18-642 Recitation #7

March 2, 2018
Updates

• Homework:
  – Homeworks #18a/18b/20 Graded
  – Homework #21 due Sunday 3/4 @ 11pm
    • Review: Only need to do lectures #4-#19 and #21,#23
• Project:
  – Project 6 due tonight @ 11 PM
  – Project 7 will be due 3/30 with 3/23 checkpoint
• Exam coming up on 3/7
• No recitation 3/9 or 3/16 (mid-semester and spring break)
• Be sure to fill out question sheet to receive attendance points today!
Today

• Project 7 Introduction
• State Chart Review
Project 7

• State chart and requirements
• Traceability
• Implementation
• Two peer reviews
State Charts and Requirements

• Revise your requirements from Project 5 if need be

• Make a state chart that describes your implementation
  – Do not look at your code while doing this
  – Include a table of input/output/internal variables
State Chart Reminders

- Number the states
- Name the states
- Input to system causes state transition
- Each state sets all output variables
- Avoid complex behaviors within state subroutine
- Avoid actions on transition

Picture and code snippet in following slides from:
http://www.ece.cmu.edu/~ece642/lectures/08_modalstatechart.pdf
State Chart Justifications

• Number the states
  – Easier to refer to in traceability

• Give them meaningful names
  – Easier to refer to when talking about (like in peer review)
  – Humans looking at statechart can figure out what the state means

• Just be sure name and number matches up in documentation!
State Chart Justifications

• Input to system causes state transition
  – Self-loops (inputs that don’t cause a change in state) are implicit
  – Can check that each state handles all possible inputs
  – Allows for consistency in code and traceability

```c
case MEDIUM: // State S3
    speed(SpdMed); // take action
    if (SpdButton() == TRUE) {CurrState = FAST;}
    if (OnOffButton() == TRUE) {CurrState = OFF;}
    break;
```
State Chart Justifications

• Each state sets all output variables
  – Easy to review/check traceability
    • Checklist item: “Each state sets all output variables”
  – Avoids bugs where an output is set in one state but not another
    • Say there’s an assumption that State 1 always goes to State 2, so State 2 doesn’t set an output variable
    • What if requirements change and State 3 also goes to State 2 now?

```c
case MEDIUM: // State S3
    speed(SpdMed); // take action
    if (SpdButton() == TRUE) {CurrState = FAST;}
    if (OnOffButton() == TRUE) {CurrState = OFF;}
    break;
```
State Chart Justifications

• Avoid complex behaviors within state subroutine
  – Makes peer review/traceability harder if internal variables are manipulated in complex ways and output is based on internal variables
  – Always consider testability, maintainability, and readability 😊

• Avoid actions (i.e. setting output) on transition
Traceability

- One row for each state and each transition
- One column for each requirement
- All columns and rows must have at least one cell checked off

### Requirements-to-Statecharts Traceability

<table>
<thead>
<tr>
<th>States</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2.1 IDLE</td>
<td>x</td>
</tr>
<tr>
<td>S2.2 EMPTY</td>
<td>x x</td>
</tr>
<tr>
<td>S2.3 VEND</td>
<td>x x x</td>
</tr>
<tr>
<td>S2.4 FLASH_OFF</td>
<td>x x x</td>
</tr>
<tr>
<td>S2.5 FLASH_ON</td>
<td>x x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transitions</th>
<th>R2.1</th>
<th>R2.2</th>
<th>R2.3</th>
<th>R2.4a</th>
<th>R2.4b</th>
<th>R2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2.1</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2.4</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>T2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>T2.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

[http://www.ece.cmu.edu/~ece642/lectures/17_traceability.pdf](http://www.ece.cmu.edu/~ece642/lectures/17_traceability.pdf)
Implementing State Chart

• Switch statement to check current state
• Set all outputs in each state block
  – Recommended to do this at the beginning of the block
  – Helps peer reviews
• Test inputs/variables to transition at the end
• We’ll show an example next recitation
Project 7 Peer Reviews

- Two Peer reviews
- Groups assigned on Canvas
- Review #1 (due 3/23): State chart + requirements
- Review #2 (due 3/30): Code + Traceability
- Same deal as in Project 4
  - Assign a leader and scribe for each review
  - Follow the peer review guidelines
Project #7 Questions?
Minute responses

• What was your favorite and least favorite lecture of the semester so far, and why?