

Embedded Systems

Bernd Finkbeiner Calogero Zarba

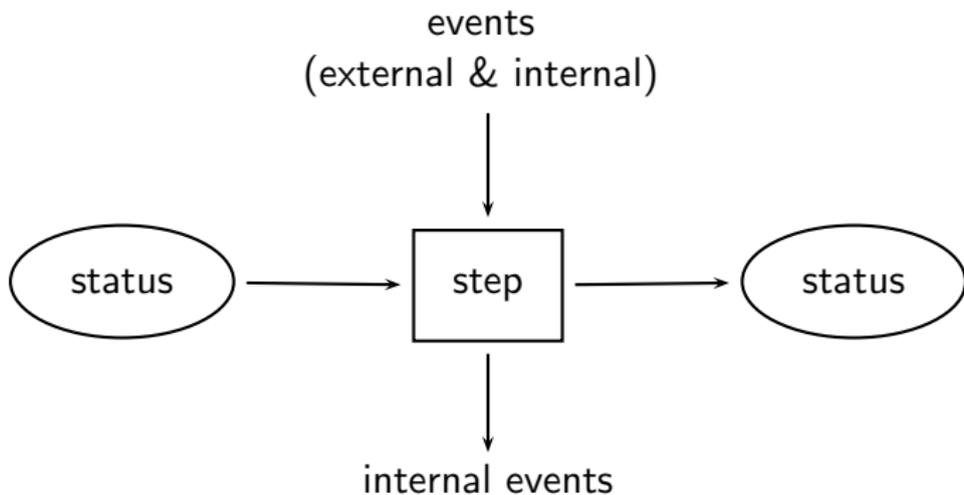
Universität des Saarlandes

Summer Semester 2007

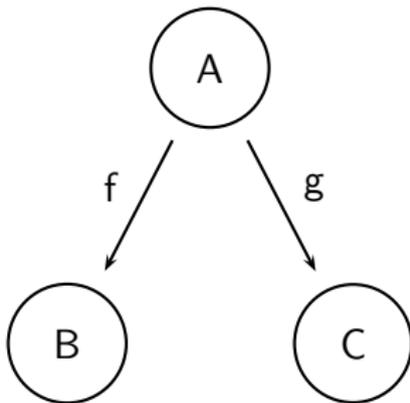
StateCharts

- Introduced by [Harel, 1987]
- Extend **finite state machines** with
 - ① Hierarchy
 - ② Concurrency
 - ③ Time
- Memory is shared, i.e., parallel processes see changes to variables immediately
- Applications: local, control dominated systems; not suited for distributed systems.

