

Statecharts: Supplemental Notes

Justin Ray

18-649 Spring 2009

Clarifications

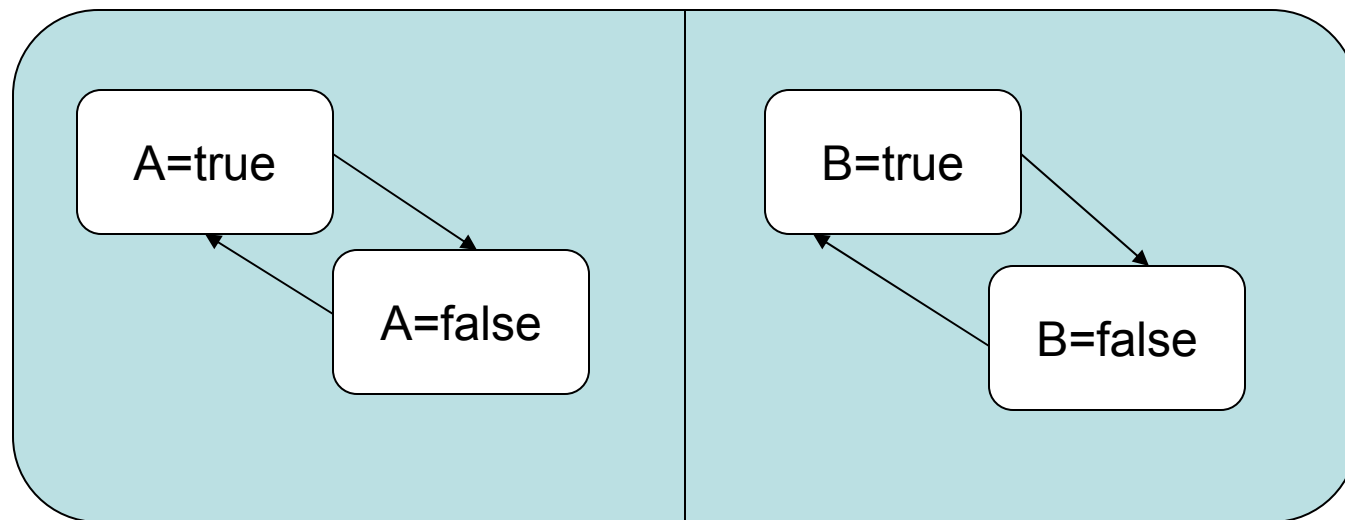
- Transition conditions
- Concurrent statecharts
- Nested statecharts
- States vs. state variables
- Changes to projects

Transition Conditions

- Statechart conditions shall contain
 - Guard conditions **only!**
 - No actions on transitions
- Even though actions are allowed by UML, this forces the time-triggered design
- If you feel you must execute an action on a transition, use an intermediate state.

