COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

EMBEDDED SYSTEM DESIGN

Composition of State Machines

Doan Duy, Ph. D.

Email: duyd@uit.edu.vn

COMPUTER ENGINEERING

Objectives

- Definition of Composite State Machine
- Kinds of Composite State Machine

COMPUTER ENGINEERING

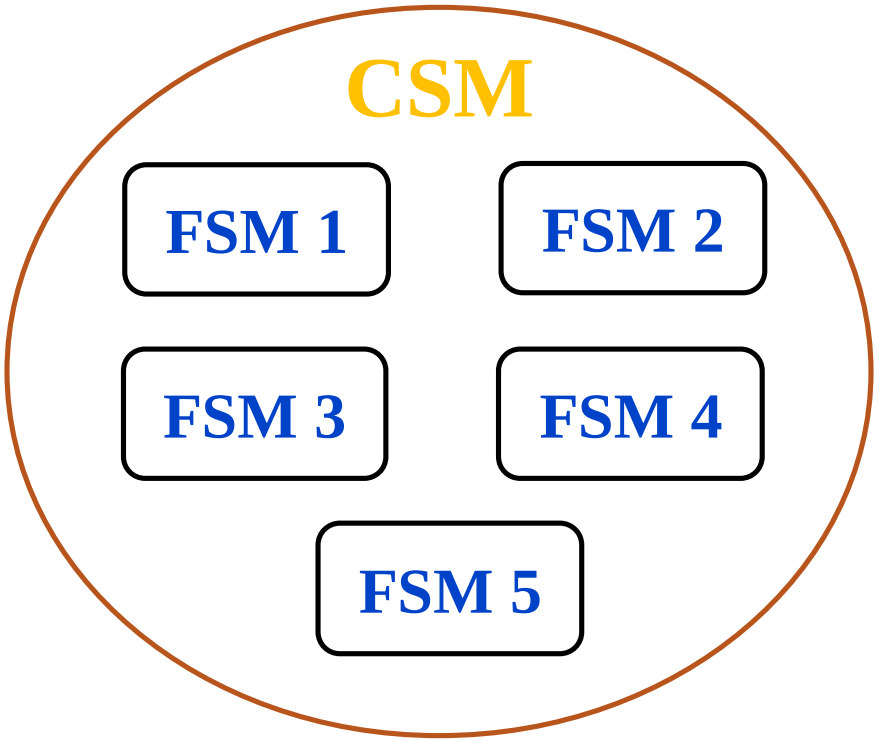
Contents

1. Introduction to CSM
2. Kinds of CSM:
   - Spatial CSM vs Temporal CSM
   - Synchronous vs Asynchronous

Introduction to Composition of State Machine

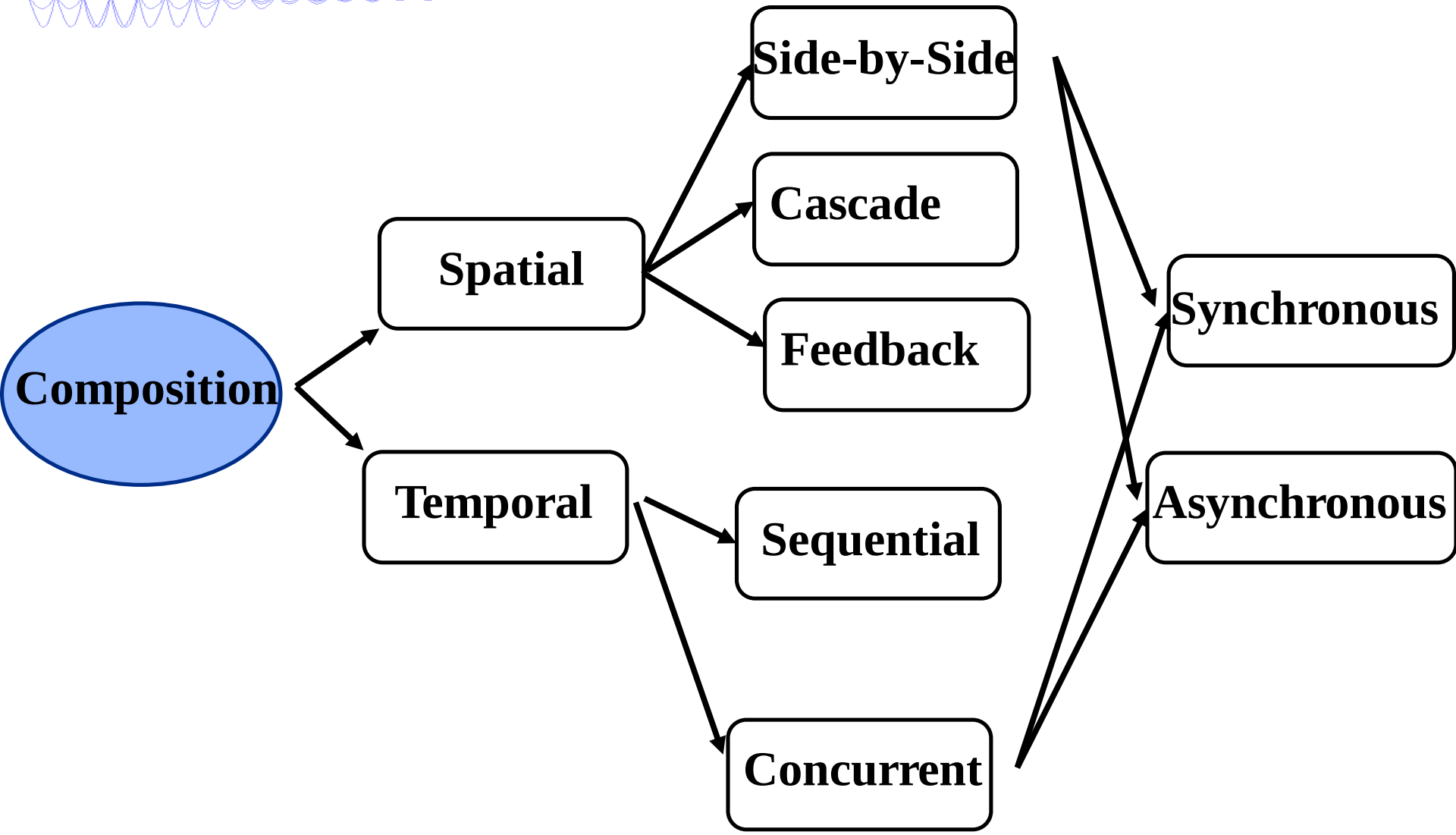
Introduction of Composite State Machine

- How do we construct complex state machines out of simpler "building blocks"?



