Causality Interfaces and Compositional Causality Analysis

Edward A. Lee Haiyang Zheng Ye Zhou

{eal,hyzheng,zhouye}@eecs.berkeley.edu
Center for Hybrid and Embedded Software Systems (CHESS)
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley

Workshop on Foundations of Interface Technologies
Satellite to CONCUR 2005

Objectives

- Focus on concurrent software components:
  - "actors", which are in charge of their actions ✓
  - "objects", which are acted upon
- Develop *causality interfaces* for actors in actor-oriented design and an *algebra* for composing these interfaces.
- Deploy causality interfaces to determine existence and uniqueness of behaviors of compositions of actors under certain *models of computation*.

Outline

1. Actor-Oriented Design
2. Causality
   - The Tagged Signal Model
   - Composition of Actors
3. Causality Interfaces
   - Definitions
   - Compositional Analysis
   - Dynamic Causality Interfaces

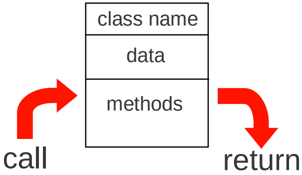
Outline

1. Actor-Oriented Design
2. Causality
   - The Tagged Signal Model
   - Composition of Actors
3. Causality Interfaces
   - Definitions
   - Compositional Analysis
   - Dynamic Causality Interfaces

Actor-Oriented Design

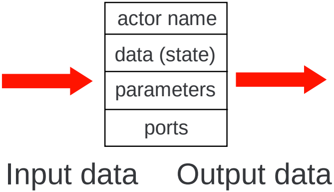
Offers complementary approaches (to object-oriented design) for modeling compositions of concurrent components.

Object orientation:



what flows through an object is sequential control.

Actor orientation:



what flow through an actor are streams of data.

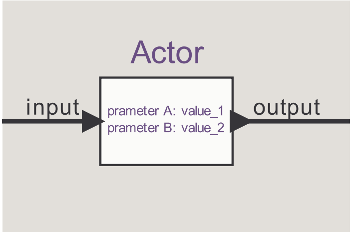
Models of Computation

The patterns of interactions between concurrent components are called *models of computation*.

- Classical ”actor model” [Agha, 1990][Hewitt, 1977]
- Synchronous languages [Benveniste, Berry, 1991] ✓
- Discrete event models [Cassandras, 1993] ✓
- Dataflow models ✓
  - Kahn-MacQueen process networks [Kahn, MacQueen, 1977]
  - Dennis-style dataflow [Dennis, 1974]

Interfaces for Actors

Analogous to the type signatures of an abstract data type in object-oriented design:



Static structure interface:

- ports
- parameters
- their type constraints

Richer Interfaces for Actors

- Interaction semantics [Talcott, 1996]
- Behavioral subtyping [Liskov, Wing, 1999]
- Interface theories [de Alfaro, Henzinger, 2001]
- Behavioral type systems [Lee, Xiong, 2004]
- Abstract behavioral types [Arbab, 2005]

Note

These interface theories cover behavioral properties,
not just static structure.

Causality Interfaces for Actors

- A special family of behavioral interfaces that capture the causality properties of actors, which reflect the dependency of particular outputs having on particular inputs.
- Useful for determining existence and uniqueness of behaviors of compositions under certain models of computation.
- Related to causalities properties of stream functions in [Broy, 1995]

Outline

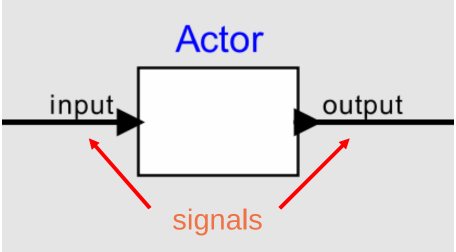
1. Actor-Oriented Design
2. Causality
   - The Tagged Signal Model
   - Composition of Actors
3. Causality Interfaces
   - Definitions
   - Compositional Analysis
   - Dynamic Causality Interfaces

The Tagged Signal Model

- We use a formal and *denotational* framework for studying and comparing actor-oriented models of computation, which is called the Tagged Signal Model [Lee, Sangiovanni-Vincentelli, 1998].

