Dynamics

The Phases of PLs

Static Phase: Well-formedness of programs; e.g., parsing and type checking.

Dynamic Phase: Execution of programs.

Transition Systems

A *transitions systems* is specified by the following four forms of judgment:

1. s state, asserting that s is a *state* of the transition system.

2. s final, where s, state, asserting that s is a final state.

3. s inital, where s state, asserting that s is an initial state.

4. $s_1 \mapsto s_2$, where s_1 state and s_2 state, asserting that state s_1 may transition to the state s_2 .

Transition Systems

1. No transition can transition out of a final state.

2. A state where from which no transition is possible is called a *stuck state*.

3. Final states are stuck by definitions, but others may exist.

4. A transition system is *deterministic* iff for every state s, there exists exactly one transition $s \mapsto s'$ for any state s'. Otherwise, it is *non-deterministic*.

Transition Sequence

A sequence of states $s_0, \ldots, s_n$ such that s_0 initial, and $s_i \mapsto s_{i+1}$ for $0 \leq i < n$.

- *maximal* iff there is no s such that $s_n \mapsto s$.
- *complete* iff it is maximal and, s_n final
- $s \downarrow$ asserts that there is a complete sequence starting with s.

Transition Judgment

$$\frac{}{s \longmapsto^* s}\ \text{Refl}$$

$$\frac{s \longmapsto s' \qquad s' \longmapsto^* s''}{s \longmapsto^* s''}\ \text{Step}$$

Transition Judgment: Rule Induction

Suppose we want to show $P(s, s')$ whenever $s \mapsto s'$ holds. Then show:

1. $P(s, s)$
2. if $s \mapsto s'$ and $P(s', s'')$, then $P(s, s'')$ (head expansion)

Transition Judgment: n-times Iterated

$$\frac{}{s \mapsto^{0} s}\ \textsf{Refl}$$

$$\frac{s \mapsto s' \quad s' \mapsto^{n} s''}{s \mapsto^{n+1} s''}\ \textsf{Step}$$

Transition Judgment: n-times Iterated

For all states s_1 and s_2, $s_1 \mapsto^* s_2$ iff $s_1 \mapsto^k s_2$ for some $k \geq 0$.

$$\frac{}{s \mapsto^0 s}\ \text{Refl} \qquad\qquad \frac{s \mapsto s' \quad s' \mapsto^n s''}{s \mapsto^{n+1} s''}\ \text{Step}$$

Structural Dynamics

Described by a transition system on abstract syntax:

1. States are closed expressions.
2. All states are initial.
3. Final states are closed values.

Structural Dynamics

$$\frac{}{\texttt{num}[n]\ \mathsf{val}}\ \text{numVal}$$

$$\frac{}{\texttt{str}[s]\ \mathsf{val}}\ \text{strVal}$$

$$\frac{n_1 + n_2 = n\ \mathsf{nat}}{\texttt{plus}(\texttt{num}[n_1];\texttt{num}[n_2]) \mapsto \texttt{num}[n]}\ \text{plusVal}$$

$$\frac{e_1 \mapsto e_1'}{\texttt{plus}(e_1;e_2) \mapsto \texttt{plus}(e_1';e_2)}\ \text{plus1}$$

$$\frac{e_1\ \mathsf{val} \quad e_2 \mapsto e_2'}{\texttt{plus}(e_1;e_2) \mapsto \texttt{plus}(e_1;e_2')}\ \text{plus2}$$

Structural Dynamics

$$\frac{s_1 \mathbin{\hat{}} s_2 = s \ \mathsf{str}}{\texttt{cat}(\texttt{str}[s_1];\texttt{str}[s_2]) \mapsto \texttt{str}[s]} \ \text{catVal}$$

$$\frac{e_1 \mapsto e_1'}{\texttt{cat}(e_1;e_2) \mapsto \texttt{cat}(e_1';e_2)} \ \text{cat1}$$

$$\frac{e_1 \ \mathsf{val} \quad e_2 \mapsto e_2'}{\texttt{cat}(e_1;e_2) \mapsto \texttt{cat}(e_1;e_2')} \ \text{cat2}$$

Structural Dynamics

$$\dfrac{[e_1 \ \mathsf{val}]}{\mathtt{let}(e_1; x\,.\,e_2) \mapsto [e_1/x]e_2}\ \text{letVal}$$

$$\left[\dfrac{e_1 \mapsto e_1'}{\mathtt{let}(e_1; x\,.\,e_2) \mapsto \mathtt{let}(e_1'; x\,.\,e_2)}\right]\text{let}$$

Structural Dynamics: Example

$$\mathsf{let}(\mathsf{str}[\texttt{"C"}], s_1\,.\,\mathsf{let}(\mathsf{str}[\texttt{"S"}], s_2\,.\,\mathsf{len}(\mathsf{cat}(s_1, s_2)) \mapsto \mathsf{num}[2]$$