StateCharts semantics

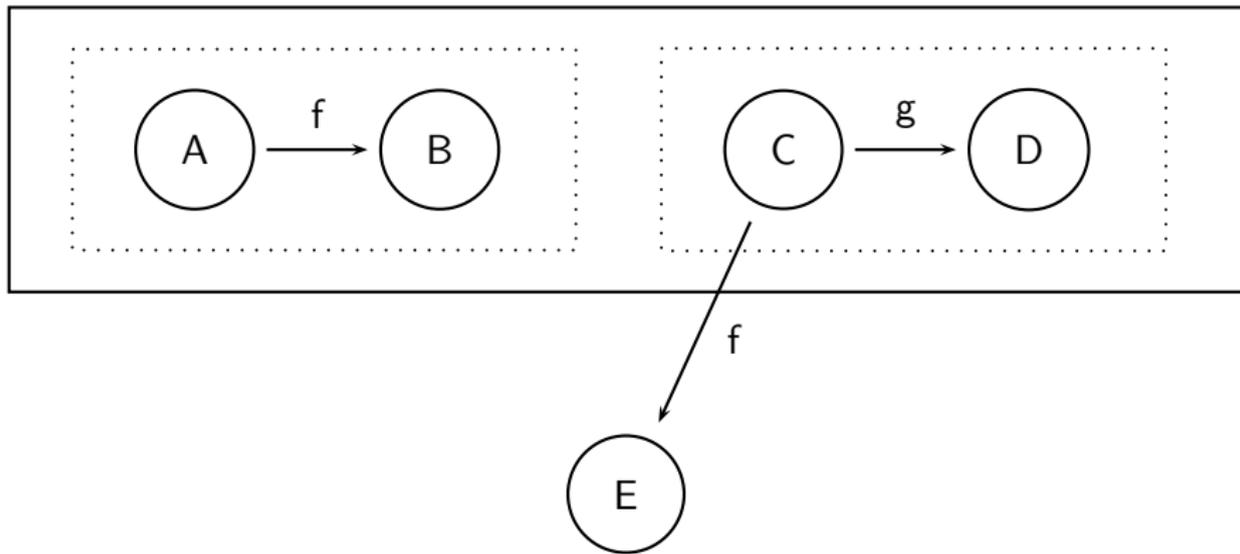


Nondeterminism



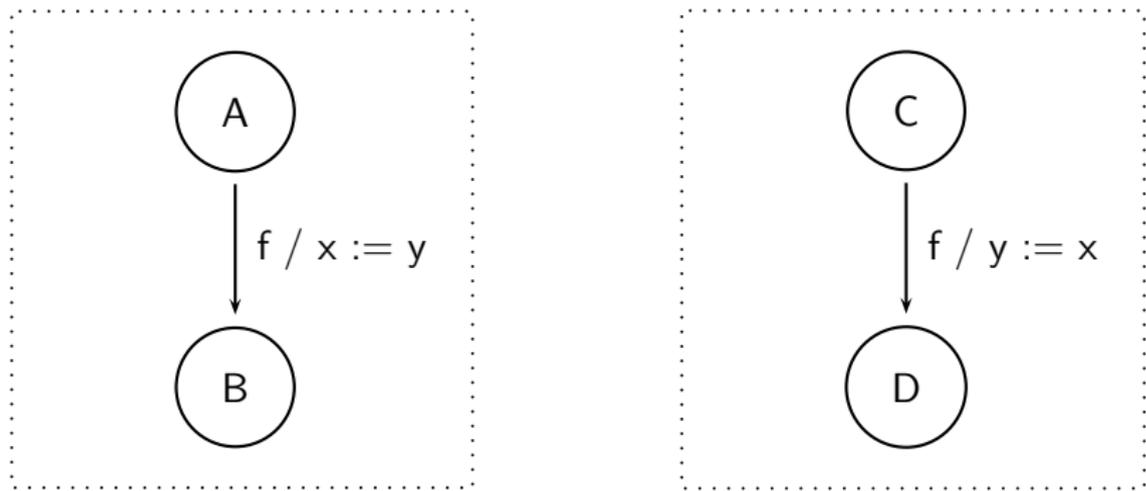
- Current status: A is active
- Assume both events f and g are alive
- Next status: Either B or C is active **but not both**

Priority



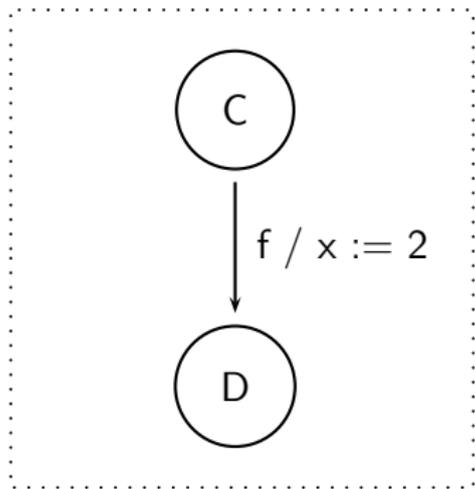
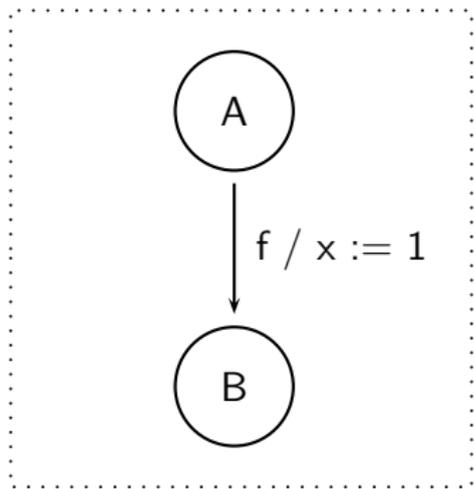
- Current status: AND-superstate is active
- Assume event f is alive
- Next status: AND-superstate is not active; **right transition has higher priority than left one**

