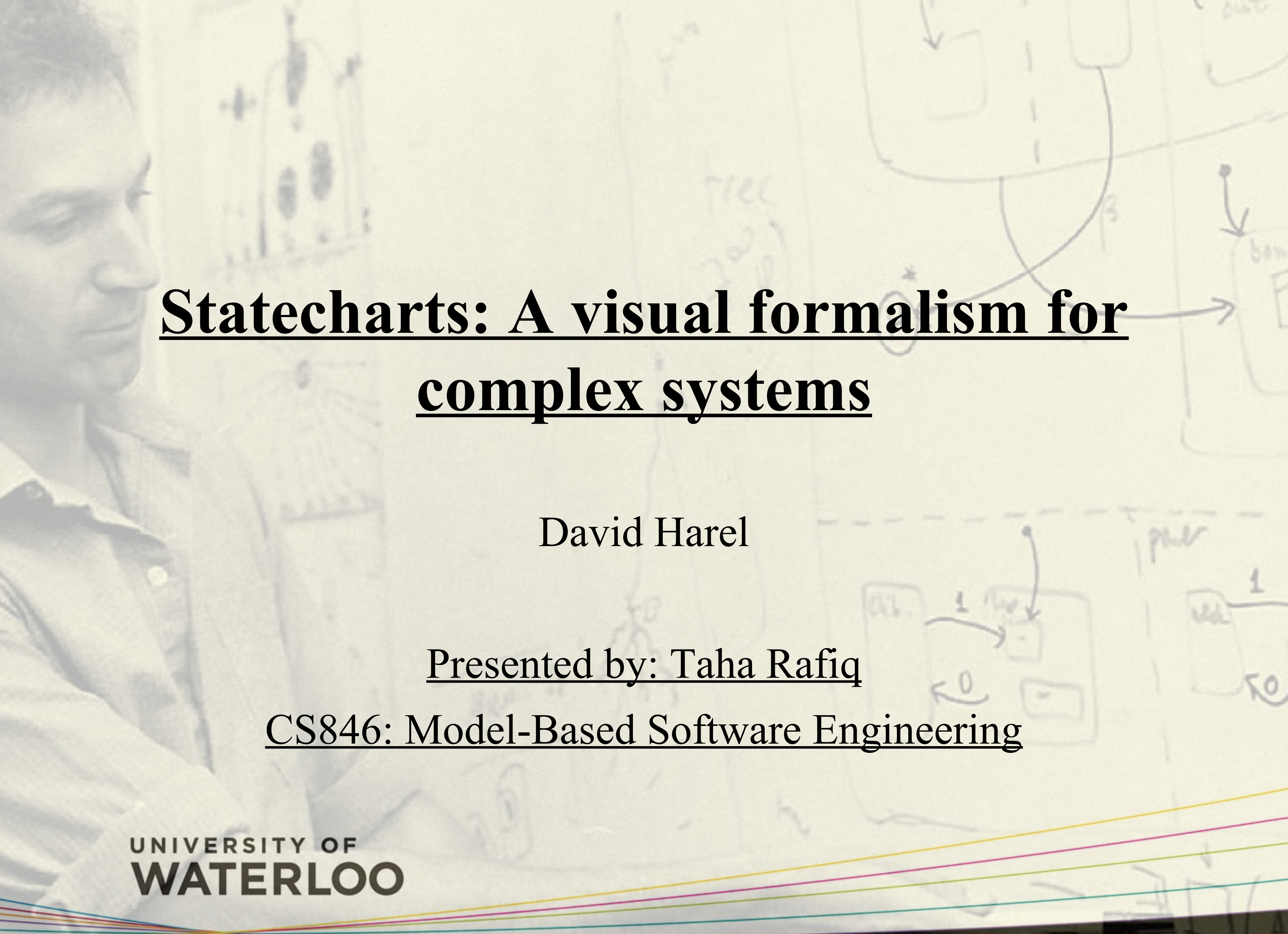


Statecharts: A visual formalism for complex systems

David Harel

Presented by: Taha Rafiq

CS846: Model-Based Software Engineering

Image Source: Statecharts in the making: A personal account

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Motivation

Source: Wikipedia

Motivation

- The author was a consultant for IAI
- Involved with design specification of fighter aircraft – the Lavi
- Interactions with the avionics team
- What happens when you press a button under a certain set of circumstances?
    - Incomplete/Inconsistent/Incomprehensible specification – who decides?

Motivation

"How should an engineering team specify the behavior of such a complex reactive system in an intuitively clear yet mathematically rigorous fashion? This was what I aimed to try to answer."

- David Harel, Statecharts in the making: A personal account

UNIVERSITY OF
WATERLOO

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

UNIVERSITY OF WATERLOO

What is a Reactive System?

- Main behavior – Reactivity
- Event-driven, control-driven, event-response nature
- Often highly parallel behavior
- Behavior is specified by set of allowed
  - Input/Output events
  - Conditions
  - Actions
  - Timing constraints

Specifying the Behavior of a Reactive System

- States & Events – natural medium
- General form
  - When event ***a*** occurs in state ***A***, if condition ***C*** is true, the system transfers to state ***B***
- **Finite State Machines** = formal mechanism for describing such interactions

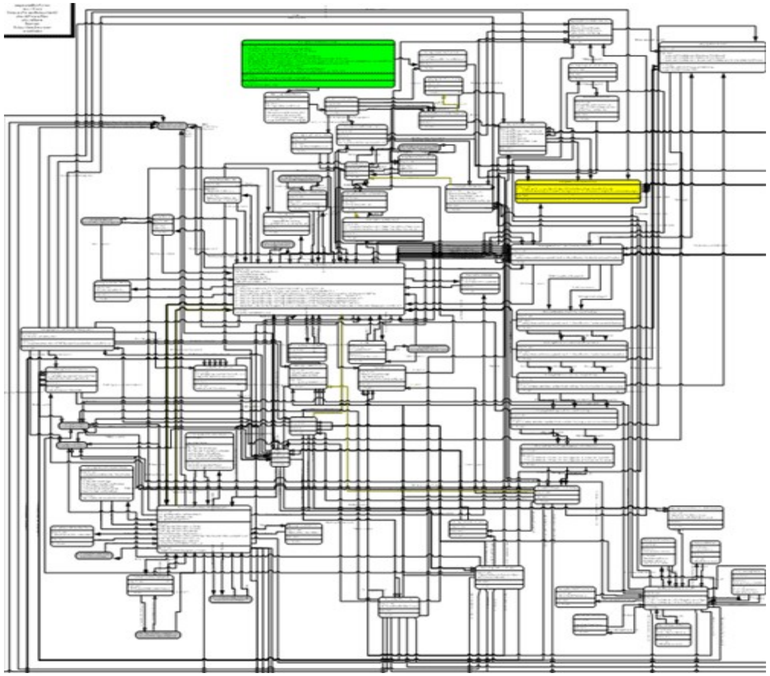
Problems with FSMs

- Complex system (fighter aircraft)
  - Unmanageable, exponentially growing states
  - Flat, unstructured and chaotic diagram

What are Statecharts?

- Extension of traditional state diagrams
- Visual formalism for states and transitions
  - Modular
  - Clustering
  - Concurrency
  - Levels of abstraction
- **Statecharts = state-diagrams + depth + orthogonality + broadcast-medium**

What are Statecharts?



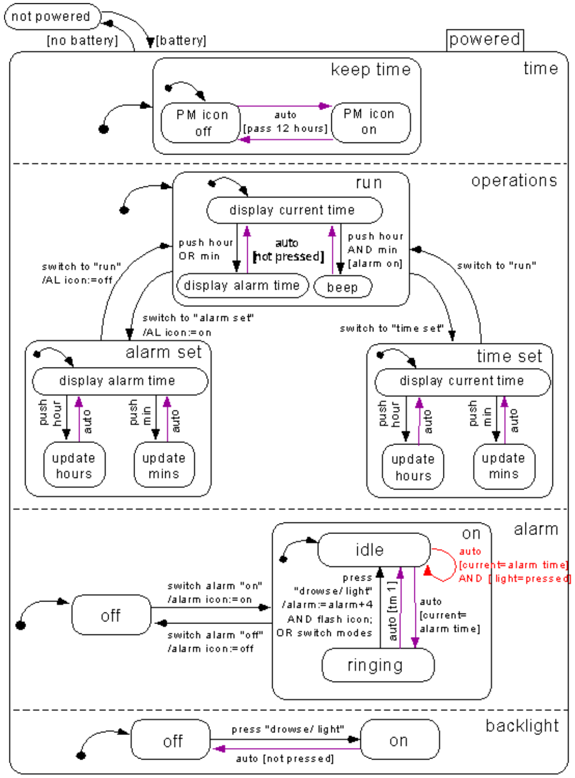
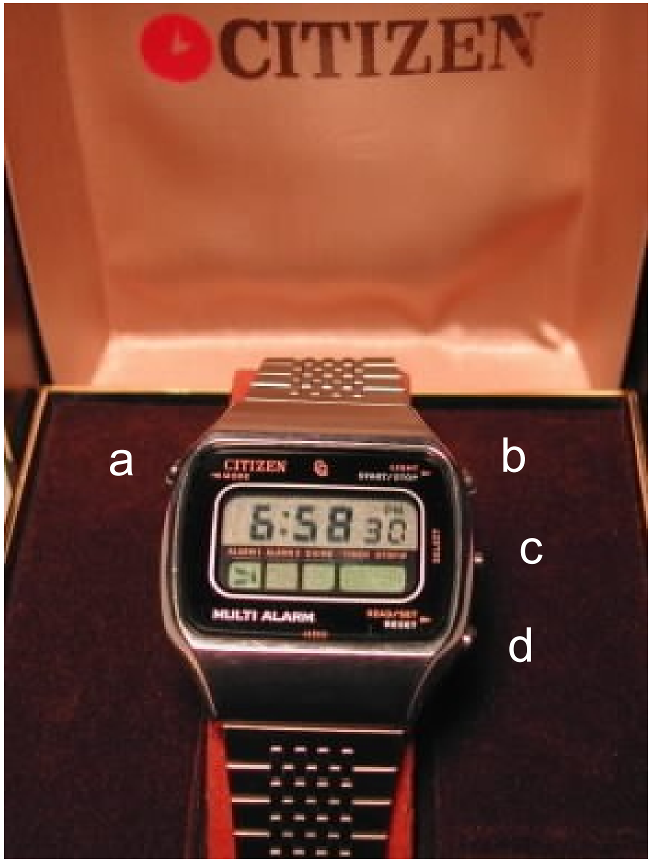


Figure 3. The Conceptual Model

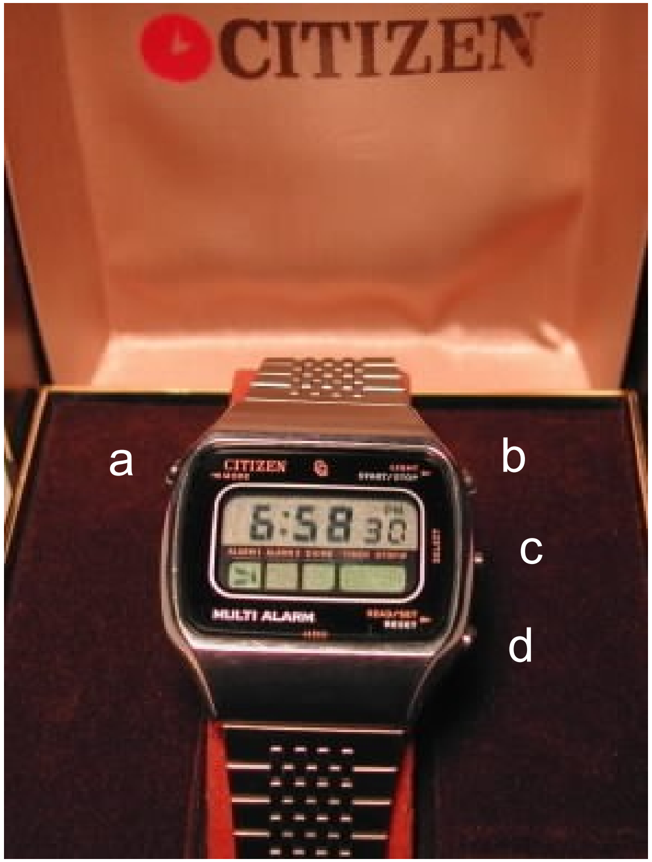
Running Example



Citizen Quartz Multi-Alarm III Wristwatch

- 4 buttons: *a*, *b*, *c*, *d*
- Time + date
- Chime (hour beep)
- 2 alarms
- Stopwatch
- Light
- Weak battery indication
- Beeper test

