CIS 4930/6930: Principles of Cyber-Physical Systems

Chapter 5: Composition of State Machines

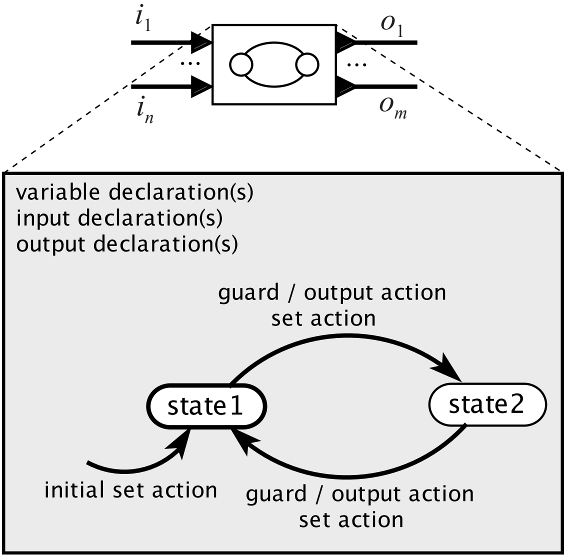
Hao Zheng

Department of Computer Science and Engineering
University of South Florida

Introduction

- State machines are useful for modeling system behaviors.
- How to represent a system for systematic analysis?
- Complete systems though often have a very large state space.
- Can represent complicated system as composition of simpler systems.
  - Modular approaches are always needed to handle large complex problems.
- Care must be taken though as the same **syntax** (model notation) often has different **semantics** (meaning).

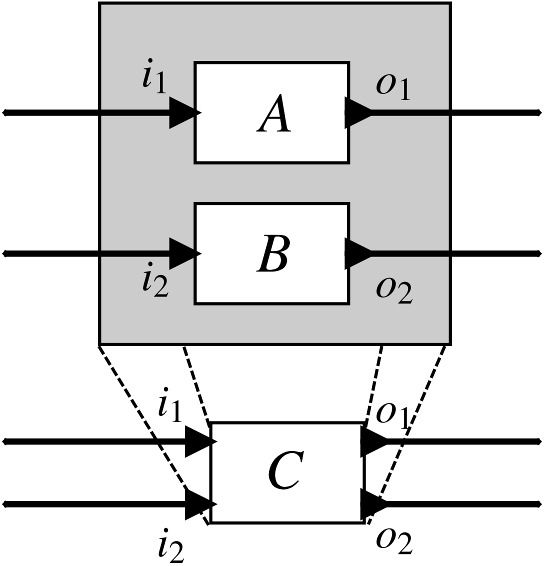
Actor Model and Extended SM Notation



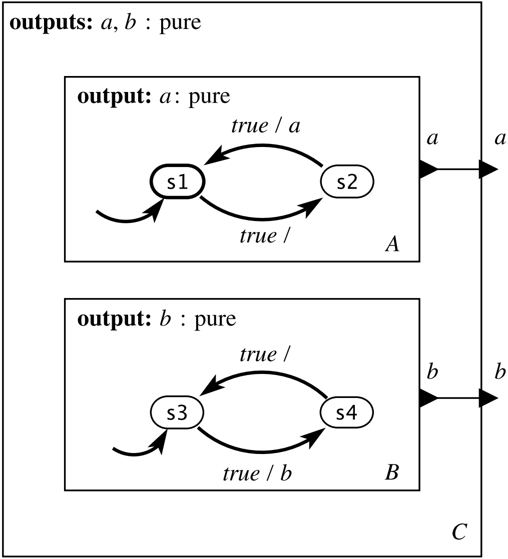
Overview

- Side-by-side **synchronous** composition (simultaneous reactions).
- Side-by-side **asynchronous** composition (independent reactions).
- Communication through shared variables.
- Cascade (serial) composition.
- General composition that combines side-by-side and cascade.
- Hierarchical state machines.

