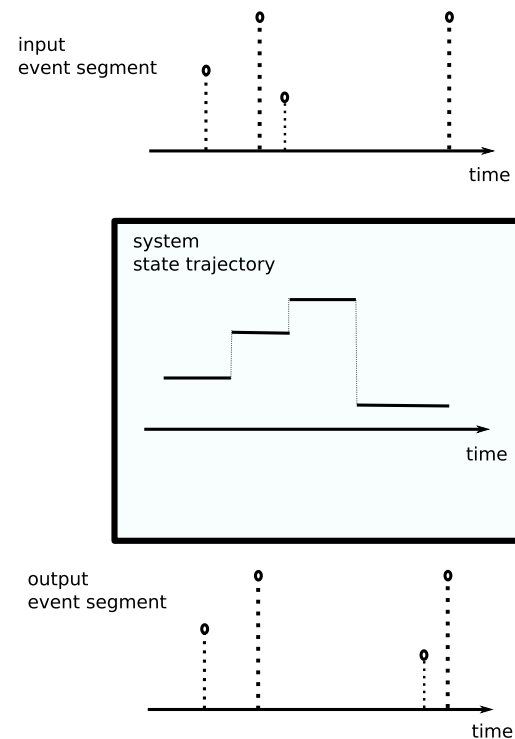# Statecharts

Used to model *behaviour* of Autonomous and Reactive Systems
(event driven, react to internal –timeouts– and external stimuli)

Statecharts

- Visual formalism yet formally and rigourously defined for

  1. documentation

  2. analysis (model checking as well as testing)

  3. simulation

  4. code synthesis

- Solves FSA problems (by combining with Higraphs):

  1. flat $\Rightarrow$ hierarchy ($\Rightarrow$ re-use)

  2. represent large number of transitions concisely

  3. represent large number of states concisely

Statecharts references

- David Harel. **Statecharts: A Visual Formalism for Complex Systems.** Science of Computer Programming. vol. 8. 1987. pp. 231 - 274.

- David Harel. **On Visual Formalisms.** Communications of the ACM. Volume 31, No. 5. 1988. pp. 514 - 530.

- David Harel and Amnon Naamad. **The STATEMATE semantics of statecharts.** ACM Transactions on Software Engineering and Methodology (TOSEM) vol. 5. Issue 4. October 1996. pp.293 - 333.

