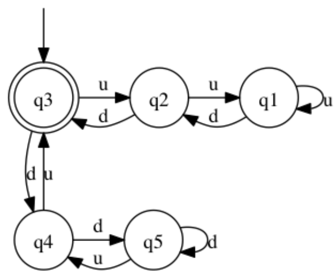
Tutorial Week 8

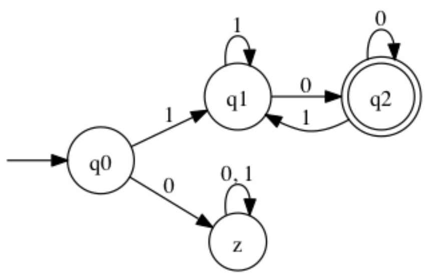
1. The formal description of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{\text{u,d}\}, \delta, q3, \{q3\})$ where δ is given by the following table. Give the state diagram of this machine.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

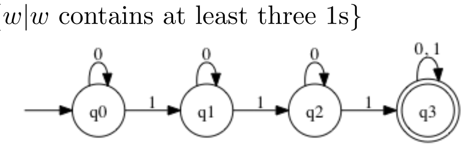


2. Give state diagrams of DFAs that recognize the following languages. In all parts the alphabet is 0, 1

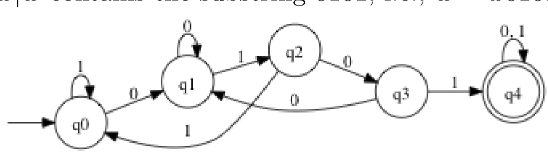
a. $\{w|w$ begins with a 1 and ends with a 0$\}$



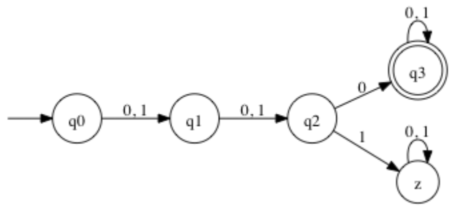
b. $\{w|w$ contains at least three 1s$\}$



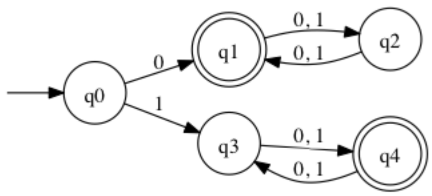
c. $\{w|w$ contains the substring 0101, i.e., $w = x0101y$ for some strings $x, y\}$



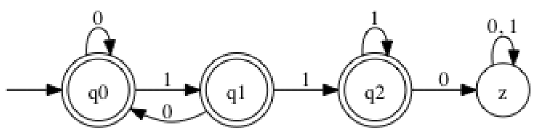
d. $\{w|w$ has length at least 3 at its third symbol is a 0$\}$



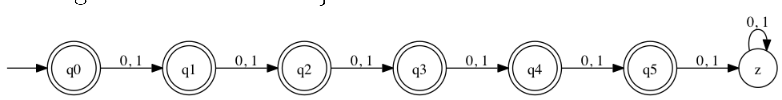
e. $\{w|w$ starts with a 0 and has odd length, or starts with a 1 and has even length$\}$



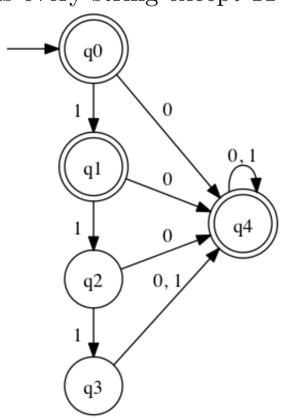
f. $\{w|w$ doesn't contain the substring 110$\}$



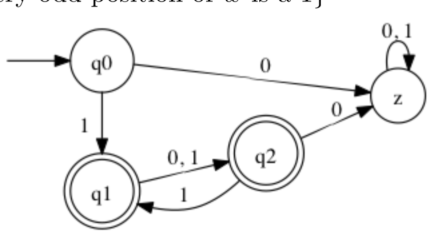
g. $\{w|$ the length of w is at most 5$\}$



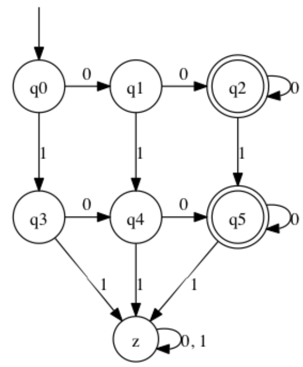
h. $\{w|w$ is every string except 11 and 111$\}$