Structural Dynamics: Example

$$\mathsf{let}(\mathsf{str}[\text{"C"}], s_1 \,.\, \mathsf{let}(\mathsf{str}[\text{"S"}], s_2 \,.\, \mathsf{len}(\mathsf{cat}(s_1, s_2)))) \mapsto^{\mathsf{LetVal}} \mathsf{let}(\mathsf{str}[\text{"S"}], s_2 \,.\, \mathsf{len}(\mathsf{cat}(\mathsf{str}[\text{"C"}], s_2))$$

$$\mapsto^{\mathsf{LetVal}} \mathsf{len}(\mathsf{cat}(\mathsf{str}[\text{"C"}], \mathsf{str}[\text{"S"}]))$$

$$\mapsto^{\mathsf{Len1,CatVal}} \mathsf{len}(\mathsf{str}[\text{"CS"}])$$

$$\mapsto^{\mathsf{LenVal}} \mathsf{num}[2]$$

Examples

$2x + 3y$, where $x = 4$ and $y = 2$

- Translate it into our abstract syntax.
- Type it.
- Evaluate it using both methods.

Examples

Translation:

$$\mathrm{let}(4, x\ .\ \mathrm{let}(2, y\ .\ \mathrm{plus}(\mathrm{mult}(2, x), \mathrm{mult}(3, y))$$

$$\dfrac{\varnothing \vdash 4 : \mathsf{Num} \qquad \dfrac{\dfrac{}{x : \mathsf{Num} \vdash 2 : \mathsf{Num}}\,\textsf{Num} \qquad \dfrac{\mathscr{D}_1 \qquad \mathscr{D}_2}{x : \mathsf{Num}, y : \mathsf{Num} \vdash \mathsf{plus}(\mathsf{mult}(2,x), \mathsf{mult}(3,y)) : \mathsf{Num}}\,\textsf{Plus}}{x : \mathsf{Num} \vdash \mathsf{let}(2, y\,.\,\mathsf{plus}(\mathsf{mult}(2,x), \mathsf{mult}(3,y))) : \mathsf{Num}}\,\textsf{Let}}{\varnothing \vdash \mathsf{let}(4, x\,.\,\mathsf{let}(2, y\,.\,\mathsf{plus}(\mathsf{mult}(2,x), \mathsf{mult}(3,y)))) : \mathsf{Num}}\,\textsf{Let}$$

$\mathscr{D}_1$:

$$\dfrac{\dfrac{}{x : \mathsf{Num}, y : \mathsf{Num} \vdash 2 : \mathsf{Num}}\,\textsf{Num} \qquad \dfrac{}{x : \mathsf{Num}, x : \mathsf{Num} \vdash y : \mathsf{Num}}\,\textsf{Num}}{x : \mathsf{Num}, y : \mathsf{Num} \vdash \mathsf{mult}(2,x) : \mathsf{Num}}\,\textsf{Mult}$$

$\mathscr{D}_2$:

$$\dfrac{\dfrac{}{x : \mathsf{Num}, y : \mathsf{Num} \vdash 3 : \mathsf{Num}}\,\textsf{Num} \qquad \dfrac{}{x : \mathsf{Num}, y : \mathsf{Num} \vdash y : \mathsf{Num}}\,\textsf{Num}}{x : \mathsf{Num}, y : \mathsf{Num} \vdash \mathsf{mult}(3,y) : \mathsf{Num}}\,\textsf{Mult}$$

$$
\begin{aligned}
& \mathsf{let}(4, x\,.\,\mathsf{let}(2, y\,.\,\mathsf{plus}(\mathsf{mult}(2, x), \mathsf{mult}(3, y)))) \\
\overset{\mathsf{LetVal}}{\longmapsto}\ & \mathsf{let}(2, y\,.\,\mathsf{plus}(\mathsf{mult}(2,4), \mathsf{mult}(3, y))) \\
\overset{\mathsf{LetVal}}{\longmapsto}\ & \mathsf{plus}(\mathsf{mult}(2,4), \mathsf{mult}(3,2)) \\
\overset{\mathsf{Plus1}}{\longmapsto}\ & \mathsf{plus}(8, \mathsf{mult}(3,2)) \\
\overset{\mathsf{Plus2}}{\longmapsto}\ & \mathsf{plus}(8,6) \\
\overset{\mathsf{PlusVal}}{\longmapsto}\ & 14
\end{aligned}
$$

$$\dfrac{\dfrac{\dfrac{}{4\ \mathsf{val}}\,\textsf{Num}}{\mathsf{let}(4,x\,.\,\mathsf{let}(2,y\,.\,\mathsf{plus}(\mathsf{mult}(2,x),\mathsf{mult}(3,y)))) \mapsto \mathsf{let}(2,y\,.\,\mathsf{plus}(\mathsf{mult}(2,4),\mathsf{mult}(3,y)))}\,\textsf{LetVal} \qquad \mathscr{D}_1}{\mathsf{let}(4,x\,.\,\mathsf{let}(2,y\,.\,\mathsf{plus}(\mathsf{mult}(2,x),\mathsf{mult}(3,y)))) \mapsto^* 14}\,\textsf{Step}$$