Running Example



Main Events

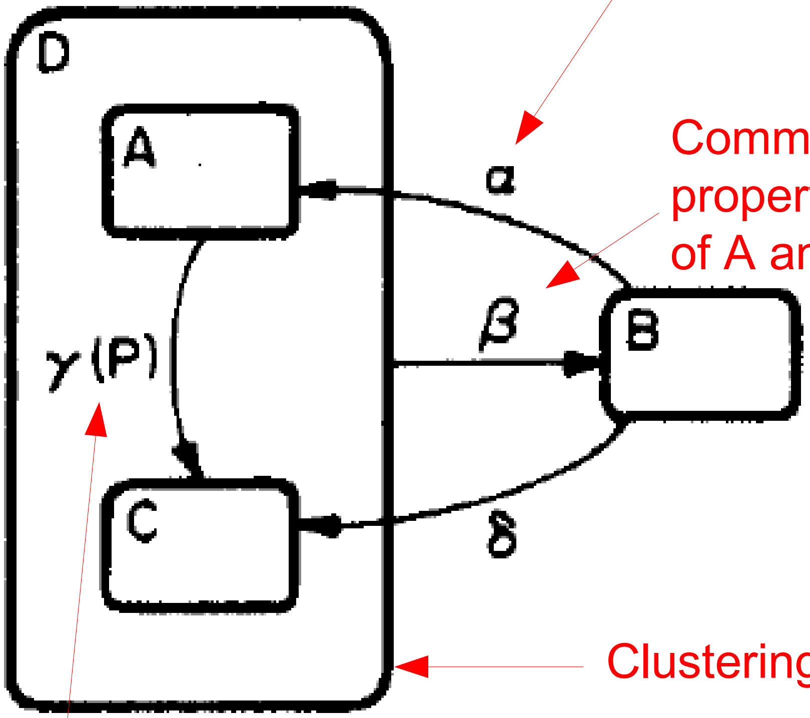
- Depressing of button (*a*)
- Releasing of button (*â*)
- Internal events
  - Timed events
  - Battery events

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Basics

- Encapsulation expresses hierarchy
- Arrows originate and terminate at any level
- Clustering represents *XOR* (Abstraction)
  - *D* is *XOR* of *A* and *C*



Event

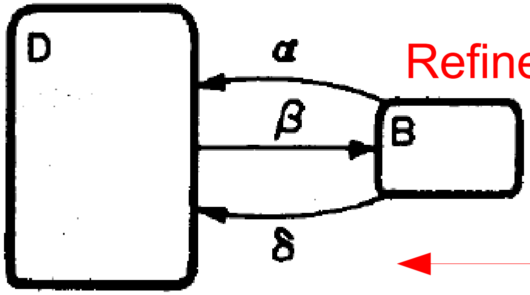
Common property of A and C

Clustering

Condition

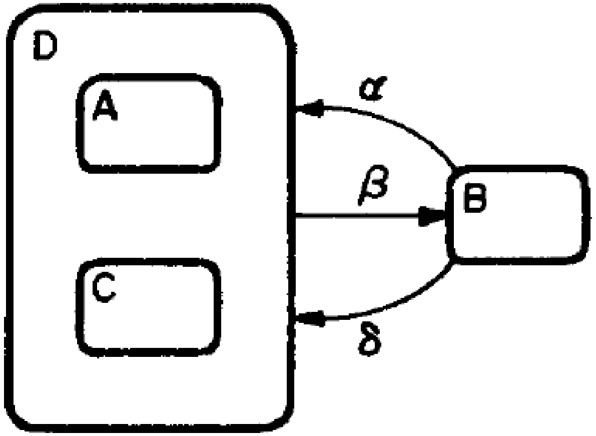
Zooming In and Zooming Out

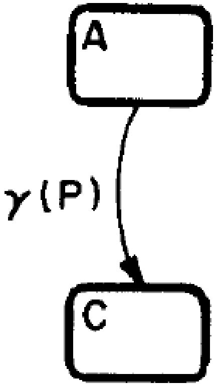
Zooming out of *D*



Refinement

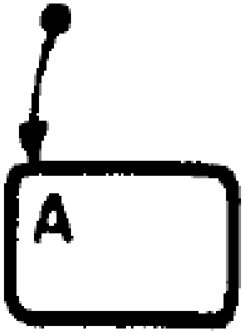
Abstraction



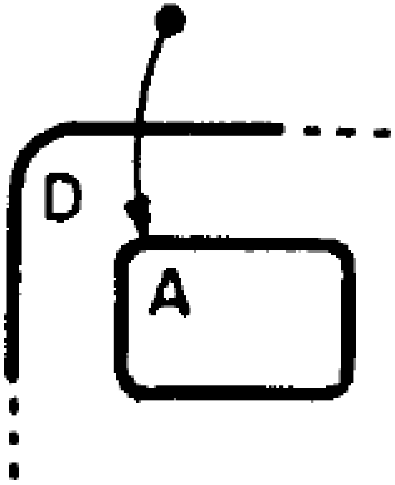


Zooming into *D*

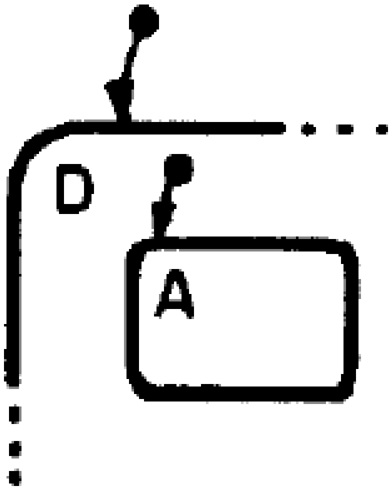
Default States



(i)



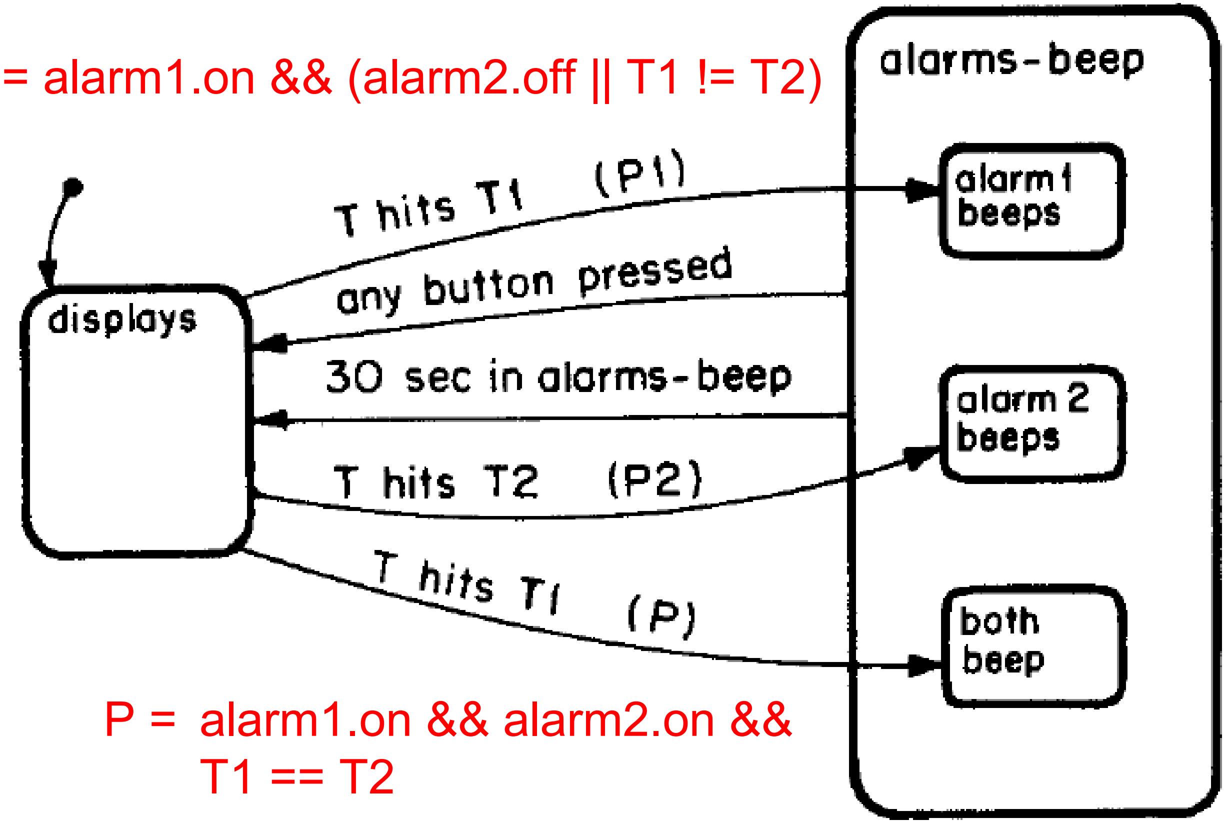
(ii)



(iii)

Advantageous for zooming

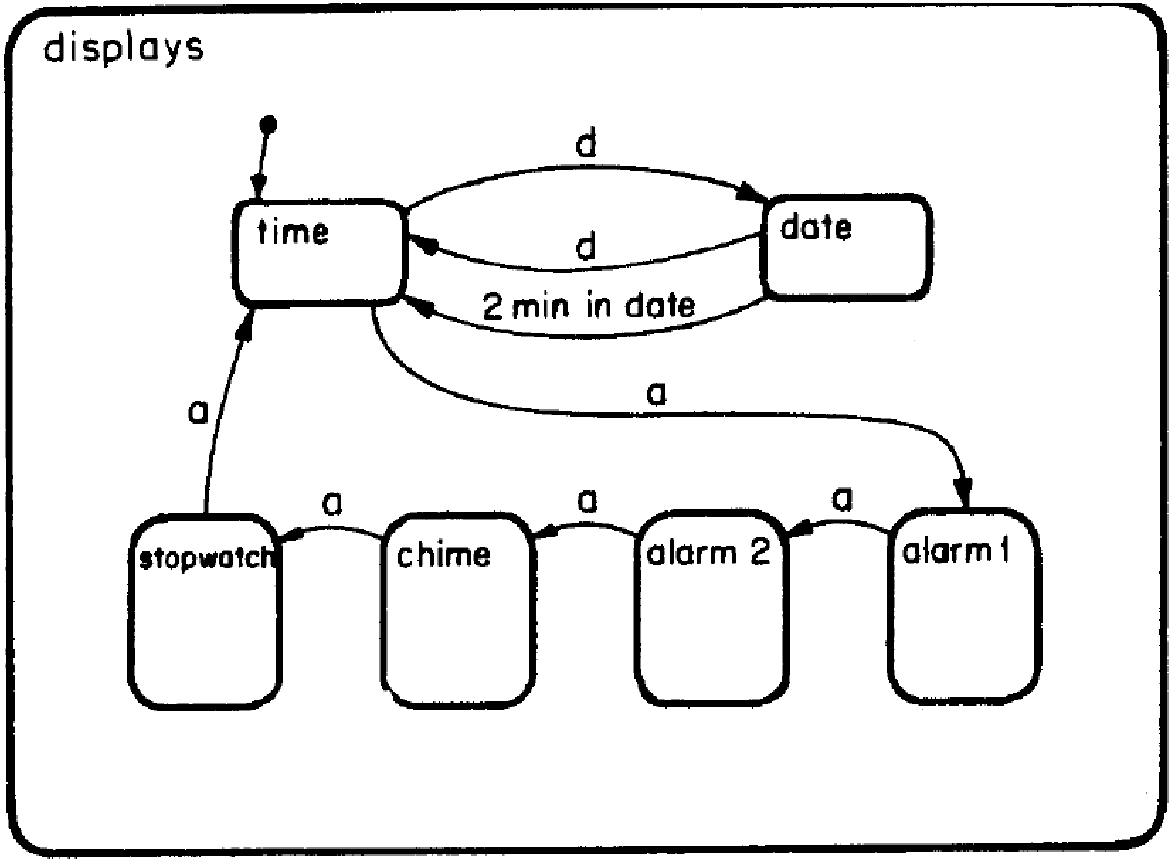
Watch Example



P1 = alarm1.on && (alarm2.off || T1 != T2)