- David Harel and Hillel Kugler. **The Rhapsody Semantics of Statecharts (or, On the Executable Core of the UML).** Springer, Lecture Notes in Computer Science 3147. 2004. pp. 325 - 354.
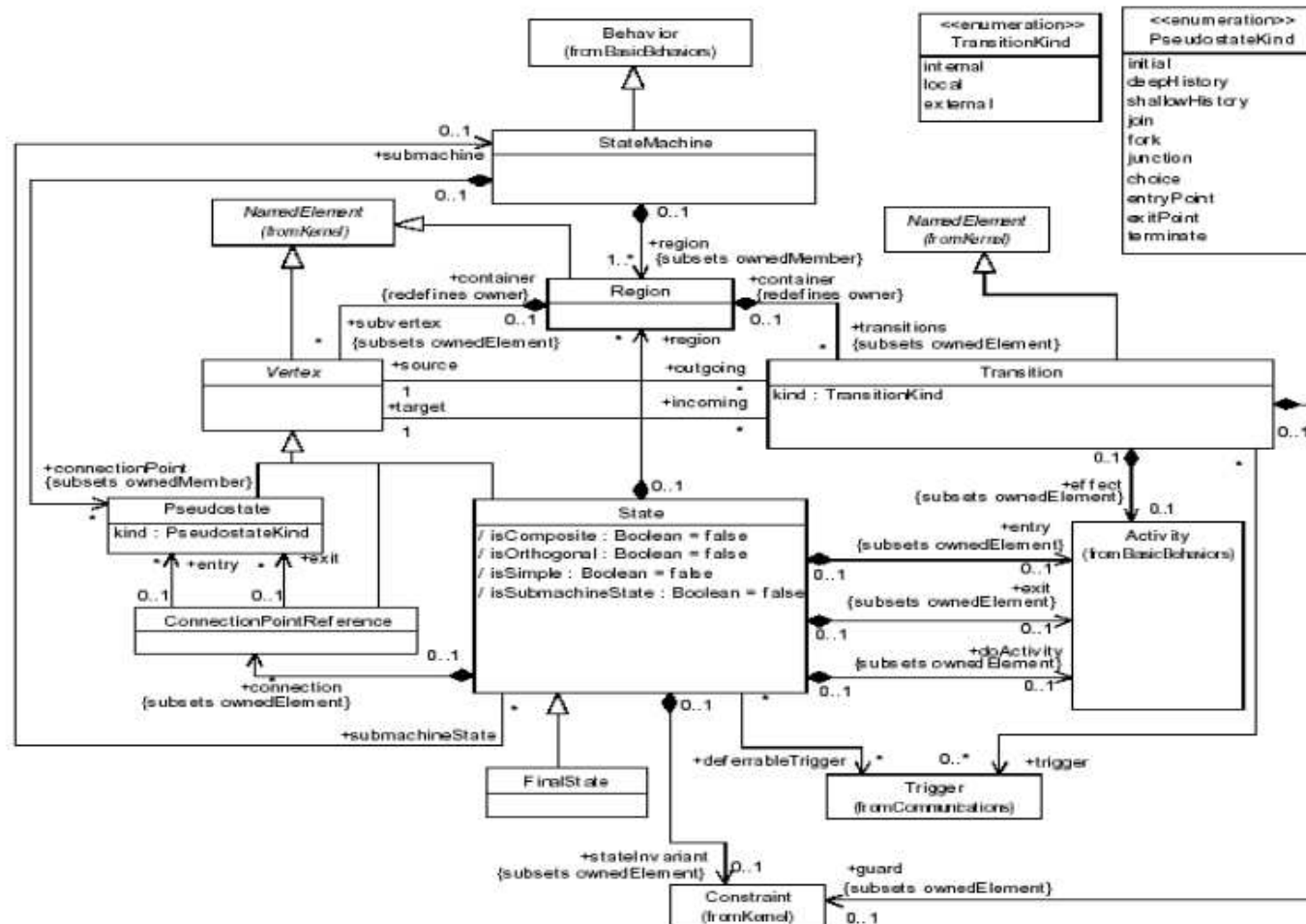
Statecharts tools

- STATEMATE [http://www-03.ibm.com/software/products/en/ratistat]

- Rhapsody [https://en.wikipedia.org/wiki/Rational_Rhapsody]

- Stateflow [http://www.mathworks.com/products/stateflow/]

- Yakindu [https://www.itemis.com/en/yakindu/state-machine/]

- ArgoUML [http://argouml.tigris.org/]

- SVM/SCC [http://msdl.cs.mcgill.ca/people/tfeng/research.html]

- Papyrus-RT (UML-RT, not Statecharts!)
  [https://www.eclipse.org/papyrus-rt/]

