Notations for specifications

Introduction

Classification

Automata and Statecharts

Specification

- The term is used here as a synonym for "precise high-level description"
  - The notations we introduce can be used for descriptions than apply to "goals", "requirements" and "domain properties" in the Jackson-Zave terminology
- In general, specification stands for
  - precise definition of a "thing" that acts as a contract between the producer and the consumer of that "thing"

Specification vs. program (the product)

Specification

- What, requirements
  - Abstract
  - High-level
  - May be:
    - Non-executable
    - Non-deterministic

Program

- How, implementation
  - Concrete
  - Low-level
  - Should/must be:
    - Executable
    - Deterministic

Levels of abstraction

- 25 lines of informal requirements
- 250 lines of (formal) specification
- 2500 lines of design description
- 25000 lines of high-level program code
- 250000 machine instructions of object code
- 2500000 CMOS transistors in hardware!

Properties of a specification

- Clear, precise, non ambiguous, understandable, consistent, complete (internal, external), incremental
- Natural language is NOT
  - "Selecting is the process for designating areas of your document that you want to work on. Most editing and formatting actions require two steps: first you select what you want to work on, such as text or graphics; then you initiate the appropriate action."
  - "The whole text should be kept in lines of equal length. The length is specified by the user;… Unless the user gives an explicit hyphenation command, a carriage return should occur only at the end of a word."

Notations

- Informal, semi-formal, formal
- Operational
  - Behavior specification in terms of some abstract machine
- Descriptive
  - Behavior described via properties

An example

- "Let a be an array of n elements. The result of its sorting is an array b of n elements such that the first element of b is the minimum of a (if several elements of a have the same value, any one of them is acceptable); the second element of b is the minimum of the array of n-1 elements obtained from a by removing its minimum element; and so on until all n elements of a have been removed."
- "The result of sorting a is an array b which is a permutation of a and is sorted."

How to *verify* a specification?

- "Observe" dynamic behavior of specified system (simulation, prototyping, "testing" specs)
- Analyze properties of the specified system
- Analogy with traditional engineering
  - physical model of a bridge
  - mathematical model of a bridge

UML

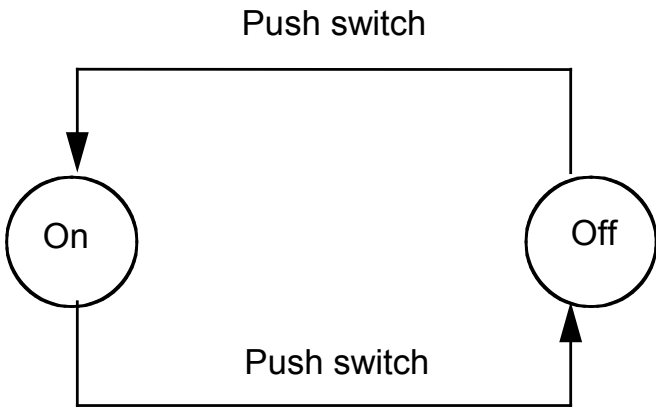
- Descriptive
  - Use Case Diagram
  - Class Diagram
- Operational
  - State Diagram
  - Sequence Diagram

Finite state machines

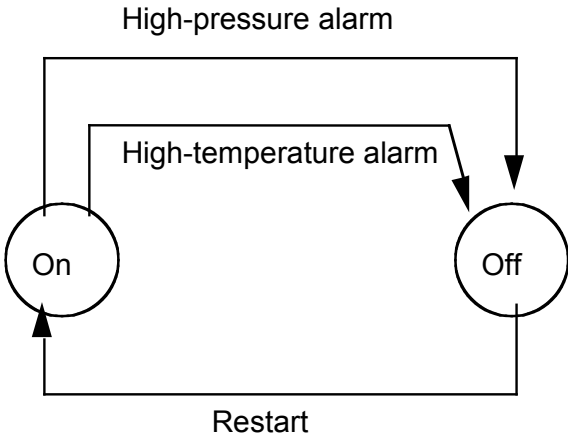
- They are the typical example of an operational specification notation
  - S a finite set of states
  - I a finite set of inputs
  - δ a state transition function
- Appealing graphical representation
- "Animation" straightforward