Side-by-side Composition



Side-by-side Composition Example



Synchronous Side-by-side Composition

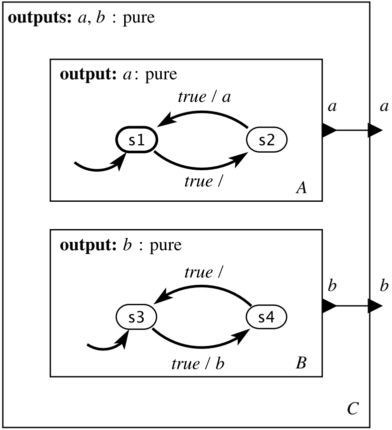
$$\begin{aligned} A &= (\text{States}_A, \text{Inputs}_A, \text{Outputs}_A, \text{update}_A, \text{initialState}_A) \\ B &= (\text{States}_B, \text{Inputs}_B, \text{Outputs}_B, \text{update}_B, \text{initialState}_B) \end{aligned}$$

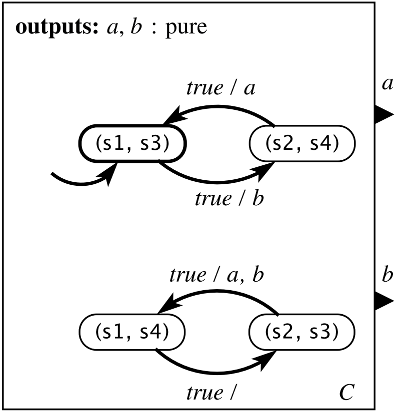
The synchronous side-by-side composition C is given by:

$$\begin{aligned} \text{States}_C &= \text{States}_A \times \text{States}_B \\ \text{Inputs}_C &= \text{Inputs}_A \times \text{Inputs}_B \\ \text{Outputs}_C &= \text{Outputs}_A \times \text{Outputs}_B \\ \text{initialState}_C &= (\text{initialState}_A, \text{initialState}_B) \\ \text{update}_C((s_A, s_B), (i_A, i_B)) &= ((s'_A, s'_B), (o_A, o_B)) \\ \text{where} & \\ (s'_A, o_A) &= \text{update}_A(s_A, i_A) \\ (s'_B, o_B) &= \text{update}_B(s_B, i_B) \end{aligned}$$

for all $s_A \in \text{States}_A$, $s_B \in \text{States}_B$, $i_A \in \text{Inputs}_A$, and $i_B \in \text{Inputs}_B$.

Synchronous Side-by-side Composition





Asynchronous Side-by-side Composition

- **Semantics 1**: a reaction of C is a reaction of one of A or B, where the choice is nondeterministic (**interleaving semantics**).

$$\text{update}_C((s_A, s_B), (i_A, i_B)) \quad = \quad ((s'_A, s'_B), (o'_A, o'_B))$$

where either

$$(s'_A, o'_A) = \text{update}_A(s_A, i_A) \text{ and } s'_B = s_B \text{ and } o'_B = \text{absent}$$

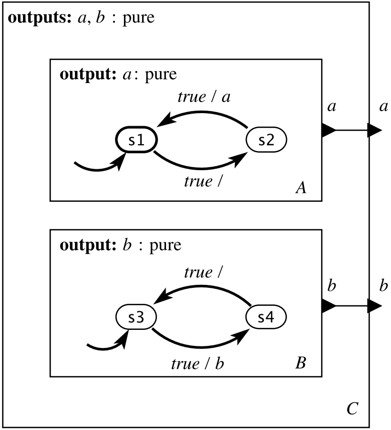
or

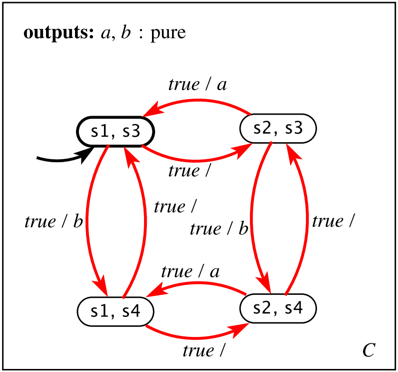
$$(s'_B, o'_B) = \text{update}_B(s_B, i_B) \text{ and } s'_A = s_A \text{ and } o'_A = \text{absent}$$

for all $s_A \in \text{States}_A$, $s_B \in \text{States}_B$, $i_A \in \text{Inputs}_A$, and $i_B \in \text{Inputs}_B$.

