Lecture 11: The Simply Typed Lambda-Calculus

Additional types

Polyvios Pratikakis

Computer Science Department, University of Crete

Type Systems and Programming Languages

Last time

- The unit type
- Pairs, the product type $T \times T$
- Binary unions, the sum type $T + T$

The unit type

- Syntax

$$\begin{array}{rcl} e & ::= & \ldots \mid () \mid e_1; e_2 \\ v & ::= & \ldots \mid () \\ T & ::= & \ldots \mid \mathit{Unit} \end{array}$$

- Typing

$$[\text{T-Unit}]\,\frac{}{\Gamma \vdash () : \mathit{Unit}} \qquad [\text{T-Seq}]\,\frac{\Gamma \vdash e_1 : \mathit{Unit} \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1; e_2 : T}$$

- Semantics

$$\frac{}{(); e_2 \to e_2} \qquad \frac{e_1 \to e_1'}{e_1; e_2 \to e_1'; e_2}$$

Pairs: the product type

- Syntax

$$
\begin{array}{lcl}
e & ::= & \ldots \mid e.1 \mid e.2 \mid (e, e) \\
v & ::= & \ldots \mid (v, v) \\
T & ::= & \ldots \mid T \times T
\end{array}
$$

- Typing

$$
[\text{T-PAIR}]\ \frac{\Gamma \vdash e_1 : T_1 \qquad \Gamma \vdash e_2 : T_2}{\Gamma \vdash (e_1, e_2) : T_1 \times T_2}
$$

$$
[\text{T-FST}]\ \frac{\Gamma \vdash e : T \times T'}{\Gamma \vdash e.1 : T} \qquad [\text{T-SND}]\ \frac{\Gamma \vdash e : T \times T'}{\Gamma \vdash e.2 : T'}
$$

Pairs: the product type (cont'd)

- Semantics

$$\frac{}{(v_1, v_2).1 \rightarrow v_1} \qquad \frac{}{(v_1, v_2).2 \rightarrow v_2}$$

$$\frac{e_1 \rightarrow e_1'}{(e_1, e_2) \rightarrow (e_1', e_2)} \qquad \frac{e_2 \rightarrow e_2'}{(v_1, e_2) \rightarrow (v_1, e_2')}$$

$$\frac{e \rightarrow e'}{e.1 \rightarrow e'.1} \qquad \frac{e \rightarrow e'}{e.2 \rightarrow e'.2}$$

Binary unions: the sum type

- Syntax

$$
\begin{array}{rcl}
e & ::= & \ldots \mid \text{inl } e \mid \text{inr } e \mid \text{case } e \text{ of inl } x => e \mid \text{inr } x => e \\
v & ::= & \ldots \mid \text{inl } v \mid \text{inr } v \\
T & ::= & \ldots \mid T + T
\end{array}
$$

- Typing

$$
[\text{T-INL}]\ \frac{\Gamma \vdash e : T_1}{\Gamma \vdash \text{inl } e : T_1 + T_2}
\qquad
[\text{T-INR}]\ \frac{\Gamma \vdash e : T_2}{\Gamma \vdash \text{inr } e : T_1 + T_2}
$$

$$
[\text{T-CASE-UNION}]\ \frac{\begin{array}{c}\Gamma \vdash e : T_1 + T_2 \\ \Gamma, x_1 : T_1 \vdash e_1 : T \\ \Gamma, x_2 : T_2 \vdash e_2 : T\end{array}}{\Gamma \vdash \text{case } e \text{ of inl } x_1 \Rightarrow e_1 \mid \text{inr } x_2 \Rightarrow e_2 : T}
$$

Binary unions: the sum type (cont'd)

- Semantics

$$\frac{e \to e'}{\text{inl } e \to \text{inl } e'} \qquad \frac{e \to e'}{\text{inr } e \to \text{inr } e'}$$

$$\frac{e \to e'}{\begin{array}{l}\text{case } e \text{ of inl } x_1 => e_1 \mid \text{inr } x_2 => e_2 \to \\ \qquad \text{case } e' \text{ of inl } x_1 => e_1 \mid \text{inr } x_2 => e_2\end{array}}$$

$$\frac{}{\text{case } (\text{inl } v) \text{ of inl } x_1 => e_1 \mid \text{inr } x_2 => e_2 \to e_1[v/x_1]}$$

$$\frac{}{\text{case } (\text{inr } v) \text{ of inl } x_1 => e_1 \mid \text{inr } x_2 => e_2 \to e_2[v/x_2]}$$