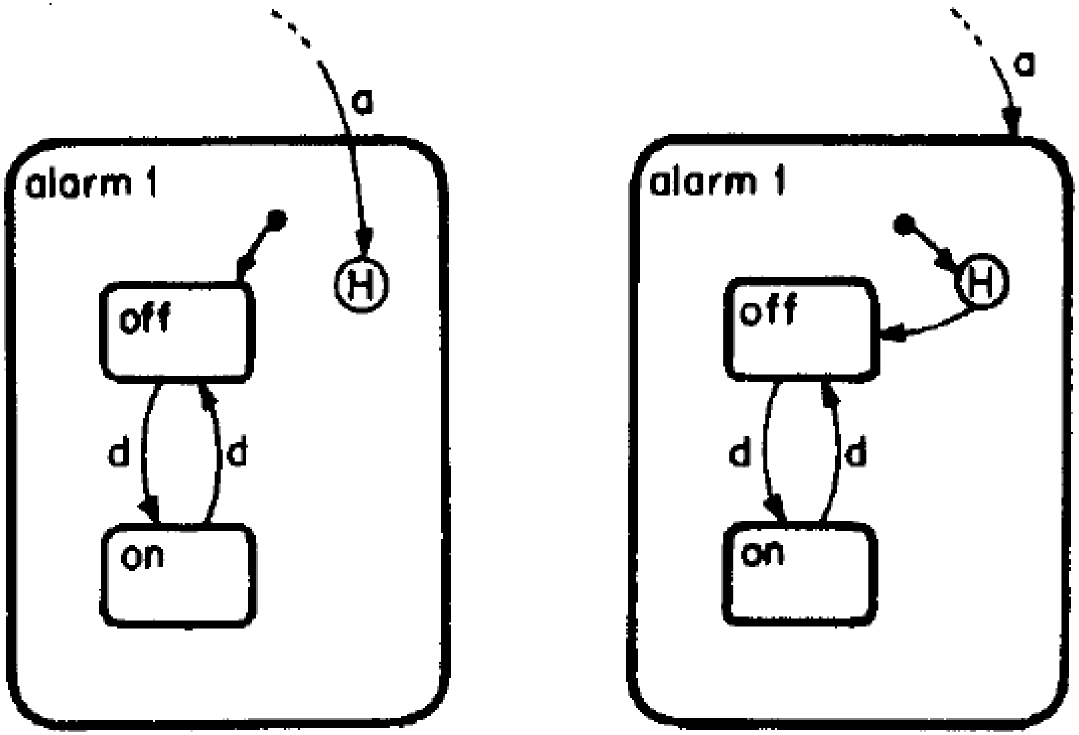
P = alarm1.on && alarm2.on &&
T1 == T2

Refinement of Displays State



History Connective

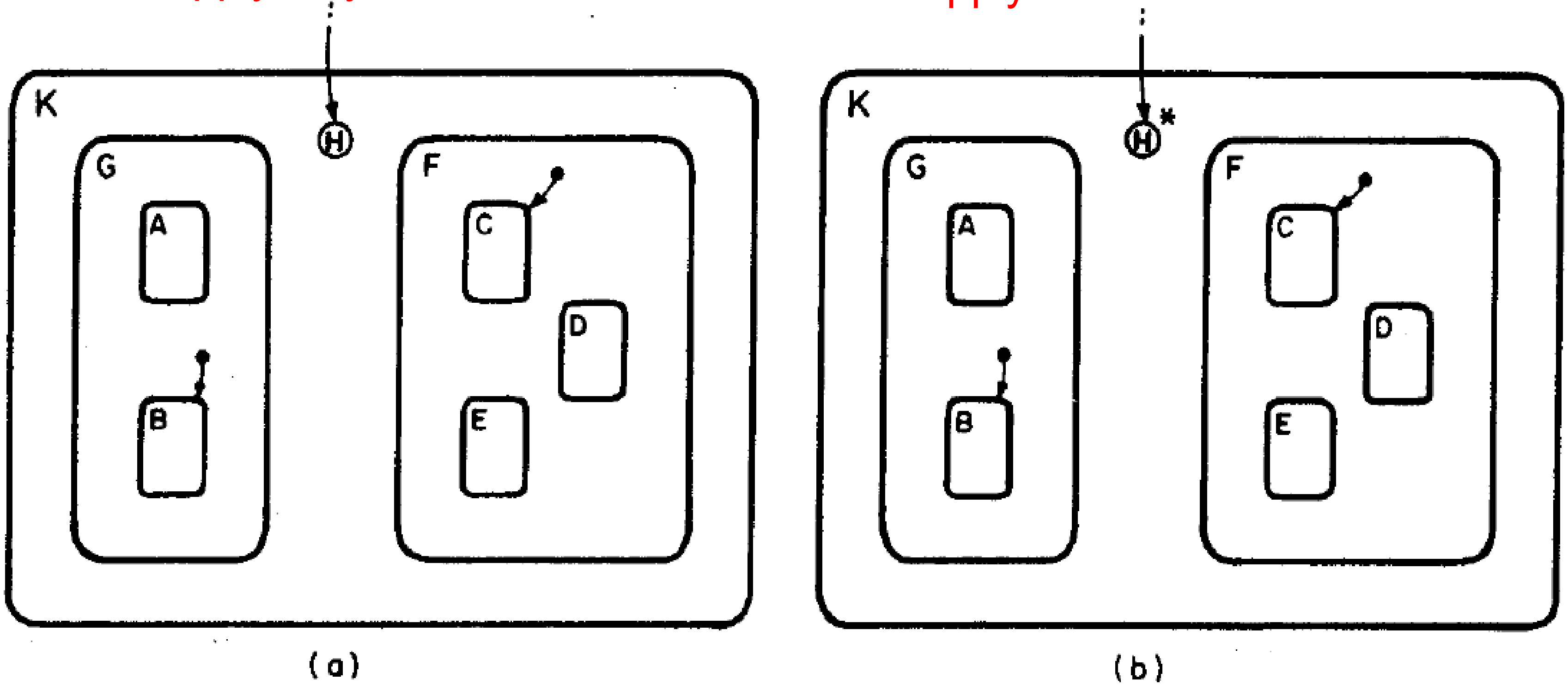
Enter *off* first time, else enter last visited state



(a)

(b)

History Connective - Levels



Apply only at level *K*

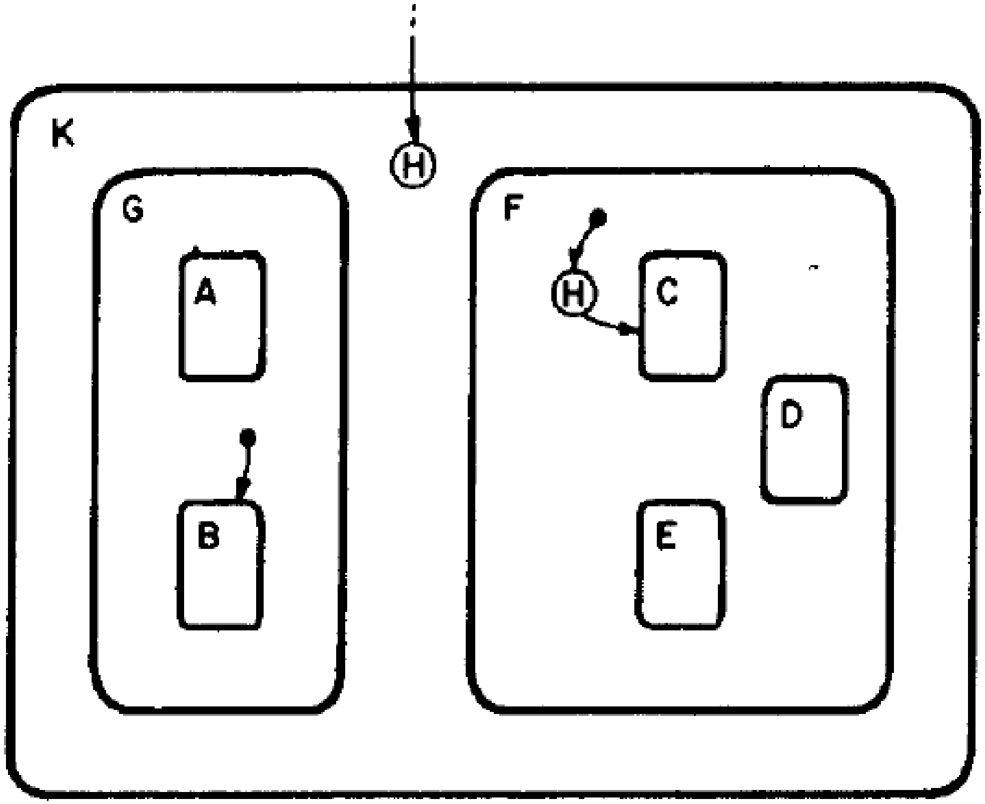
Apply at all contained levels

(a)

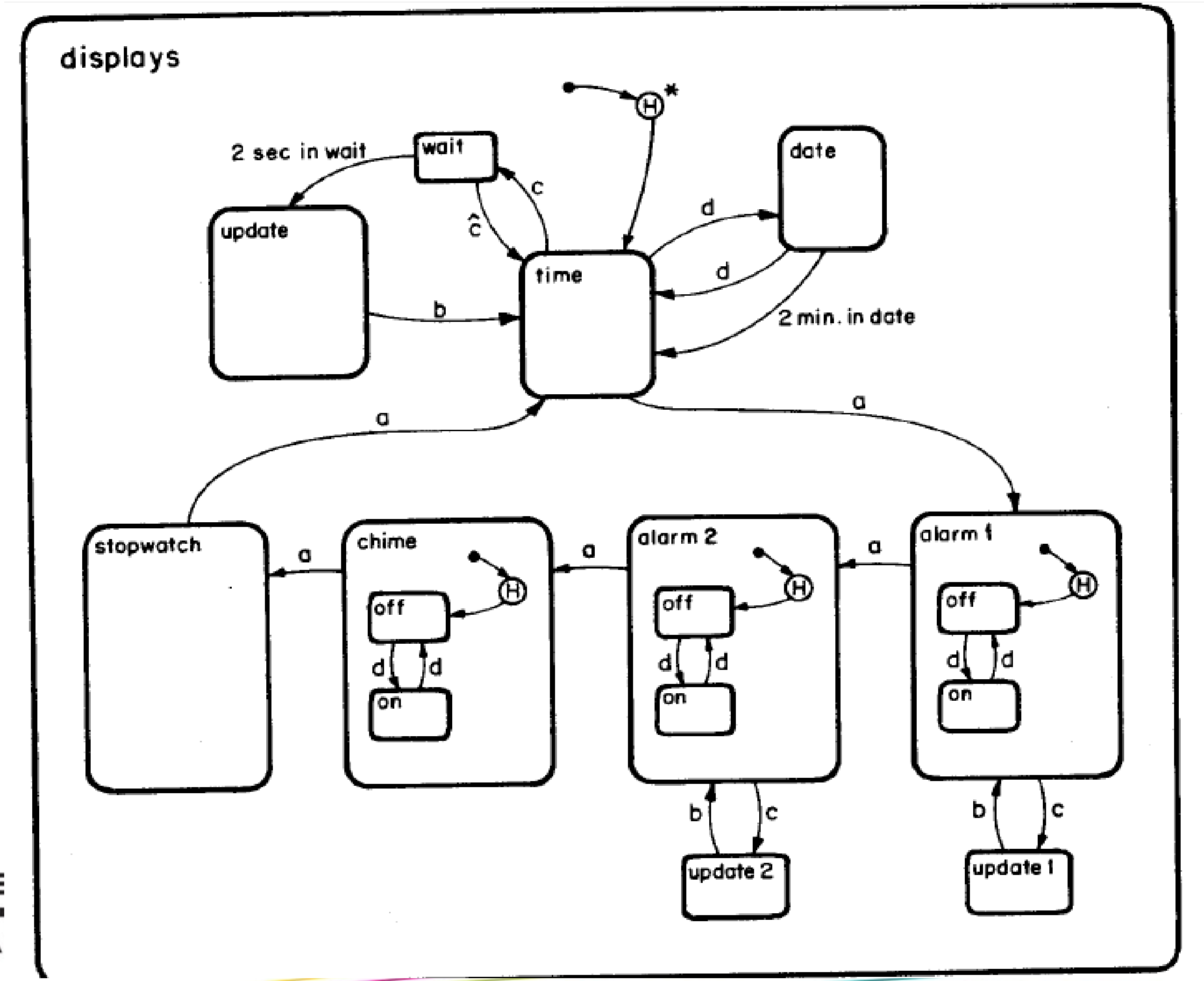
(b)