- **Semantics 2**: a reaction of C is a reaction of A, B, or both A and B, where the choice is nondeterministic.

Asynchronous Side-by-side Composition

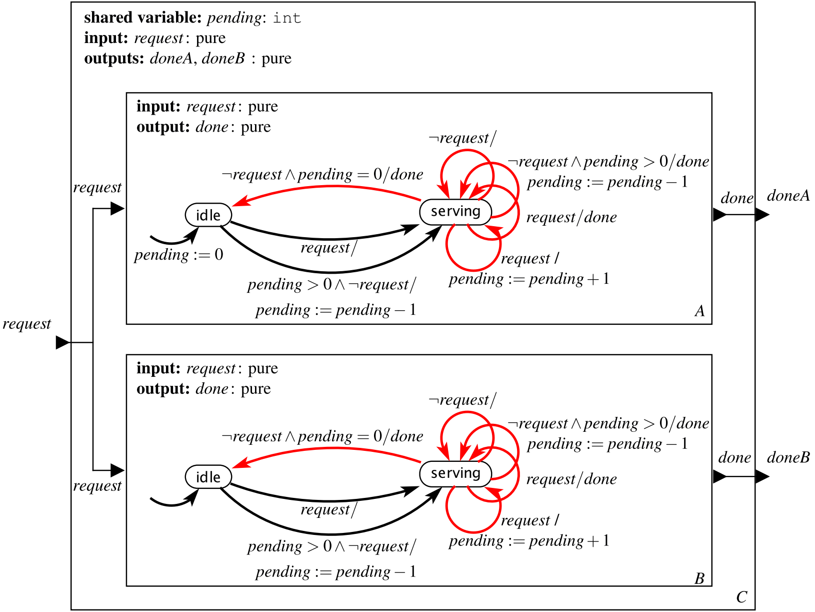




Shared Variables

- Extended state machines have variables that are read/written by transitions.
- These can be shared when composing state machines.
- Useful when modeling interrupts and threads.
- Ensuring correct semantics though can be challenging.

