Introduction to Embedded Systems

Chapter 6: Hierarchical State Machines

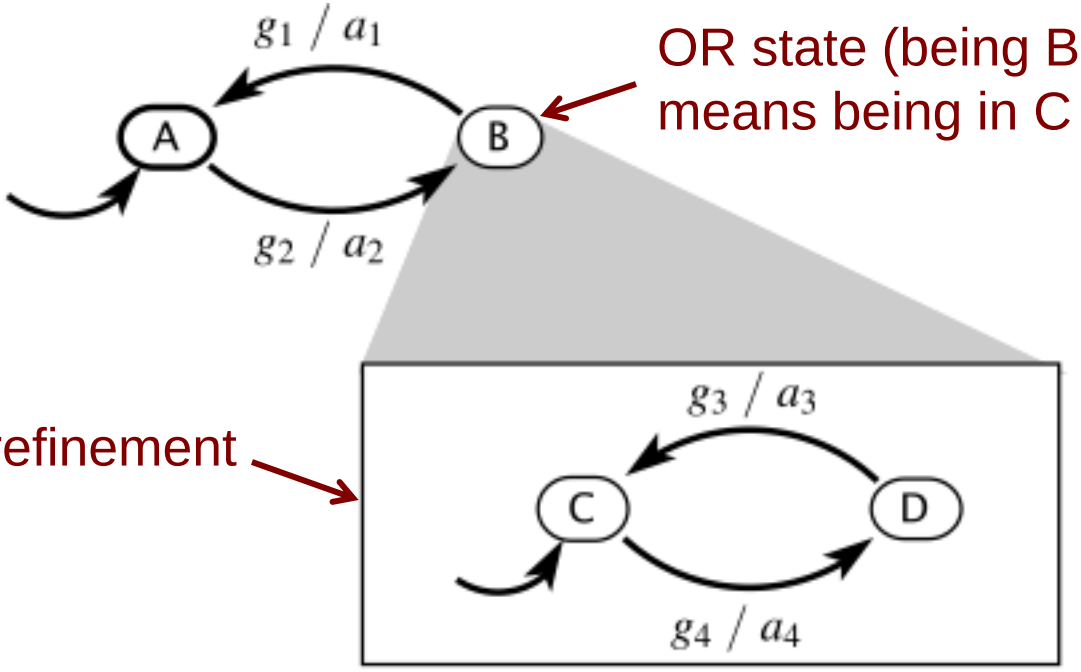
Sanjit A. Seshia
UC Berkeley
EECS 149/249A
Fall 2015

Modifications by Nicolae Cleju: in this color

Hierarchical State Machines

- Hierarchical state machines:
  - A state in a top-level FSM can be implemented ("refined") as an internal/embedded state machine
    - The top level state = "super-state"
    - An internal state inside it = "sub-state"
- Problems:
  - Which sub-state is entered?
  - What transitions are executed and in what order?

Hierarchical State Machines



OR state (being B means being in C or D)

refinement

Reaction:

1. First, the refinement of the current state (if any) reacts.
2. Then the top-level machine reacts.

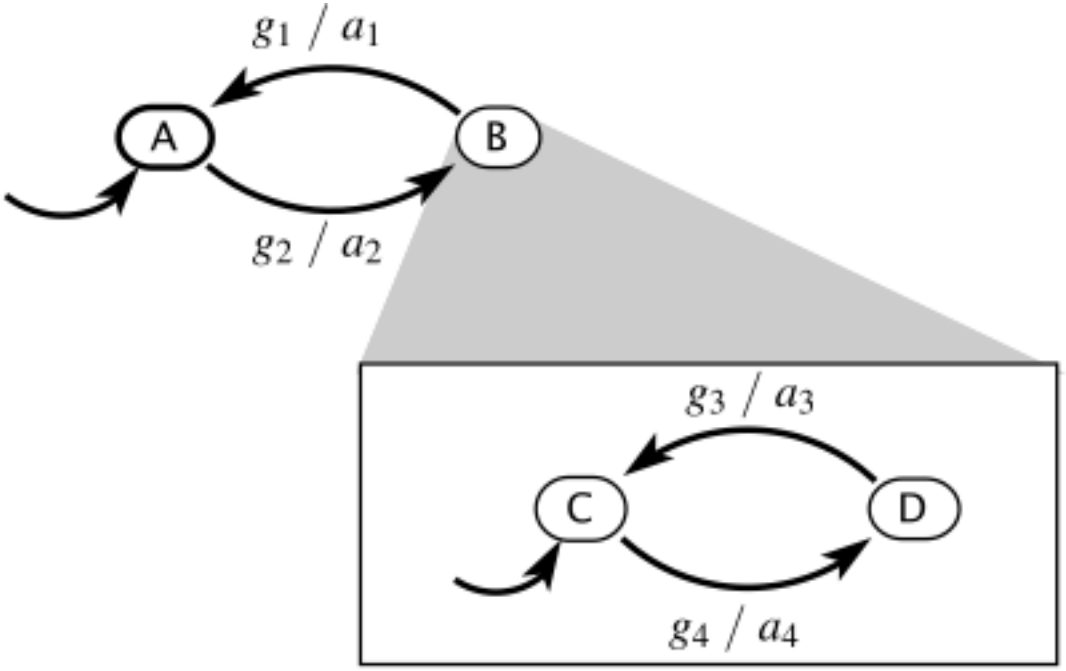
If both produce outputs, they are required to not conflict. The two steps are part of the same reaction.