History Connective - Levels

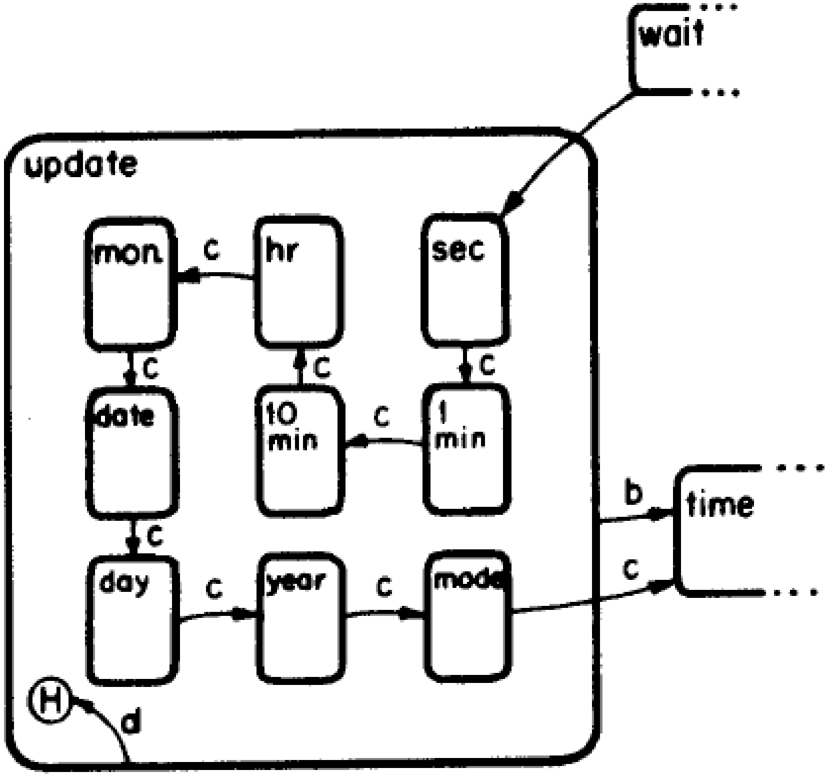
Something between 'one' and 'all' extremes



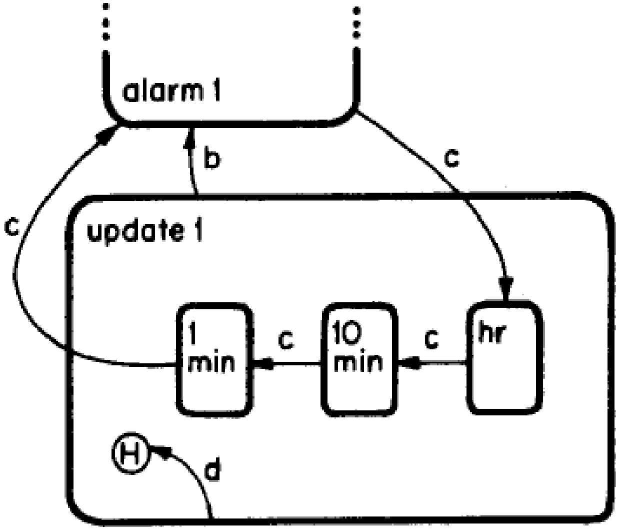
Watch Example – History + Update Capability



Watch Example – Refinement of Update States



Depressing *d* brings back to previous substate



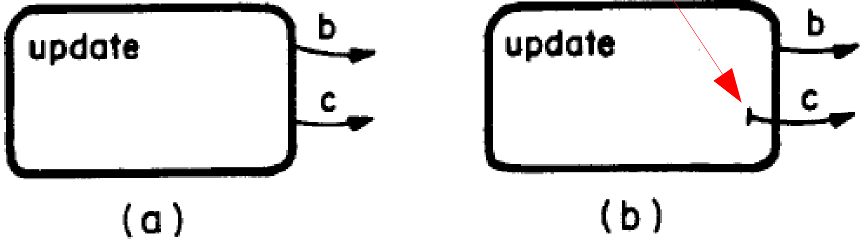
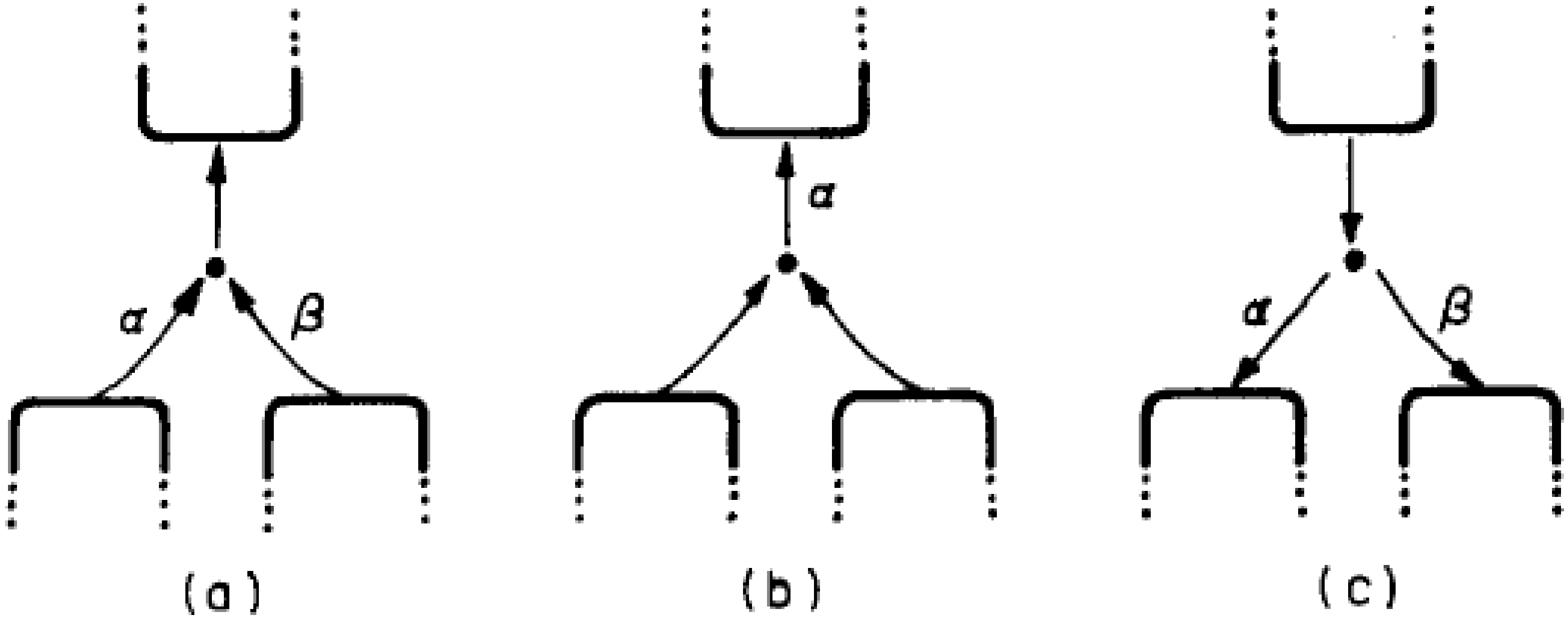


Fig. 15.

