPR?
- Do we have any expressions that contain variables in EXPR?

Derivation in EXPR

$$\text{triangle)}\ \dfrac{\text{(Val circle)}\ \dfrac{\text{(Val N)}\ \dfrac{}{5 : Number} \quad \text{(Val triangle)}\ \dfrac{\text{(Val N)}\ \dfrac{}{3 : Number}}{\triangle(3) : Number}}{5 \odot (\triangle(3)) : Number}}{\triangle(5 \odot (\triangle(3))) : Number}$$