[Statecharts, David Harel, 1987]

Hierarchical State Machines

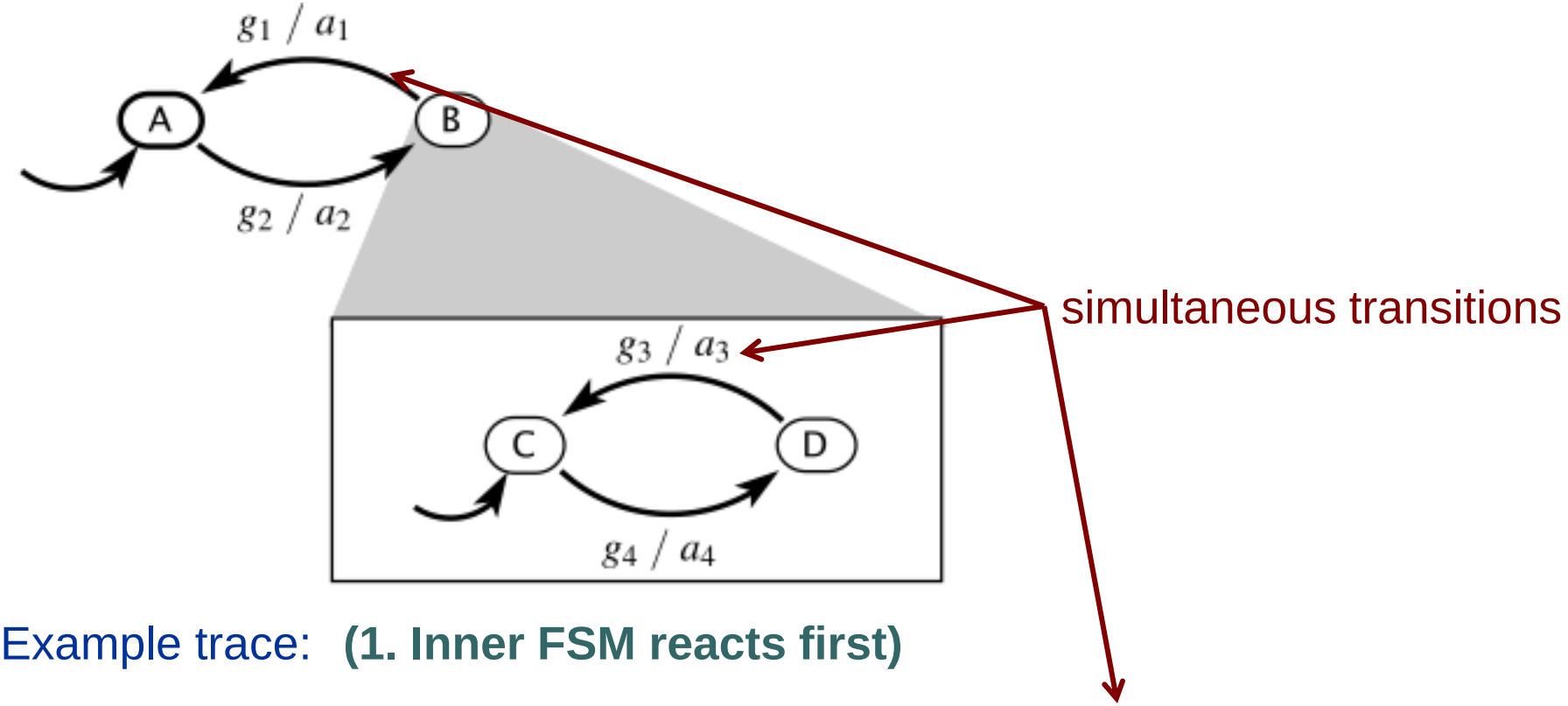
- Which FSM reacts first? The inner one or the outer one?
- 2 solutions:
  1. [Statecharts language] Inner FSM reacts first, outer FSM reacts later.
     - The two reactions are considered simultaneous
     - The output actions are required to not conflict
  2. [Stateflow, Matlab] Outer FSM reacts first, inner FSM reacts later (if at all)
     - If state is left, the inner FSM will not react at all