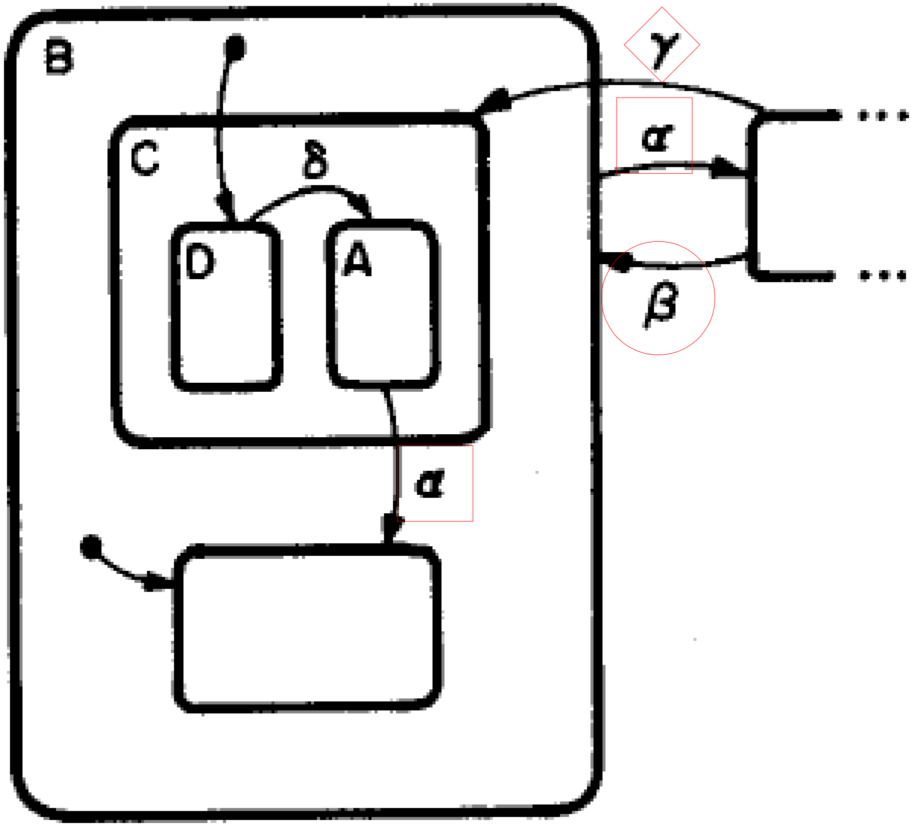
c applies to certain parts of update

Common Source/Target Arrows



Contradiction: Non-deterministic behavior

Subtle Contradictions - Example

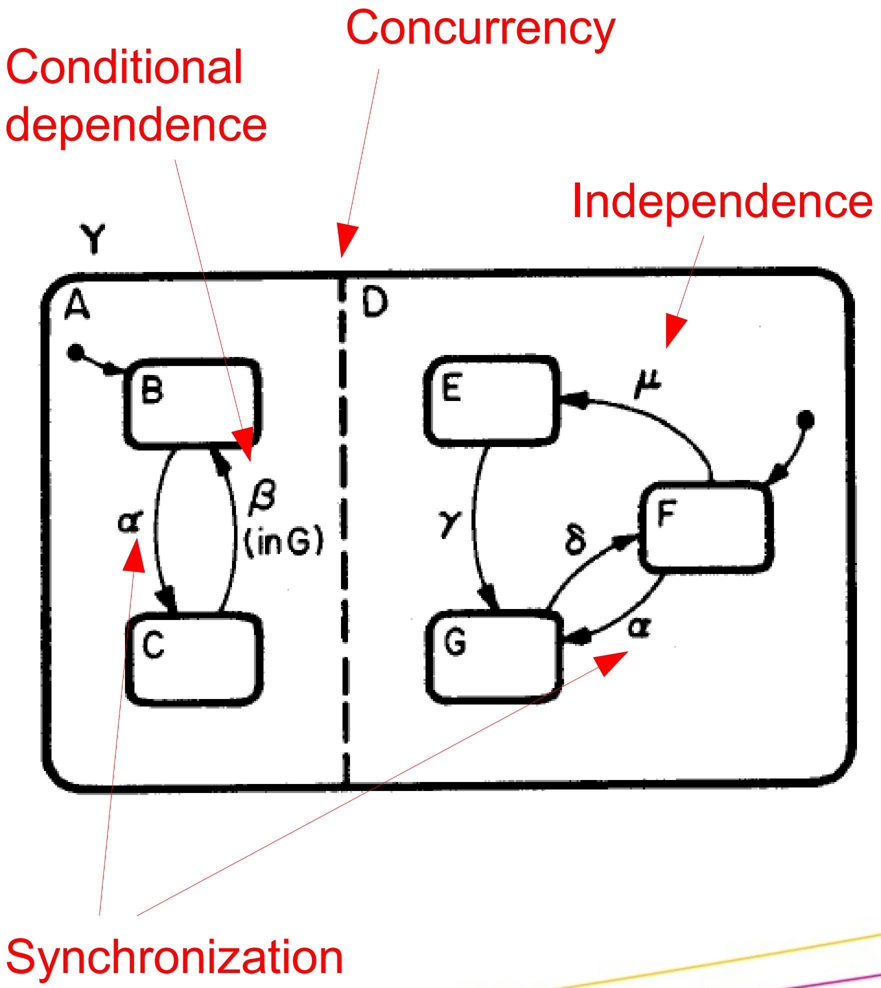


Outline

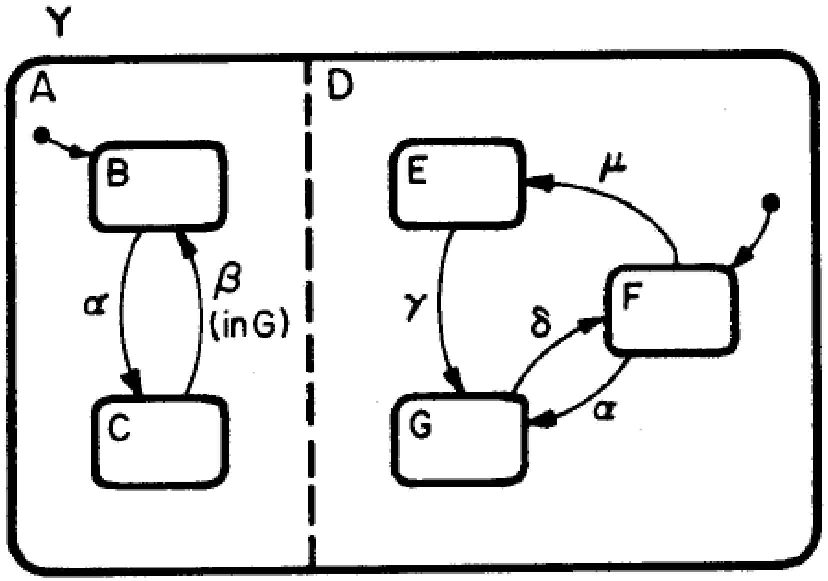
- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Basics

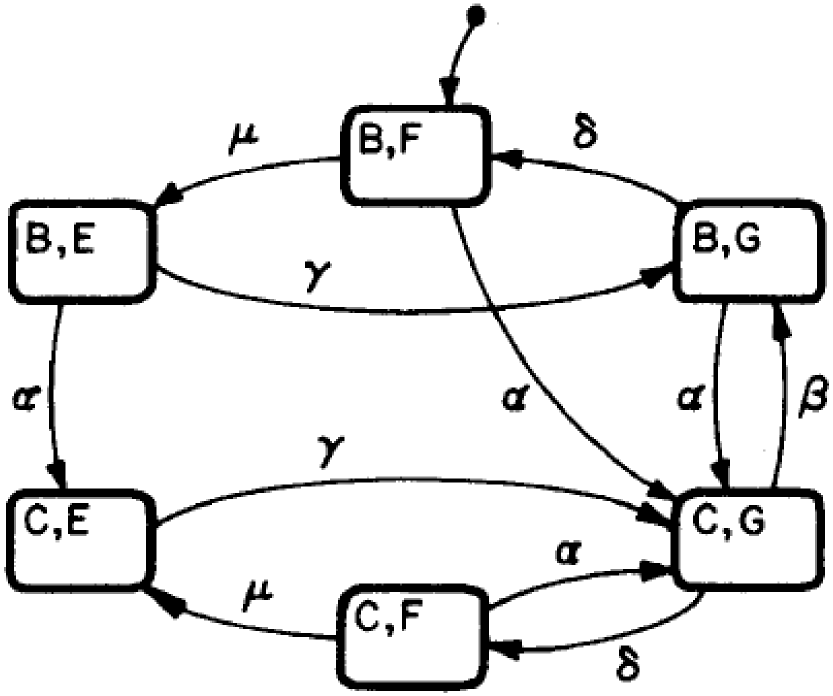
- *AND* decomposition
- System must be in **<u>all</u>** of its *AND* components
- *Y* is an orthogonal product of *A* and *D*



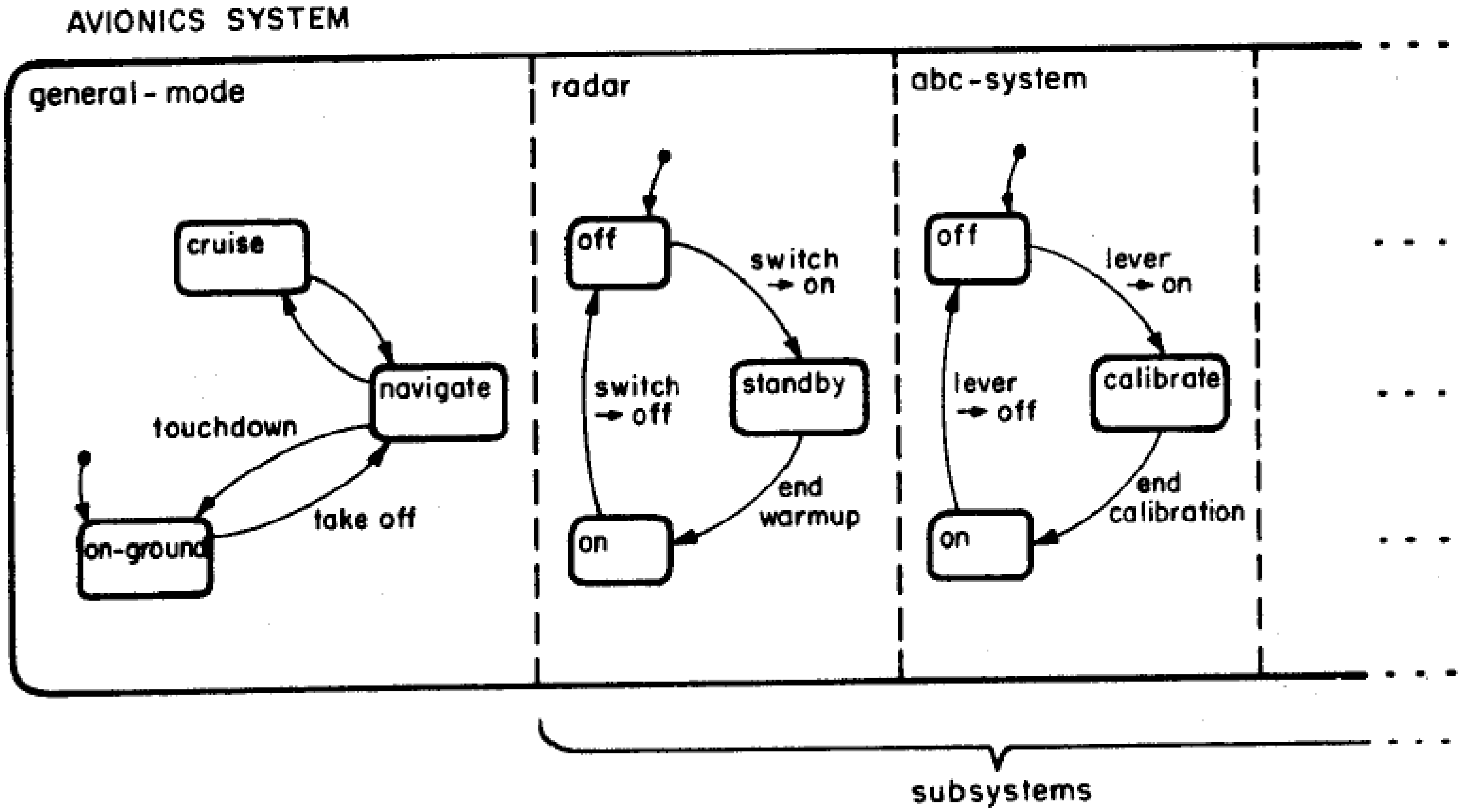
AND-Free Equivalence



Much cleaner and easier to understand!