Let bindings

- Syntax

$$e \quad ::= \quad \ldots \mid \mathsf{let}\ x = e\ \mathsf{in}\ e$$

- Typing

$$[\textsc{T-Let}]\ \frac{\Gamma \vdash e_1 : T_1 \qquad \Gamma, x : T_1 \vdash e_2 : T_2}{\Gamma \vdash \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 : T_2}$$

- Semantics

$$\frac{}{\mathsf{let}\ x = v\ \mathsf{in}\ e \to e[v/x]}$$

$$\frac{e_1 \to e_1'}{\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \to \mathsf{let}\ x = e_1'\ \mathsf{in}\ e_2}$$

Let bindings (cont'd)

- In untyped calculus defined as syntactic sugar (preprocessed):

$$\text{let } x = e_1 \text{ in } e_2 \doteq (\lambda x.e_2)\ \ e_1$$

- Now, functions are annotated with types:

$$\text{let } x = e_1 \text{ in } e_2 \doteq (\lambda x : T_1.e_2)\ \ e_1$$

- What is the type T_1?
- Answer: ask the typechecker!
- Let-binding not exactly syntactic sugar in typed calculus

Tuples

- Generalize pairs to tuples of n items
- Syntax

$$\begin{array}{rcl} e & ::= & \ldots \mid e.i \mid \{e_1, \ldots, e_n\} \\ v & ::= & \ldots \mid \{v_1, \ldots, v_n\} \\ T & ::= & \ldots \mid \{T_1 \times \ldots \times T_n\} \end{array}$$

- Typing

$$[\text{T-TUPLE}]\frac{\forall 1 \leq i \leq n,\ \Gamma \vdash e_i : T_i}{\Gamma \vdash \{e_1, \ldots, e_n\} : \{T_1 \times \ldots \times T_n\}}$$

$$[\text{T-PROJ}]\frac{\Gamma \vdash e : \{T_1 \times \ldots \times T_n\}}{\Gamma \vdash e.i : T_i}$$

Tuples (cont'd)

- Semantics

$$\frac{e \to e'}{e.i \to e'.i} \qquad \frac{}{\{v_1, \ldots, v_n\}.i \to v_i}$$

$$\frac{e_j \to e_j'}{\{v_1, \ldots, v_{j-1}, e_j, \ldots e_n\} \to \left\{v_1, \ldots, v_{j-1}, e_j', \ldots e_n\right\}}$$

Records

- Generalize tuples with arbitrary labels: structs
- Syntax

$$\begin{array}{rcl} e & ::= & \ldots \mid e.l \mid \{l_1 = e_1, \ldots, l_n = e_n\} \\ v & ::= & \ldots \mid \{l_1 = v_1, \ldots, l_n = v_n\} \\ T & ::= & \ldots \mid \{l_1 : T_1, \ldots, l_n : T_n\} \end{array}$$

- Typing

$$[\text{T-Record}]\ \frac{\forall 1 \leq i \leq n,\ \Gamma \vdash e_i : T_i}{\Gamma \vdash \{l_1 = e_1, \ldots, l_n = e_n\} : \{l_1 : T_1, \ldots, l_n : T_n\}}$$

$$[\text{T-Proj}]\ \frac{\Gamma \vdash e : \{l_1 : T_1, \ldots l_n : T_n\}}{\Gamma \vdash e.l_i : T_i}$$

Records (cont'd)

- Semantics

$$\frac{e \to e'}{e.l_i \to e'.l_i} \qquad \frac{}{\{l_1 = v_1, \ldots, l_n = v_n\}.l_i \to v_i}$$

$$\frac{e_j \to e'_j}{\begin{array}{c}\{l_1 = v_1, \ldots, l_{j-1} = v_{j-1}, l_j = e_j, \ldots l_n = e_n\} \to \\ \left\{l_1 = v_1, \ldots, l_{j-1} = v_{j-1}, l_j = e'_j, \ldots l_n = e_n\right\}\end{array}}$$