Hierarchical State Machines



Example trace:

Hierarchical State Machines

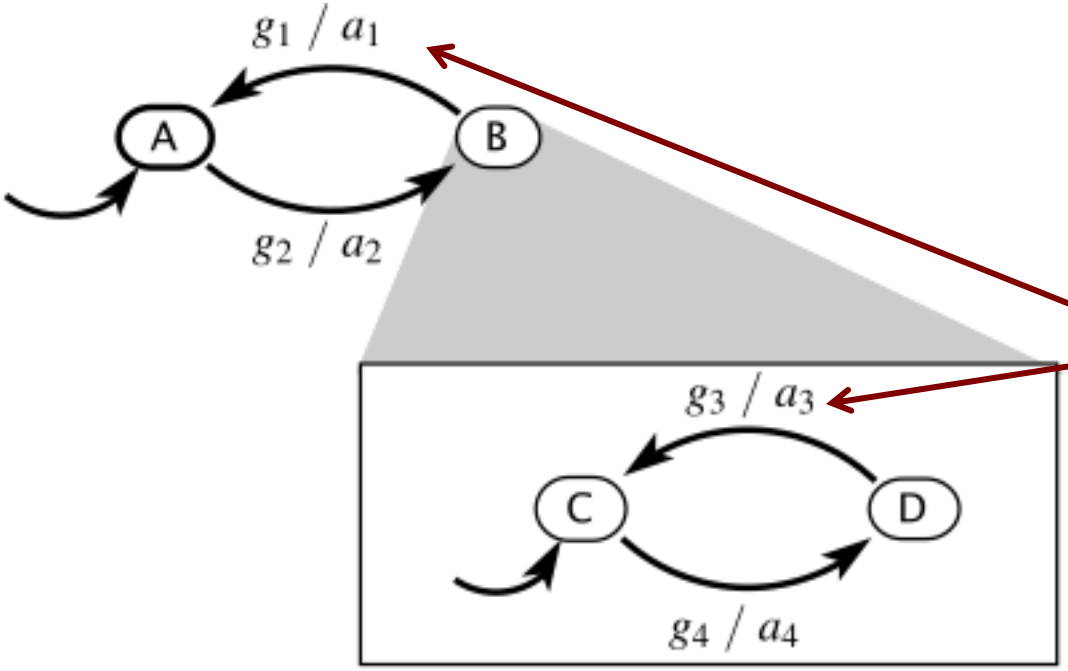


Example trace: **(1. Inner FSM reacts first)**

$$A \xrightarrow{g_2/a_2} C \xrightarrow{g_4/a_4} D \xrightarrow{g_1/a_1} A \xrightarrow{g_2/a_2} D \xrightarrow{g_3 \wedge g_1 / a_3, a_1} A \cdots$$

Simultaneous transitions can produce multiple outputs. These are required to not conflict.