- Related to
  - Interaction semantics [Talcott, 1996]
  - Behavioral types [Arbab, 2005]

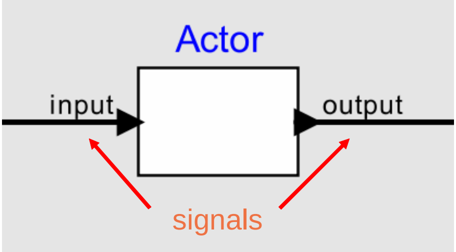
Actors, Formally



Event e: (t, v) pair, where $t \in \mathcal{T}$ and $v \in \mathcal{V}$.
$\mathcal{T}$ is a partially or totally ordered set. $\mathcal{V}$ is a set of possible values.

$$e \in \mathcal{E} = \mathcal{T} \times \mathcal{V}, \quad \text{where } \mathcal{E} := \text{the set of all events.}$$

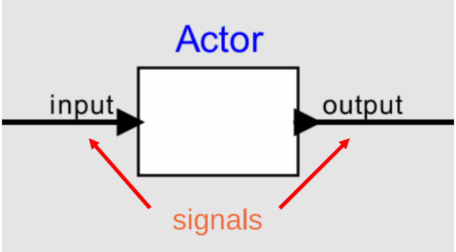
Actors, Formally



Signal s: a set of events
For example, a token stream of infinite length

$s \subset \mathcal{E}$, $S :=$ the set of all signals.
$s \in S = \mathcal{P}(\mathcal{E})$, the power set.

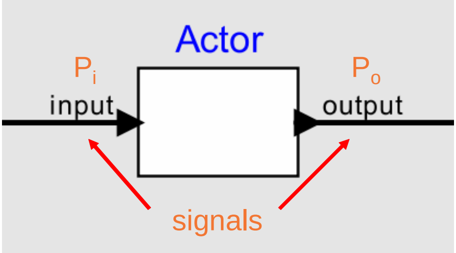
Actors, Formally



An actor operates on signals.
Behavior σ: a total function from ports to signals
Actor *a*: a set of behaviors

$$\sigma : P \to S, \; a \subset [P_a \to S].$$

Actors, Formally

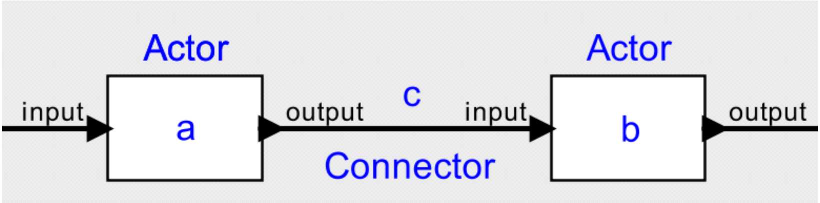


Functional actor:

- has input ports P_i and output ports P_o
- defines a function from input behaviors to output behaviors

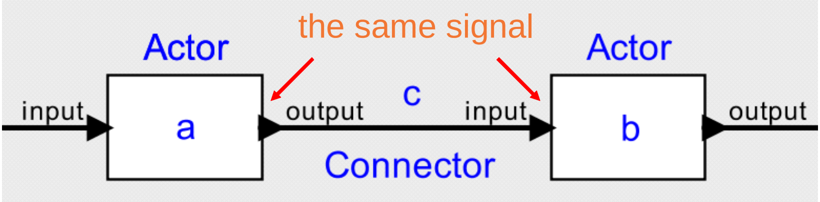
$$F_a \colon [P_i \to S] \rightharpoonup [P_o \to S].$$

Composition of Actors



Actors are composed by connecting ports with connectors.

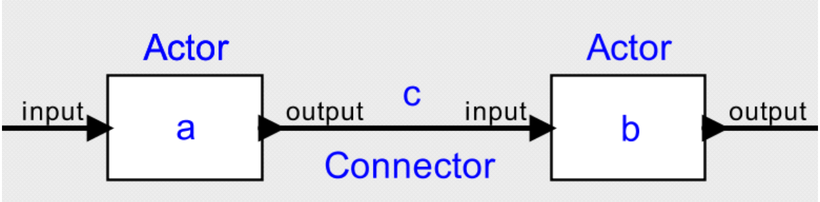
Composition of Actors



A *connector* c between ports P_c is the constraint on ports.
It is also a set of behaviors.

$$c \subset [P_c \to S]$$
$$\forall \sigma \in c, \exists s \in S \; s.t. \; \forall \; p \in P_c, \;\; \sigma(p) = s$$