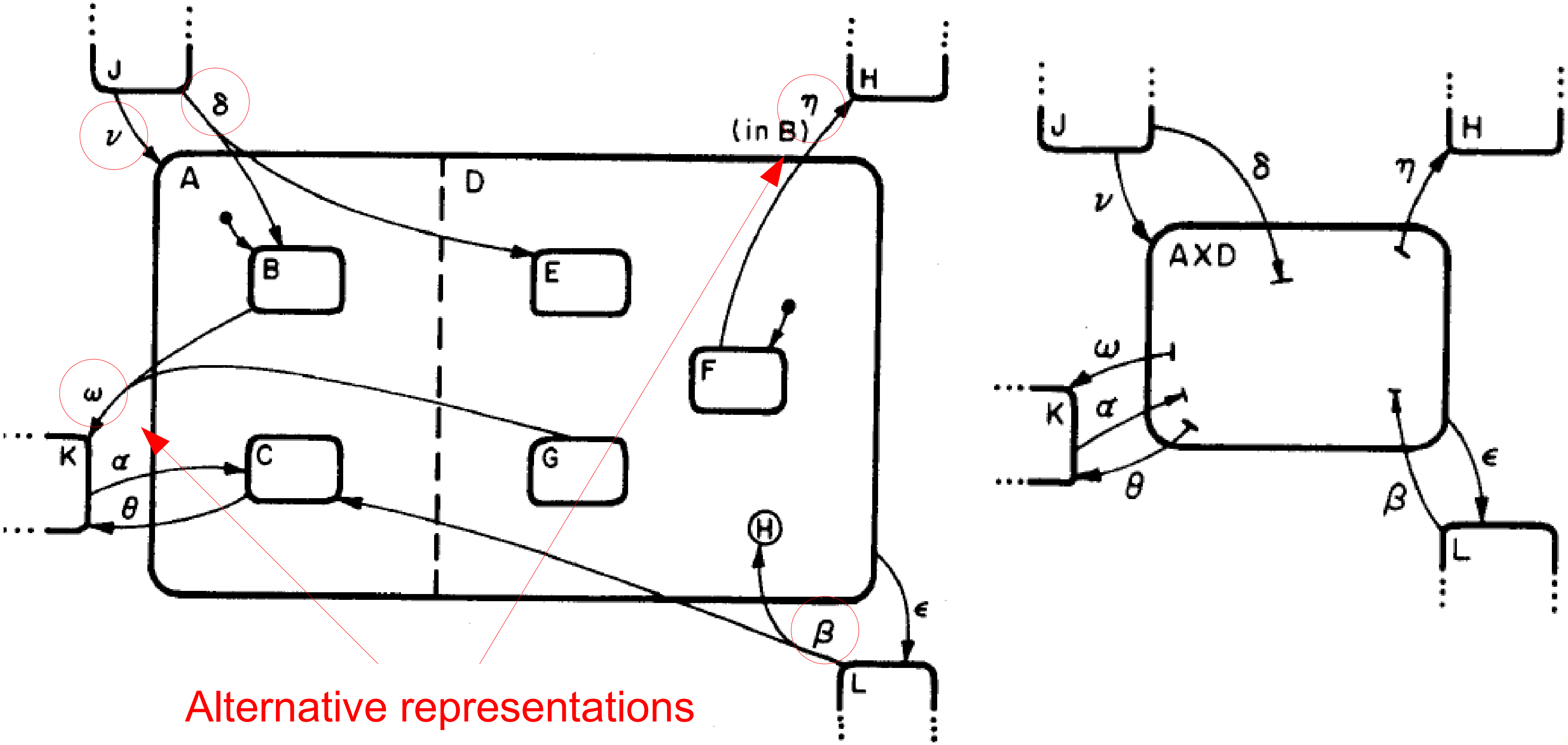


Example Application – Avionics System



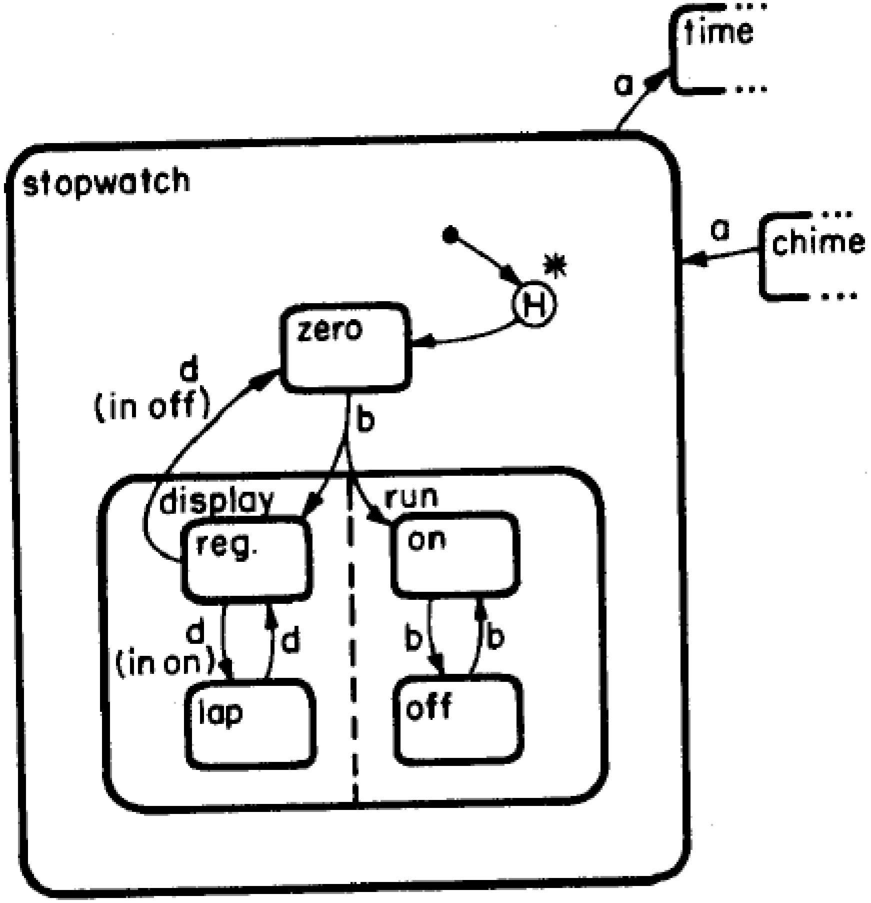
UNIVERSITY OF WATERLOO

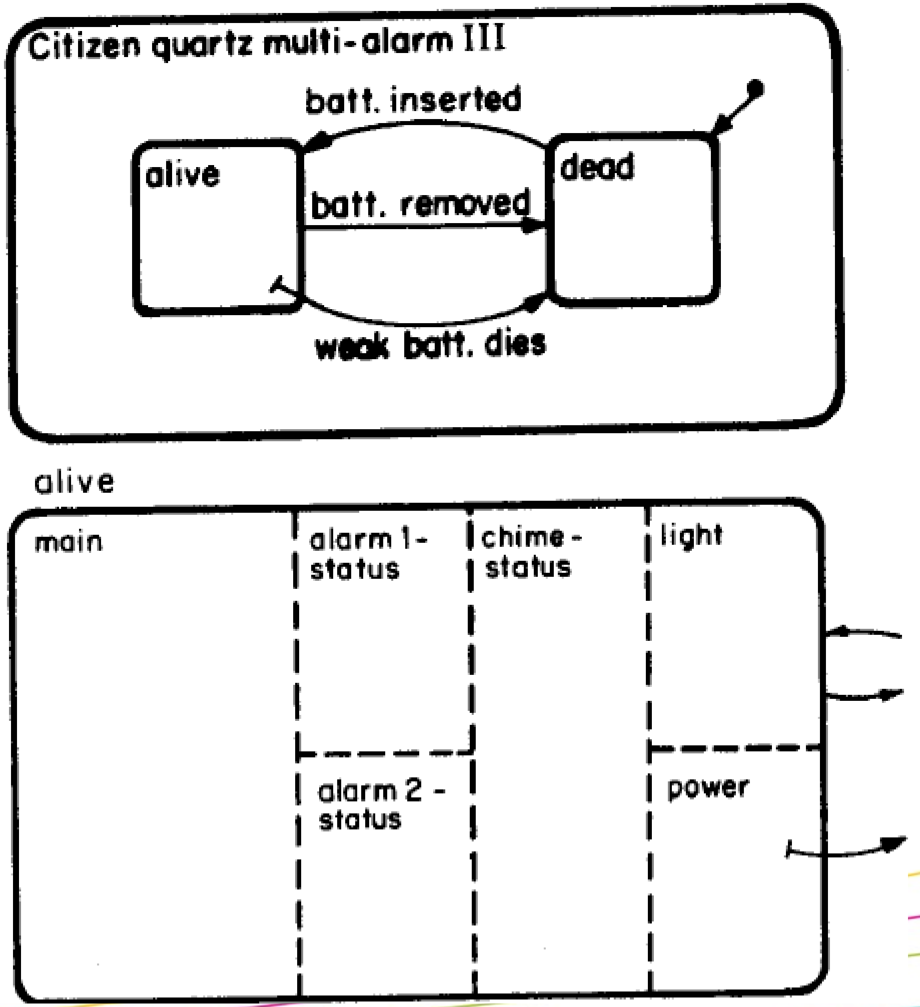
Orthogonal States - Exits and Entrances



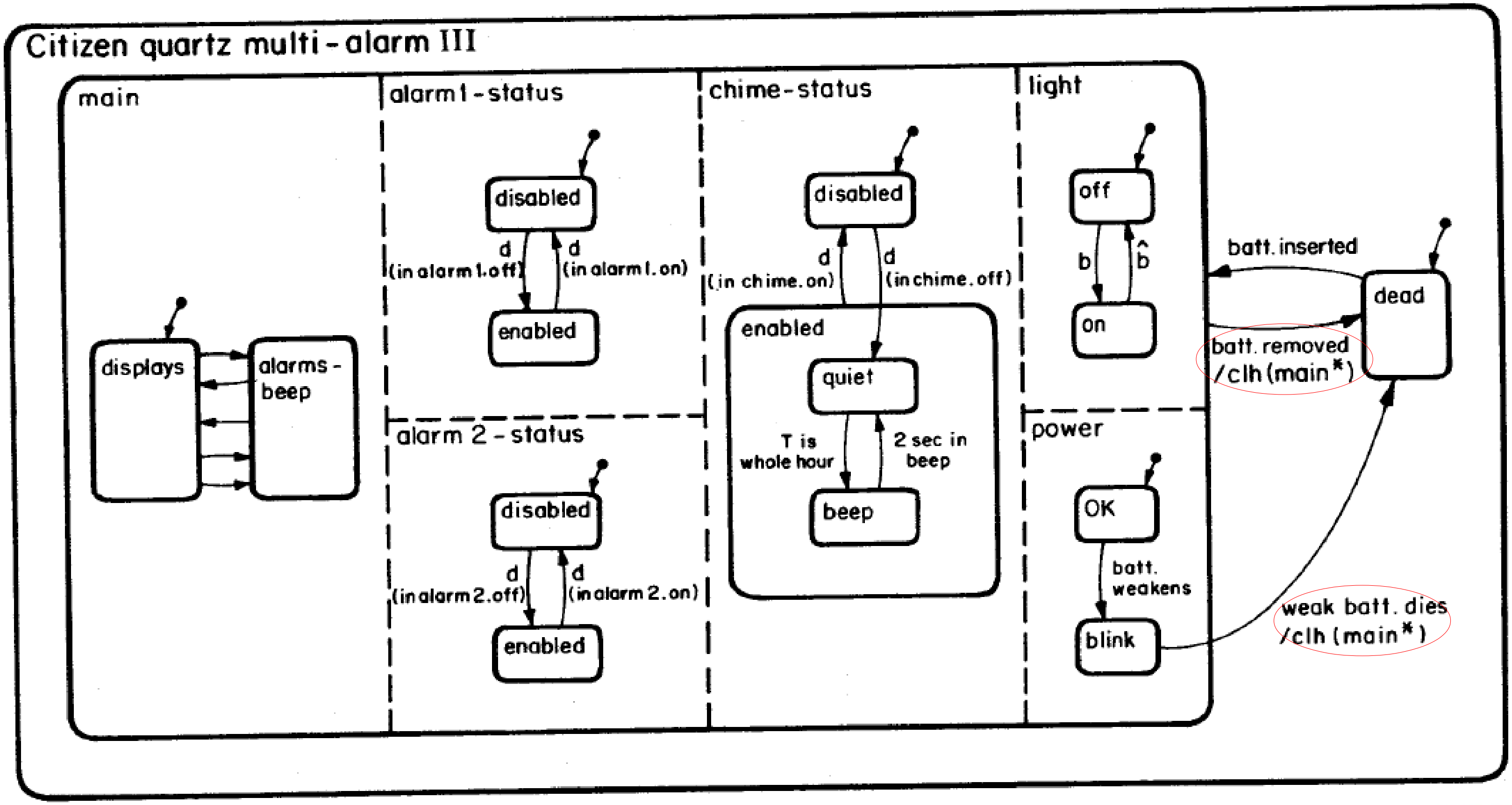
Alternative representations

Orthogonality – Watch Example

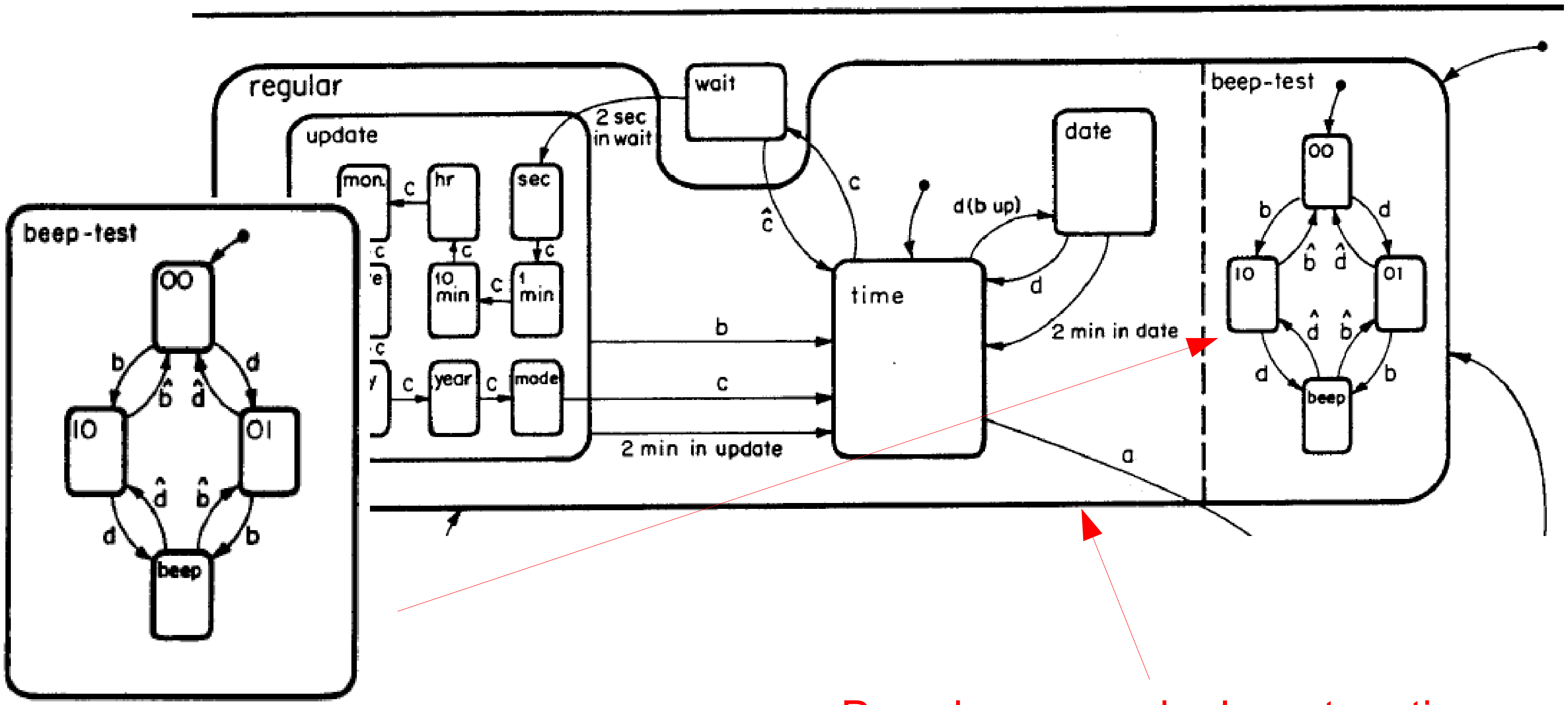




Orthogonality – Watch Example



Adding a Feature – Watch Example



Draw box around relevant portions

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

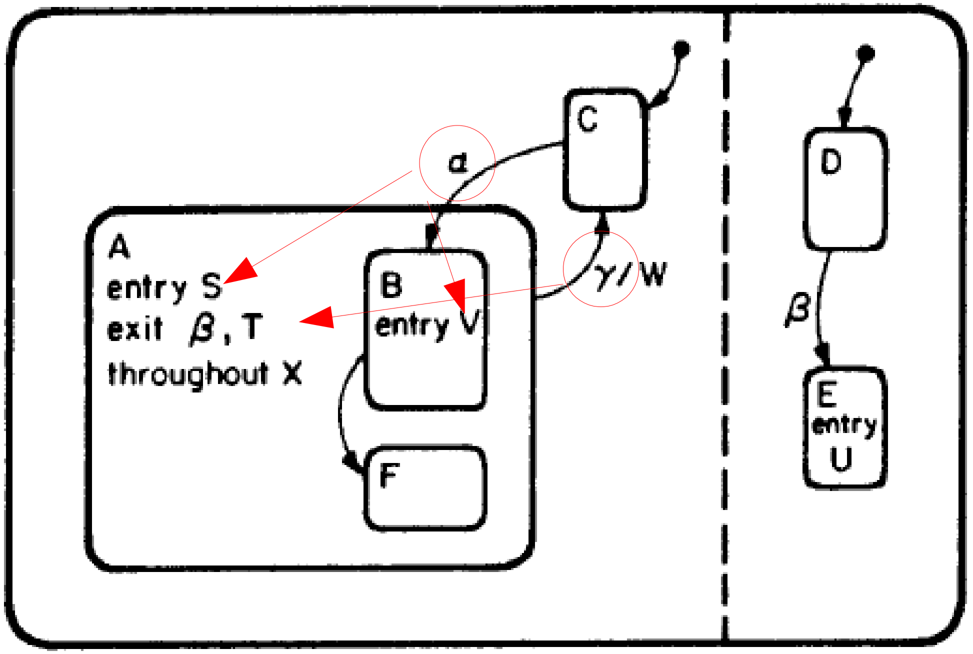
Basics

- Expressing reactivity
  - Generating events