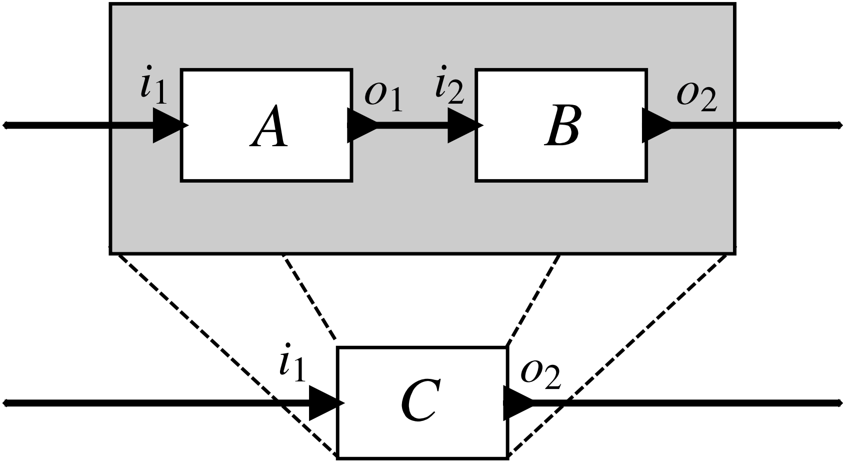
Shared Task Queue Example



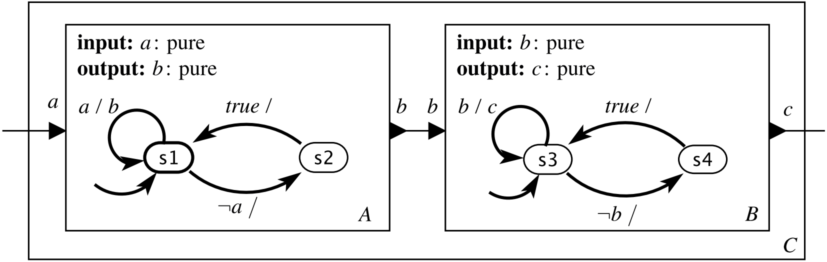
Semantic Subtleties

- Interleaving semantics makes accesses to the shared variable atomic.
  - Tricky to satisfy in practice.
- What if both machines react or machines use synchronous semantics?
  - Leads to non-deterministic outputs.

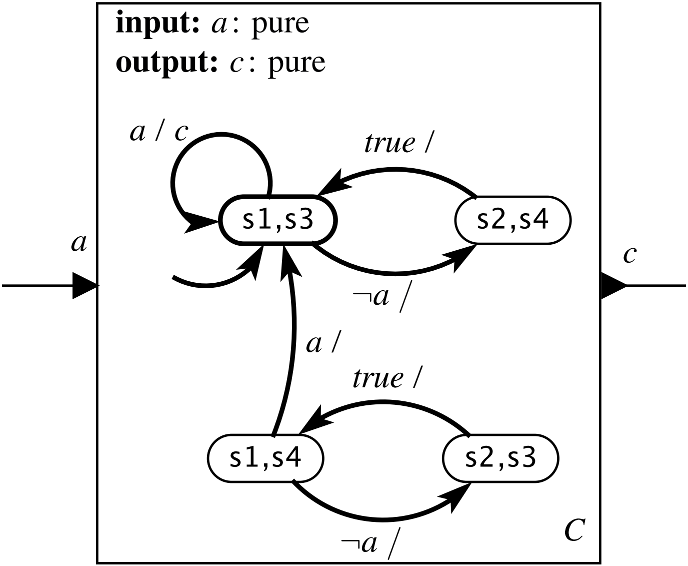
Cascade Composition



Cascade Composition Example

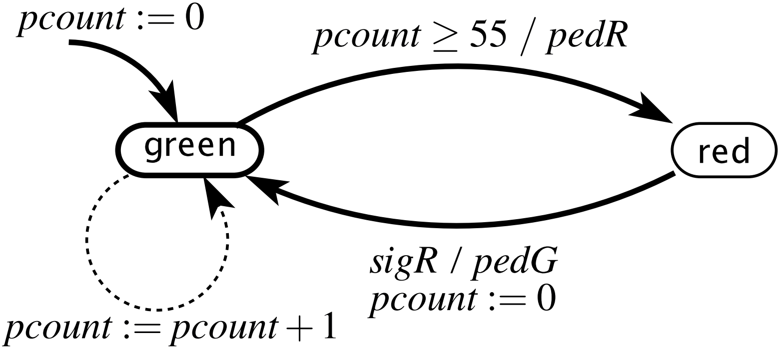


Synchronous Cascade Composition Example



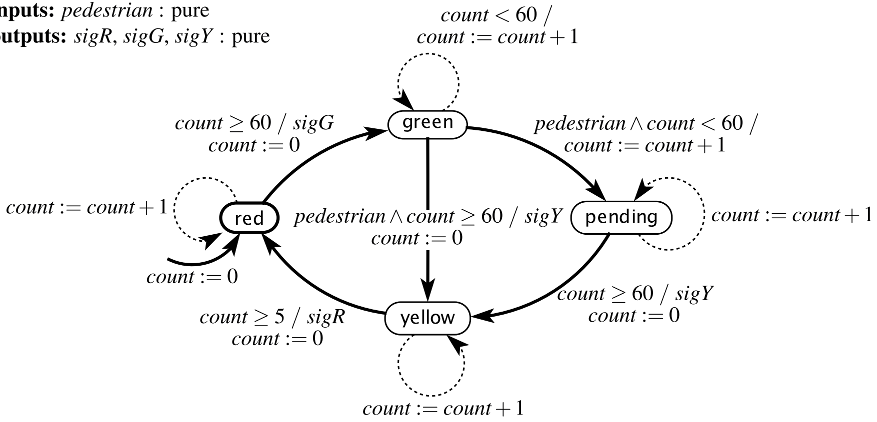
A Model of a Pedestrian Crossing Light

variable: $pcount$: $\{0, \cdots, 55\}$
input: $sigR$: pure
outputs: $pedG$, $pedR$: pure