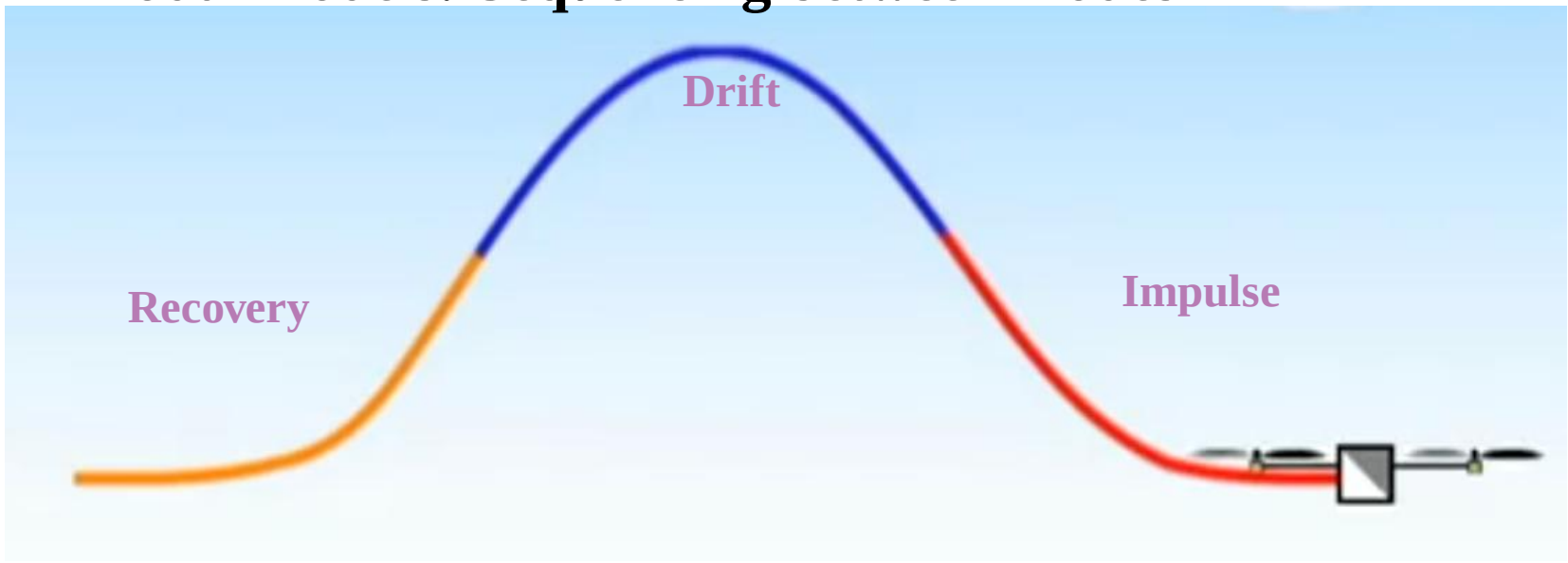
COMPUTER ENGINEERING

Kinds of Composition



Sequential Composition

Modal models: Sequencing between modes

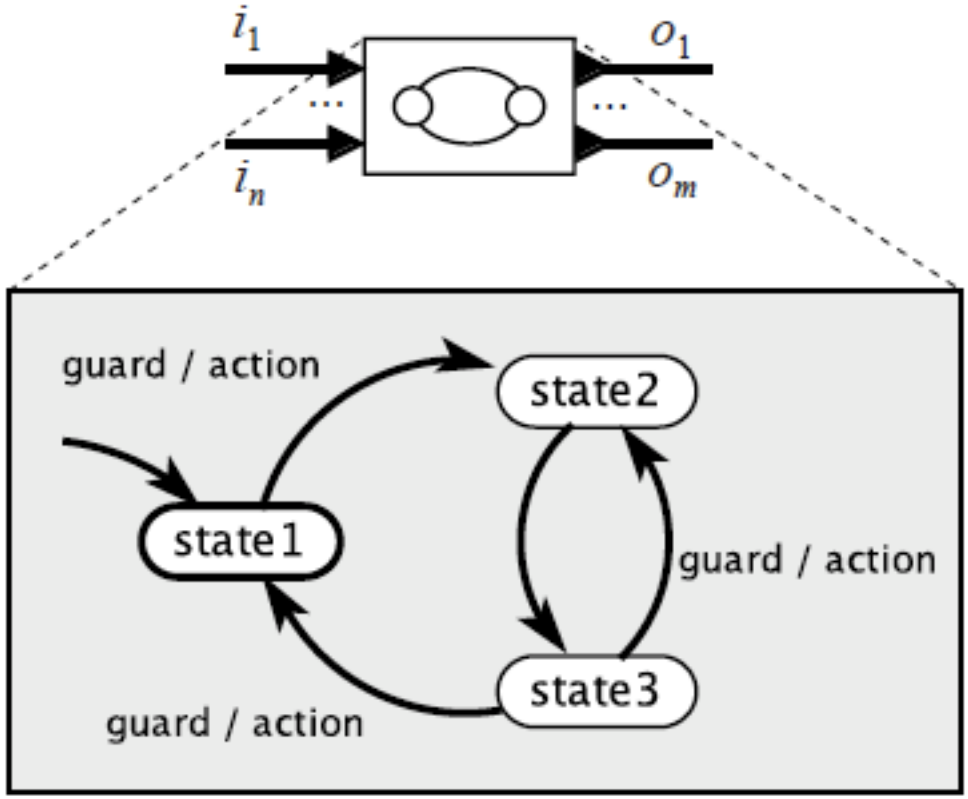


https://www.youtube.com/watch?v=iD3QgGpzzIM

[Tomlin et al.]

Concurrent Composition

- Expose inputs and outputs, enabling concurrent composition:

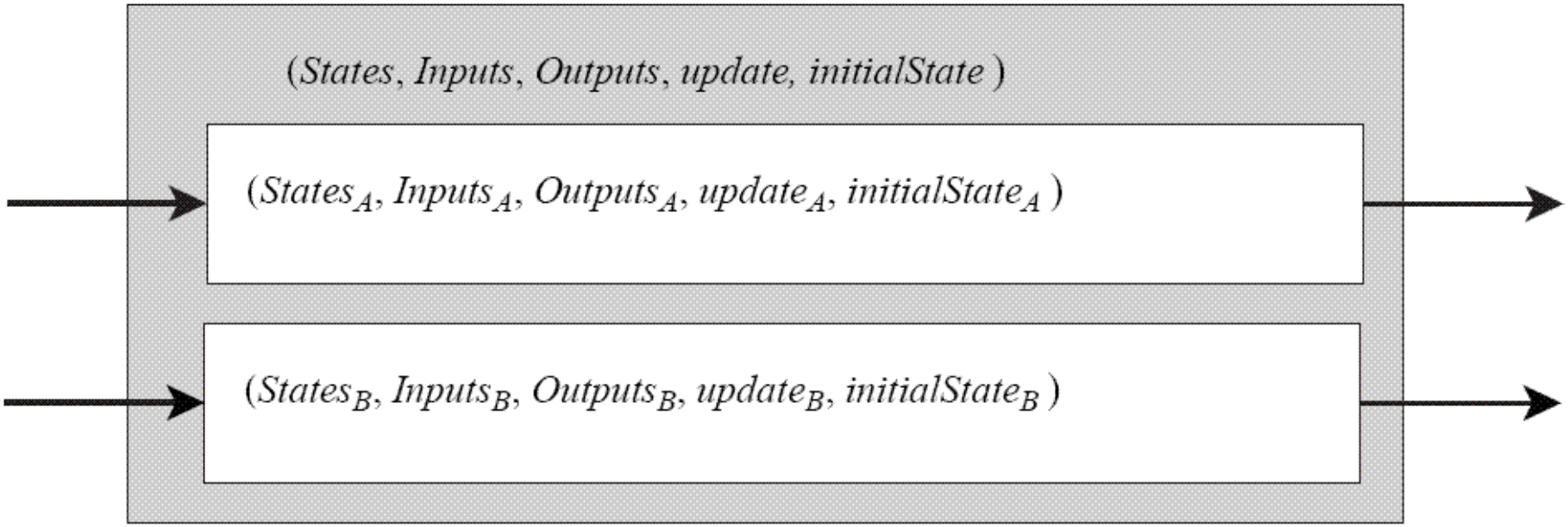


COMPUTER ENGINEERING

Kinds of Spatial Composition

- Side-by-side composition
- Cascade composition
- Feedback composition

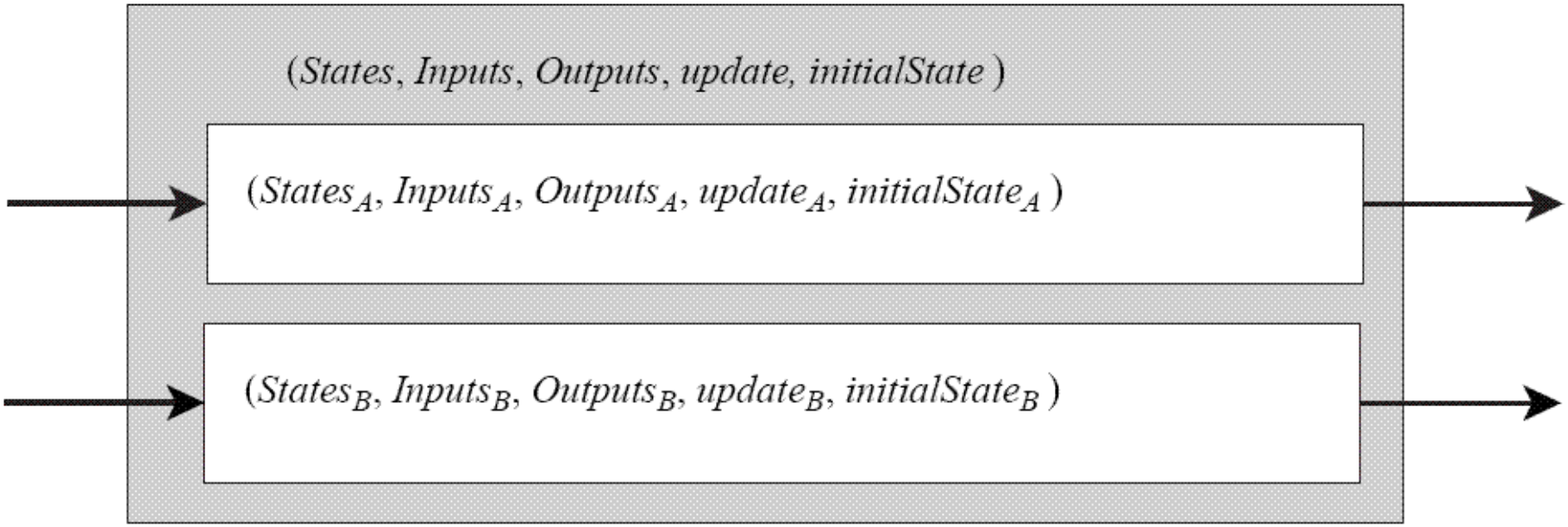
Side-by-Side Composition



A key question: When do these machines react?

How the reactions of composed machines is coordinated is called a “Model of Computation” (MoC).

Side-by-Side Composition (cont.)

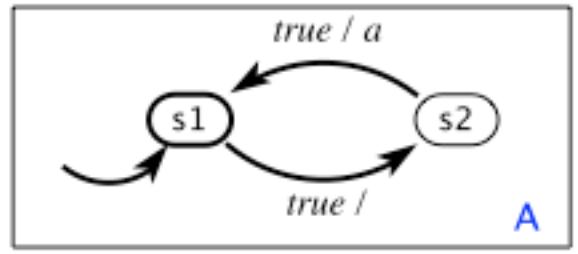


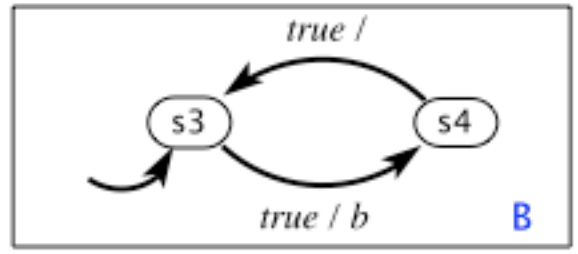
When do these machines react? Two of many possibilities:

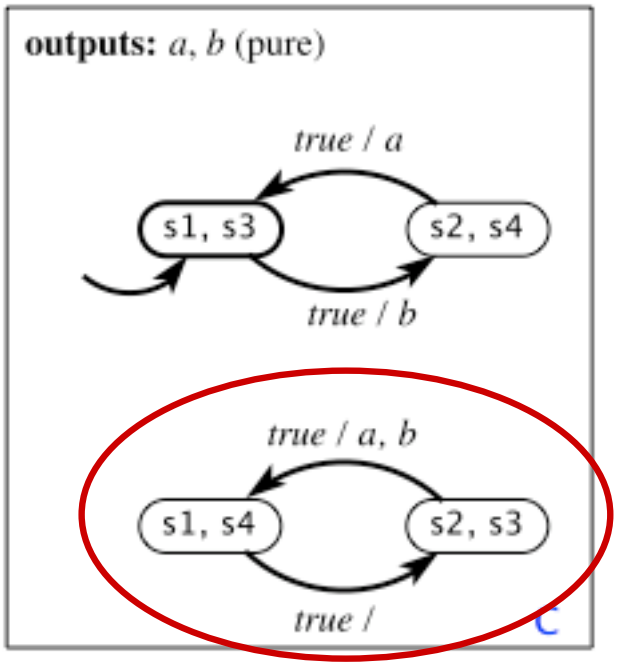
- Together, in lock step (synchronous, concurrent composition)
- Independently (asynchronous, concurrent composition)

Synchronous Composition

$$S_C \subseteq S_A \times S_B$$





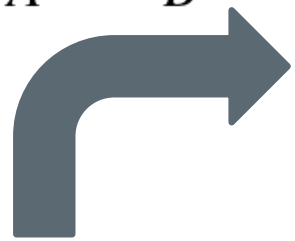


Synchronous composition

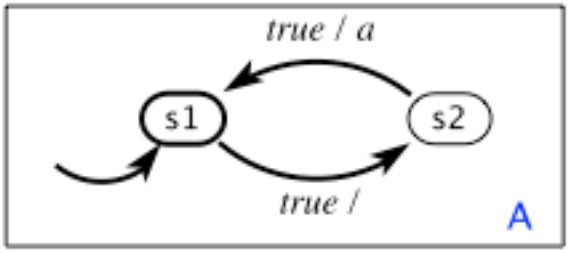
Note that these two states are not reachable.

