Causality Interfaces and Compositional Causality Analysis

Edited and presented by
Rachel Zhou
UC Berkeley

Chess Review
November 21, 2005
Berkeley, CA
Actor-Oriented Design

- Actors are in charge of their own actions.
- Actors interact with each other by exchanging data through ports.
- The patterns of interactions between actors are called *model of computation*.

```
<table>
<thead>
<tr>
<th>class name</th>
<th>data</th>
<th>methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>actor name</th>
<th>data (state)</th>
<th>parameters</th>
<th>ports</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```

Object-Oriented

Actor-Oriented

"Causality Interfaces", Zhou

Chess Review, Nov. 21, 2005
Causality Interfaces

- A special family of behavioral interfaces that capture the causality properties of actors and their connections.

- Useful for determining constructive semantics of compositions under certain models of computation.
  - Synchronous Languages
  - Discrete-Event Models
  - Synchronous Dataflow
Causality Interfaces as Functions

- A function that maps a pair of ports to an element in the dependency set $D$.
  - For an actor:
    \[ \delta : P_i \times P_o \rightarrow D \]
  - For a connector:
    \[ \delta : P_o \times P_i \rightarrow D \]
Dependency Set

- Dependency set $D$ is an ordered set with two binary operators $\oplus$ and $\otimes$.

- **Synchronous Languages**
  - $D = \{\text{true, false}\}$, false $< \text{true}$.
  - $\oplus$ is *logical and*, $\otimes$ is *logical or*.

- **Discrete-Event Models**
  - $D = \mathbb{R}_+ \cup \{\infty\}$, “$<$” as numerical ordering.
  - $\oplus$ is the *minimum* function, $\otimes$ is addition.
Compositional Analysis

- Use $\otimes$ operator for serial compositions and $\oplus$ operator for parallel compositions.

\[
\delta(p_1, p_4) = \delta(p_1, p_5) \otimes (\delta(p_5, p_2) \otimes \delta(p_2, p_4)) \\
\oplus \delta(p_5, p_7) \otimes \delta(p_7, p_6) \otimes \delta(p_6, p_3) \otimes \delta(p_3, p_4))
\]
Determining Constructive Semantics

- A constructive behavior exists if there exists no port that has an immediate dependency on itself.

- Synchronous Languages
  - $\forall p \in P, \delta(p, p) > false$

- Discrete-Event Models
  - $\forall p \in P, \delta(p, p) > 0$
Conclusions

- An interface theory for causality interfaces of actors and their composition.
- An algebraic procedure to determine whether a composition has a constructive semantics.
- Applied to synchronous languages and discrete-event models.
- On-going work: synchronous dataflow, continuous time, rendezvous.
- Joint work with Prof. Edward A. Lee and Haiyang Zheng