THE FSM DOMAIN

main contributors
Alain Girault
Bilung Lee
Edward Lee

other contributors
Wan-Teh Chang
Stephen Edwards

OUTLINE

1. Motivations

2. Specification of FSMs

3. Embedding of FSMs:
   • into the SDF domain
   • into the DDF domain
   • into the SR domain

4. Adding hierarchy

5. Comparison with ARGOS
**SPECIFICATION**

\[ \langle I, O, Q, q_0, T \rangle \]

where:

- \( I \) is a set of input signals,
- \( O \) is a set of output signals,
- \( Q \) is a set of states,
- \( q_0 \in Q \) is the initial state, and
- \( T \) is a set of transitions of the form guard\_part/action\_part

**MOTIVATIONS**

**Control:** clean control structure in PTOLEMY

**Heterogeneity:** keep the usual PTOLEMY philosophy: a new domain

**DOMAIN of VALUES**

Each signal has 3 states: “unknown” (\( \perp \)), “absent” (\( \varepsilon \)) and “present with value \( v \)” (\( v \))
**EXAMPLE**

```
\[
\begin{align*}
1 & \quad a = 1/c(\text{default}) \\
2 & \quad \alpha > 1 \wedge b = 3/ \\
2 & \quad a \neq 1/ \\
3 & \quad a \neq 1/c(-2) \\
\end{align*}
\]
```

<table>
<thead>
<tr>
<th>current state</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>...</td>
</tr>
<tr>
<td>( b )</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>-4</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>next state</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
</tr>
<tr>
<td>( c )</td>
<td>( \varepsilon )</td>
<td>4</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>( \varepsilon )</td>
<td>( \varepsilon )</td>
<td>4</td>
<td>...</td>
</tr>
</tbody>
</table>

**EMBEDDING into SDF**

SDF semantics: a token **must** be produced on transition from state 1 to state 2 \( \Rightarrow \) label \( a \neq 1/c(\text{default}) \)
NON HOMOGENEOUS CASE

SDF semantics: two tokens **must** be produced on transition from state 1 to state 2 ⇒ label \( a \neq 1/c(2) \) \( c(\text{default}) \)

EMBEDDING into DDF

DDF semantics: additional tokens are **never** produced
Need to consume a number of token sufficient to evaluate all the guards
EMBEDDING into SR

Handling instantaneous loops:
1. partial order on the values of the wires
2. block functions must be monotonic w.r.t. this partial order

Partial evaluation algorithm for FSMs

PARTIAL EVALUATION

state 1: functions $O_b^1$ and $O_c^1$

\[
\begin{array}{c|ccc|c}
  a & -\infty & 1 & +\infty \\
  b & 2 & 2 & 2 \\
  c & \varepsilon & 4 & \varepsilon \\
\end{array}
\]

For any state $s$ and any output $o$:

\[
\begin{cases}
  \text{if } \exists v : \forall x, O_o^s(x) = v \text{ then } O_o^s(\perp) = v \\
  \text{else } O_o^s(\perp) = \perp
\end{cases}
\]

state 1: extended functions $O_b^1$ and $O_c^1$

\[
\begin{array}{c|ccc|c}
  a & -\infty & 1 & +\infty & \perp \\
  b & 2 & 2 & 2 & 2 \\
  c & \varepsilon & 4 & \varepsilon & \perp \\
\end{array}
\]
EXECUTION EXAMPLE

1. start with \((a, b, c) = (\bot, \bot, \bot)\)
2. execute \(B\): \(a = O_a^1(\bot, \bot) = \bot\)
3. execute \(A\): \(b = O_b^1(\bot) = 2\) and \(c = O_c^1(\bot) = \bot\)
4. execute \(B\): \(a = O_a^1(2, \bot) = 1\)
5. execute \(A\): \(b = O_b^1(1) = 2\) and \(c = O_c^1(1) = 4\)

Theorem: The extended output functions are monotonic w.r.t. to the partial order

ADDING HIERARCHY

Executing an FSM: execute the current state & select and fire a transition (same set of inputs, distinct sets of outputs, internal events)

Executing a data-flow network: perform a complete execution cycle of the network (hierarchical blocks are executed according to their respective domain)

Executing an SR network: find the behavior (least fixed point) of the network (each block must compute a monotonic function)
The SR scheduler will generate $A.B.A$ or $B.A.B$

In both cases, the network goes from state $(A_1, B_1)$ to $(A_2, B_2)$