State Machine Syntax and Semantics
6.005 Elements of Software Construction

Basic elements

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e</td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

transition on event e from state A to C

Semantics

State machine consists of:
- set of states S
- initial state I ∈ S
- set of event classes E
- transition relation R ⊆ S×E×S

Semantics of state machine is:
- set of traces T ⊆ E*
- the empty sequence is a trace <> ∈ T leading to the initial state
- if trace t can lead to state s, and (s,e,s') ∈ R, then t^<e> ∈ T is a trace that can lead to state s'

Parallel combination:
- given machines (S1,E1,R1,T1) and (S2,E2,R2,T2)
- a sequence t in (E1∪E2)* is a trace if t restricted to the events in E1 (E2) is in T1 (T2)

Defining a state machine

Definition should include:
- state machine diagram
- designations of events

Sample designation:
offhook: user hangs up phone by pressing END button

I/O shorthand

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>i/o</td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

short for:
in state A, machine can perform event i and then event o and end in state B

Statechart notation

S is a superstate (OR) grouping A and B
S is a superstate (AND) grouping A and B in parallel

transition on event e from any state in S to C

I is the initial state

-- set of traces T ⊆ E*
-- the empty sequence is a trace <> ∈ T leading to the initial state
-- if trace t can lead to state s, and (s,e,s') ∈ R, then t^<e> ∈ T is a trace that can lead to state s'

Parallel combination:
- given machines (S1,E1,R1,T1) and (S2,E2,R2,T2)
- a sequence t in (E1∪E2)* is a trace if t restricted to the events in E1 (E2) is in T1 (T2)

Defining a state machine

Definition should include:
- state machine diagram
- designations of events

Sample designation:
offhook: user hangs up phone by pressing END button

history (H): on entry to state S (eg, by event e from state C), return to last state visited in S

initialization: on entry to S (eg, by event e from state C), enter state A