COMPSCI 3MI3 - Principles of Programming Languages

Data-Structures

J. Carette

McMaster University

Fall 2023

Adapted from "Types and Programming Languages" by Benjamin C. Pierce

Adding data-structures: pairs

$$\langle t \rangle ::= \ldots \mid \{\text{t, t}\} \mid \{\text{t, t}\}.1 \mid \{\text{t, t}\}.2$$

$$\langle v \rangle ::= \ldots \mid \{\text{v, v}\}$$

$$\frac{t \to t'}{t.1 \to t'.1} \quad \text{(E-Proj1)}$$

$$\frac{t \to t'}{t.2 \to t'.2} \quad \text{(E-Proj2)}$$

$$\frac{t_1 \to t_1'}{\{t_1, t_2\} \to \{t_1', t_2\}} \quad \text{(E-Pair1)}$$

$$\frac{t_2 \to t_2'}{\{v_1, t_2\} \to \{v_1, t_2'\}} \quad \text{(E-Pair2)}$$

$$\{v_1, v_2\}.1 \to v_1 \quad \text{(E-PairBeta1)}$$

$$\{v_1, v_2\}.2 \to v_2 \quad \text{(E-PairBeta2)}$$

Typing Pairs

$$\langle T \rangle ::= \ \ldots$$
$$\mid \quad \langle T \rangle \times \langle T \rangle$$

This is known as the **product** or the **Cartesian Product** type constructor.

$$\frac{\Gamma \vdash t_1 : T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash \{t_1, t_2\} : T_1 \times T_2} \quad \text{(T-Pair)}$$

$$\frac{\Gamma \vdash t : T_1 \times T_2}{\Gamma \vdash t.1 : T_1} \quad \text{(T-Proj1)}$$

$$\frac{\Gamma \vdash t : T_1 \times T_2}{\Gamma \vdash t.2 : T_2} \quad \text{(T-Proj2)}$$

Tuples: from 2 to n

$\langle t \rangle$ ::= ...
$\quad | \quad \{\langle t \rangle, \langle t \rangle, \ldots, \langle t \rangle\}$
$\quad | \quad$ t.i

where there are n terms in the first case, and $1 \leq i \leq n$ in the second.

$\langle v \rangle$ ::= ...
$\quad | \quad \{\langle v \rangle, \langle v \rangle, \ldots, \langle v \rangle\}$

$\langle T \rangle$ ::= ...
$\quad | \quad \{\langle T \rangle \times \langle T \rangle \times \ldots \times \langle T \rangle \}$

As this ... notation can get tiresome, we use $\vec{t}$, $\vec{v}$ and $\vec{T}$.

Evaluation Rules

$$\frac{j \in 1..n}{\{\vec{v}\}.j \to v_j} \qquad \text{(E-ProjTuple)}$$

$$\frac{t \to t'}{t.i \to t'.i} \qquad \text{(E-Proj)}$$

$$\frac{t_j \to t_j'}{\{v_1, v_2, \ldots, v_{j-1}, t_j, \ldots t_n\} \to \{v_1, v_2, \ldots, v_{j-1}, t_j', \ldots t_n\}} \qquad \text{(E-Tuple)}$$

Tuping Typles

$$\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \ldots \Gamma \vdash t_n : T_n}{\Gamma \vdash \{\vec{t}\} : \{\vec{T}\}} \qquad \text{(T-Tuple)}$$

$$\frac{j \in 1..n \qquad \Gamma \vdash t : \{\vec{T}\}}{\Gamma \vdash t.j : T_j} \qquad \text{(T-Proj)}$$

Record

Numbers are silly labels, let's use names as **labels**. $l \in \mathcal{L}$.

$$
\begin{array}{rl}
\langle t \rangle ::= & \ldots \\
\mid & \{\langle l \rangle = \langle t \rangle, \langle l \rangle = \langle t \rangle, \ldots, \langle l \rangle = \langle t \rangle\ \} \\
\mid & \langle t \rangle.\langle l \rangle
\end{array}
$$

$$
\begin{array}{rl}
\langle v \rangle ::= & \ldots \\
\mid & \{\langle l \rangle = \langle v \rangle, \langle l \rangle = \langle v \rangle, \ldots, \langle l \rangle = \langle v \rangle\ \}
\end{array}
$$

$$
\begin{array}{rl}
\langle T \rangle ::= & \ldots \\
\mid & \{\langle l \rangle : \langle T \rangle, \langle l \rangle : \langle T \rangle, \ldots, \langle l \rangle : \langle T \rangle\ \}
\end{array}
$$

structs in C, object with only fields in Java, dictionaries (sort of) in Python

Evaluation Rules

$$\frac{j \in 1..n}{\{\overrightarrow{l = v}\}.l_j \to v_j} \qquad \text{(E-ProjRcd)}$$

$$\frac{t \to t'}{t.l_i \to t'.l_i} \qquad \text{(E-Proj)}$$

$$\frac{t_j \to t'_j}{\begin{array}{c}\{l_1 = v_1, \ldots, l_{j-1} = v_{j-1}, l_j = t_j, \ldots l_n = t_n\} \to \\ \{l_1 = v_1, \ldots, l_{j-1} = v_{j-1}, l_j = t'_j, \ldots l_n = t_n\}\end{array}} \qquad \text{(E-Rcd)}$$

Note: order of labels is induced by the language somehow. Usually at type declaration time.

Typing

$$\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \ldots \Gamma \vdash t_n : T_n}{\Gamma \vdash \{\overrightarrow{f = t}\} : \{\overrightarrow{f : T}\}} \qquad \text{(T-Tuple)}$$

$$\frac{j \in 1..n \qquad \Gamma \vdash t : \{\overrightarrow{T}\}}{\Gamma \vdash t.j : T_j} \qquad \text{(T-Proj)}$$