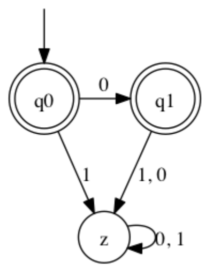
i. $\{w|$ every odd position of w is a 1$\}$



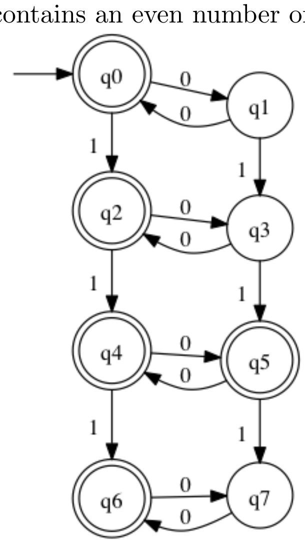
j. $\{w | w$ contains at least two 0s and at most one 1$\}$



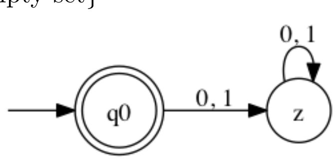
k. $\{\epsilon, 0\}$



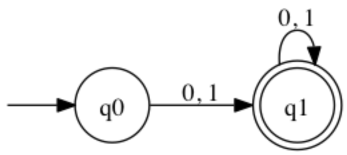
l. $\{w | w$ contains an even number of 0s, or contains exactly two 1s$\}$



m. $\{$The empty set$\}$



n. {All strings except the empty string}

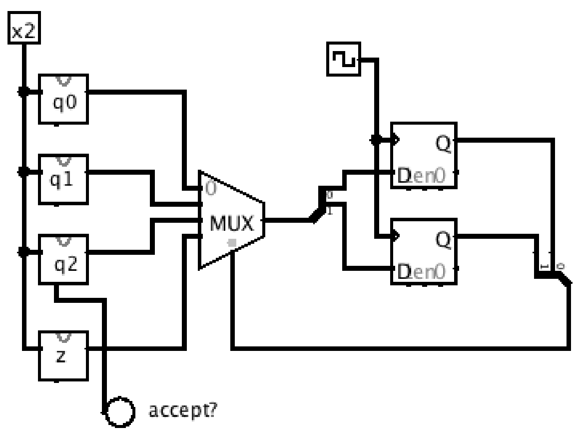


3. Convert question h. into a circuit

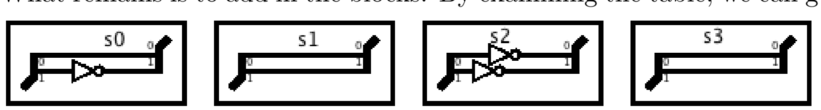
First example:
Let's start with question 1. We know there are 4 possible states, thus we need 2 bits to represent this ($log_2(4)$).
Secondly we know there are 2 possible transitions, 1 and 0.
We will label each state by its corresponding number (and make z become state 11).
Now we can construct a table of transitions.

Current State	Input	New State
00	0	11
	1	01
01	0	10
	1	01
10	0	10
	1	01
11	0	11
	1	11

Then we construct our skeleton circuit. This will have 2 flip-flops (one per state bit), and a 2-bit wire,
We then add an LED to each accepting state, to say whether or not we are accepted:



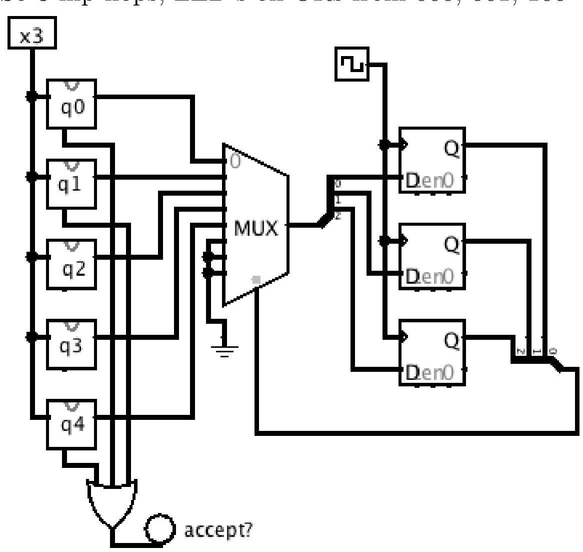
What remains is to add in the blocks. By examining the table, we can get the answers:

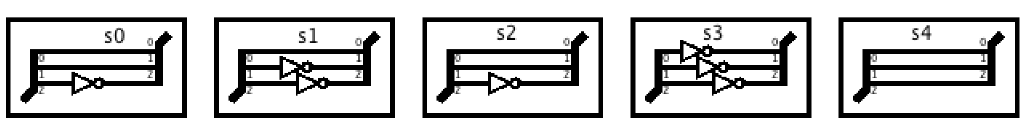


Finally, the solution to h:
5 States $\rightarrow$ 3 bits
2 inputs $\rightarrow$ 1 bit wire

Current State	Input	New State
000	0	100
	1	001
001	0	100
	1	010
010	0	100
	1	011
011	0	100
	1	100
100	0	100
	1	100
Unused		
⋮		

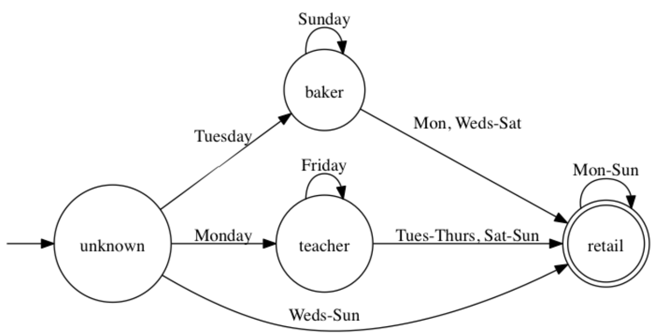
So 3 flip flops, LED's on ORs from 000, 001, 100





The accept single-bit width wires are calculated by simply creating a circuit that will output true if the state-number is inputed, i.e. for q_1 it is $\sim i_0 \wedge \sim i_1 \wedge i_2$

4. Dave has been telling everyone he has a great teaching job, but his friend Brian suspects he actually works in retail. Brian knows Dave used to work as a baker, but he thinks by looking at the day of the week Dave starts to work, then looking at the day Dave ends, he can determine what job he has based on the break.
   Teachers have weekends off, Bakers have Sunday & Monday off, and Retail workers could have any day of the week off.
   Design a DFA that will start in an "unknown" state, then take in two consecutive inputs, first the day of the week Dave starts on, then the day Dave ends on. Based on the day Dave starts your DFA should have an initial guess as to his profession. The DFA accepts Brian's guess, i.e. if Dave is a retail worker.



5. Now design the outline for a circuit for the last DFA. Complete everything but the boxes for the state transitions:

   4 states $\rightarrow$ 2 bits
   7 inputs $\rightarrow$ $log_2(8) = 3$ bit wire

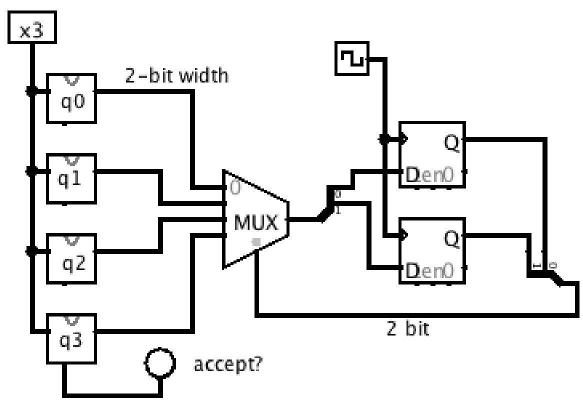
   Assign values to each input:

Monday	Tuesday	Wednesday	...	Sunday
0$\rightarrow$000	1$\rightarrow$001	2$\rightarrow$010	3...5	6$\rightarrow$110

   Then state:

Current State	Input	New State
00	000	10
(unknown)	001	01
	010-110	11
01	110	01
(baker)	000	11
	010-101	11
10	100	10
(teacher)	001-011	11
	101-110	11
11	000-110	11
(retail)		

So 2 flip flops, LED on state 3. The final circuit:



6. Now generate regular expressions based on the previous DFAs in Question 2

   a. 1+(.)*0
   b. 0*1+0*1+0*1+0*
   c. (.)*0101(.)*
   d. (.){2}0(.)*
   e. (0|(1(.)))((.)(.))*
   f. 0*(10+)*1*
   g. (.){,5}
   h. 0*1*0+1*0*|14,
   i. (1(.))*
   j. (0+1?0+)|(100+)|(00+1)
   k. ϵ|0
   l. (00)*|(0*10*10*)
   m. ^[01]*
   n. (.)+