An example

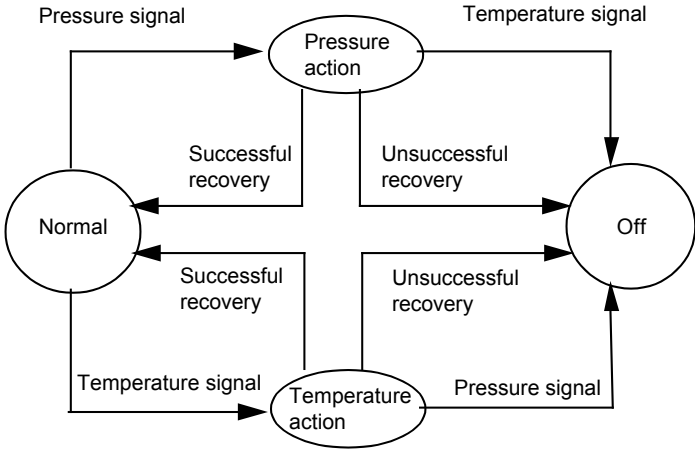
- Behavior of a lamp



Another example: a plant control system



A refinement



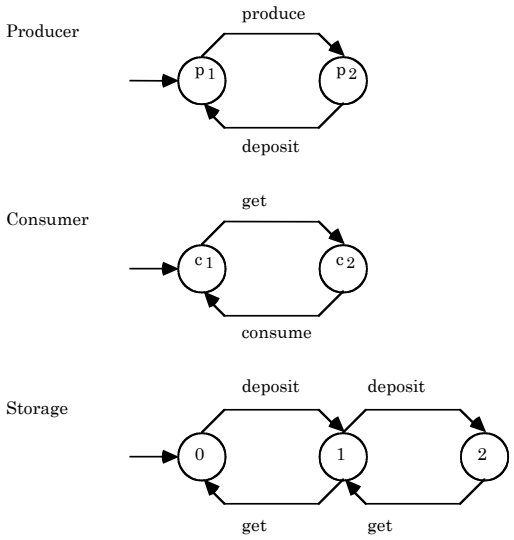
Classes of FSMs

- Deterministic/nondeterministic
- FSM as recognizers
- FSM as transducers
- . . .

Limitations

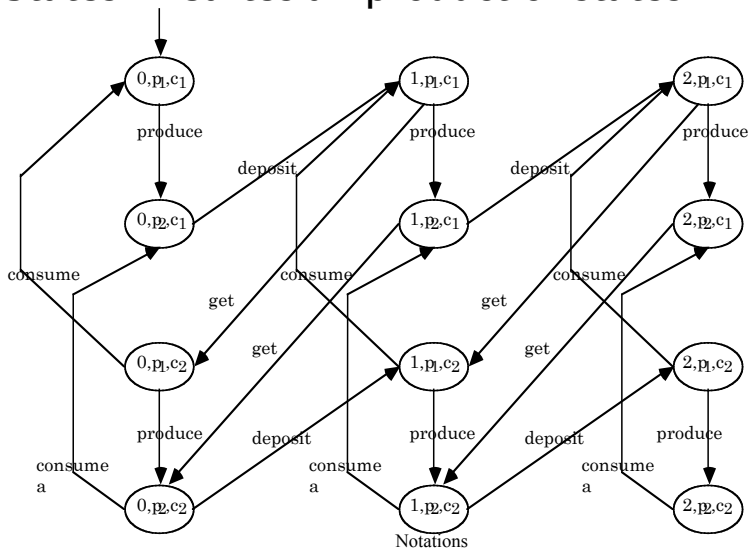
- Finite memory
- State explosion
  - Given a number of FSMs with k_1, k_2, … k_n states, their composition is a FSM with k_1 * k_2 *… * k_n. This growth is exponential with the number of FSMs, not linear (we would like it to be k_1 + k_2 +… + k_n)

State explosion: an example



How to combine into an FSM?

States = Cartesian product of states



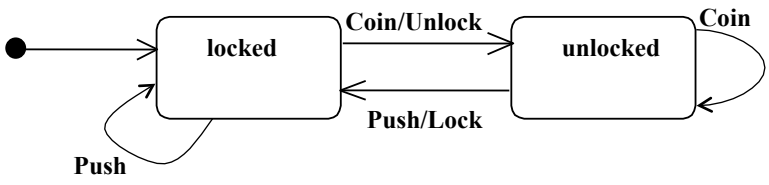
How to solve these problems?

- Statecharts
  - cooperating finite state machines
  - used in UML as state diagram
- Petri nets

