Lecture #17: Typed λ-calculi

CS 6371: Advanced Programming Languages

March 12, 2020

There are many typed λ-calculi, of which we will study two. The first is known as the *simply-typed λ-calculus* or $\lambda^{\rightarrow}$. Here are some of the operations commonly included in $\lambda^{\rightarrow}$.

$e ::= n$	integers
$\mid v$	variables
$\mid \lambda v\!:\!\tau \,.\, e$	abstraction
$\mid e_1 e_2$	application
$\mid \texttt{true} \mid \texttt{false}$	booleans
$\mid e_1 \; aop \; e_2$	arithmetic ops
$\mid e_1 \; bop \; e_2$	boolean ops
$\mid e_1 \; cmp \; e_2$	integer comparison
$\mid (e_1, e_2)$	pairs
$\mid \pi_1 e \mid \pi_2 e$	projection
$\mid ()$	unit
$\mid \texttt{in}_1^{\tau_1+\tau_2} e \mid \texttt{in}_2^{\tau_1+\tau_2} e$	injection
$\mid (\texttt{case}\ e\ \texttt{of}\ \texttt{in}_1(v_1) \rightarrow e_1 \mid \texttt{in}_2(v_2) \rightarrow e_2)$	case distinction

The type system for the above language is as follows.

$\tau ::= int$	integer
$\mid bool$	boolean
$\mid \tau_1 \rightarrow \tau_2$	function
$\mid \tau_1 \times \tau_2$	product type
$\mid unit$	unit type
$\mid \tau_1 + \tau_2$	sum type
$\mid void$	uninhabited type

$$\Gamma \vdash n : \mathit{int} \tag{1}$$

$$\Gamma \vdash v : \Gamma(v) \tag{2}$$

$$\frac{\Gamma[v \mapsto \tau_1] \vdash e : \tau_2}{\Gamma \vdash (\lambda v{:}\tau_1 \,.\, e) : \tau_1 \to \tau_2} \tag{3}$$

$$\frac{\Gamma \vdash e_1 : \tau \to \tau' \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 e_2 : \tau'} \tag{4}$$

$$\Gamma \vdash \texttt{true} : \mathit{bool} \tag{5}$$

$$\Gamma \vdash \texttt{false} : \mathit{bool} \tag{6}$$

$$\frac{\Gamma \vdash e_1 : \mathit{int} \qquad \Gamma \vdash e_2 : \mathit{int}}{\Gamma \vdash e_1 \;\mathit{aop}\; e_2 : \mathit{int}} \tag{7}$$

$$\frac{\Gamma \vdash e_1 : \mathit{bool} \qquad \Gamma \vdash e_2 : \mathit{bool}}{\Gamma \vdash e_1 \;\mathit{bop}\; e_2 : \mathit{bool}} \tag{8}$$

$$\frac{\Gamma \vdash e_1 : \mathit{int} \qquad \Gamma \vdash e_2 : \mathit{int}}{\Gamma \vdash e_1 \;\mathit{cmp}\; e_2 : \mathit{bool}} \tag{9}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2} \tag{10}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2 \qquad i \in \{1, 2\}}{\Gamma \vdash \pi_i e : \tau_i} \tag{11}$$

$$\Gamma \vdash () : \mathit{unit} \tag{12}$$

$$\frac{\Gamma \vdash e : \tau_i \qquad i \in \{1, 2\}}{\Gamma \vdash \texttt{in}_i^{\tau_1 + \tau_2} e : \tau_1 + \tau_2} \tag{13}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma[v_1 \mapsto \tau_1] \vdash e_1 : \tau \qquad \Gamma[v_2 \mapsto \tau_2] \vdash e_2 : \tau}{\Gamma \vdash (\texttt{case}\; e\; \texttt{of}\; \texttt{in}_1(v_1) \to e_1 \;\texttt{|}\; \texttt{in}_2(v_2) \to e_2) : \tau} \tag{14}$$

Since simply-typed λ-calculus is strongly normalizing, it is sometimes augmented with an explicit fixed-point operator μ:

$$e ::= \cdots \mid \mu v{:}\tau \,.\, e$$

$$\mu v{:}\tau \,.\, e \to_1 e[(\mu v{:}\tau . e)/v]$$

$$\frac{\Gamma[v \mapsto \tau] \vdash e : \tau}{\Gamma \vdash (\mu v{:}\tau \,.\, e) : \tau} \tag{15}$$

System F [Girard 1972, Reynolds 1974], also known as the *polymorphic λ-calculus*, extends the simply-typed λ-calculus with *type variables*. Valid System F expressions contain no free variables and no free type variables.

$e ::= \cdots \mid \Lambda\alpha.e$	polymorphic abstraction
$\quad\mid e[\tau]$	polymorphic instantiation
$\tau ::= \cdots \mid \alpha$	type variables
$\quad\mid \forall\alpha.\tau$	universal types

$$(\Lambda\alpha.e)[\tau] \to_1 e[\tau/\alpha]$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash \Lambda\alpha.e : \forall\alpha.\tau} \tag{16}$$

$$\frac{\Gamma \vdash e : \forall\alpha.\tau'}{\Gamma \vdash e[\tau] : \tau'[\tau/\alpha]} \tag{17}$$