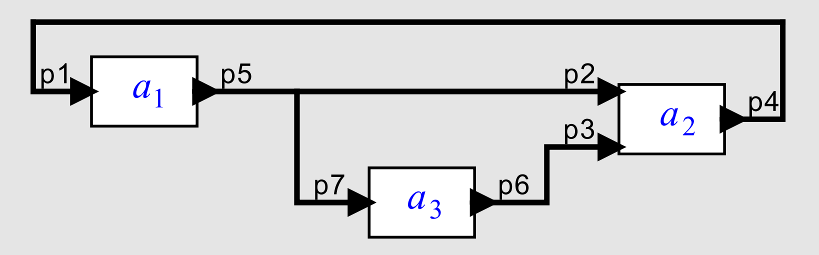
Composition of Actors



A *composition behavior* is the intersection of the actor and connector behaviors.

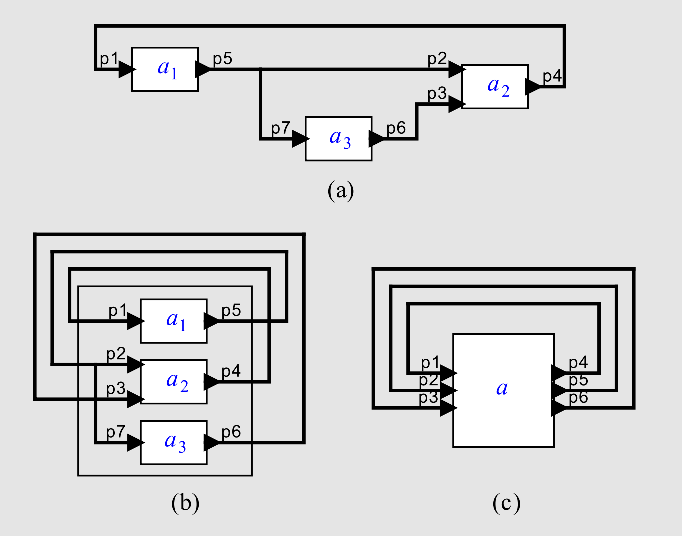
$$a \wedge b \wedge c = \{\sigma \mid \sigma \downarrow_{P_a} \in a \quad \text{and} \quad \sigma \downarrow_{P_b} \in b \quad \text{and} \quad \sigma \downarrow_{P_c} \in c\}.$$

Example of Composition

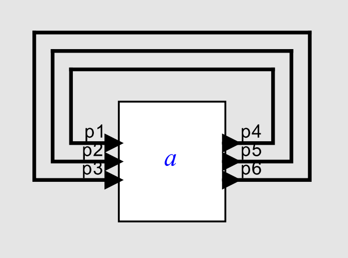


- series composition
- parallel composition
- feedback composition

Feedback Composition



Fixed Point Behavior



Recall that F_a denotes the behaviors of actor a:

$$F_a \colon [P_i \to S] \rightharpoonup [P_o \to S], \; P_i \cup P_o = P_a$$

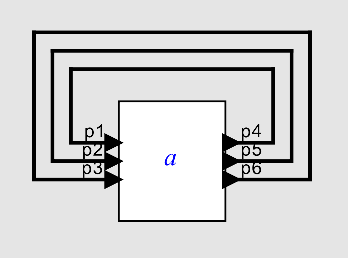
The input behavior of the left feedback composition is a function:

$$f \colon \{p1, p2, p3\} \to S \, .$$

The above behavior is a fixed point of F_a. That is,

$$F_a(f) = f \, .$$

Questions about Fixed Point Behavior



$$F_a(f) = f$$

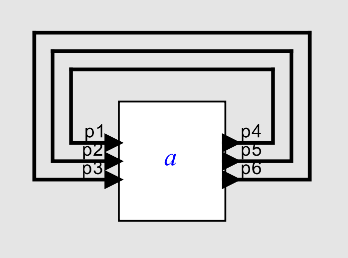
Problem (1)

Does such a fixed point f exist?

Problem (2)

Is the fixed point f unique?

Causality is the Key



- Synchronous Languages
  No causality loop.
- Discrete-Event Models
  Contraction map.
- Dataflow Models
  No deadlock.

The key is to determine the causality properties of actor *a* from the causality properties of its components, the actors and connectors contained inside.