Hierarchical State Machines



Outer transition first, inner transition not at all

Example trace: **(2. Outer FSM reacts first)**

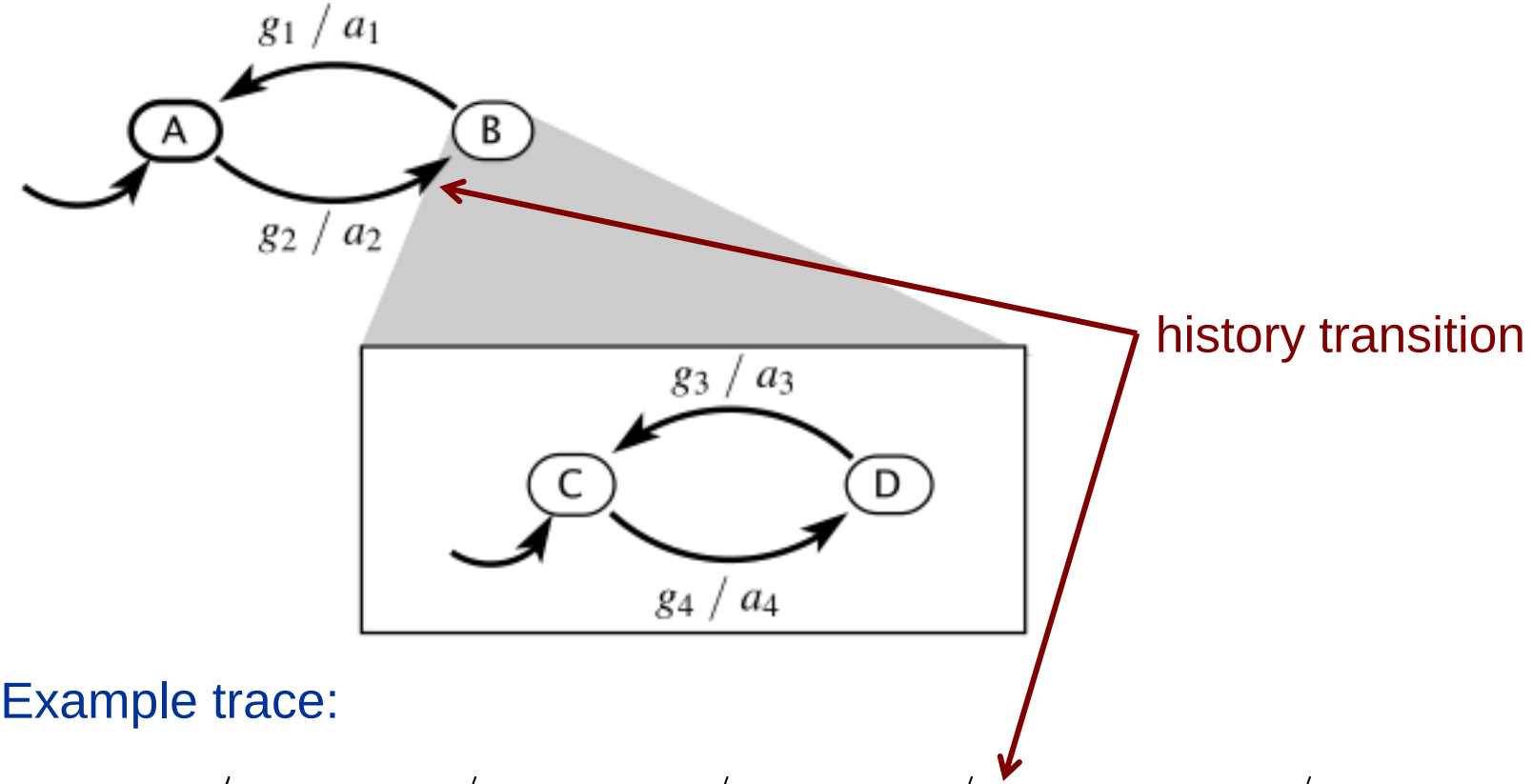
$$A \xrightarrow{g_2/a_2} C \xrightarrow{g_4/a_4} D \xrightarrow{g_1/a_1} A \xrightarrow{g_2/a_2} D \xrightarrow{g_3 \wedge g_1 / \cancel{g_3}, a_1} A \cdots$$

Simultaneous transitions can produce multiple outputs. These are required to not conflict.

History transitions

- When entering a super-state, which sub-state is entered?
- 2 solutions:
  1. Enter the last sub-state you were in, when you last left the super-state
     - Represented as a “history transition” (full black arrow on these schematics / a H sign in Matlab)
  2. Enter the default sub-state every time
     - Known as a “reset transition” (white arrow on these schematics / default behavior in Matlab)

Hierarchical State Machines



Example trace:

$$A \xrightarrow{g_2/a_2} C \xrightarrow{g_4/a_4} D \xrightarrow{g_1/a_1} A \xrightarrow{g_2/a_2} D \xrightarrow{g_3 \wedge g_1 / a_3, a_1} A \cdots$$

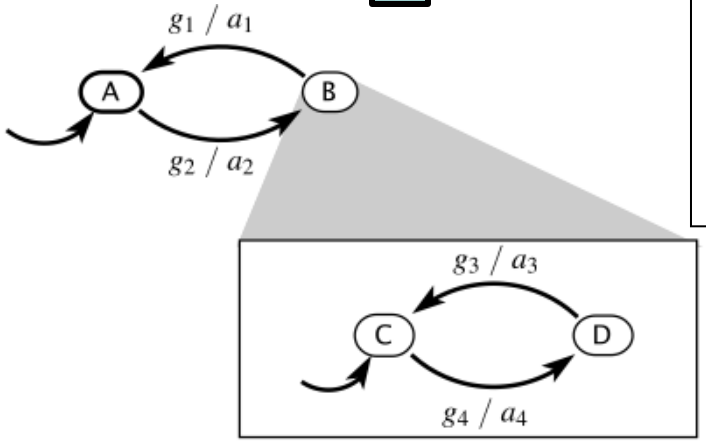
A history transition implies that when a state with a refinement is left, it is nonetheless necessary to remember the state of the refinement.

