Example Transitions

\begin{align*}
\text{sitting} &\_\text{at}(\text{rob, o109}). \\
\text{sitting} &\_\text{at}(\text{parcel, storage}). \\
\text{sitting} &\_\text{at}(\text{k1, mail}). \\
\text{move}(\text{rob, o109, storage}) &\rightarrow \\
\text{sitting} &\_\text{at}(\text{rob, storage}). \\
\text{sitting} &\_\text{at}(\text{parcel, storage}). \\
\text{sitting} &\_\text{at}(\text{k1, mail}). \\
\text{pickup}(\text{rob, parcel}) &\rightarrow \\
\text{sitting} &\_\text{at}(\text{rob, storage}). \\
\text{carrying}(\text{rob, parcel}). \\
\text{sitting} &\_\text{at}(\text{k1, mail}).
\end{align*}

MSO applied to STRIPS

\(\forall x \forall y \ S(x, y) \supseteq \bigvee_{\alpha \in \text{Act}} (\text{Pre}_\alpha(x) \land \text{Add}_\alpha(y) \land \text{Del}_\alpha(y) \land \text{Inertia}_\alpha(x, y))\)

\[
\begin{align*}
\text{Pre}_\alpha(x) &:= \bigwedge \{P_a(x) \mid a \in \text{precondition-list}(\alpha)\} \\
\text{Add}_\alpha(y) &:= \bigwedge \{P_a(y) \mid a \in \text{add-list}(\alpha)\} \\
\text{Del}_\alpha(y) &:= \bigwedge \{\neg P_a(y) \mid a \in \text{delete-list}(\alpha) - \text{add list}(\alpha)\}
\end{align*}
\]

\[
\text{Inertia}_\alpha(x, y) := \bigwedge \{ (P_a(x) \equiv P_a(y)) \mid a \in \Sigma - (\text{add-list}(\alpha) \cup \text{delete-list}(\alpha)) \}
\]

no action other than \(\alpha\) executes between \(x\) and \(y\)
MSO and finite automata: an example with $\Sigma = \{a, b\}$

![Automaton diagram](image)

Accepting runs:
$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \in F$
$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_2 \in F$

... 

regexp $ab^*b$ denoting the regular language $\{ab, abb, \ldots\}$

MSO-sentence:

$$\exists x \exists y \ P_a(x) \land P_b(y) \land (\forall z)(\neg S(z, x) \land (z = x \lor P_b(z)))$$

Strings in $\Sigma^+$ as MSO$_\Sigma$-models

$MSO_\Sigma = \text{monadic second-order logic with}$

- a binary relation symbol (successor) $S$
- a unary relation symbol $P_\sigma$ for each symbol $\sigma \in \Sigma$

$abbc \leadsto MSO_\Sigma$-model

$$\langle D_4, S_4, [P_a], [P_b], [P_c] \rangle$$

where

$D_4 := \{1, 2, 3, 4\}$
$S_4 := \{(1, 2), (2, 3), (3, 4)\}$
$[P_a] := \{1\}$
$[P_b] := \{2, 3\}$
$[P_c] := \{4\}$

$$(\exists x_1)(\exists x_2)(\exists x_3)(\exists x_4) \ P_a(x_1) \land P_b(x_2) \land P_b(x_3) \land P_c(x_4) \land S(x_1, x_2) \land S(x_2, x_3) \land S(x_3, x_4) \land \neg(\exists x)(S(x, x_1) \lor S(x_4, x))$$
Büchi, Elgot, Trakhtenbrot: Reg = MSO

For any finite alphabet $\Sigma$,

regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in MSO}_\Sigma$

(i) For every regular $L \subseteq \Sigma^+$, there is an MSO$_\Sigma$-sentence $\varphi$ s.t.

$L = \{ s \in \Sigma^+ \mid \text{Mod}(s) \models \varphi \}.$

(ii) For every MSO$_\Sigma$-sentence $\varphi$,

$\{ s \in \Sigma^+ \mid \text{Mod}(s) \models \varphi \}$ is regular.

N.B. (ii) is proved by induction on MSO$_\Sigma$-formulas $\varphi$ with free variables interpreted in an MSO$_\Sigma$-model $M$ by a function $f$

$M, f \models \varphi$

From tuples-as-symbols to sets-as-symbols

Recall: $abacb + \text{assignment } f(X) = \{1, 3, 4\}, f(x) = 2$

$$
\begin{pmatrix}
  a \\
  1 \\
  0 \\
\end{pmatrix}
\begin{pmatrix}
  b \\
  0 \\
  1 \\
\end{pmatrix}
\begin{pmatrix}
  a \\
  1 \\
  0 \\
\end{pmatrix}
\begin{pmatrix}
  c \\
  1 \\
  0 \\
\end{pmatrix}
\begin{pmatrix}
  b \\
  0 \\
  0 \\
\end{pmatrix}
$$

with 2nd component (row) for $X$, and 3rd for $x$.

$$
\begin{array}{c|c|c|c|c|c|c}
  a & X & b & x & a & X & c & X & b \\
\end{array}
\in \text{Pow}\{a, b, c, x, X\}^*
$$

boxing a set rather than putting it between curly braces.