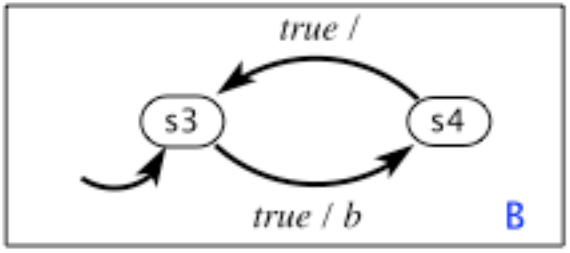
Asynchronous Composition

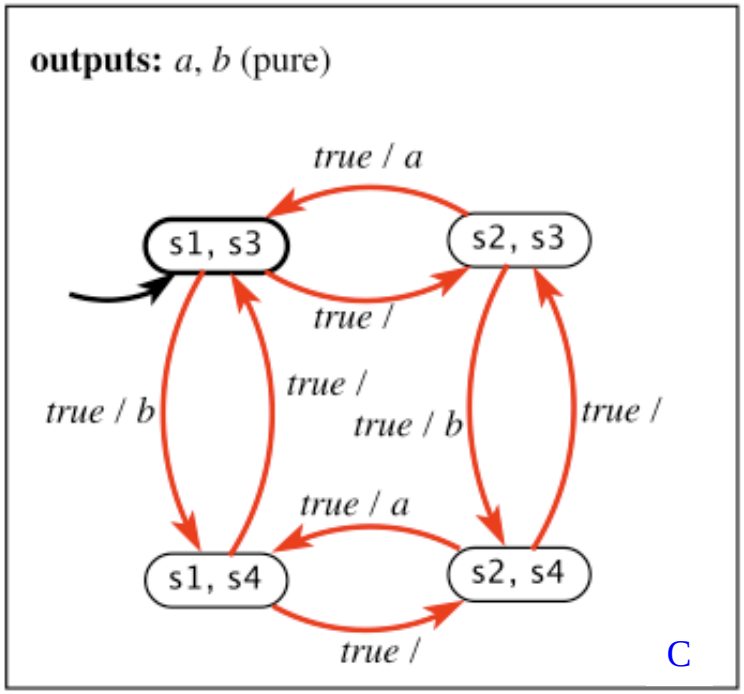
$$S_C \subseteq S_A \times S_B$$



outputs: *a*, *b* (pure)



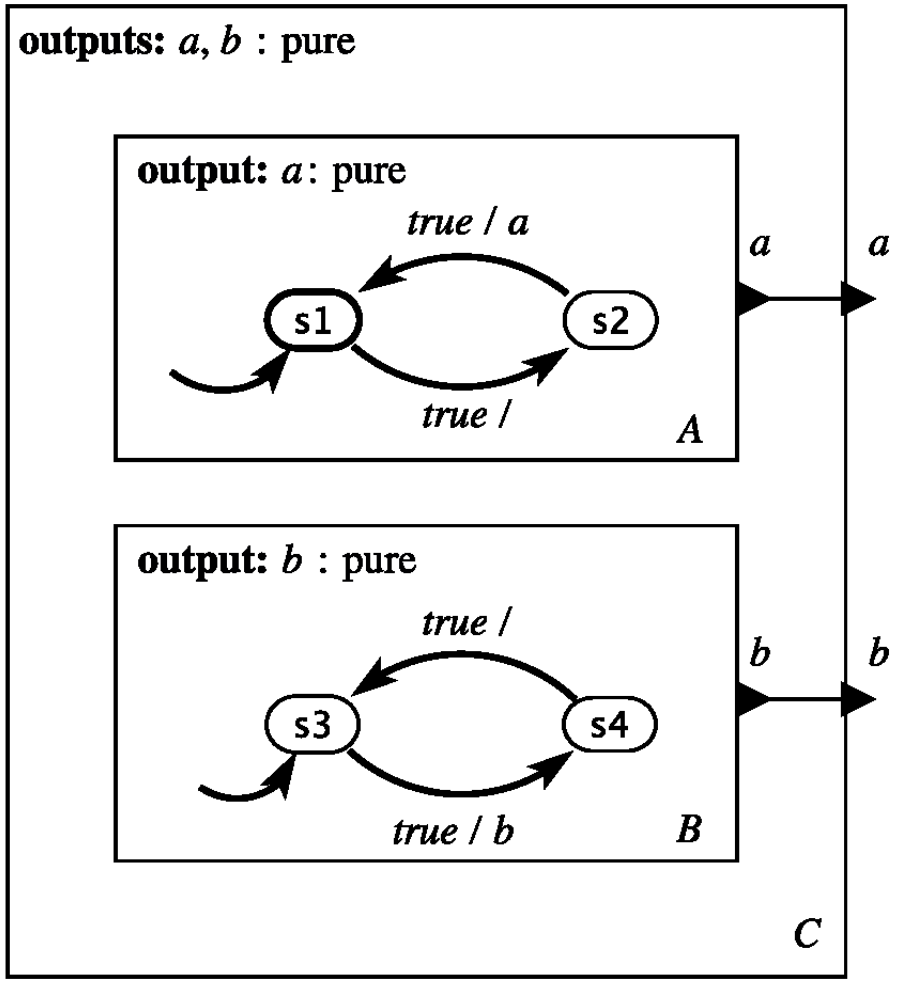




Asynchronous composition using interleaving semantics

Note that now all states are reachable.