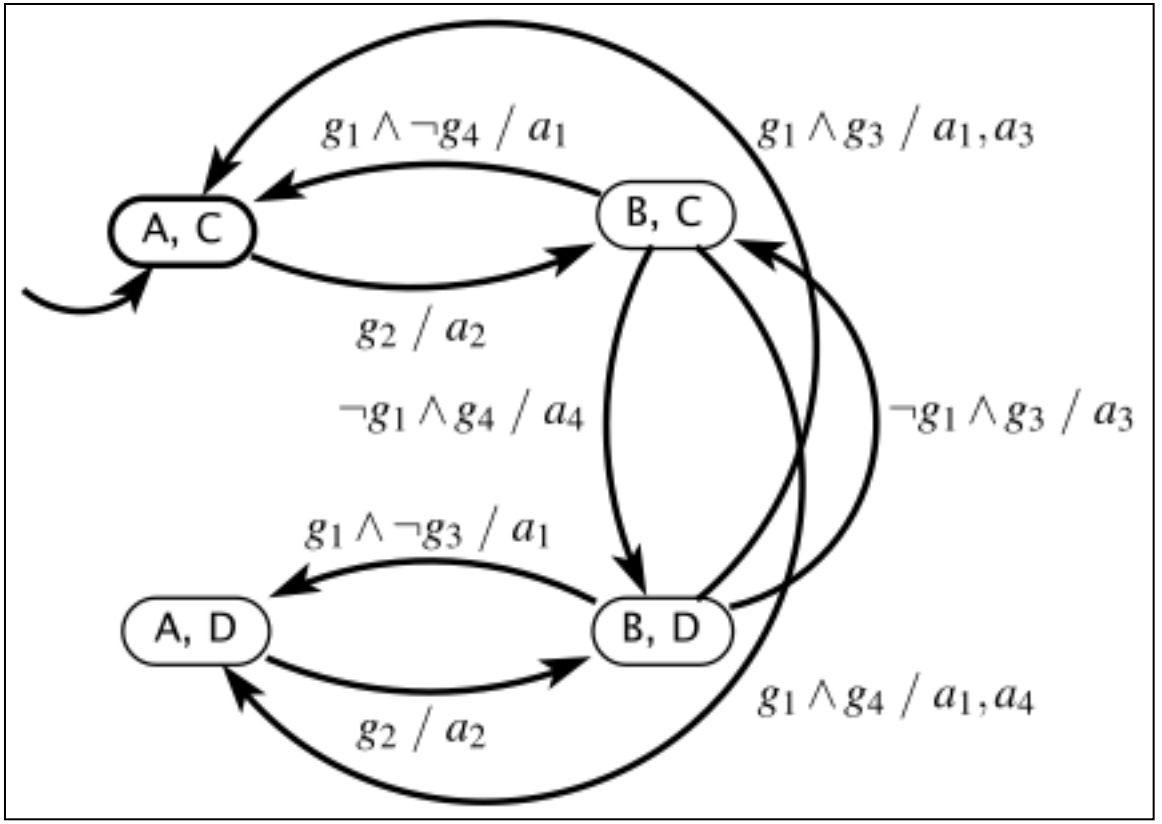
Equivalent Flattened State Machine

Every hierarchical state machine can be transformed into an equivalent “flat” state machine.

This transformation can cause the state space to blow up substantially.