Statecharts =

1. Deterministic Finite State Automata (FSA, DFA)

2. Depth

3. Orthogonality

4. Broadcast

5. History

6. Time

7. syntactic sugar: entry/exit actions, . . .
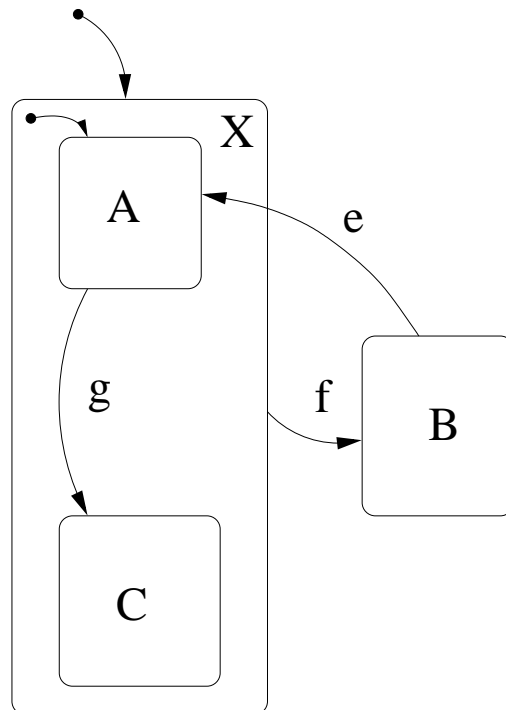
UML 2.0 Statecharts abstract syntax
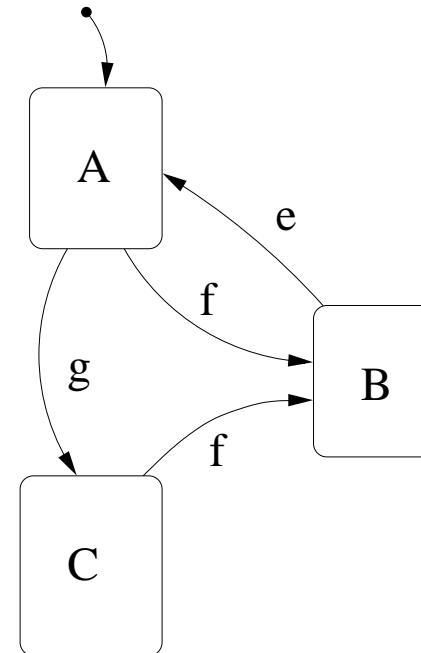
Deterministic Finite State Automata (FSA, DFA)

State

A

A
entry/ ...
do/ ...
exit/ ...

A

e

f/a

g [guard]

B

g(2,"a")

C

e

Not allowed in DFA !

Default state (pseudo state)

transition

event(params) [guard] /output_action(params)
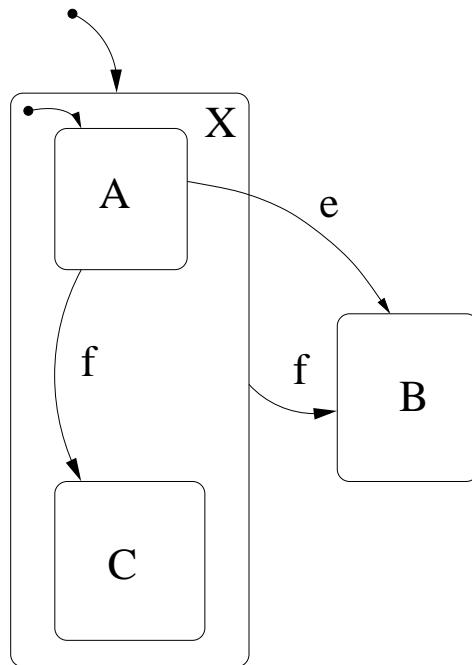
Depth (XOR)

give meaning
by "flattening"