$\mathscr{D}_1$:

$$\dfrac{\dfrac{\dfrac{}{2\ \mathsf{val}}\,\textsf{Num}}{\mathsf{let}(2,y\,.\,\mathsf{plus}(\mathsf{mult}(2,4),\mathsf{mult}(3,y))) \mapsto \mathsf{plus}(\mathsf{mult}(2,4),\mathsf{mult}(3,2))}\,\textsf{LetVal} \qquad \mathscr{D}_2}{\mathsf{let}(2,y\,.\,\mathsf{plus}(\mathsf{mult}(2,4),\mathsf{mult}(3,y))) \mapsto^* 14}\,\textsf{Step}$$

$\mathscr{D}_2$:

$$\dfrac{\dfrac{\dfrac{2 * 4 = 8}{\mathsf{mult}(2,4) \mapsto 8}\ \textsf{MultVal}}{\mathsf{plus}(\mathsf{mult}(2,4), \mathsf{mult}(3,2)) \mapsto \mathsf{plus}(8,\mathsf{mult}(3,2))}\ \textsf{Plus1} \qquad \mathscr{D}_3}{\mathsf{plus}(\mathsf{mult}(2,4), \mathsf{mult}(3,2)) \mapsto^* 14}\ \textsf{Step}$$

$\mathscr{D}_3$:

$$\dfrac{\dfrac{\dfrac{}{8\ \mathsf{val}}\ \textsf{Num} \qquad \dfrac{3 * 2 = 6}{\mathsf{mult}(3,2) \mapsto 6}\ \textsf{MultVal}}{\mathsf{plus}(8,\mathsf{mult}(3,2)) \mapsto \mathsf{plus}(8,6)}\ \textsf{Plus2} \qquad \mathscr{D}_4}{\mathsf{plus}(8,\mathsf{mult}(3,2)) \mapsto^* 14}\ \textsf{Step}$$

$\mathscr{D}_4:$

$$
\dfrac{\dfrac{8+6=14}{\mathsf{plus}(8,6) \mapsto 14}\;\textsf{PlusVal} \qquad \dfrac{}{14 \mapsto^* 14}\;\textsf{Refl}}{\mathsf{plus}(8,6) \mapsto^* 14}\;\textsf{Step}
$$

Lemma. If $e_1 \mapsto e_2$ and $e_1 \mapsto e_3$, then $e_2 =_\alpha e_3$

Proof. By induction on the form of $e_1 \mapsto e_2$.

Case: $$\frac{n_1 + n_2 = n \ \mathsf{nat}}{\mathsf{plus}\,(\mathsf{num}[n_1]; \mathsf{num}[n_2]) \mapsto \mathsf{num}[n]} \ \textsc{PlusVal}$$

In this case $e_1 = \mathsf{plus}(\mathsf{num}[n_1]; \mathsf{num}[n_2])$ and $e_2 = \mathsf{num}[n]$. It suffices to determine which expression e_3 is. The only possibilities are $e_3 = \mathsf{plus}(e_3'; e_3'')$ for some expressions e_3' and e_3'', or $e_3 = \mathsf{num}[n]$. We know that it is impossible for the former to be true, because e_1's parameters are values, and e_3 must be reachable via e_1. Thus, $e_3 = \mathsf{num}[n] = e_2$.

Lemma. If $e_1 \mapsto e_2$ and $e_1 \mapsto e_3$, then $e_2 =_\alpha e_3$

Case: $$\frac{e_4 \mapsto e_4'}{\mathsf{plus}\,(e_4; e_5) \mapsto \mathsf{plus}\,(e_4'; e_5)}\ \textsc{Plus1}$$

In this case $e_1 = \mathsf{plus}(e_4; e_5)$ and $e_2 = \mathsf{plus}(e_4'; e_5)$. It suffices to determine the form of e_3. By assumption, $e_4 \mapsto e_4'$, and thus, e_4 is not a value. Thus, it must be the case that $e_3 = \mathsf{plus}(e_4''; e_5)$ and $e_1 \mapsto e_3$ ends in the Plus1 rule as follows:

$$\frac{e_4 \mapsto e_4''}{\mathsf{plus}\,(e_4; e_5) \mapsto \mathsf{plus}\,(e_4''; e_5)}\ \textsc{Plus1}$$

At this point we know by assumption that $e_4 \mapsto e_4'$ and from our analysis above $e_4 \mapsto e_4''$. Hence, by the IH $e_4' =_\alpha e_4''$ which implies that $e_2 = \mathsf{plus}(e_4'; e_5) =_\alpha \mathsf{plus}(e_4'; e_5) = e_3$.

Lemma. If $e_1 \mapsto e_2$ and $e_1 \mapsto e_3$, then $e_2 =_\alpha e_3$

All other cases are similar to the previous two.