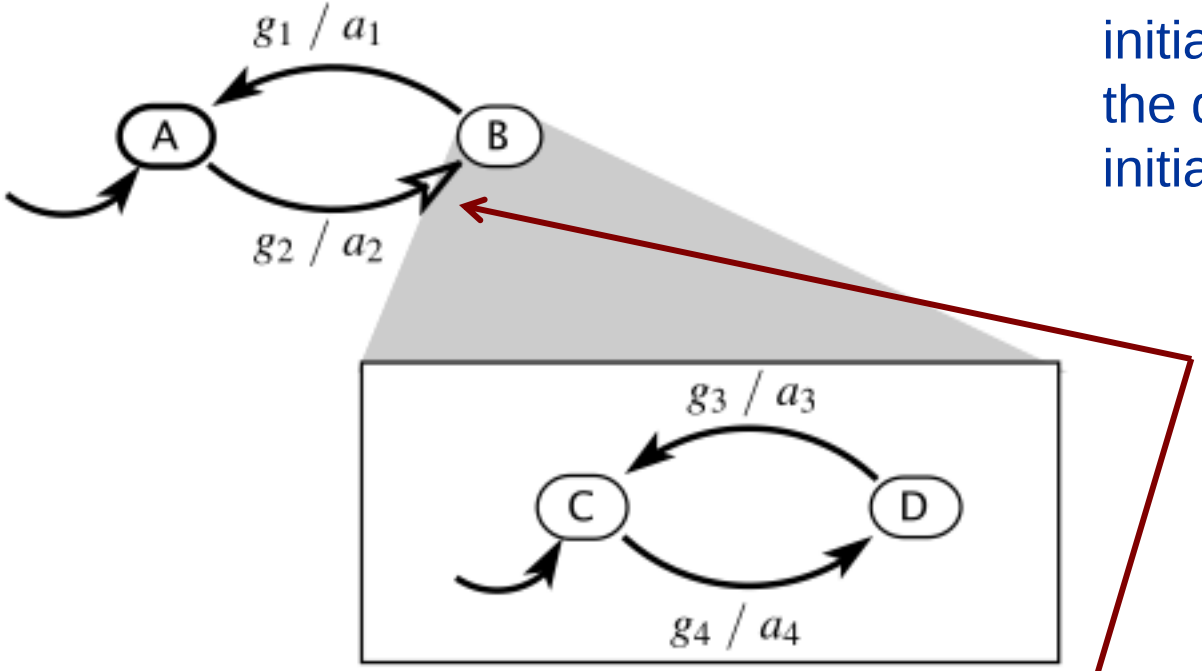
Flattening the state machine (assuming history transitions):





A history transition implies that when a state with a refinement is left, it is nonetheless necessary to remember the state of the refinement. Hence A,C and A,D.

Hierarchical State Machines with Reset Transitions



A reset transition always initializes the refinement of the destination state to its initial state.