State automaton
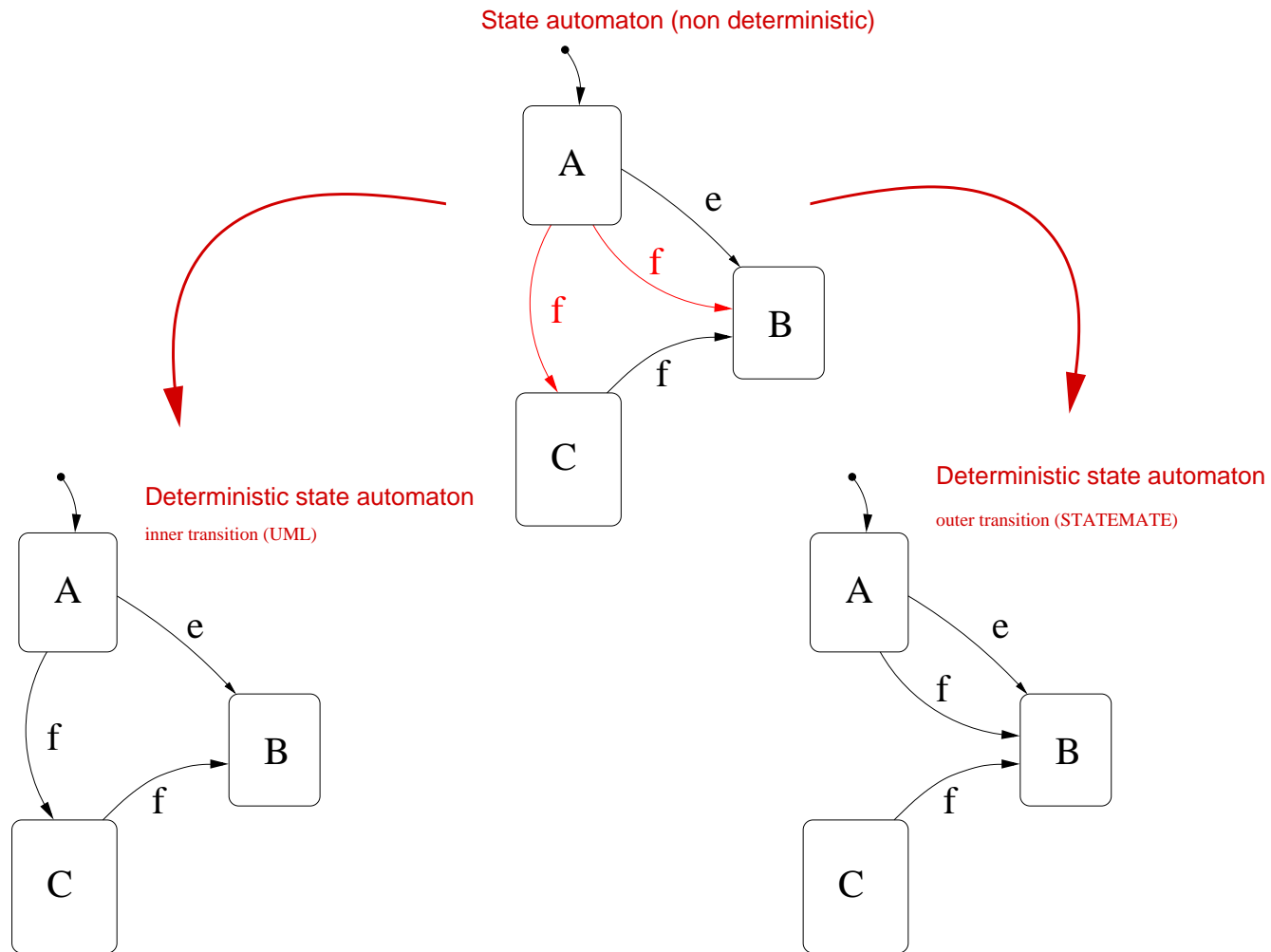
Depth and determinism

Statechart with depth

