5. Statechart Diagrams
5.1 Examples for Statecharts

- To follow: examples from the UML notation guide
- plus other examples
5.1 Examples for Statecharts

Statechart Syntax – Overview

<transition> ::=<state>  
  { <event> }\textsuperscript{1}  { [ <guard> ] }\textsuperscript{1}  { / <action> }\textsuperscript{1}  
  <state>

<state> ::=<state-name>  
  { entry / <action> }\textsuperscript{1}  
  { do / <action> }\textsuperscript{1}  
  { <event> / <action> }\textsuperscript{*}  
  { exit / <action> }\textsuperscript{0}  

<state-with-sequential-substates> ::=<state-name>  
  <state>  
  <transition>  
  <state>  

<state-with-concurrent-substates> ::=<state-name>  
  <state>  
  <transition>  
  <state>  

pseudo states: initial, final, synchronisation, deep history, shallow history, stub
Statechart Diagram (3-71)
Statechart Diagram (3-72)

Typing Password

- entry / set echo invisible
- exit / set echo normal
- character / handle character
- help / display help
Statechart Diagram (3-74)

HiddenComposite

entry/ start dial tone
exit/ stop dial tone
5.1 Examples for Statecharts

Statechart Diagram (3-75)
Statechart (3-77)
Statechart Diagram (3-78)

may be abstracted as
Statechart Diagram (3-79)
Statechart Diagram (3-80)

State0

\[ e_2[b < 0] \]

\[ [a < 0] \]

State2

State1

\[ e_1[b < 0] \]

\[ [a > 7] \]

State3

\[ [a = 5] \]

State4
Statechart Diagram (3-81)

State1

\[ e1[b < 0]/a := f(m) \]

[a < 0]

[else]

[a = 5]

State2

State3

State4
5.1 Examples for Statecharts

Statechart Diagram (3-83)
Submission Status for Scientific Paper

- inPreparation
  - submitted
  - inRefereeing
    - refereed
      - accepted
      - rejected
        - inRevision
          - published
        - hopeless
      - inPolishing

Education Status for People (1)

- atSchool
- atUniversity
- inEducation
- professional
Education Status for People (2)
Education Status for People (3)
Education Status for People (4)

- Apprentice
- Journeyman
- Master
Education Status for People (5)
Civil Status for People

- single
- non-single
- married
- divorced
- widowed
One-shot State Diagram for Chess Game
5.1 Examples for Statecharts

State Diagram with Guarded Transitions

- **North/south may go straight**
  - Time-out [cars in N/S left lanes]

- **East/west may turn left**
  - Time-out [no cars in N/S left lanes]

- **North/south may turn left**
  - Time-out [no cars in N/S left lanes]

- **East/west may go straight**
  - Time-out [cars in E/W left lanes]
Actions for Pop-up Menu

- Idle
  - right button down / display popup menu
  - right button up / erase popup menu

- Menu visible
  - cursor moved / highlight menu item
Vending Machine Model

Idle

collect in(amount) / set balance

cancel / refund coins

Collecting money

coins in(amount) / add to balance

select(item)

[item empty]

[change=0]

[change>0]

do: dispense item

do: make change
Dispense Item Activity of Vending Machine
Select Item Transition of Vending Machine
An Aggregation and its Concurrent State Diagrams (1)
An Aggregation and its Concurrent State Diagrams (2)
5.1 Examples for Statecharts

Entering a PIN Number (1)
Entering a PIN Number (2)
Entering a PIN Number (3)

NoInput

0..9

UnDo

Reset

DigitsIn

OK

ToBeVerified
Traffic Light (1)

- **red** → pushRed → RED
- **yellow** → pushYellow → YELLOW
- **green** → pushGreen → GREEN
Traffic Light (2)
5.2 Graph Transformation for Statecharts

Explanation 1 (SC Diagram Semantics)

• explain state expansion in nested UML state diagrams
  1. adding boundary nodes introducing a precise interface for the state to be expanded
  2. expanding the state
  3. removing the boundary nodes
• our approach: intermediate step between original UML diagrams and a general comprehensive semantical framework
Explanations (Cont’d SC Diagram Semantics)

- graph notation: as close as possible to original UML representation, but a representation forcing an unambiguous interpretation
- resulting graphs (in the view of this section, the semantics of the UML state diagrams) can be translated into various semantical frameworks like temporal logics, streams, or (again) graph transformation systems (among other approaches)
Car Transmission – UML High Level Diagram
Car Transmission – High Level Graph

- Forward
- Neutral
- Reverse

Transitions:
- pushF: Forward
- pushN: Neutral
- pushR: Reverse
Resulting Low Level Graph

- Neutral
  - pushF
  - pushN
  - stop
  - up
  - down
  - pushN

- First
  - pushF
  - pushN
  - stop
  - up
  - down
  - pushN

- Second
  - pushN
  - stop
  - up
  - down
  - pushN

- Third
  - pushN
  - stop

- Reverse
  - pushR
  - pushN
Graph Transformation System for Introducing the Boundary Nodes

Conventions for graph production layout

- nodes in the *top* of a production represent nodes *outside* the part to be expanded
- nodes in the *bottom* of a production represent nodes *inside* the part to be expanded
Car Transmission – Explicit Boundaries in the High Level Graph

- Neutral
  - pushF
  - pushN

- Forward

- Reverse
  - pushR
  - pushN
Graph Transformation System for the Car Transmission Example
Applying the Rule in the High Level Graph
Graph Transformation System for Removing the Boundary Nodes
Stubbed Transitions – UML High and Low Level Diagram

StubbedTransitions-H

StubbedTransitions-L
5.2 Graph Transformation for Statecharts

Stubbed Transitions - Explicit Stubs and Boundaries in the High Level Graph

[Diagram with nodes labeled A, B, C, D, and W, with arrows indicating transitions and stubs]
Graph Transformation System for Introducing the Final Node
Graph Transformation System for the Stubbed Transitions Example
Stubbed Transitions – Applying the Rule in the High Level Graph
Graph Transformation System for Removing the Stubs
Graph Transformation System for Removing the Final Node
Stubbed Transitions – Resulting Low Level Graph
General Steps for Statechart Diagram Semantics

General steps

• **Step 0**: Consider productions for boundary and stub node insertion and deletion

• **Step 1**: Make explicit part to be expanded by introducing boundary nodes

• **(Step 1*)**: If needed, make explicit stub nodes

• **Step 2**: Define graph transformation production for state expansion
General Steps for Statechart Diagram Semantics (cont’d)

- Step 3: Apply graph transformation production
- (Step 4*): If needed, remove stub nodes
- Step 4: Remove boundary nodes