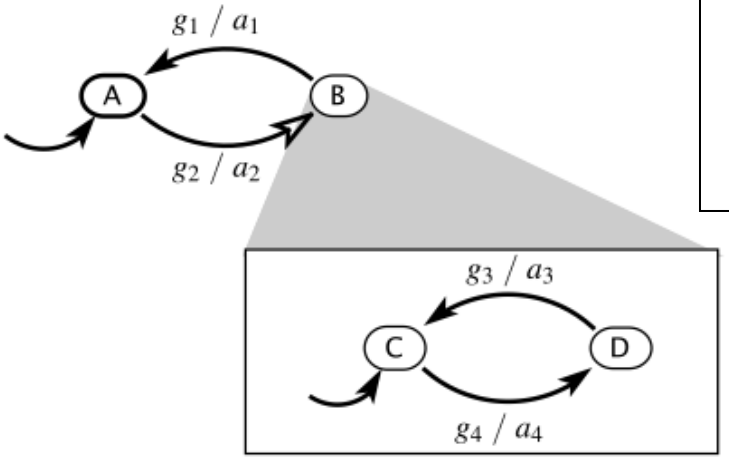
reset transition

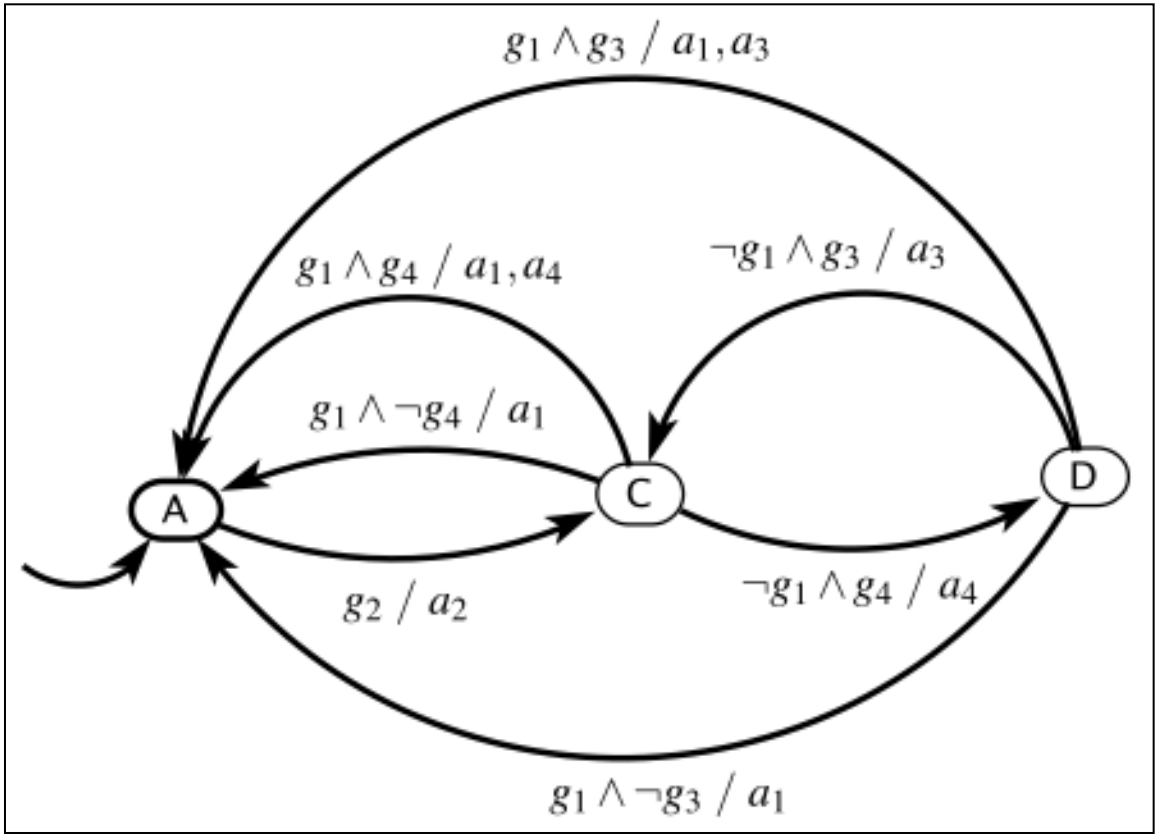
Example trace:

$$A \xrightarrow{g_2/a_2} C \xrightarrow{g_4/a_4} D \xrightarrow{g_1/a_1} A \xrightarrow{g_2/a_2} C \xrightarrow{g_4 \wedge g_1/a_4,a_1} A \cdots$$

A reset transition implies that when a state with a refinement is left, you can forget the state of the refinement.

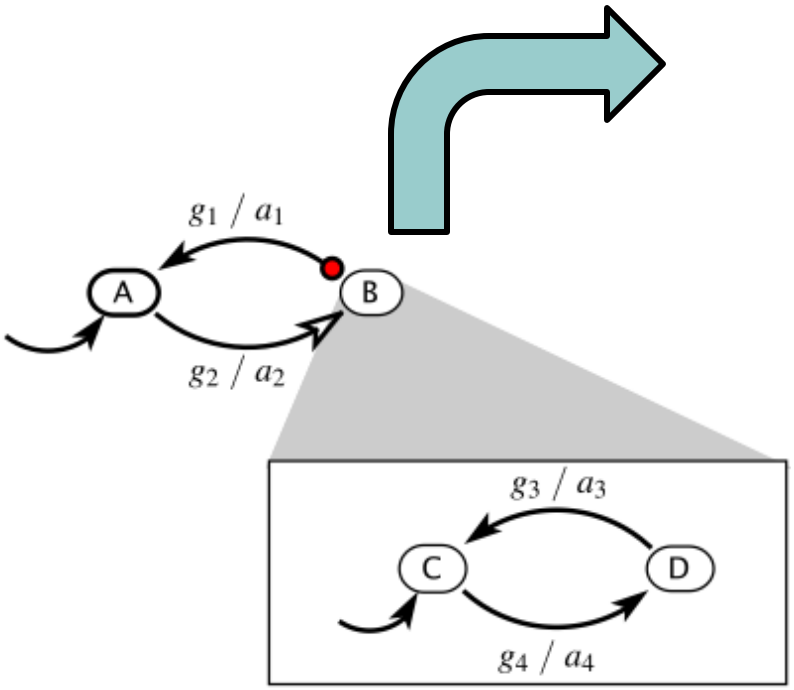
Flattening the state machine (assuming reset transitions):

