

Regarding Directed Graphs and Binary Sequences

Ethan Jensen

October 19th, 2019

1 Finite State Machines as Directed Graphs

The follow is an expert from “Finite-State Machines.” Digital Design: an Embedded Systems Approach Using Verilog, by Peter J. Ashenden

”**State machines** are defined by a sequence of inputs, a sequence of outputs, a set of states, a transition function that governs transitions between states, and an output function”

”A **state transition diagram** is an abstract diagrammatic representation of a finite-state machine. It uses a circle, or ”bubble,” to represent each state. Directed arcs between state bubbles represent transitions from one state to another. An arc may be labeled with a combination of input values that allow the transition to occur”.¹

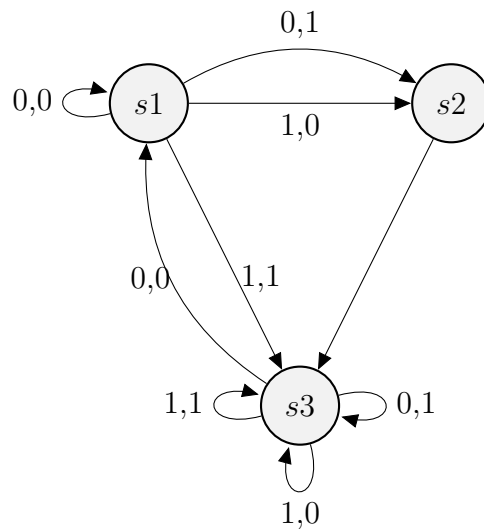


Figure 1: A state transition diagram.

2 Finite-State Machines in Sequence Detection

Finite-state machines can be used to detect sequences of numbers. To simplify, this paper will only be studying finite-state machines that detect sequences of 0s and 1s. This also gives the finite-state machines a digital analog. Consider the following example.

Ex. A system is given an a semi-random input stream of 0s and 1s. 10111100111000101
Draw a state transition diagram that detects if and when the sequence "110" appears when iterating through the input stream.

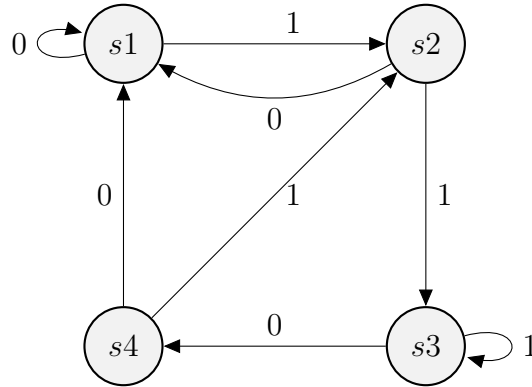


Figure 2: Sequence "110" detector.

In Figure 2, s1 represents the initial state of the state-machine. When the state-machine gets a new input, it follows the corresponding arrow to the corresponding state. Reaching the final state, s4 represents the system successfully detecting "110". We will call these state-machines detectors.

Intuitively, every three-bit sequence will have its own unique detector. Each detector is a directed graph with very special properties. However, any given directed graph with 4 vertices will almost certainly not be a detector for any 3-bit sequence. There are 8 possible 3-bit sequences. In contrast, there are an infinite number of possible directed graphs with 4 vertices! The list of conditions for directed graphs to be a detector is very steep. One can compile a list of conditions a directed graph will need to satisfy to become a "detector", and in this way, an isomorphism can be established between binary numbers and the set of all "detector" graphs.

3 Properties of Detectors

Definition 1.1. A **detector** is a directed graph defined by a sequence of 0s and 1s, an initial vertex, and a final vertex. Progression through the graph is possible, by referring to a series of 1 bit binary inputs. The final vertex is reached if and only if that sequence of inputs appear.

Theorem 1.1. An n -bit detector has exactly $n+1$ vertices.

Proof. Consider an n -bit detector.