  - Changing conditions
- **Action**: Split second occurrence
  - Display balance
- **Activity:** Take non-zero time
  - Beep for 30 seconds
- Each activity *X* associated associated with two actions: *start(X)* and *stop(X)*

UNIVERSITY OF
WATERLOO

Basics

- Actions are allowed with
  - Transitions
  - Entering a state
  - Exiting a state
- Difficult to define semantics

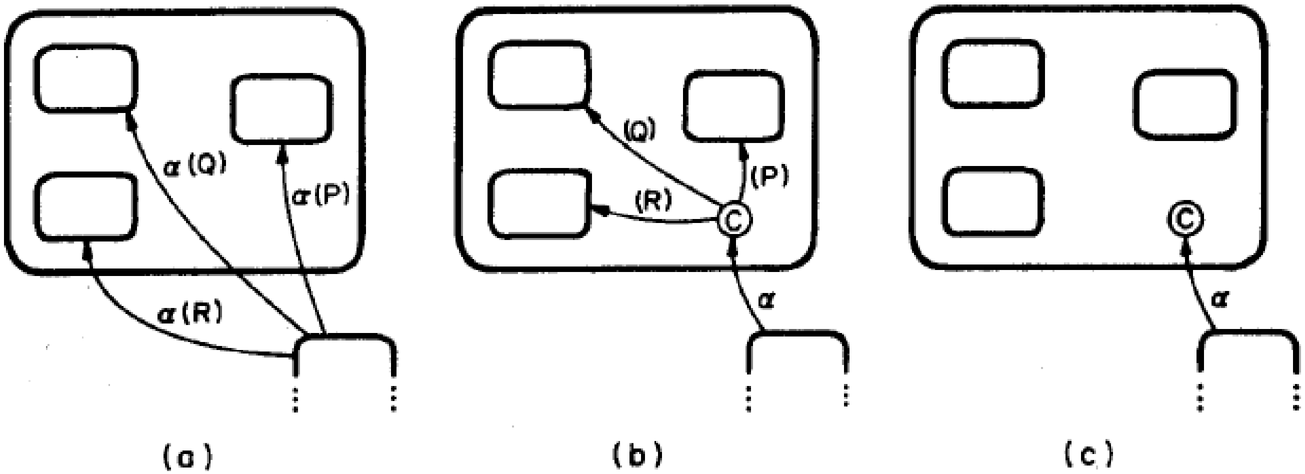


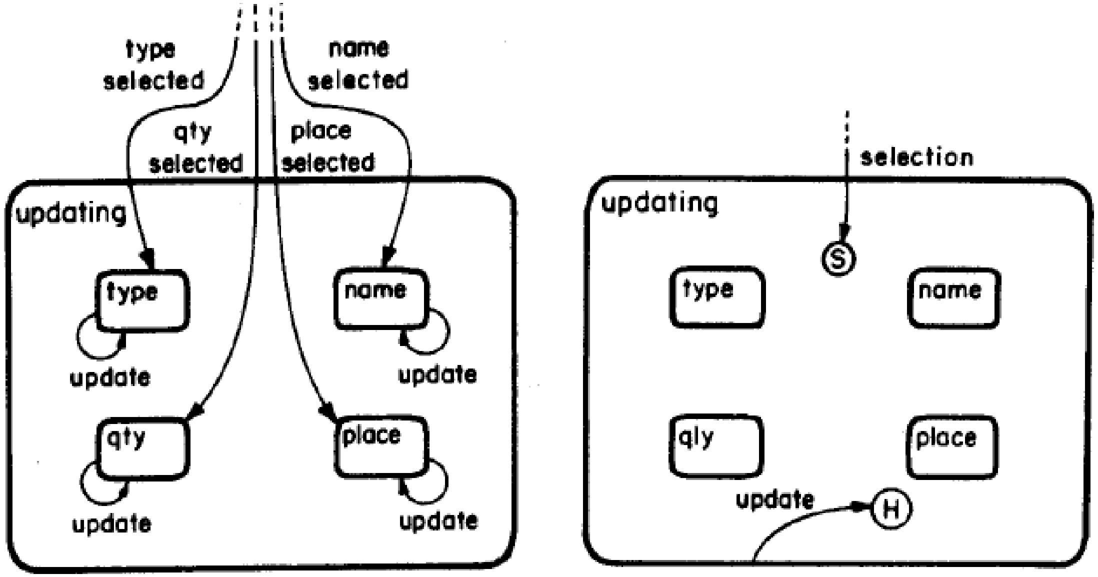
UNIVERSITY OF WATERLOO

Outline

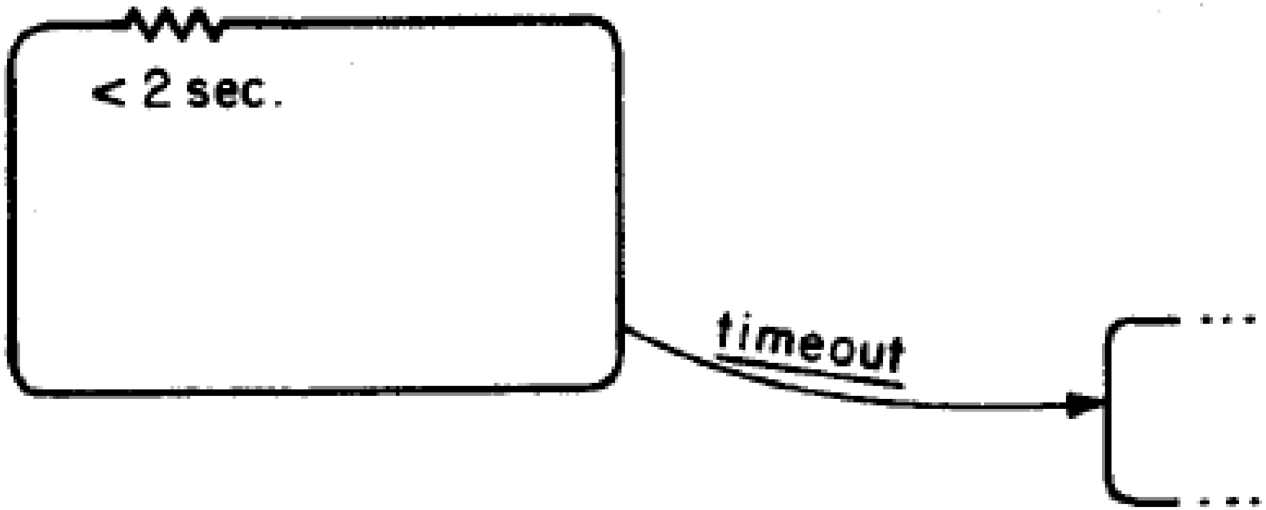
- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Condition and Selection Entrances