Outline

1 Actor-Oriented Design

2 Causality
- The Tagged Signal Model
- Composition of Actors

3 Causality Interfaces
- Definitions
- Compositional Analysis
- Dynamic Causality Interfaces

Causality Interfaces

- D is an ordered set with elements called *dependencies*.
- A *causality interface* for an actor a with input ports P_i and output ports P_o is a function

$$\delta_a \colon P_i \times P_o \to D$$

- A *causality interface* for a connector c that links output ports P_o to input ports P_i is a function

$$\delta_c \colon P_o \times P_i \to D$$

Interpretation

$\delta(p_1, p_2)$ denotes the dependency that port p_2 has on p_1.

Dependency Set D as an Algebra

D is an ordered set with two binary operations $\otimes$ and $\oplus$ that satisfy the axioms given below.

- (Associativity)

$$\forall d_1, d_2, d_3 \in D, \quad (d_1 \oplus d_2) \oplus d_3 = d_1 \oplus (d_2 \oplus d_3),$$
$$\forall d_1, d_2, d_3 \in D, \quad (d_1 \otimes d_2) \otimes d_3 = d_1 \otimes (d_2 \otimes d_3).$$

- (Commutativity)

$$\forall d_1, d_2 \in D, \quad d_1 \oplus d_2 = d_2 \oplus d_1,$$
$$\forall d_1, d_2 \in D, \quad d_1 \otimes d_2 = d_2 \otimes d_1.$$

- (Distributivity)

$$\forall d_1, d_2, d_3 \in D, \quad d_1 \otimes (d_2 \oplus d_3) = (d_1 \otimes d_2) \oplus (d_1 \otimes d_3).$$

- (Null and Identity Elements) $\exists\, \mathbf{0}, \mathbf{1} \in D$, such that $\forall\, d \in D$,

$$d \oplus \mathbf{0} = d, \quad d \oplus d = d,$$
$$d \otimes \mathbf{0} = \mathbf{0}, \quad d \otimes \mathbf{1} = d.$$

Examples of Dependency Sets

- Boolean Dependencies
  - $D = \{\text{true}, \text{false}\}$, where false $<$ true, $\mathbf{0}$ = true, and $\mathbf{1}$ = false.
  - $\oplus$ is *logical and*, $\otimes$ is *logical or*.

$\delta(p_1, p_2) = \text{false}$ means that P_2 depends immediately on P_1.

- Weighted Dependencies [1]
  - $D = \mathbb{R}_+ \cup \{\infty\}$, where $<$ is as usual, $\mathbf{0} = \infty$, and $\mathbf{1} = 0$.
  - $\oplus$ is the *minimum* function, $\otimes$ is addition.

$\delta(p_1, p_2) = 0$ means that P_2 depends immediately on P_1.

The $\otimes$ identity, **1**, means immediate dependency.

[1] This set is also called a min-plus algebra [Baccelli, 1992].

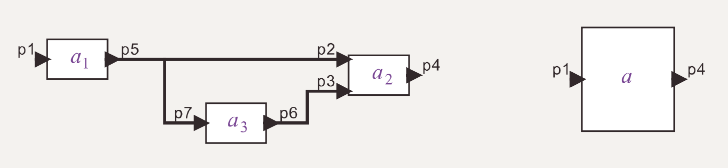
Compositional Analysis

Problem

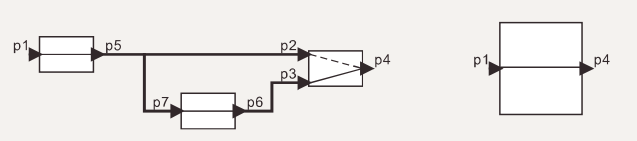
Given a set of actors, a set of connectors, and their causality interfaces, determine the causality interface of their composition.

In the following example, we want to determine the function:

$$\delta_a \colon \{p1\} \times \{p4\} \to D.$$



Solving the Example



Causalities interfaces of actors:

$$\delta_1(p1, p5) = \mathbf{1} \quad \delta_3(p7, p6) = \mathbf{1}$$
$$\delta_2(p3, p4) = \mathbf{1} \quad \delta_2(p2, p4) \neq \mathbf{1}$$