Syntax vs. Semantics

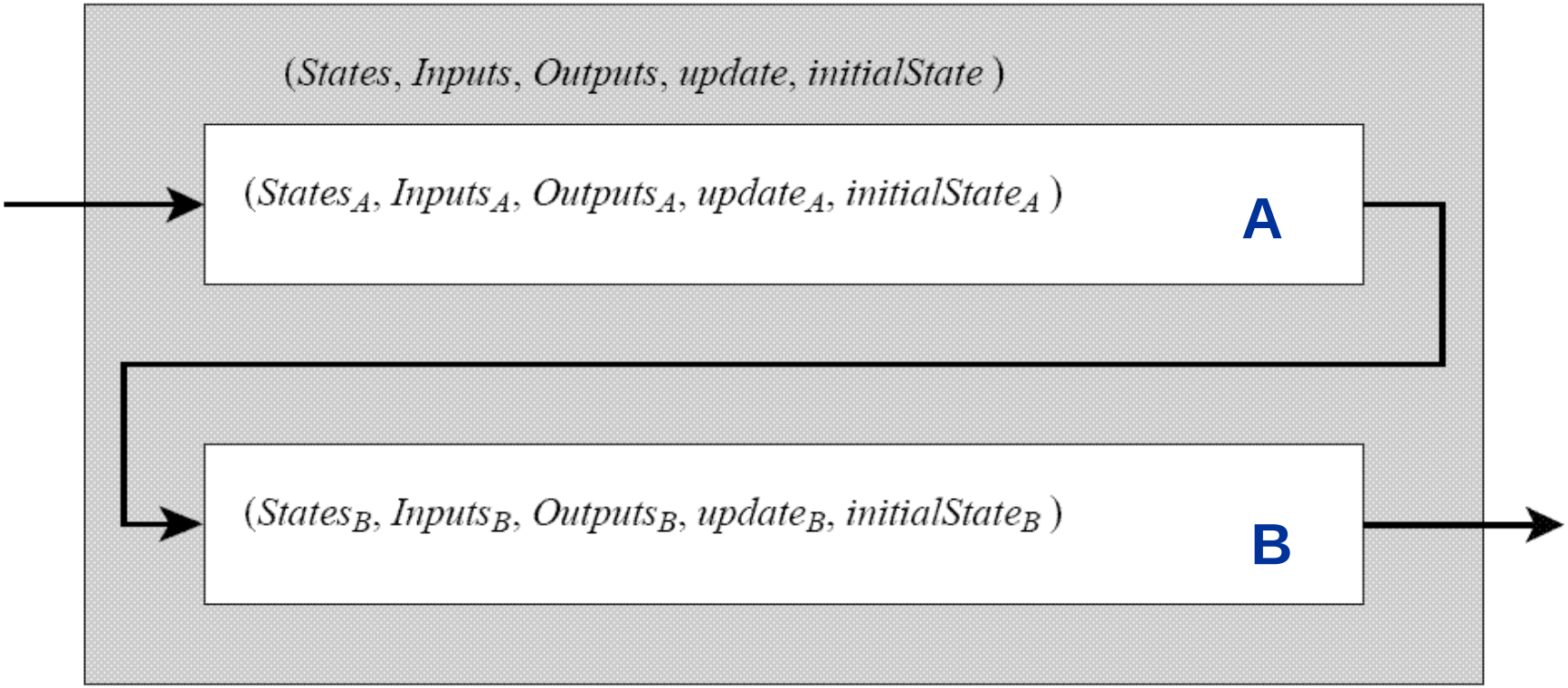


The answers to these questions defines the MoC being used.

Synchronous or Asynchronous composition?

If asynchronous, does it allow simultaneous transitions in A & B? How to choose whether A or B reacts when C reacts?