give meaning by "flattening"
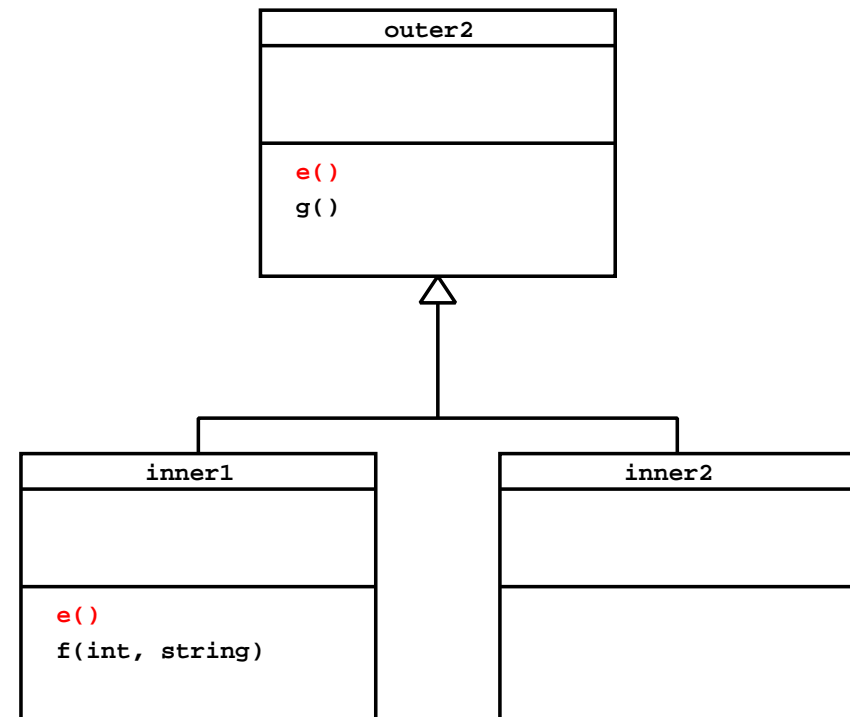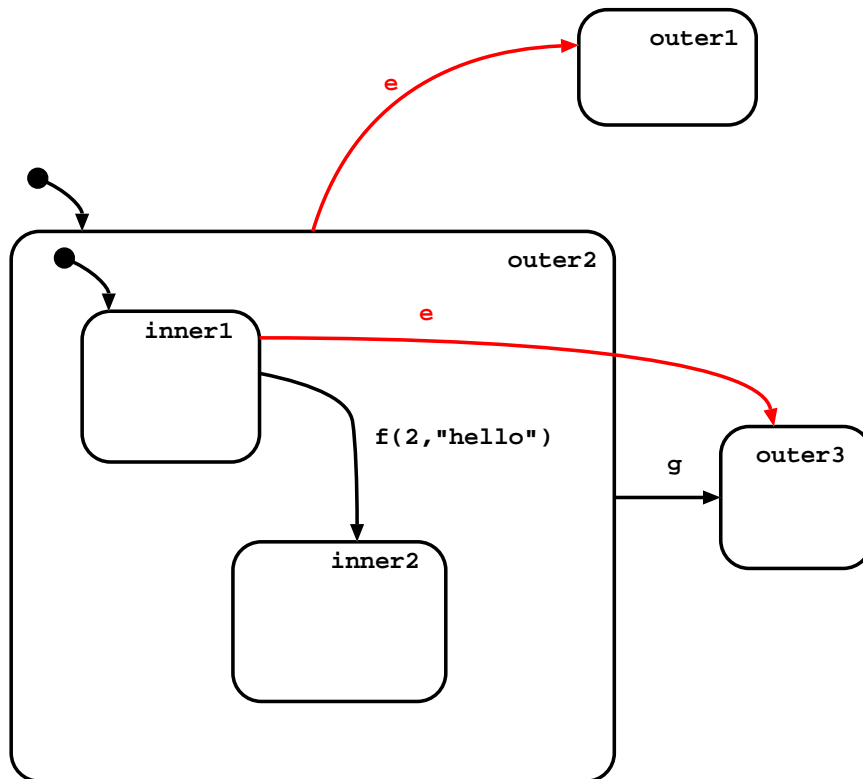