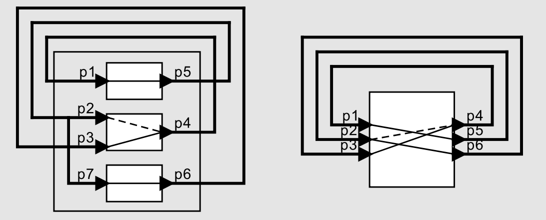
Causalities interfaces of connectors:

$$\forall p \in P_o,\ p' \in P_i,\ P_i \cup P_o = P_c,$$
$$\delta_c(p, p') = \mathbf{1}$$

Combine the interfaces of connectors and actors using the $\otimes$ operator for *series* compositions and the $\oplus$ operator for *parallel* compositions.

$$\begin{aligned}\delta_a(p1, p4) &= \delta_1(p1, p5) \otimes (\delta_c(p5, p2) \otimes \delta_2(p2, p4) \\ &\qquad \oplus \delta_c(p5, p7) \otimes \delta_3(p7, p6) \otimes \delta_c(p6, p3) \otimes \delta_2(p3, p4)) \\ &= \mathbf{1}\end{aligned}$$

Existence and Uniqueness of Behaviors

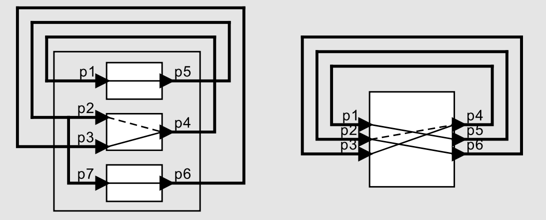


Problem

Given the causality properties of a composition of a set of actors and connectors, determine the existence and uniqueness of the behaviors of the composition.

A unique behavior exists if there exists no port that has an immediate dependency ($\otimes$ identity, "**1**") on itself.

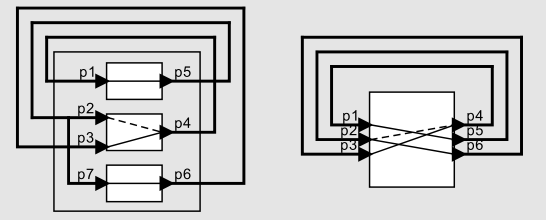
Application to Synchronous Languages



- Apply boolean dependencies, where **0** $=$ true and **1** $=$ false. Note that $\oplus$ is *logical and*, $\otimes$ is *logical or*.
- A unique behavior exists if there exists no port that has a dependency on itself with value "false".

$$\begin{aligned}\delta(p4, p4) &= \delta_c(p4, p1) \otimes \delta_a(p1, p5) \otimes \delta_c(p5, p2) \\ &\quad \otimes(\delta_a(p2, p4) \oplus \delta_a(p2, p6) \otimes \delta_c(p6, p3) \otimes \delta_a(p3, p4)) \\ &= \text{false} \otimes \text{false} \otimes \text{false} \otimes (\text{true} \oplus \text{false} \otimes \text{false} \otimes \text{false}) \\ &= \text{false}.\end{aligned}$$

Application to Discrete-Event Models



- Apply weighted dependencies, where $\mathbf{0} = \infty$ and $\mathbf{1} = 0$. Note that $\oplus$ is the minimum function, and $\otimes$ is addition.
- A unique behavior exists if there exists no port that has a dependency on itself with value "0".

$$\begin{aligned}
\delta(p4, p4) &= \delta_c(p4, p1) \otimes \delta_a(p1, p5) \otimes \delta_c(p5, p2) \\
&\quad \otimes(\delta_a(p2, p4) \oplus \delta_a(p2, p6) \otimes \delta_c(p6, p3) \otimes \delta_a(p3, p4)) \\
&= 0 \otimes 0 \otimes 0 \otimes (2.0 \oplus 0 \otimes 0 \otimes 0) \\
&= 0.
\end{aligned}$$

Dynamic Causality Interfaces

- Causality interfaces of a modal mode may vary due to the change of its internal states. Let X denote the set of states, then the causality interfaces are given by a function

$$\delta' : P_i \times P_o \times X \rightarrow D \, .$$

- A simple static and conservative analysis approach:
for an input port $p_i \in P_i$ and an output port $p_o \in P_o$ of an actor,

$$\delta(p_i, p_o) = \bigoplus_{x \in X} \delta'(p_i, p_o, x) \, .$$

- This simple method may be too conservative in practice. A more precise static analysis may not be always available.

Summary

- An interface theory for causality interfaces of actors and their composition.
- Applications to determining existence and uniqueness of behaviors for synchronous languages and discrete-event models.