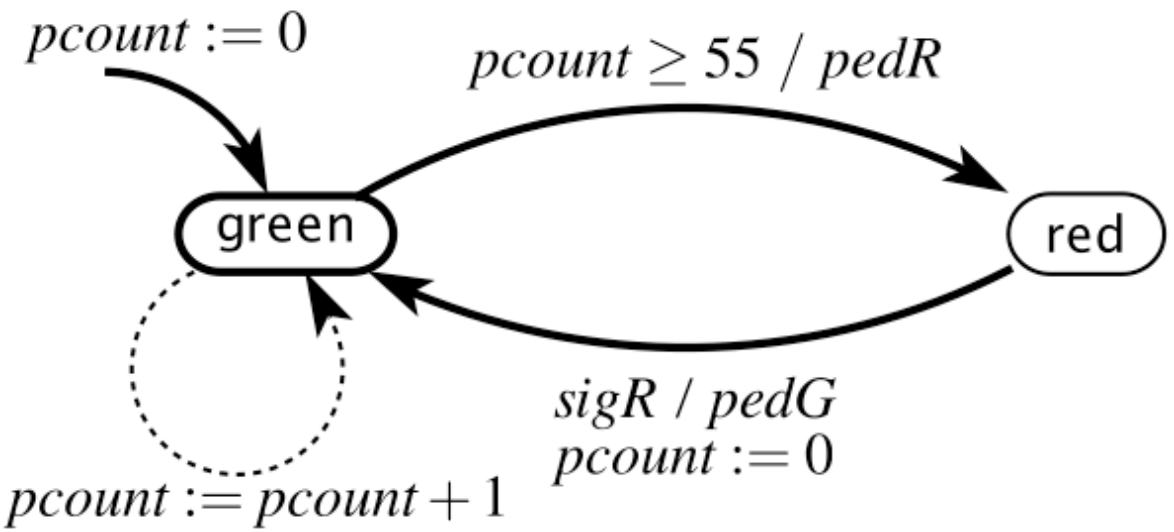
Cascade Composition



Output port(s) of A connected to input port(s) of B

Time-Triggered Pedestrian Light

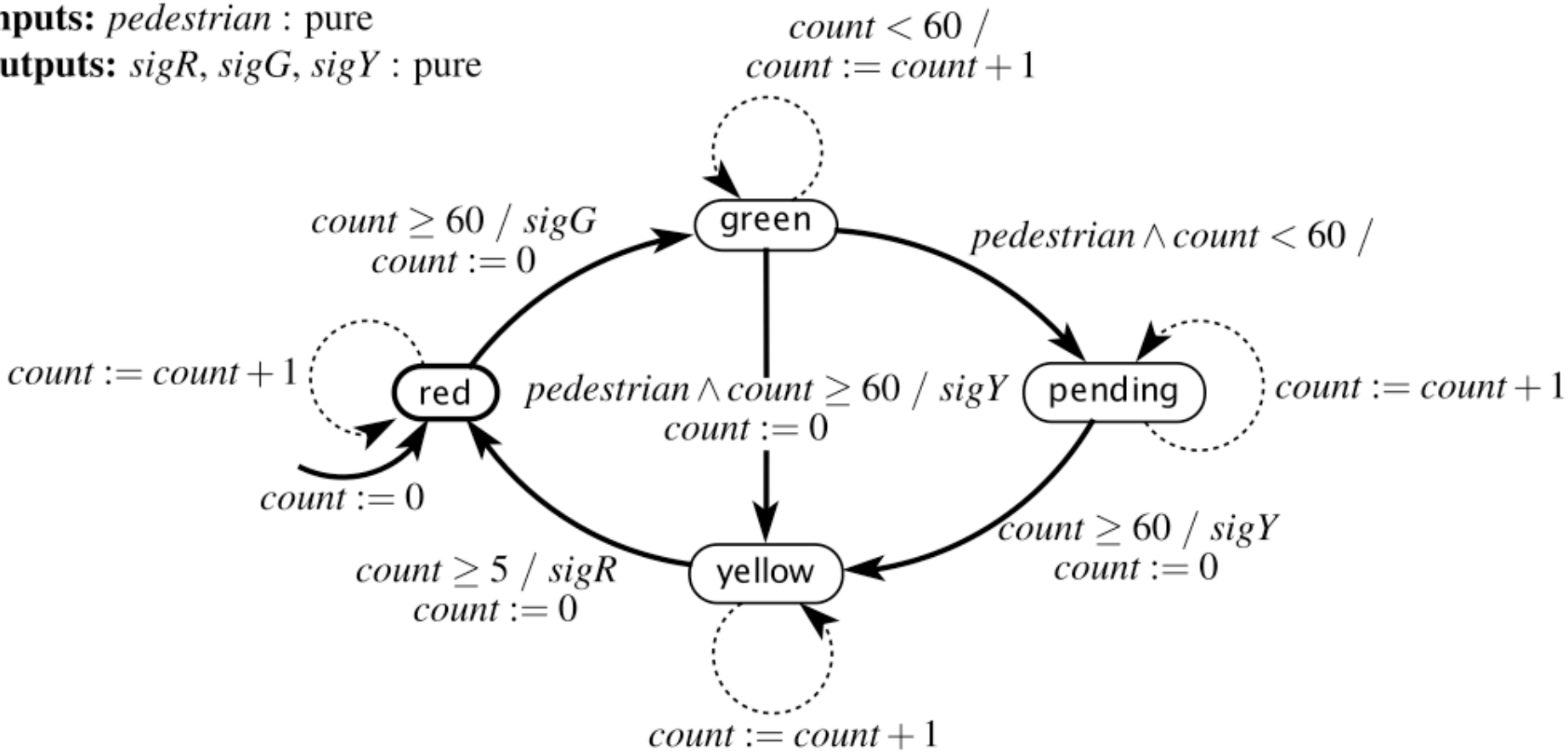
variable: *pcount*: $\{0, \cdots, 55\}$
input: *sigR*: pure
outputs: *pedG*, *pedR*: pure



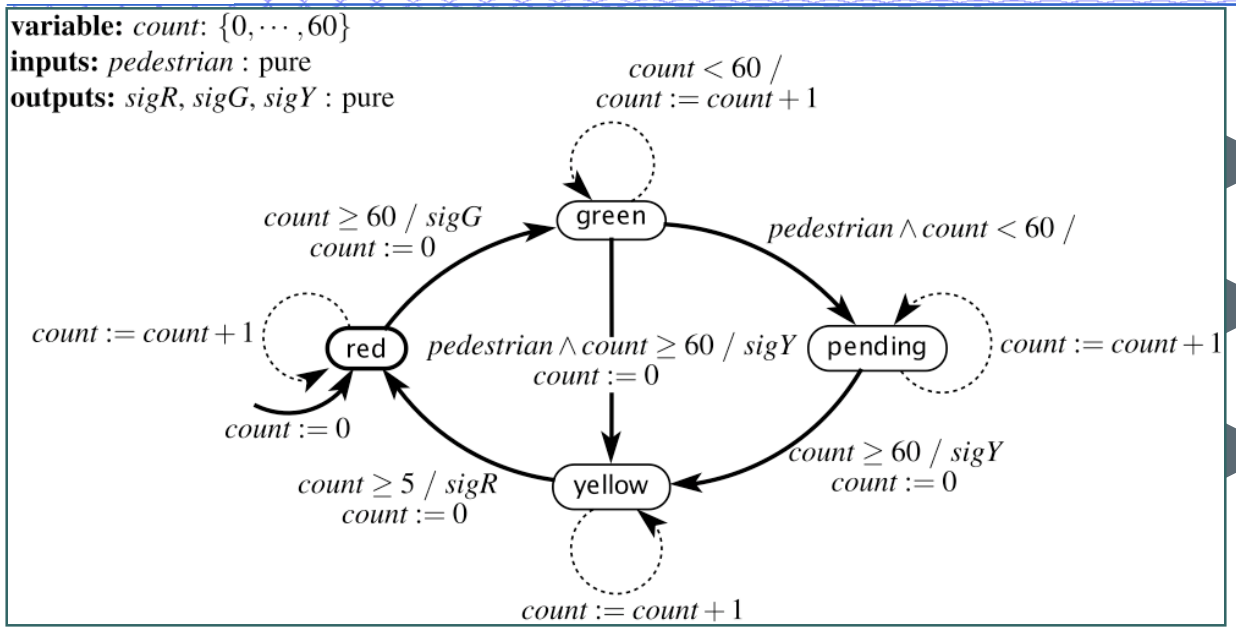
This light stays green for 55 seconds, then goes red.
Upon receiving a sigR input, it repeats the cycle.

