Chapter #8: Finite State Machine Design

8.5 Finite State Machine Word Problems
Finite State Machine Word Problems

Mapping English Language Description to Formal Specifications

Four Case Studies:

- Finite String Pattern Recognizer
- Complex Counter with Decision Making
- Traffic Light Controller
- Digital Combination Lock

*We will use state diagrams and ASM Charts*
Finite State Machine Word Problems

1. Finite String Pattern Recognizer

A finite string recognizer has one input (X) and one output (Z). The output is asserted whenever the input sequence \( ...010... \) has been observed, as long as the sequence 100 has never been seen.

Step 1. Understanding the problem statement

Sample input/output behavior:

\[
\begin{align*}
X: & \quad 00101010010... \\
Z: & \quad 00010101000... \\
X: & \quad 11011010010... \\
Z: & \quad 00000001000... \\
X: & \quad 11011010010... \\
Z: & \quad 00000001000...
\end{align*}
\]

The outputs have been written to lag behind the inputs.
Finite State Machine Word Problems

Finite String Recognizer

Step 2. Draw State Diagrams/ASM Charts for the strings that must be recognized. I.e., 010 and 100.

Moore State Diagram
Reset signal places FSM in S0

Outputs 1

Loops in State

© R.H. Katz  Transparency No. 16-4
Finite State Machine Word Problems

Finite String Recognizer

Exit conditions from state S3:
- have recognized ...010
- if next input is 0 then have ...0100!
- if next input is 1 then have ...0101 = ...01 (state S2)

![Finite State Machine Diagram]
Finite State Machine Word Problems

**Finite String Recognizer**

Exit conditions from S1: recognizes strings of form \( \ldots 0 \) (no 1 seen)
loop back to S1 if input is 0

Exit conditions from S4: recognizes strings of form \( \ldots 1 \) (no 0 seen)
loop back to S4 if input is 1
Finite State Machine Word Problems

Finite String Recognizer

S2, S5 with incomplete transitions

S2 = ...01; If next input is 1, then string could be prefix of (01)1(00)
S4 handles just this case!

S5 = ...10; If next input is 1, then string could be prefix of (10)1(0)
S2 handles just this case!

Final State Diagram
Finite State Machine Word Problems

Finite String Recognizer

Verify I/O behavior of your state diagram to insure it functions like the specification

Sample input/output behavior:

X: 00101010010...
Z: 00010101000...

X: 11011010010...
Z: 00000001000...

X: 11011010010...
Z: 00000001000...
Finite State Machine Word Problems

2. Complex Counter

A sync. 3 bit counter has a mode control M. When M = 0, the counter counts up in the binary sequence. When M = 1, the counter advances through the Gray code sequence.

Binary: 000, 001, 010, 011, 100, 101, 110, 111
Gray: 000, 001, 011, 010, 110, 111, 101, 100

Example Valid I/O behavior:

<table>
<thead>
<tr>
<th>Mode Input M</th>
<th>Current State</th>
<th>Next State (Z2 Z1 Z0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>0</td>
<td>001</td>
<td>010</td>
</tr>
<tr>
<td>1</td>
<td>010</td>
<td>110</td>
</tr>
<tr>
<td>1</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>1</td>
<td>111</td>
<td>101</td>
</tr>
<tr>
<td>0</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>0</td>
<td>110</td>
<td>111</td>
</tr>
</tbody>
</table>
Finite State Machine Word Problems

Complex Counter

Derive the State Diagram using Moore FSM
Finite State Machine Word Problems

3. Traffic Light Controller

A busy highway is intersected by a little used farmroad. Detectors sense the presence of cars waiting on the farmroad.

With no car on farmroad, light remain green in highway direction. If vehicle on farmroad, highway lights go from Green to Yellow to Red, allowing the farmroad lights to become green. These stay green only as long as a farmroad car is detected but never longer than a set interval. When these are met, farm lights transition from Green to Yellow to Red, allowing highway to return to green. Even if farmroad vehicles are waiting, highway gets at least a set interval as green.

Assume you have an interval timer that generates a short time pulse (TS) and a long time pulse (TL) in response to a set (ST) signal. TS is to be used for timing yellow lights and TL for green lights.
Finite State Machine Word Problems

Traffic Light Controller

Picture of Highway/Farmroad Intersection:
Finite State Machine Word Problems

Traffic Light Controller

- Tabulation of Inputs and Outputs:

<table>
<thead>
<tr>
<th>Input Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset</td>
<td>place FSM in initial state</td>
</tr>
<tr>
<td>C</td>
<td>detect vehicle on farmroad</td>
</tr>
<tr>
<td>TS</td>
<td>short time interval expired</td>
</tr>
<tr>
<td>TL</td>
<td>long time interval expired</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HG, HY, HR</td>
<td>assert green/yellow/red highway lights</td>
</tr>
<tr>
<td>FG, FY, FR</td>
<td>assert green/yellow/red farmroad lights</td>
</tr>
<tr>
<td>ST</td>
<td>start timing a short or long interval</td>
</tr>
</tbody>
</table>

- Tabulation of Unique States: Some light configuration imply others

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>Highway green (farmroad red)</td>
</tr>
<tr>
<td>S1</td>
<td>Highway yellow (farmroad red)</td>
</tr>
<tr>
<td>S2</td>
<td>Farmroad green (highway red)</td>
</tr>
<tr>
<td>S3</td>
<td>Farmroad yellow (highway red)</td>
</tr>
</tbody>
</table>
Traffic Light Controller

Refinement of ASM Chart:

Start with basic sequencing and outputs:

![Diagram showing the states and transitions for a traffic light controller]
Finite State Machine Word Problems

Traffic Light Controller
Determine Exit Conditions for S0:
Car waiting and Long Time Interval Expired - C • TL

Equivalent ASM Chart Fragments

© R.H. Katz  Transparency No. 16-15
Finite State Machine Word Problems

Traffic Light Controller

S1 to S2 Transition:
Set ST on exit from S0
Stay in S1 until TS asserted
Similar situation for S3 to S4 transition
Finite State Machine Word Problems

Traffic Light Controller

S2 Exit Condition: no car waiting OR long time interval expired
Finite State Machine Word Problems

Traffic Light Controller

Compare with state diagram:

Advantages of State Charts:

- Concentrates on paths and conditions for exiting a state
- Exit conditions built up incrementally, later combined into single Boolean condition for exit
- Easier to understand the design as an algorithm
Finite State Machine Word Problems

4. Digital Combination Lock

"3 bit serial lock controls entry to locked room. Inputs are RESET, ENTER, 2 position switch for bit of key data. Locks generates an UNLOCK signal when key matches internal combination. ERROR light illuminated if key does not match combination. Sequence is: (1) Press RESET, (2) enter key bit, (3) Press ENTER, (4) repeat (2) & (3) two more times."

Problem specification is incomplete:

• how do you set the internal combination?

• exactly when is the ERROR light asserted?

Make reasonable assumptions:

• combination hardwired into next state logic vs. stored in internal register

• assert as soon as error is detected vs. wait until full combination has been entered

==> Our design: registered combination plus error after full combination
Finite State Machine Word Problems

**Digital Combination Lock**

Understanding the problem: draw a block diagram …

![Block Diagram of Digital Combination Lock](image)

**Inputs:**
- Reset
- Enter
- Key-In
- L0, L1, L2

**Outputs:**
- Unlock
- Error
Finite State Machine Word Problems

*Digital Combination Lock*

Derive the ASM or the State Diagram Implementation
HW #16 -- Section 8.5