Swapping two variables



- Current status: A and C are active, $x = 1$, $y = 2$
- Assume event f is alive
- Next status: Both B and D are active; $x = 2$, $y = 1$; **variables get swapped**

Write racing conditions



- Current status: A and C are active
- Assume event f is alive
- Next status: Either $x = 1$ or $x = 2$; write race condition causes nondeterministic behavior

Status

- Current time
- Values of variables
- Values of history connectors
- Active events, generated either internally in the previous step, or externally by the environment
- A consistent set of active states, that is
 - ① If the chart is in OR-superstate S , then the chart is in **exactly one** of the substates of S
 - ② If the chart is in AND-superstate S , then the chart is in **all** of the substates of S

Step

- **First stage: Select**

Select, nondeterministically, a maximal set of enabled transitions that leads to a consistent set of states.

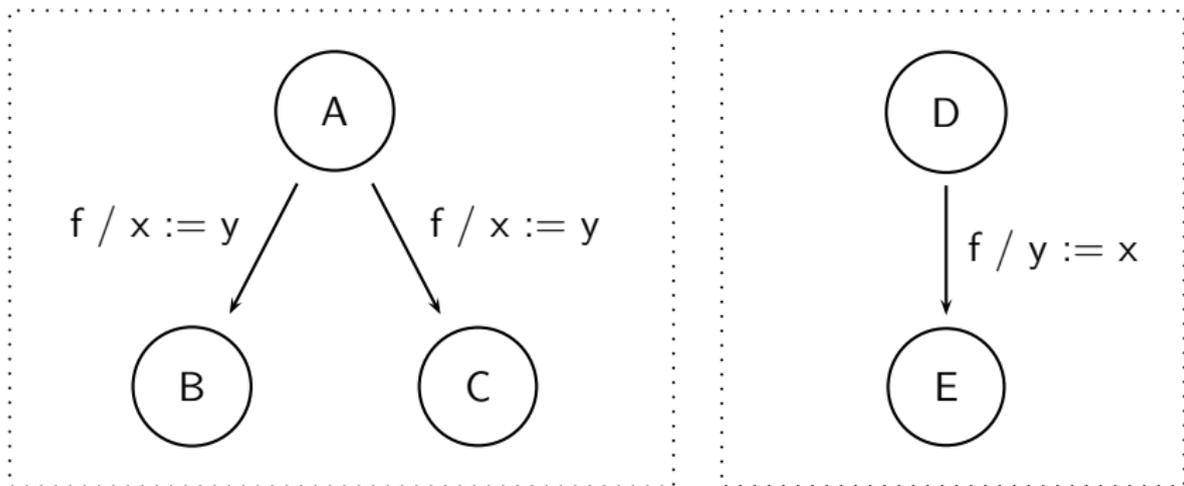
- **Second stage: Compute**

Compute the next states, the generated events, and the right hand sides of assignments (do not yet perform assignments)

- **Third stage: Activate**

Assignments become effective, history connectors are updated, next states become active.

Example



- Current status: A and D are active, $x = 1$, $y = 2$
- Assume event f is alive
- **First stage:** Select one transition on the left, and the one on the right
- **Second stage:** Let $x' = 2$ and $y' = 1$
- **Third stage:** Next status becomes effective; either B or C is active but not both, E is active, $x = 2$, $y = 1$