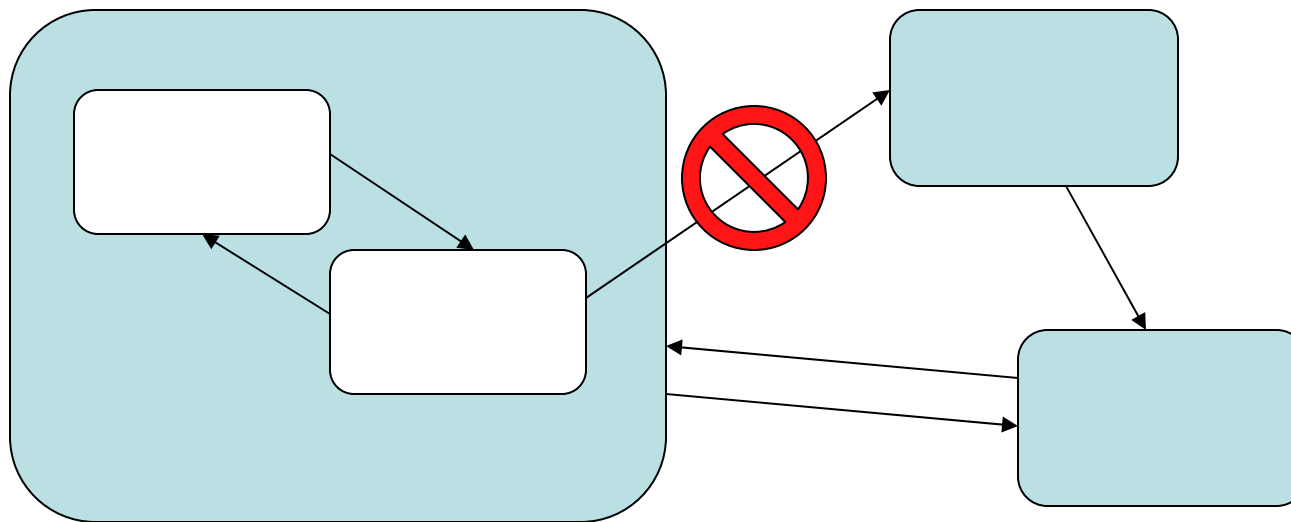
Concurrent Statecharts

- OK to have two or more statecharts executing in parallel
- Parallel statecharts shall not write to the same outputs or state variables



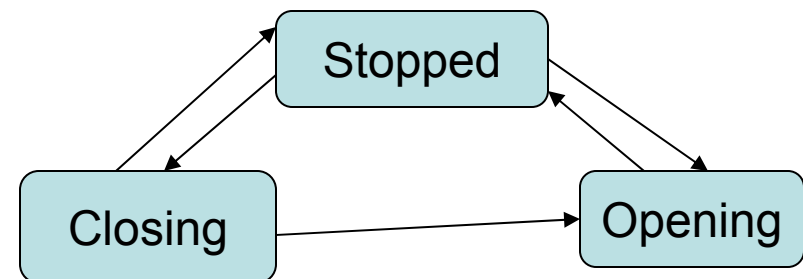
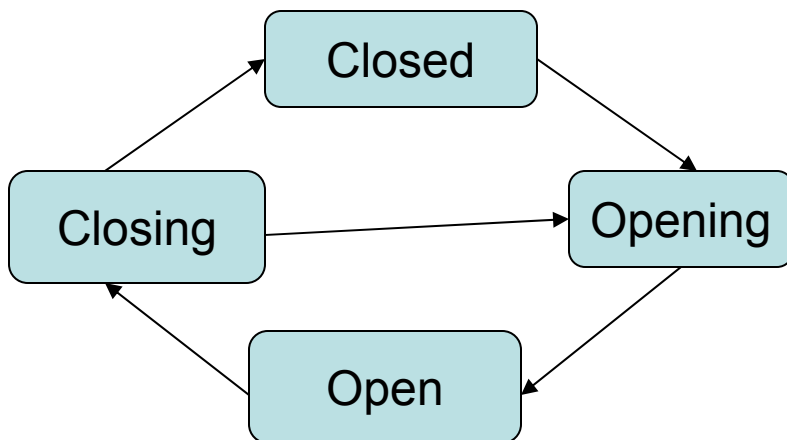
Nested Statecharts

- Avoid using them!
 - Difficult to implement in code
 - Requires multiple, nested switch statements
- If you *must* use them
 - you may not have a transition out of the superstate from an inner state



States vs. State Variables

- State variables are appropriate for:
 - Integers (counters, floors numbers, etc)
- NOT suitable for:
 - Boolean flags (doorsClosed)
- Statechart should represent the state of the door, not the state of the door motor



Changes to Projects

- It is to your advantage to update statecharts now, before you write code:
 - Fewer bugs
 - Update sooner → Less rework later
- You **must** fix your statechart by Project 7
- Your submitted code and design must meet all traceability requirements!