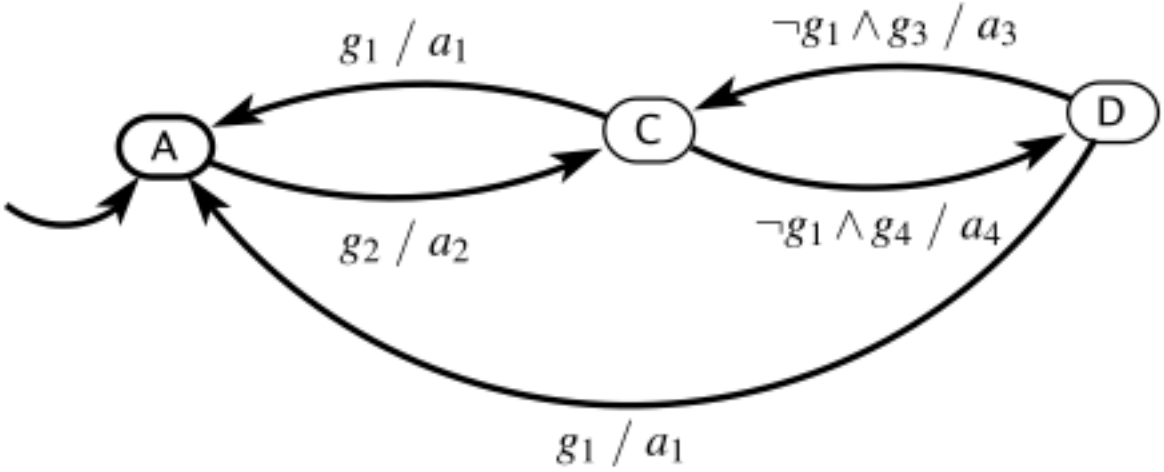




A reset transition implies that when a state with a refinement is left, it is not necessary to remember the state of the refinement. Hence there are fewer states.

Preemptive Transitions





A preemptive transition specifies that the guard should be evaluated before the current state refinement reacts, and if it is true, then the current state should not react.

This is the default behavior in Matlab!

Summary of Key Concepts

States can have refinements (other modal models)

- OR states
- AND states

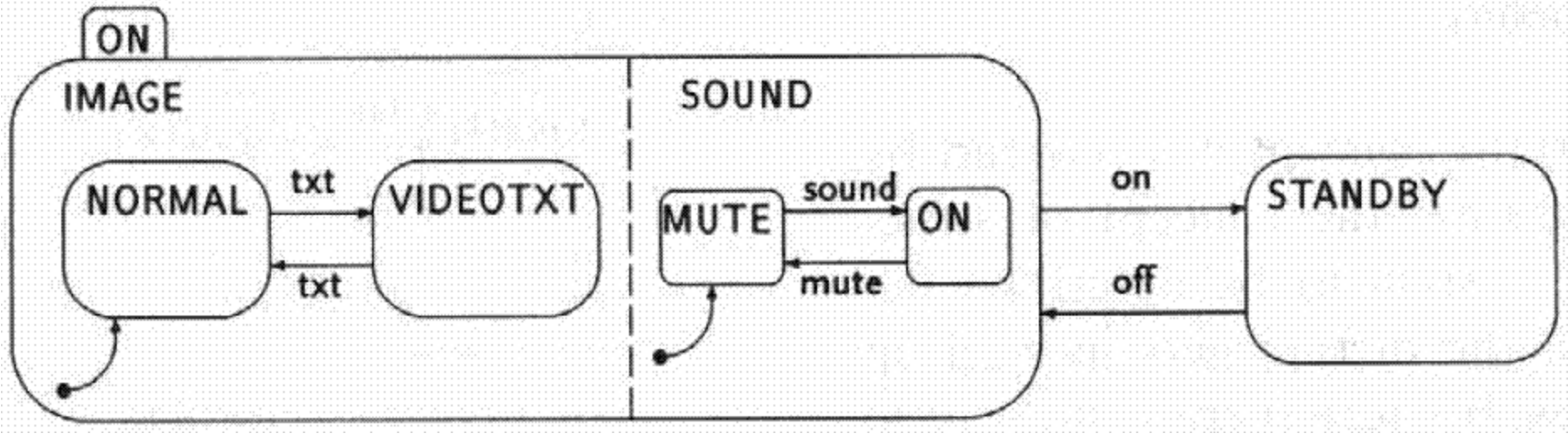
Different types of transitions:

- History
- Reset
- Preemptive

Hierarchical FSMs + Synchronous Composition: Statecharts [Harel 87]

Modeling with

- Hierarchy (OR states)
- Synchronous composition (AND states)
- Broadcast (for communication)



Example due to Reinhard von Hanxleden

Summary

- Composition enables building complex systems from simpler ones.
- Hierarchical FSMs enable compact representations of large state machines.
- These can be converted to single flat FSMs, but the resulting FSMs are quite complex and difficult to analyze by hand.
- Algorithmic techniques are needed to analyze large state spaces (e.g., *reachability analysis* and *model checking*, see Chapter 13 of Lee & Seshia).