CS6848 - Principles of Programming Languages

Principles of Programming Languages

V. Krishna Nandivada

IIT Madras

Recap

- Structural subtyping
- Unification algorithm

Recall

$$
\begin{array}{lll}
e & ::= & x \mid \lambda x.e \mid e_1 e_2 \mid c \mid succ\ e \\
x & \in & \text{Identifier (infinite set of variables)} \\
c & \in & \text{Integer} \\
v & :: = & c|\lambda x.e \\
t & :: = & \text{Int} \mid t -> t
\end{array}
$$

Extending a language

- Extend the language grammar that will lead to new terms.
- Extend the allowed values.
- Extend the types.
- New operational semantics.
- New typing rules.

Pairs

- Expressions
$$e ::= \cdots | (e_1, e_2) | e.1 | e.2$$
- Values
$$v ::= \cdots | (v_1, v_2)$$
- Types
$$t ::= \cdots | t_1 \times t_2$$

New operational semantics

- First element:
$$(Pair\ \beta 1)\ \ (v_1, v_2).1 -> v_1$$
- Second element:
$$(Pair\ \beta 2)\ \ (v_1, v_2).2 -> v_2$$
- $$[Projection\ 1\ ] e -> e'.e.1 -> e'.1$$
- $$[Projection\ 2\ ] e -> e'.e.2 -> e'.2$$
- $$[Pair\ Evaluation\ 1\ ] e_1 -> e_1'.(e_1, e_2) -> (e_1', e_2)$$
- $$[Pair\ Evaluation\ 2\ ] e_2 -> e_2'.(v_1, e_2) -> (v_1, e_2')$$

Properties of pairs

- The components are evaluated left to right.
- The pair must be fully evaluated to get the components.
- A pair that is passed as an argument will be fully evaluated, before the function starts executing(in call by value semantics).

Tuples

- Expressions
$$e ::= \cdots | (e_i^{i \in 1..n}) | e.i$$
- Values
$$v ::= \cdots | (v_i^{i \in 1..n})$$
- Types
$$t ::= \cdots | (t_i^{i \in 1..n})$$

New operational semantics

- Element j:

$$(\beta)(v_i^{i\in 1..n}).j -> v_j$$

- $$[Projection\ 1]e -> e'e.i -> e'.i$$

- $$[Tuple\ Evaluation]e_j -> e'_j(v_i^{i\in 1..j-1}, e_j, e_k^{k\in j+1..n}) -> (v_i^{i\in 1..j-1}, e'_j, e_k^{k\in j+1..n})$$

Typing rules for tuples

- $$[Tuple]A \vdash \forall i\ e_i : t_i A \vdash (l_i = e_i^{i\in 1..n}) : (l_i : t_i^{i\in 1..n})$$

- $$[Projection]A \vdash e : (l_i : t^{i\in 1..n})A \vdash e.l_j : t_j$$