The detector must have an initial vertex. (Def. 1.1)

Progress to the final vertex can be measured discretely by starting at the front of the input sequence and counting how many bits are matched by the detector sequence. For each discrete step of progression, one vertex must be defined. Since the detector sequence is n -bit, n progression vertices must be defined.

Thus, the number of all vertices in an n -bit detector-the initial and the progression vertices-is $n + 1$.

■

Theorem 1.2 An n -bit detector has $2n$ edges.

Proof. Consider an n -bit detector.

The detector must have n vertices. (Thm. 1.1)

The input sequence is binary (Def. 1.1) so each vertex has exactly 2 edges coming out. Thus, an n -bit detector has $2n$ edges.

■

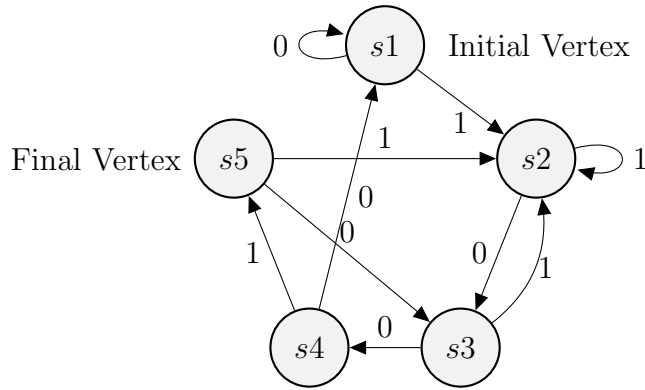


Figure 3: Sequence "1001" detector.

Note that the 4-bit detector shown in Figure 3 has 5 vertices. s_1 is the initial vertex, while s_2 , s_3 , s_4 , and s_5 are the progression vertices. Also note that the Figure has 10 edges, where each vertex has 2 pointing.

4 Properties of Detectors cont.

Theorem 1.3 Every vertex s_k in an n -bit detector, except for s_n , has one output going to $s_k + 1$ and one output going to a some vertex s_h where $0 \leq h \leq k$

Proof. Consider a k -bit detector defined with some sequence of bits $b_0 b_1 \dots b_{k-1}$

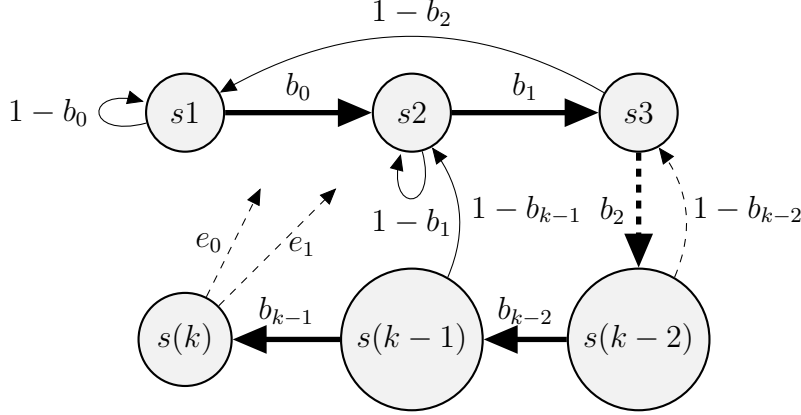


Figure 4: Some k bit detector.

In order to detect the sequence, we need a the sequence of edges $b_0, b_1, b_2 \dots b_{k-2}, b_{k-1}$ to chain the first vertex, s_1 to the final vertex $s(k)$.

By Def. 1.1, each vertex has two edges coming out of it. With one edge taken care of, each vertex has one free edge to go to another vertex.

This free edge cannot connect to any be directed at any vertex further down.

This would allow the detector to reach the final state in less than k steps.

Thus, the second edge of each vertex must be directed at either itself or a previous vertex.

■

Theorem 1.4 A detector has exactly 2 loops

Proof. Consider an n -bit detector.

5 References

- (1) “Finite-State Machines.” Digital Design: an Embedded Systems Approach Using Verilog, by Peter J. Ashenden, Morgan Kaufmann, 2008, pp. 179–185.