Time-Triggered Car Light

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



Pedestrian Car Light

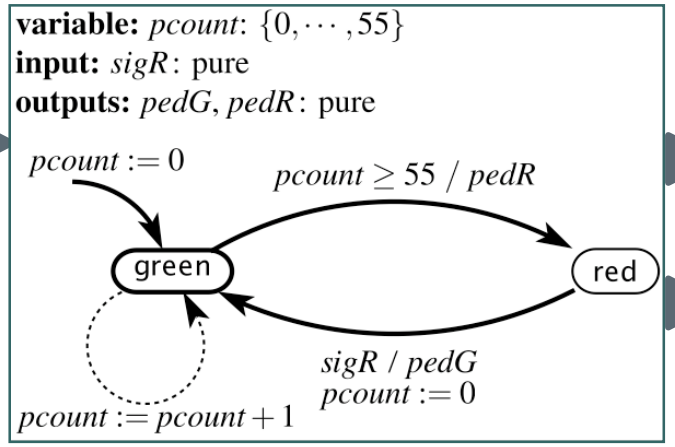


sigY

sigG

sigR

sigR



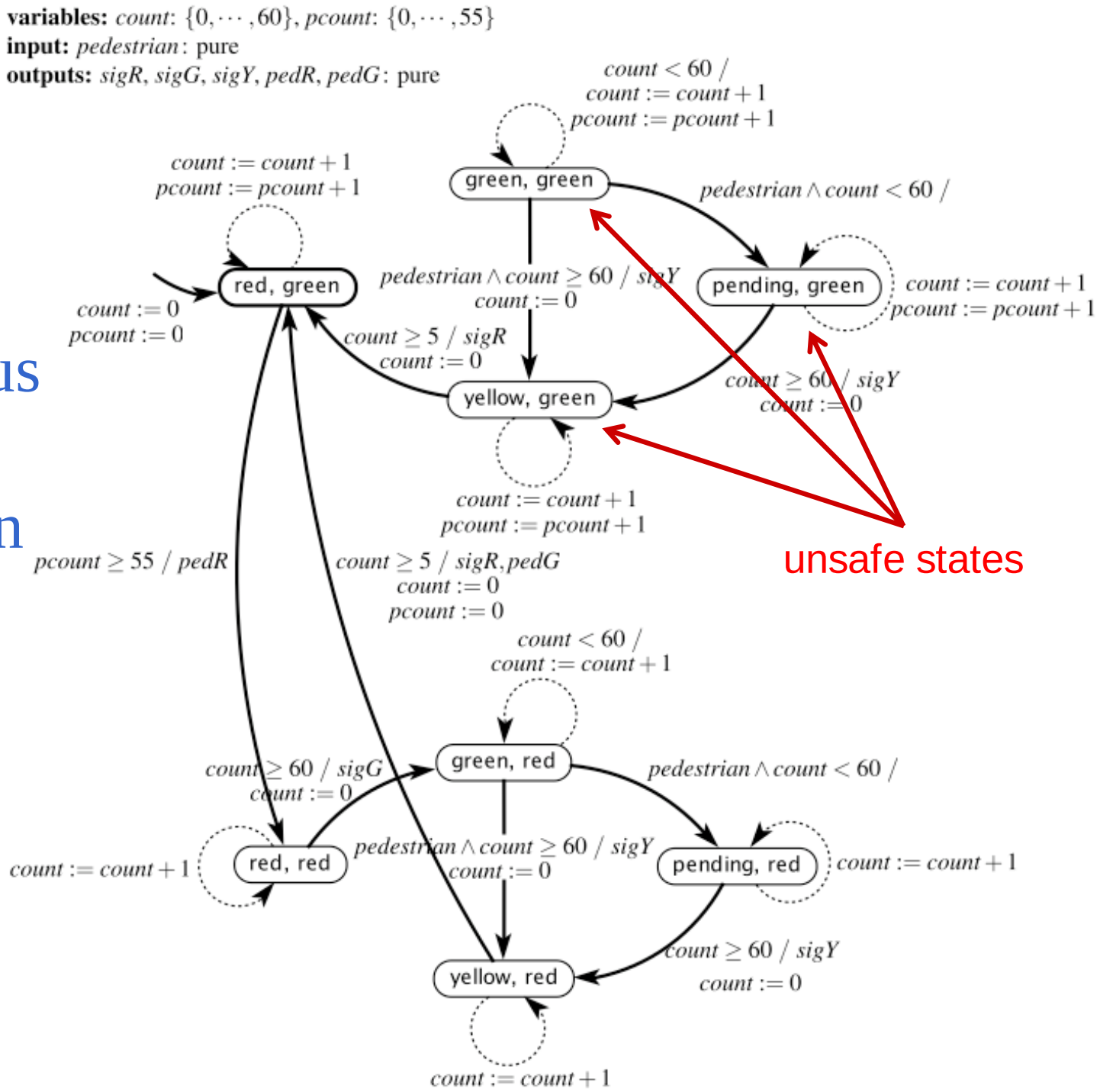
pedG

pedR

What is the size of the state space of the composite machine?

