

## Syntax

Action Fields	$f_i ::= f_{a_1} \mid \cdots \mid f_{a_k}$
Result Fields	$f_r ::= f_{r_1} \mid \cdots \mid f_{r_k}$
Fields	$f ::= f_i \mid f_r$
Actions	$i ::= \{f_{a_1} = v_{a_1}, \dots, f_{a_k} = v_{a_k}\}$
Results	$r ::= \{f_{r_1} = v_{r_1}, \dots, f_{r_k} = v_{r_k}\}$
Predicates	$a, b ::= 0 \quad \textit{Identity}$ $\mid 1 \quad \textit{False}$ $\mid f = n \quad \textit{Test}$ $\mid a + b \quad \textit{Sum}$ $\mid a \cdot b \quad \textit{Product}$ $\mid \neg a \quad \textit{Negation}$
Policies	$p, q ::= a \quad \textit{Test}$ $\mid \textit{act}(p) \quad \textit{Slice Actions}$ $\mid \textit{res}(p) \quad \textit{Slice Results}$ $\mid \textit{inj}_i \quad \textit{Injection Action}$ $\mid \textit{inj}_r \quad \textit{Injection Result}$ $\mid f \leftarrow n \quad \textit{Update}$ $\mid p + q \quad \textit{Choice}$ $\mid p \cdot q \quad \textit{Sequential Concatenation}$ $\mid p^* \quad \textit{Kleene Star}$

## Semantics

$$\begin{aligned}
\llbracket \cdot \rrbracket &: P(A) \times P(R) \rightarrow P(A) \times P(R) \\
\llbracket 0 \rrbracket(-, -) &\triangleq (\emptyset, \emptyset) \\
\llbracket 1 \rrbracket(is, rs) &\triangleq (is, rs) \\
\llbracket f = n \rrbracket(is, rs) &\triangleq (\text{filter } (f = n) \text{ } is, \text{filter } (f = n) \text{ } rs) \\
\llbracket a + b \rrbracket(is, rs) &\triangleq \text{let } (is_a, rs_a) = \llbracket a \rrbracket(is, rs) \\
&\quad \text{let } (is_b, rs_b) = \llbracket b \rrbracket(is, rs) \\
&\quad (is_a \cup is_b, rs_a \cup rs_b) \\
\llbracket a \cdot b \rrbracket(is, rs) &\triangleq \text{let } (is_a, rs_a) = \llbracket a \rrbracket(is, rs) \\
&\quad \text{let } (is_b, rs_b) = \llbracket b \rrbracket(is, rs) \\
&\quad (is_a \cap is_b, rs_a \cap rs_b) \\
\llbracket \neg a \rrbracket(is, rs) &\triangleq \text{let } (is', rs') = \llbracket a \rrbracket(is, rs) \\
&\quad (is - is', rs - rs') \\
\llbracket act(p) \rrbracket(is, rs) &\triangleq \text{let } (is', rs') = \llbracket p \rrbracket(is, rs) \text{ in } (is', rs) \\
\llbracket res(p) \rrbracket(is, rs) &\triangleq \text{let } (is', rs') = \llbracket p \rrbracket(is, rs) \text{ in } (is, rs') \\
\llbracket inj_i(i) \rrbracket(is, rs) &\triangleq (\{i\} \cup is, rs) \\
\llbracket inj_r(r) \rrbracket(is, rs) &\triangleq (is, \{r\} \cup rs) \\
\llbracket f \leftarrow n \rrbracket(is, rs) &\triangleq (\text{map } (f \leftarrow n) \text{ } is, \text{map } (f \leftarrow n) \text{ } rs) \\
\llbracket p + q \rrbracket(is, rs) &\triangleq \text{let } (is_p, rs_p) = \llbracket p \rrbracket(is, rs) \\
&\quad \text{let } (is_q, rs_q) = \llbracket q \rrbracket(is, rs) \\
&\quad (is_p \cup is_q, rs_p \cup rs_q) \\
&\quad \dots\dots\dots \\
\llbracket p \cdot q \rrbracket(is, rs) &\triangleq \text{let } (is', rs') = \llbracket p \rrbracket(is, rs) \\
&\quad \llbracket q \rrbracket(is', rs') \\
\llbracket p^* \rrbracket \varphi &\triangleq ()?
\end{aligned}$$