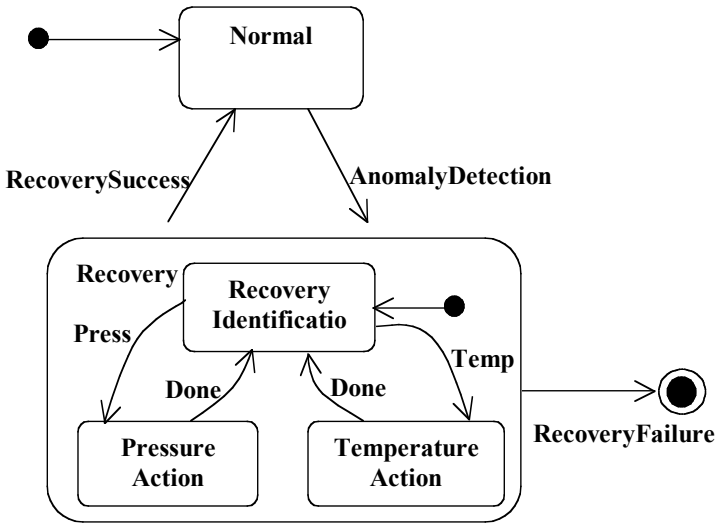
StateCharts

- A modular hierarchical notation for automata

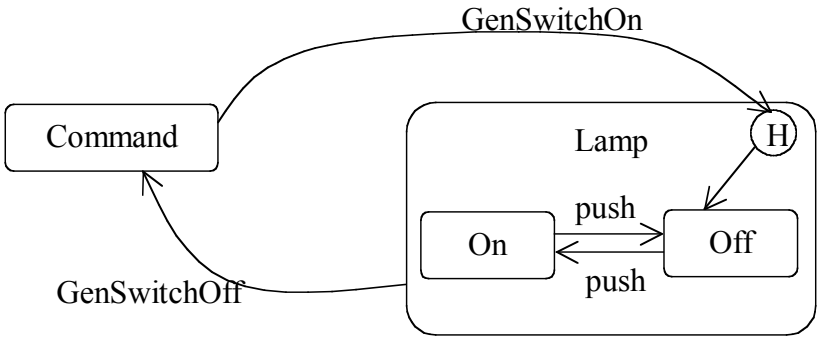
Turnstile: events and actions



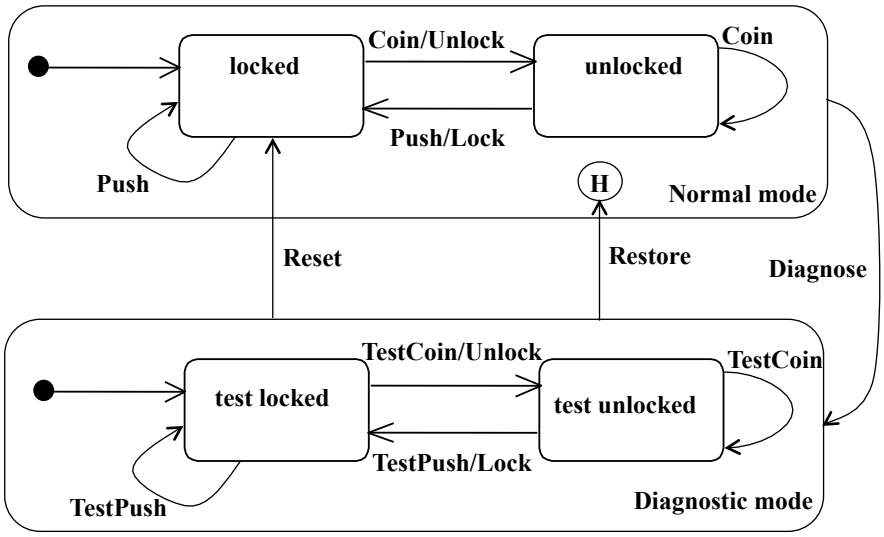
Plant control system: substates



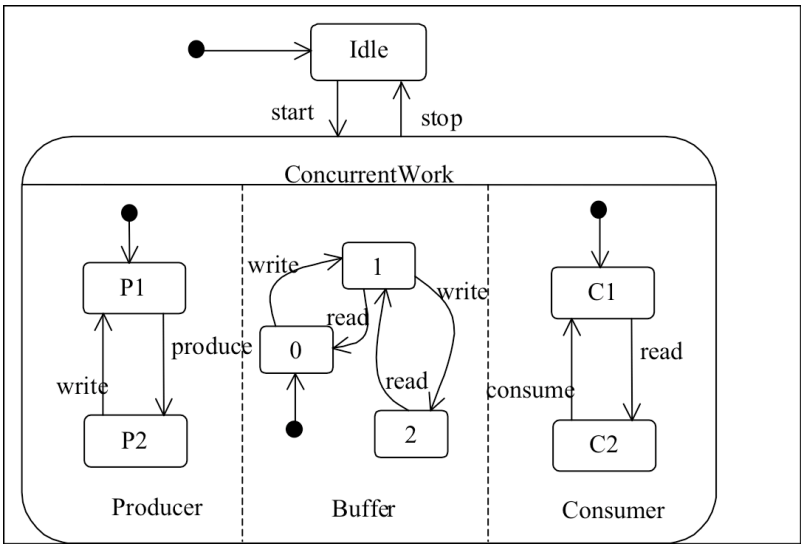
History states



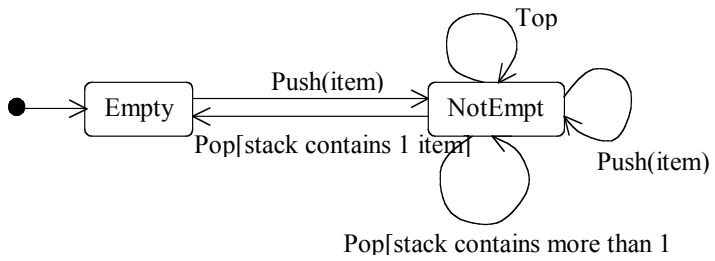
Turnstile II ver: history



Concurrent substates



State specification of abstract object



Digital clock: events, conditions, and actions