COMPUTER ENGINEERING

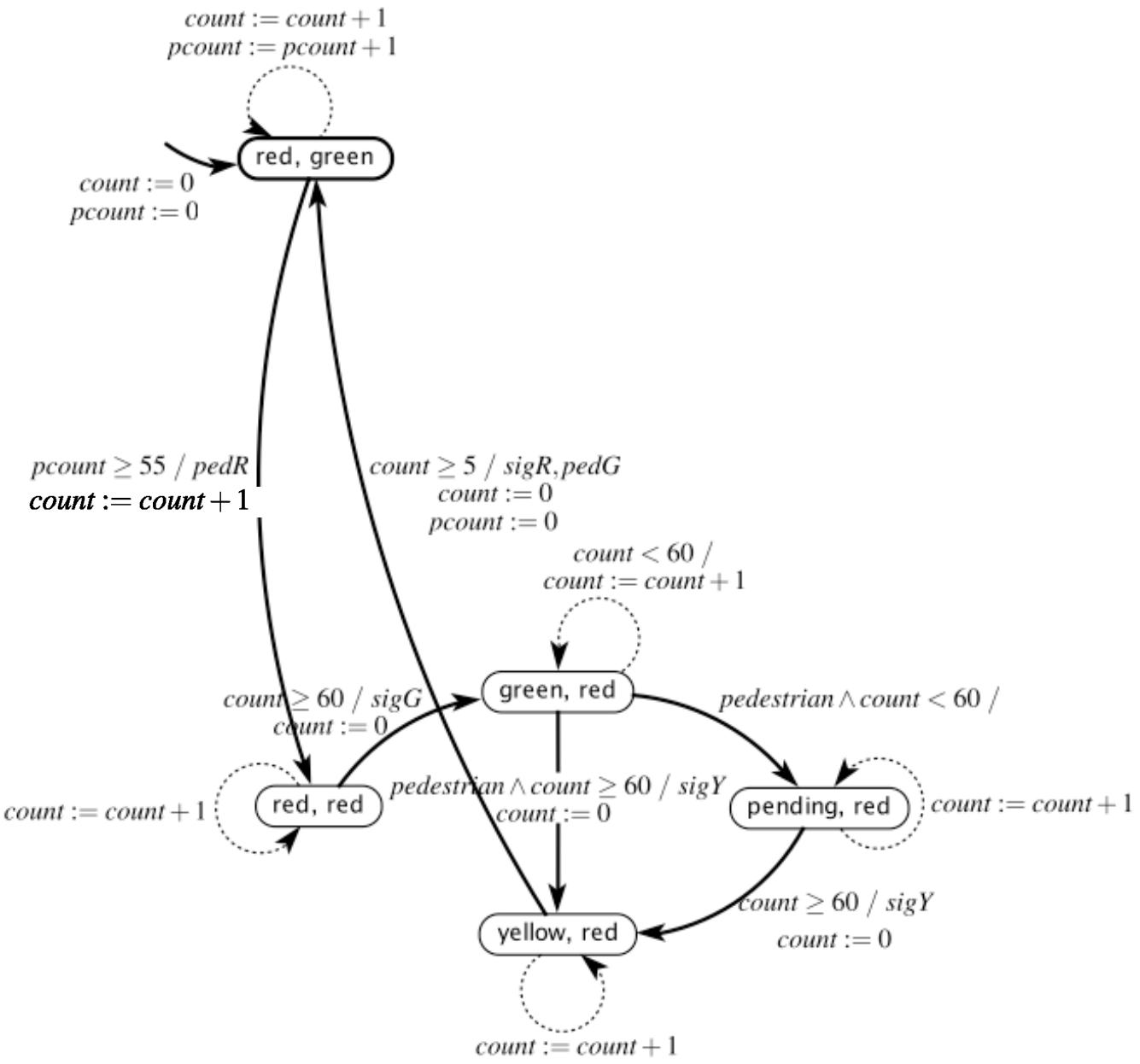
Synchronous cascade composition

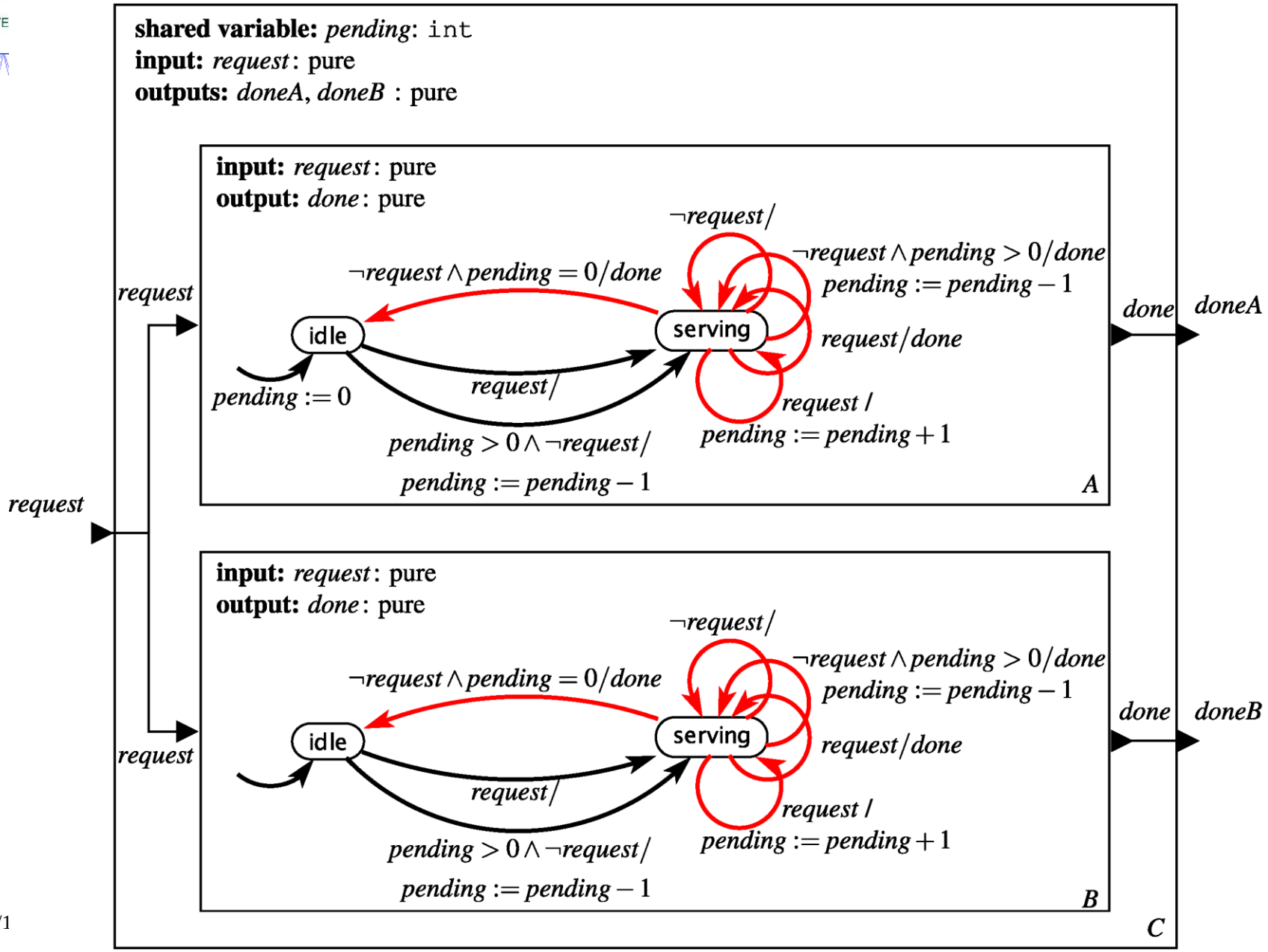


COMPUTER ENGINEERING

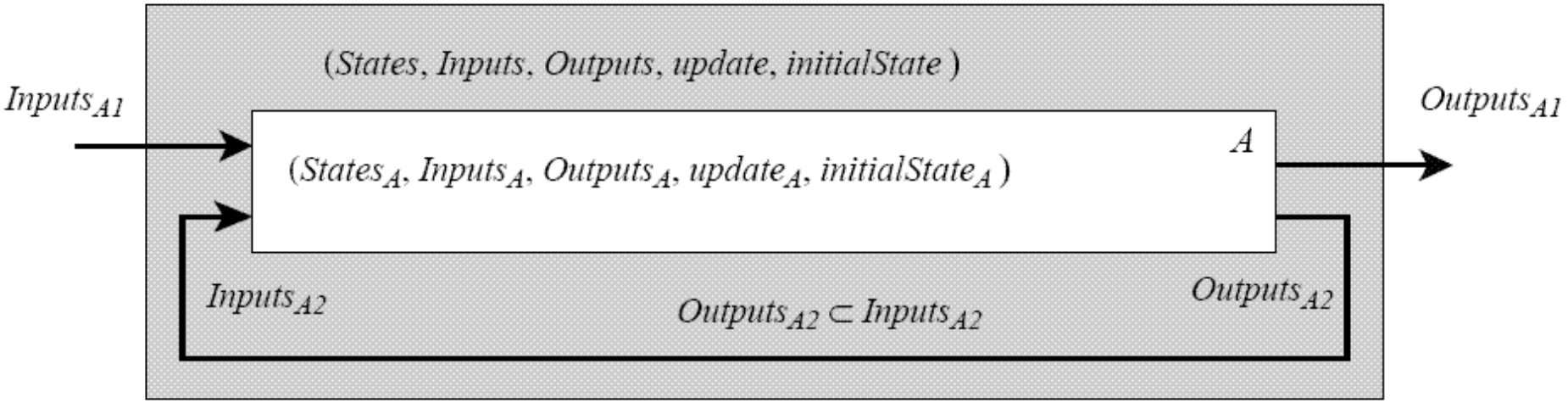
Synchronous composition with unreachable states removed

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




COMPUTE
Shared Variables
shared variable: pending: int
input: request: pure
outputs: doneA, doneB : pure
input: request: pure
output: done: pure
¬request/
¬request ∧ pending > 0/done
pending := pending − 1
¬request ∧ pending = 0/done
request
idle
serving
request/done
pending := 0
request/
request /
pending := pending + 1
pending > 0 ∧ ¬request/
pending := pending − 1
A
done
doneA
request
input: request: pure
output: done: pure
¬request/
¬request ∧ pending > 0/done
pending := pending − 1
¬request ∧ pending = 0/done
idle
serving
request/done
request/
request /
pending := pending + 1
pending > 0 ∧ ¬request/
pending := pending − 1
B
done
doneB
request
C

Example: Feedback Composition



Reasoning about feedback composition can be very subtle. (more about this later)

Kinds of Spatial Composition

- Side-by-side composition
- Cascade composition
- Feedback composition

Temporal Composition

- Sequential vs. Parallel
- Asynchronous vs. Synchronous

COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Q&A