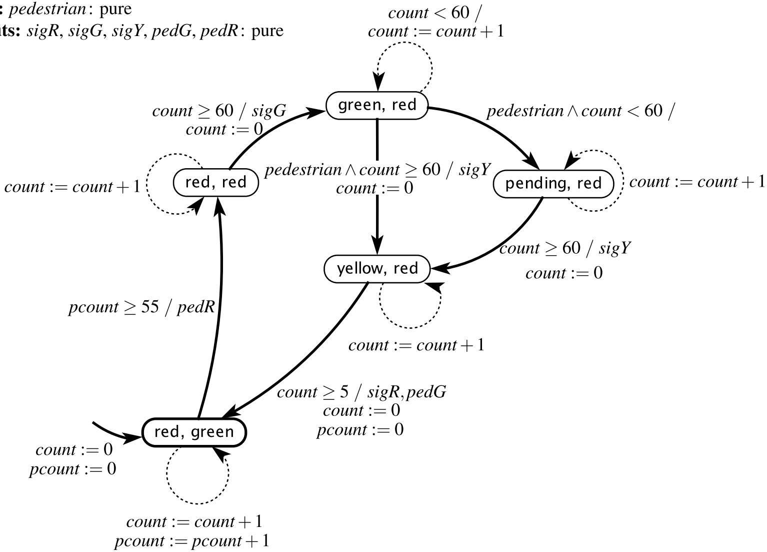
Traffic Light Model

variable: *count*: $\{0, \cdots, 60\}$
inputs: *pedestrian* : pure
outputs: *sigR*, *sigG*, *sigY* : pure