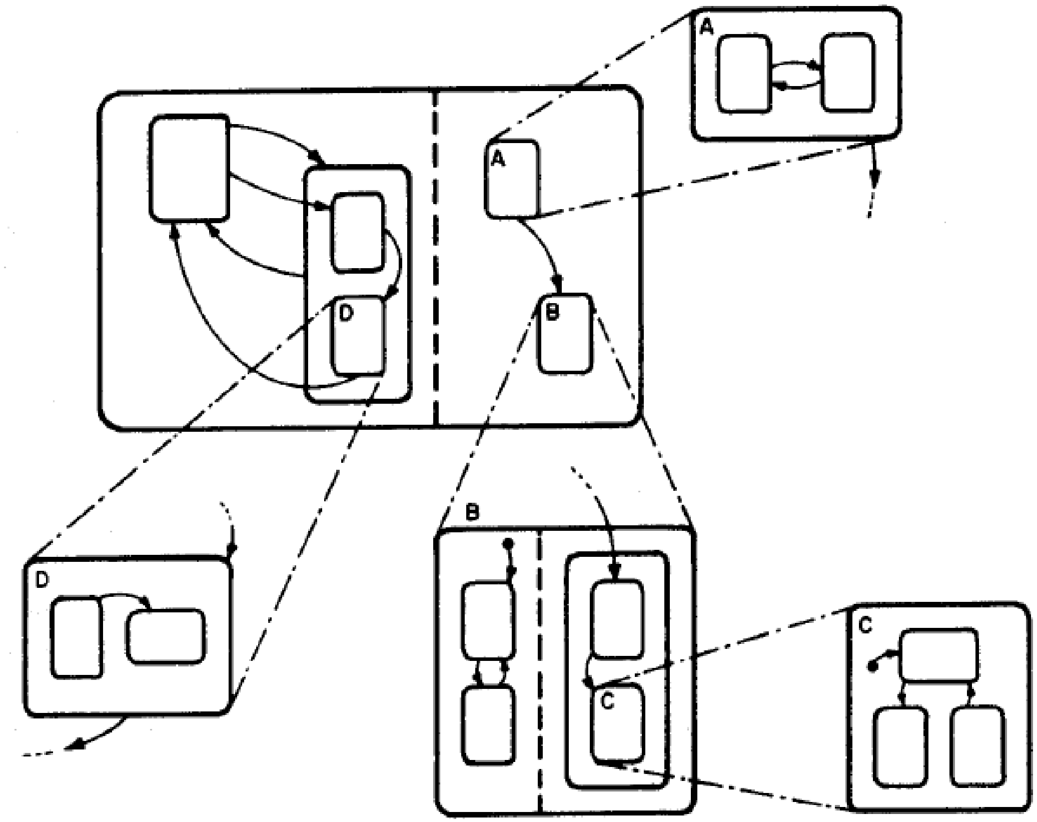




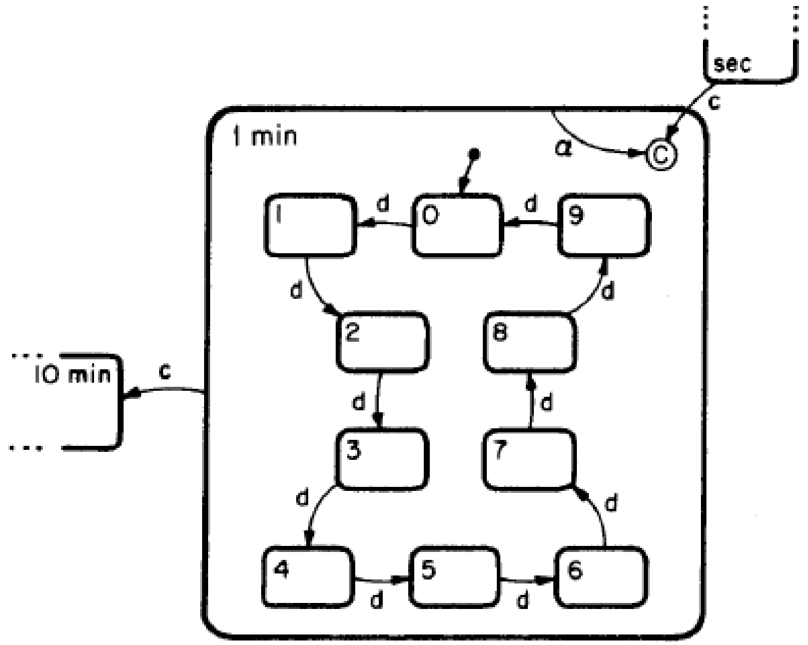
Timeouts

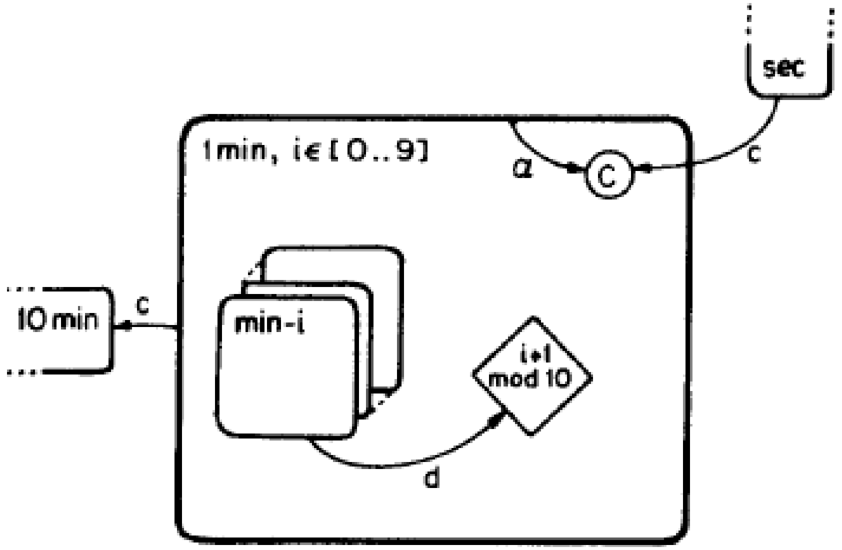


Unclustering

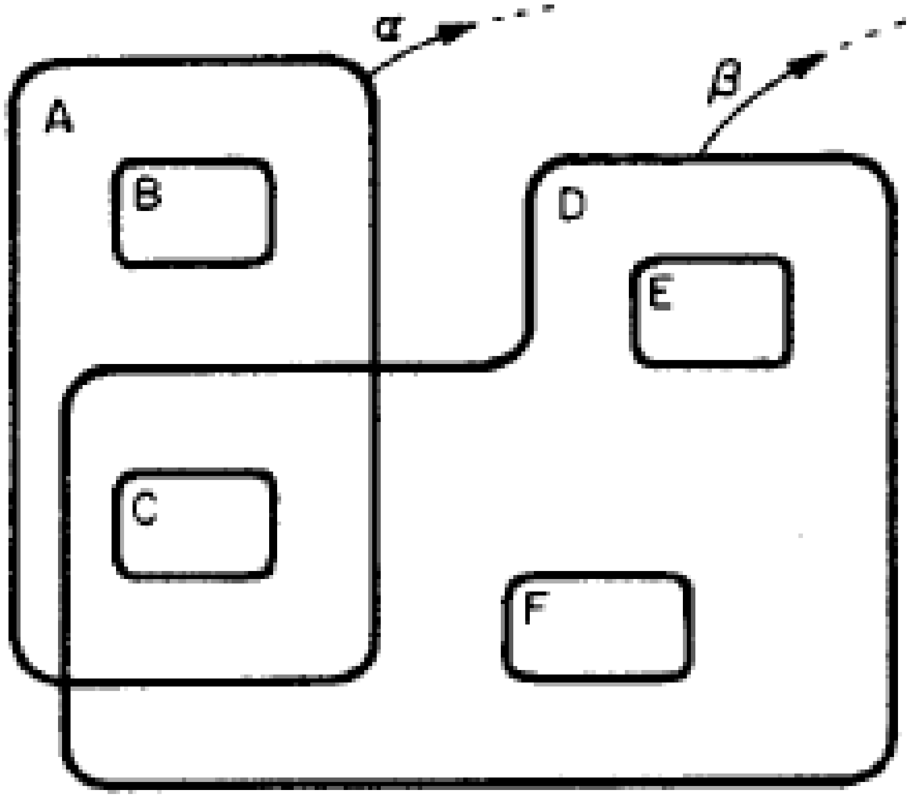


Parametrized States





Overlapping States



Temporal Logic

- Specifying constraints in TL and verification of statecharts from constraint specification

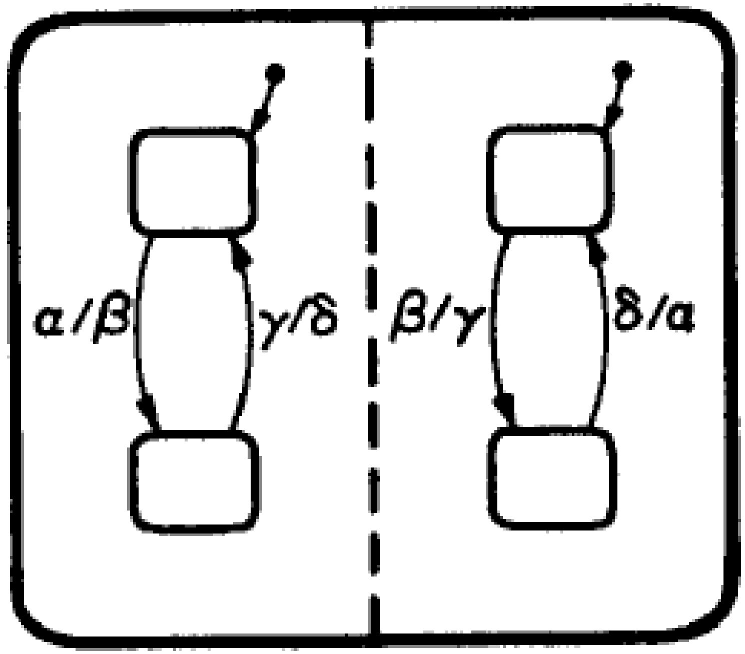
OR

- Synthesizing 'good' statecharts from TL specifications

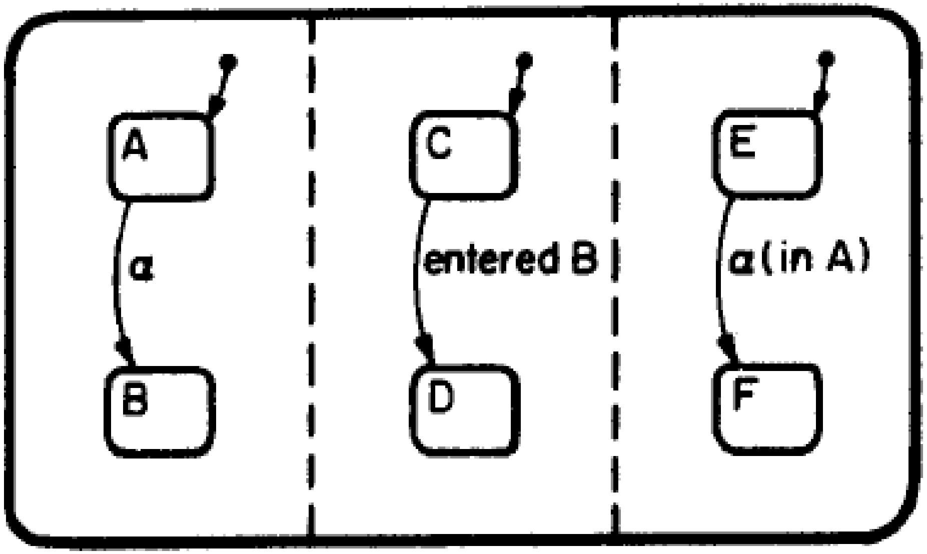
Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Some Problems



Cycles



What happens when α occurs?

Outline

- Motivation behind Statecharts
- What are Statecharts?
- Diving deeper
  - Clustering & Refinement
  - Orthogonality & Concurrency
  - Actions & Activities
- Additional features & possible extensions
- Trouble with semantics
- Discussion

Discussion

- Impact
  - 6000+ citations
  - UML statecharts are a variant of the Harel statechart
- Problems
  - Easy to make errors that lead to undefined/contradictory states
  - Unintended consequences in complex systems