CS411 Notes 9 – Types II

A. Demers

1 Mar 2001

Here we extend our little language to have an infinite set of types with some interesting structure, including function and record types, and assignable variables with aliasing.

1 Syntax

Types

$$
\begin{aligned}
\tau \ ::= \ & \text{int} \mid \text{bool} \mid \text{string} \\
& \mid \mathbf{var}(\tau) \\
& \mid \mathbf{prod}(\ \ldots \ x_i : \tau_i \ \ldots \ ) \\
& \mid \mathbf{fun}(\tau_1)\tau_2
\end{aligned}
$$

Addresses

$$
a^\tau \in \mathbf{A} \qquad \mathbf{a}^\tau \ = \ \langle i, \tau \rangle
$$

This represents a typed address constant – a^τ identifies ("is the address of") a value of type τ in the store. We assume there are infinitely many address constants of each type, and they are ordered (e.g. numerically).

Expressions

$$
\begin{aligned}
e \ ::= \ & n \mid t \mid s \mid a^\tau \\
& \mid \psi e_1 \mid e_1 \theta e_1 \mid e_1 ; e_2 \\
& \mid (\mathbf{if} \ e_1 \ \mathbf{then} \ e_2 \ \mathbf{else} \ e_3)
\end{aligned}
$$

$$
\begin{aligned}
&| \ (\textbf{while } e_1 \textbf{ do } e_2) \\
&| \ \textbf{newvar}(e_1) \mid e_1 \leftarrow e_2 \mid e_1 \uparrow \\
&| \ (\textbf{let } \ldots \ x_i \sim e_i \ \ldots \textbf{ in } e_0) \mid x \\
&| \ \langle \ \ldots \ x_i \sim e_i \ \ldots \ \rangle \mid e.x \\
&| \ (\textbf{lambda } x : \tau \textbf{ dot } e) \mid e_1(e_2)
\end{aligned}
$$

2 Typing Rules

2.1 Preliminaries

The typing rules follow the same general structure as the ones in Notes 8. That is, a type assignment π is a finite set

$$\pi \quad = \quad \{\ldots \ x_i \sim \tau_i \ \ldots\}$$

of assignments of types to names. The definitions of

$$\textbf{dom}(\pi) \qquad \pi|_S \qquad (\pi_1 \oplus \pi_2)$$

all appear in Notes 8. Judgements take the form

$$\pi \vdash e : \tau$$

and the well-typed program expressions are those e such that

$$\{\} \vdash e : \tau$$

is derivable for some τ.

2.2 The Rules

Constants

$$\frac{}{\pi \vdash n : \ \textbf{int}} \tag{T9.1}$$

$$\frac{}{\pi \vdash t : \ \textbf{bool}} \tag{T9.2}$$

$$\frac{}{\pi \vdash s : \ \mathbf{string}} \tag{T9.3}$$

$$\frac{}{\pi \vdash (a^{\tau}) : \ \mathbf{var}(\tau)} \tag{T9.4}$$

Operators

$$\frac{\pi \vdash e_1 : \ \tau_1}{\pi \vdash (\psi e_1) : \tau} \qquad \psi \text{ is } \tau_1 \to \tau \tag{T9.5}$$

$$\frac{\pi \vdash e_1 : \ \tau_1 \qquad \pi \vdash e_2 : \ \tau_2}{\pi \vdash (e_1 \theta e_2) : \tau} \qquad \theta \text{ is } \tau_1 \times \tau_2 \to \tau \tag{T9.6}$$

Control Structures

$$\frac{\pi \vdash e_1 : \ \tau_1 \qquad \pi \vdash e_2 : \ \tau_2}{\pi \vdash (e_1 ; e_2) : \tau_2} \tag{T9.7}$$

$$\frac{\pi \vdash e_1 : \ \text{bool} \qquad \pi \vdash e_2 : \ \tau \qquad \pi \vdash e_3 : \ \tau}{\pi \vdash (\mathbf{if}\ e_1\ \mathbf{then}\ e_2\ \mathbf{else}\ e_3) : \tau} \tag{T9.8}$$

$$\frac{\pi \vdash e_1 : \ \text{bool} \qquad \pi \vdash e_2 : \ \tau}{\pi \vdash (\mathbf{while}\ e_1\ \mathbf{do}\ e_2) : \ \text{bool}} \tag{T9.9}$$

Assignable Variables

$$\frac{\pi \vdash e_1 : \tau}{\pi \vdash \mathbf{newvar}(e_1) : \ \mathbf{var}(\tau)} \tag{T9.10}$$

$$\frac{\pi \vdash e_1 : \ \mathbf{var}(\tau) \qquad \pi \vdash e_2 : \ \tau}{\pi \vdash (e_1 \leftarrow e_2) : \tau} \tag{T9.11}$$

$$\frac{\pi \vdash e_1 : \ \mathbf{var}(\tau)}{\pi \vdash (e_1 \uparrow) : \tau} \tag{T9.12}$$

Let Bindings

$$\frac{\begin{array}{c}\cdots \quad \pi \vdash e_i : \ \tau_i \quad \cdots \\ (\pi \oplus \{\ldots\ x_i : \tau_i\ \ldots\}) \vdash e_0 : \tau\end{array}}{(\pi \vdash \mathbf{let}\ \ldots\ x_i \sim e_i\ \ldots\ \mathbf{in}\ e_0) : \tau} \tag{T9.13}$$

$$\frac{}{\pi \vdash x : \ \tau} \qquad \text{where } x : \tau \in \pi \tag{T9.14}$$

Products

$$\frac{\cdots \qquad \pi \vdash e_i : \ \tau_i \qquad \cdots}{\pi \vdash \langle \ldots\ x_i \sim e_i\ \ldots \rangle : \ \mathbf{prod}(\ldots\ x_i : \tau_i\ \ldots)} \tag{T9.15}$$

$$\frac{\pi \vdash e : \ \mathbf{prod}(\ldots\ x_i : \tau_i\ \ldots)}{\pi \vdash e.x_i : \tau_i} \tag{T9.16}$$

Functions

$$\frac{(\pi \oplus \{x : \tau'\}) \vdash e : \ \tau}{\pi \vdash (\mathbf{lambda}\ x : \tau'\ \mathbf{dot}\ e) : \ \mathbf{fun}(\tau')\tau} \tag{T9.17}$$

$$\frac{\pi \vdash e_1 : \ \mathbf{fun}(\tau_2)\tau \qquad \pi \vdash e_2 : \tau_2}{\pi \vdash e_1(e_2) : \tau} \tag{T9.18}$$

2.3 Properties

The following are easily provable by induction on derivations:

Prop (unique typing) For any π, e, τ, τ'

$$(\pi \vdash e : \tau) \wedge (\pi \vdash e : \tau') \quad \Rightarrow \quad (\tau \equiv \tau')$$

That is, in any type environment an expression can be typed only one way. □

Prop (support) For any π, e, τ

$$(\pi \vdash e : \tau) \quad \Rightarrow \quad ((\pi|_{FV(e)} \vdash e : \tau)$$

That is, the type of an expression depends only on the types of its free variables in the type assignment. □