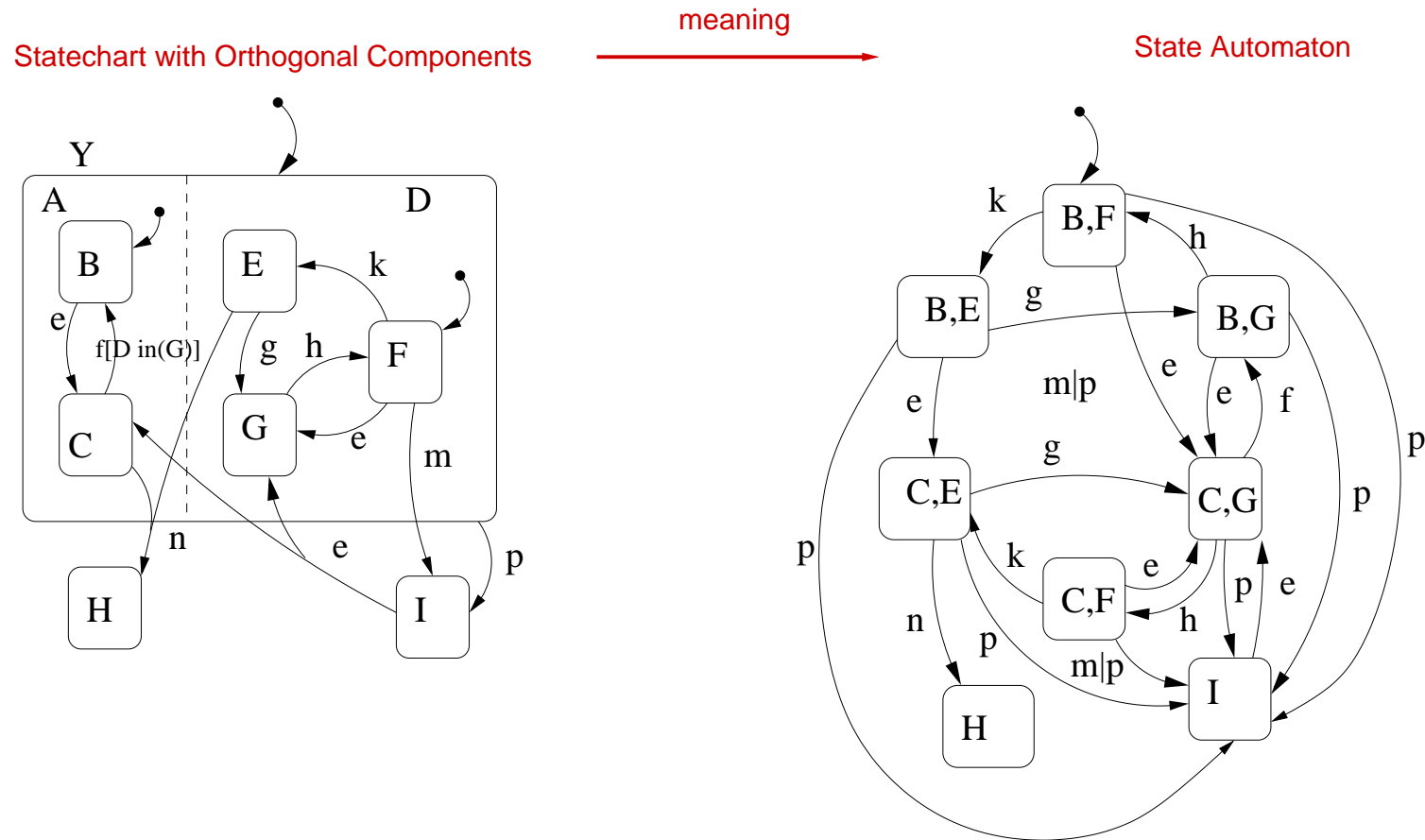
State automaton (non deterministic)