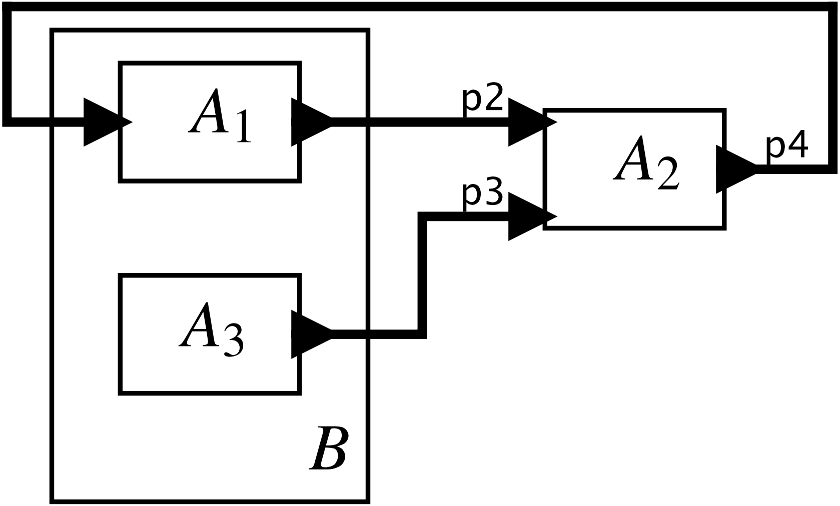


Synchronous Cascade Composition

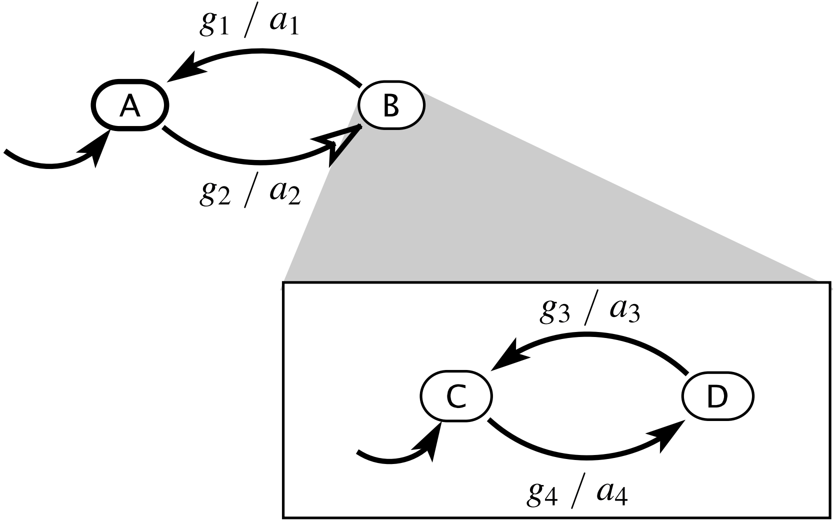
variables: *count*: $\{0, \cdots, 60\}$, *pcount*: $\{0, \cdots, 55\}$
input: *pedestrian*: pure
outputs: *sigR*, *sigG*, *sigY*, *pedG*, *pedR*: pure



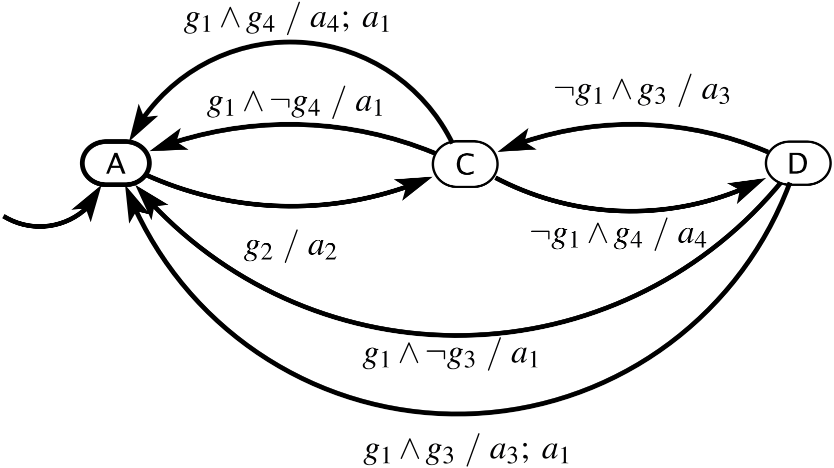
Arbitrary Interconnections of State Machines



