Hype for Types

Lecture 1: Notation, notation, notation

Logistics

- This is a Pass/Fail class
- Attendance is mandatory (StuCo's rule, not ours): if you miss more than 2 classes you fail
- There will be homeworks. They have very short mandatory portions and very long optional portions. They are graded for completion, not correctness. Homework is 50% of grade.
- There is a midterm and a final, in class. 25% each of grade.
- If you come to class and do the work, you should expect to pass :)

Course Website: hypefortypes.github.io

Syllabus/Schedule

"This course aims to go over fun and weird results in type theory that you might otherwise have to read complicated academic papers to understand, as well as to provide a foundation to help understand these fun results."

Basic Layout:

- Some classes will be more foundation
- Some classes will be more fun and weird
- "Weird" and "fun" are dependent on your level of experience and your interests. If you've seen a lot of this stuff already, it might not be too weird to you.

Question: What are types?

Judgments

A judgment is an assertion about a property or relationship.

- Examples:

$A\ \mathsf{true}$	(proposition A is true)
$e \hookrightarrow v$	(expression e evaluates to value v)
$e\ \mathsf{val}$	(expression e is a value)
$e : \tau$	(expression e has type τ)

Types are judgments about expressions

$e : \texttt{int}$ (if e evaluates to a value, that value will be an integer)

$e : \texttt{int} \rightarrow \texttt{int}$ (if e evaluates to a value, that value will be a function[1] which can only be applied to integers and only return integers)

$e : \forall \alpha . \alpha$ (For all types t, e has type t)

1 functions are values

Inference Rules

An inference rule consists of a set of judgments above the line, which are known as premises, and a single judgment below the line, known as the conclusion.

If an inference rule does not have any premises, it's an **axiom.**

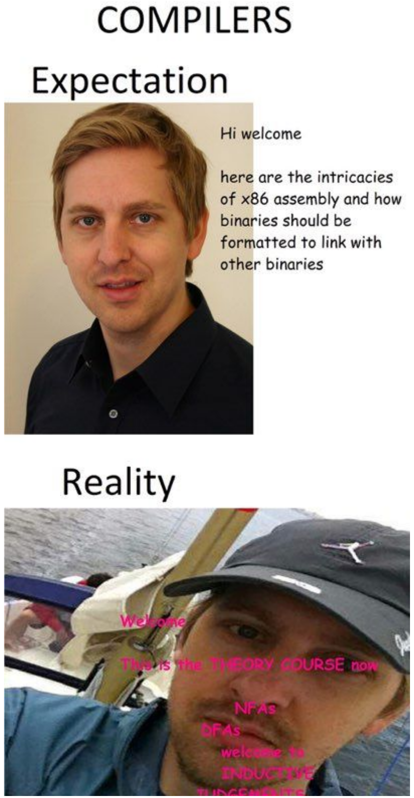
$$\frac{e_1 : \texttt{int} \qquad e_2 : \texttt{int}}{e_1 + e_2 : \texttt{int}} \qquad\qquad \frac{e_1\ \mathsf{val} \qquad e_2\ \mathsf{val}}{(e_1, e_2)\ \mathsf{val}}$$

This most important inference rule

$$\frac{}{\lambda\,(x : \tau)\,e\ \mathsf{val}}$$

Functions are values :)

Inductive Definitions



Inductive Definitions

An inductive definition is a set of inference rules that completely describes a judgment.

This is how we define what expressions have a particular type.

A simple language

$$
\begin{array}{lcll}
\textsf{exp} & ::= & n & \text{(decimal number)} \\
 & & \mid x & \text{(variable)} \\
 & & \mid e_1 + e_2 & \text{(addition)} \\
 & & \mid \lambda\,(x : \tau)\,e & \text{(function)} \\
 & & \mid e_1(e_2) & \text{(application)} \\
 & & & \\
\textsf{type} & ::= & \texttt{int} & \text{(integer)} \\
 & & \mid t_1 \to t_2 & \text{(function from } t_1 \text{ to } t_2\text{)}
\end{array}
$$

Types? Types? Types?

$$\frac{}{\bar{n} : \texttt{int}} \qquad \frac{e_1 : t_1 \to t_2 \quad e_2 : t_1}{e_1(e_2) : t_2} \qquad \frac{e_1 : \texttt{int} \quad e_2 : \texttt{int}}{e_1 + e_2 : \texttt{int}}$$

But what about $\lambda\,(x : \tau)\,e$?

Premise: "Assuming $x : \tau_1$ then $e : \tau_2$"
Conclusion: $\lambda\,(x : \tau_1)\,e : \tau_1 \to \tau_2$

How to write premise?

Context is for Kings

We keep track of a **context** which tells us the type of all variables in scope.

We can use all the types in this context when checking the type of an expression.

A context $\boldsymbol{\Gamma}$ is either empty: $\cdot$

or some set of variables with types: $x : \tau_1, y : \tau_2$, etc.

To write our premise: $\Gamma, x : \tau_1 \vdash e : \tau_2$

Final Rules

$$\frac{}{\Gamma \vdash \bar{n} : \texttt{int}}$$

$$\frac{\Gamma \vdash e_1 : \texttt{int} \quad \Gamma \vdash e_2 : \texttt{int}}{\Gamma \vdash e_1 + e_2 : \texttt{int}}$$

$$\frac{\Gamma \vdash e_1 : t_1 \to t_2 \quad \Gamma \vdash e_2 : t_1}{\Gamma \vdash e_1(e_2) : t_2}$$

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\lambda(x : \tau)e : \tau_1 \to \tau_2}$$