Depth and determinism

State automaton (non deterministic)

A

e

f

f

B

f

C

Deterministic state automaton

inner transition (UML)

A

e

f

B

f

C

Deterministic state automaton

outer transition (STATEMATE)

A

e

f

B

f

C
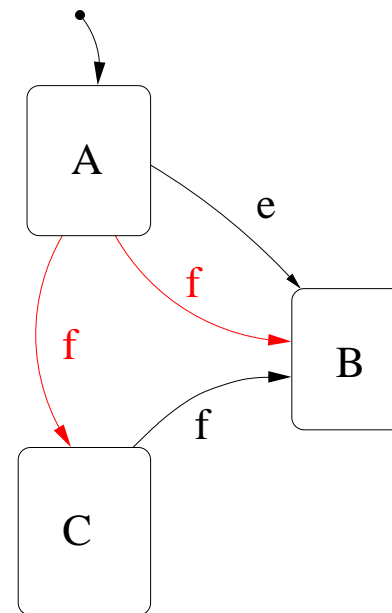
Why "inner-first" in UML Statecharts ?

Orthogonality (AND)

Statechart with Orthogonal Components
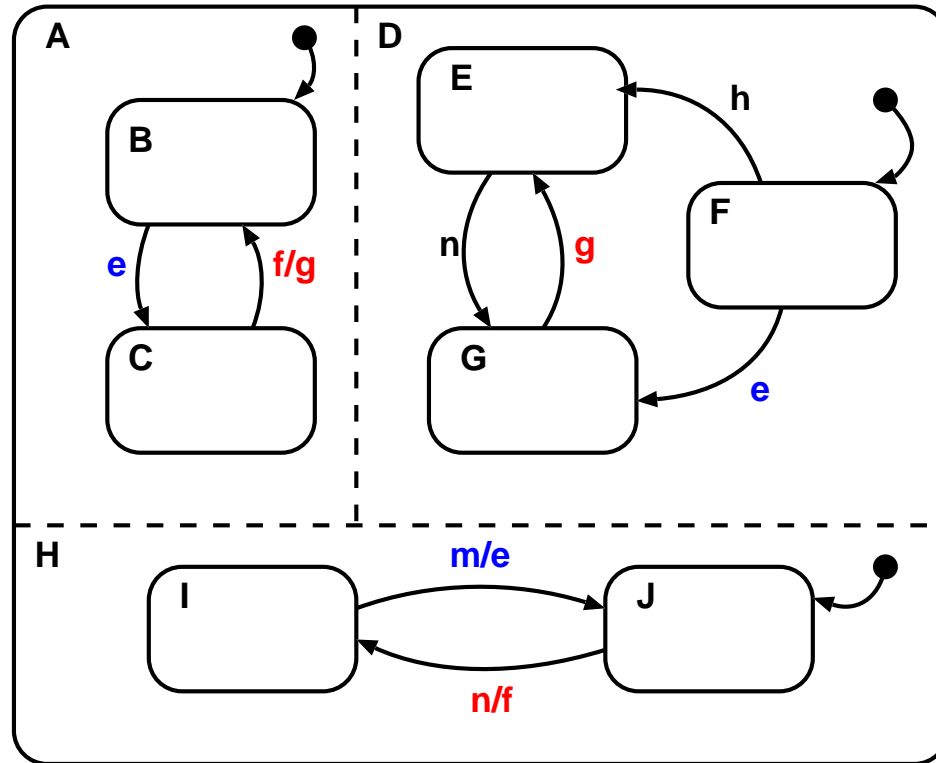
meaning