Variants

- Generalization of unions (sums)
- Syntax

$$\begin{array}{lcl} e & ::= & \ldots \mid l\,\langle e\rangle \;\text{ as }\; T \mid \text{case } e \text{ of } \{l_i\,\langle x_i\rangle \;=>\; e_i\}^{1\leq i\leq n} \\ v & ::= & \ldots \mid l\,\langle v\rangle \;\text{ as }\; T \\ T & ::= & \ldots \mid \left\langle l_i : T_i^{1\leq i\leq n}\right\rangle \end{array}$$

- Typing

$$[\text{T-VARIANT}]\;\frac{\Gamma \vdash e_j : T_j}{\Gamma \vdash l_j\,\langle e_j\rangle \;\text{ as }\; \left\langle l_i : T_i^{1\leq i\leq n}\right\rangle : \left\langle l_i : T_i^{1\leq i\leq n}\right\rangle}$$

$$[\text{T-CASE-VAR}]\;\frac{\Gamma \vdash e : \left\langle l_i : T_i^{1\leq i\leq n}\right\rangle \qquad \forall 1 \leq i \leq n,\; \Gamma, x_i : T_i \vdash e_i : T}{\Gamma \vdash \text{case } e \text{ of } \{l_i\,\langle x_i\rangle \Rightarrow e_i\}^{1\leq i\leq n} : T}$$

Variants (cont'd)

- Semantics

$$\frac{}{\text{case } (l_j \langle v_j \rangle \text{ as } T) \text{ of } \{l_i \langle x_i \rangle => e_i\}^{1 \leq i \leq n} \to e_j[v_j/x_j]}$$

$$\frac{e \to e'}{\text{case } e \text{ of } \{l_i \langle x_i \rangle => e_i\}^{1 \leq i \leq n} \to \text{case } e' \text{ of } \{l_i \langle x_i \rangle => e_i\}^{1 \leq i \leq n}}$$

$$\frac{e_i \to e_i'}{l_i \langle e_i \rangle \text{ as } T \to l_i \langle e_i' \rangle \text{ as } T}$$

Lists

- Syntax

$$
\begin{array}{lcl}
e & ::= & \ldots \mid \text{nil } [T] \mid \text{cons } [T] e\ e \mid \text{case } e \text{ of } \{\text{nil } => e \mid \text{cons } x\ x => e\} \\
v & ::= & \ldots \mid \text{nil } [T] \mid \text{cons } [T] v\ v \\
T & ::= & \ldots \mid \mathit{List}\ T
\end{array}
$$

- Typing

$$
[\text{T-Nil}]\ \frac{}{\Gamma \vdash \text{nil } [T] : \mathit{List}\ T}
\qquad
[\text{T-Cons}]\ \frac{\Gamma \vdash e : T \quad \Gamma \vdash e' : \mathit{List}\ T}{\Gamma \vdash \text{cons } [T] e\ e' : \mathit{List}\ T}
$$

$$
[\text{T-Case-List}]\ \frac{\Gamma \vdash e : \mathit{List}\ T \quad \Gamma \vdash e_1 : T' \quad \Gamma, x_h : T, x_t : \mathit{List}\ T \vdash e_2 : T'}{\Gamma \vdash \text{case } e \text{ of } \{\text{nil } \Rightarrow e_1 \mid \text{cons } x_h\ x_t \Rightarrow e_2\} : T'}
$$

Lists (cont'd)

- Semantics

$$\frac{e_1 \rightarrow e_1'}{\text{cons }[T] e_1\ e_2 \rightarrow \text{cons }[T] e_1'\ e_2} \qquad \frac{e_2 \rightarrow e_2'}{\text{cons }[T] v\ e_2 \rightarrow \text{cons }[T] v\ e_2'}$$

$$\frac{e \rightarrow e'}{\begin{array}{c}\text{case } e \text{ of } \{\text{nil } => e_1 | \text{cons } x_h\ x_t => e_2\} \rightarrow \\ \text{case } e' \text{ of } \{\text{nil } => e_1 | \text{cons } x_h\ x_t => e_2\}\end{array}}$$

$$\frac{}{\text{case nil }[T] \text{ of } \{\text{nil } => e_1 | \text{cons } x_h\ x_t => e_2\} \rightarrow e_1}$$

$$\frac{}{\text{case cons }[T] v_1\ v_2 \text{ of } \{\text{nil } => e_1 | \text{cons } x_h\ x_t => e_2\} \rightarrow e_2[v_1/x_h, v_2/x_t]}$$