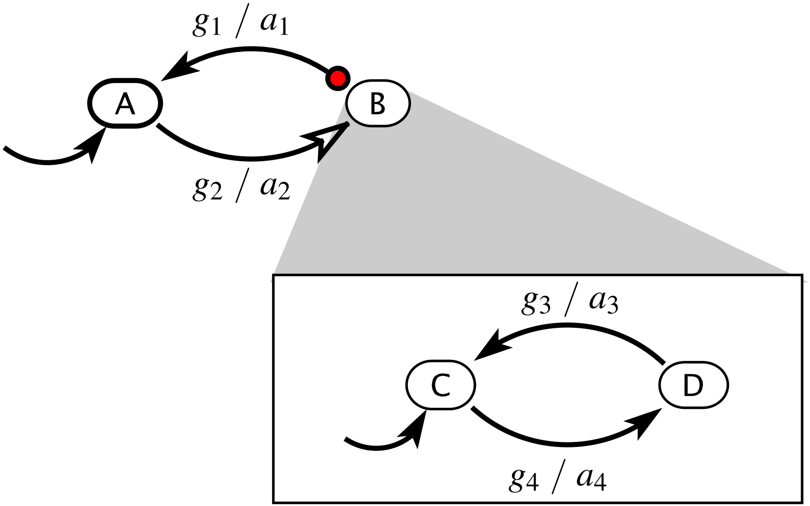
Hierarchical FSM



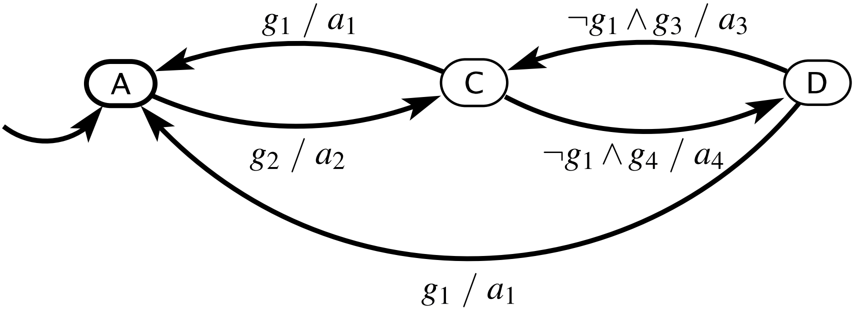
Semantics of a Hierarchical FSM



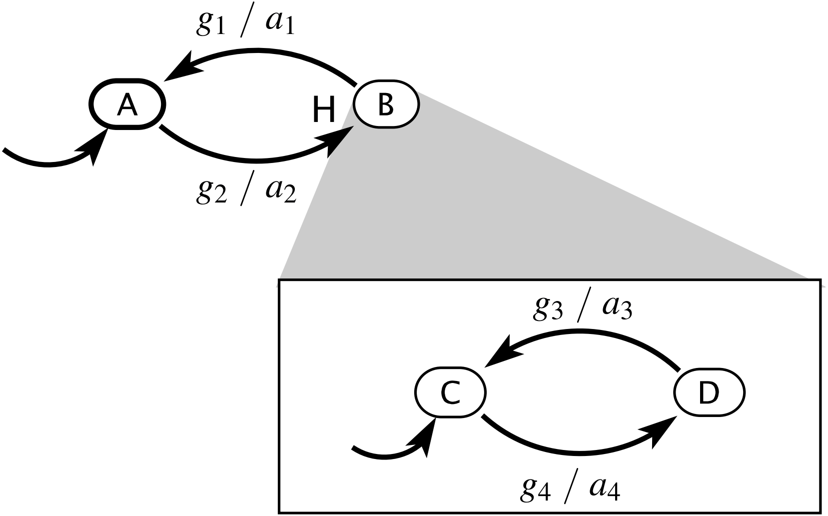
Preemptive Transition Example



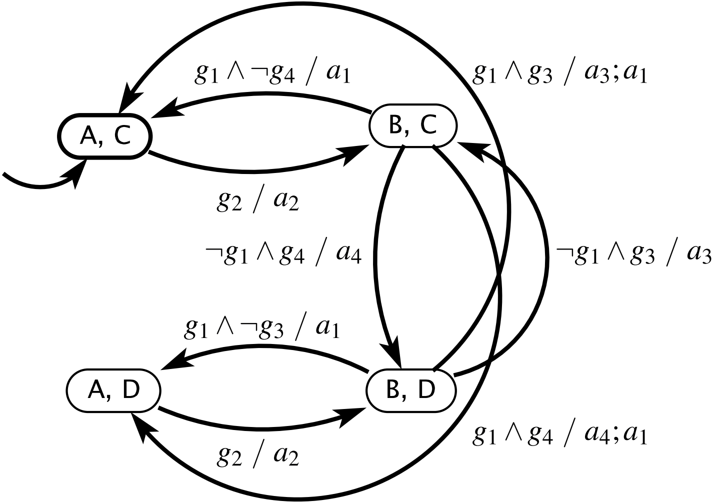
Semantics of a Preemptive Transition



History Transition Example



Semantics of a History Transition



Concluding Remarks

- Any well-engineered system is a composition of simpler components.
- Considered concurrent composition and hierarchical composition.
- For concurrent composition, introduced both **synchronous** and **asynchronous** composition.
- Several possible semantics for asynchronous composition.
- Hierarchical models similar to *Statecharts* introduced by Harel (1987).