State Automaton
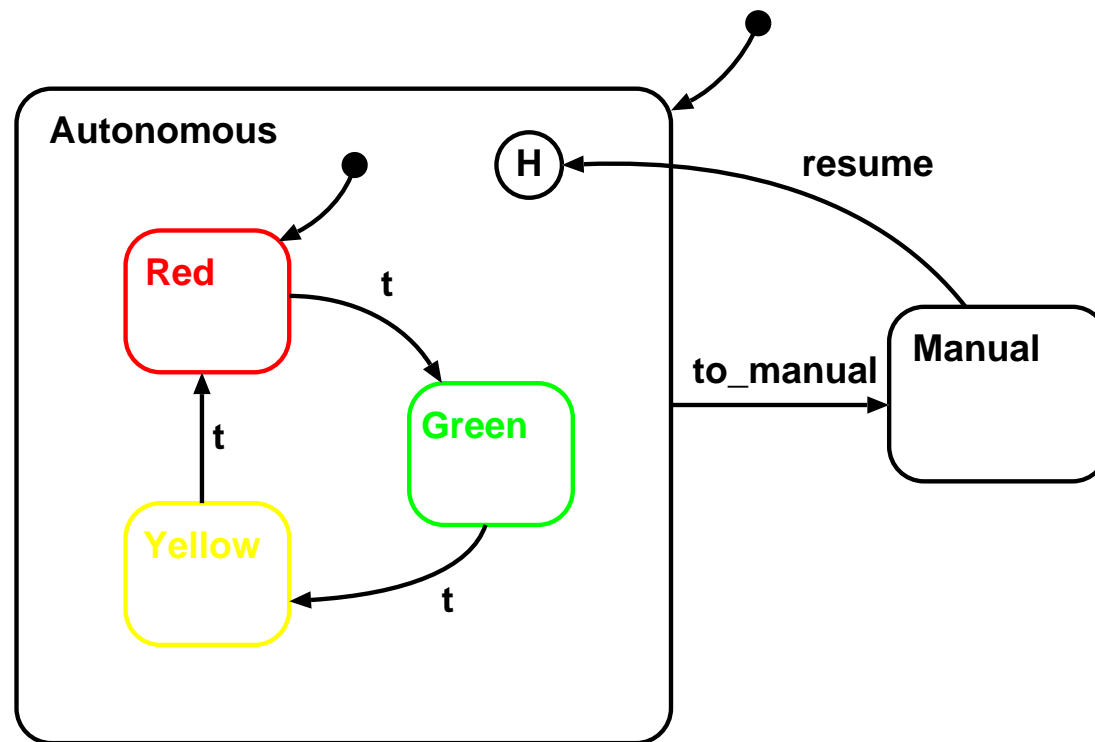
Orthogonality and synchronization
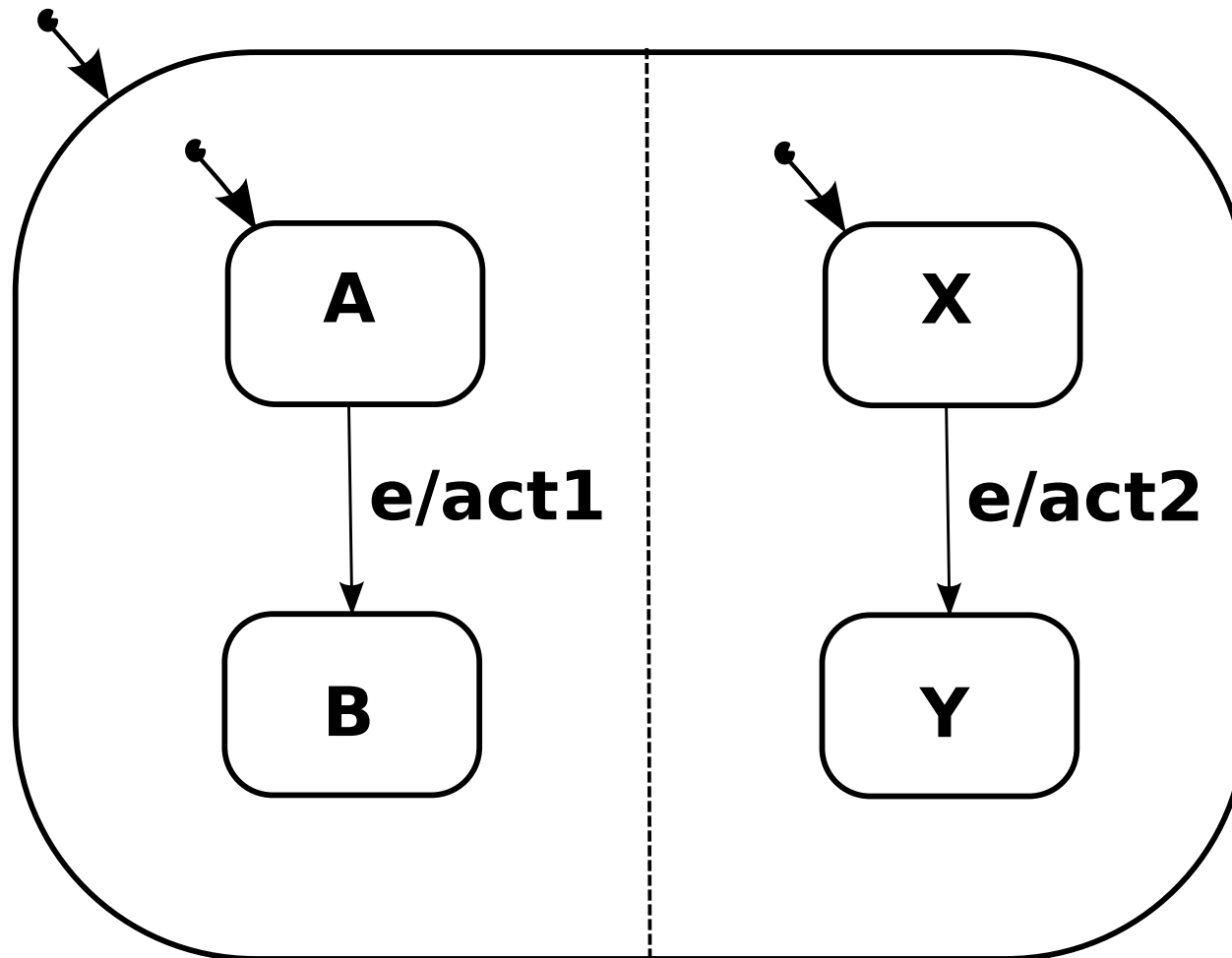
Broadcasting (output events)

Input Segment: **nmnn**

History States

Deep History