Digitalteknik EITF65

Lecture 2: Finite State Machines

August 27, 2020

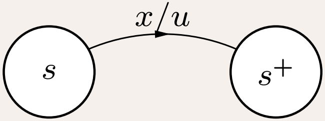
State transition graphs

Definition

A state transition graph is a graph where

- the state is an abstract (and compact) desription of the past. It contains all relevant information about the history.
- in each state there is one branch for each input.
- in each state there is one output combination for each input.

The branches between the states are drawn as arrows labeled with the input and the output.



$$\begin{tikzcd} s \arrow[r, bend left, "x/u"] & s^{+} \end{tikzcd}$$

State Machine

Definition (2.2)

A state machine is defined by the five-tuple $\mathcal{M} = \{\mathcal{I}, \mathcal{S}, \mathcal{Z}, \delta, \lambda\}$ where

- $\mathcal{I}$: is the input set.
- $\mathcal{S}$: is the state set.
- $\mathcal{Z}$: is the output set.
- δ: $\mathcal{S} \times \mathcal{I} \to \mathcal{S}$ is the *state transition* function.
- λ: $\mathcal{S} \times \mathcal{I} \to \mathcal{Z}$ is the *output* function.

The behavior of a state machine can be described by a state transition graph.

Trellis

Definition

A trellis is a (directed) graph where

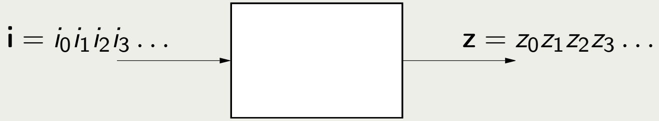
- each node represents a state at a certain depth.
- there is a set of *root vertices* that does not have any entering edges.

The depth of a root vertex is zero. The depth of other vertices are defined as the number of edges from a root vertex.

Parity check

Example 2.2

Check if a sequence has even or odd parity, i.e., if the number of ones is even or odd.



$\mathbf{i} = i_0 i_1 i_2 i_3 \ldots$
$\mathbf{z} = z_0 z_1 z_2 z_3 \ldots$

where

$$z_k = \begin{cases} \text{even} & \text{if } \sum_{\ell=0}^{k} i_\ell \text{ even} \\ \text{odd} & \text{if } \sum_{\ell=0}^{k} i_\ell \text{ odd} \end{cases}$$

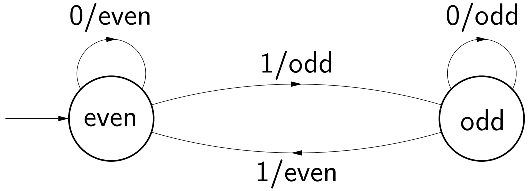
$\mathcal{I}$: $= \{0,1\}$ is the input set,

$\mathcal{Z}$: $= \{\text{even, odd}\}$ is the output set

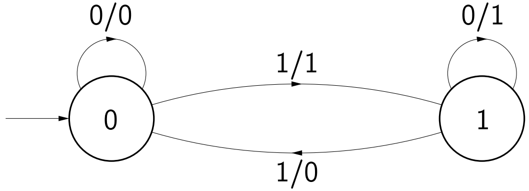
$\mathcal{S}$: $= \{\sum_{\ell=0}^{k-1} i_\ell \text{ even}, \sum_{\ell=0}^{k-1} i_\ell \text{ odd}\}$

We can use a state transition graph to define δ and λ.

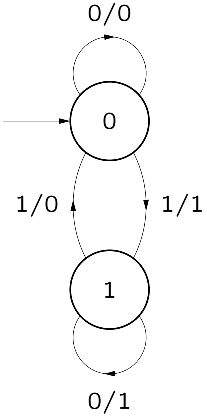
Parity check (Graph)

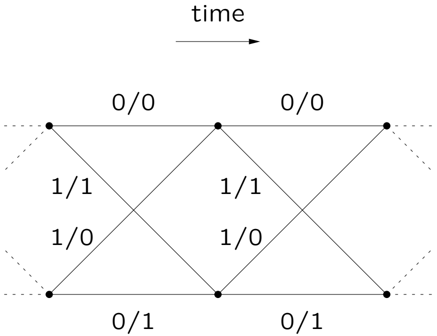


Let even $= 0$ and odd $= 1$ to get:



Parity check (Trellis)





Detector

Example 2.3

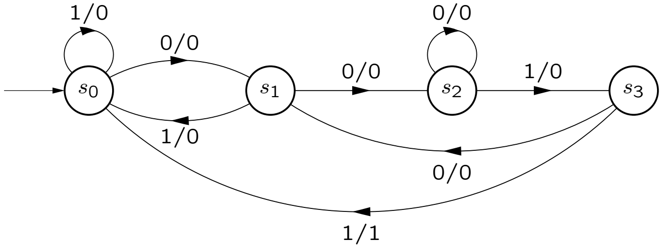
Give 1 as output if and only if the last four input bits matches 0011, otherwise 0.
E.g.,

$$\begin{array}{ll} \mathbf{i}: & \ldots 0011\ \ldots 0011\ \ldots \\ & \qquad\ \ \downarrow \qquad\quad\ \downarrow \\ \mathbf{z}: & \ldots 00010 \ldots 00010 \ldots \end{array}$$

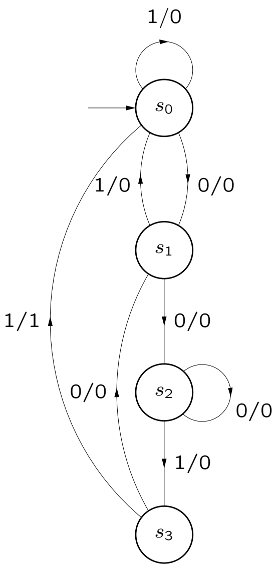
Question: What are

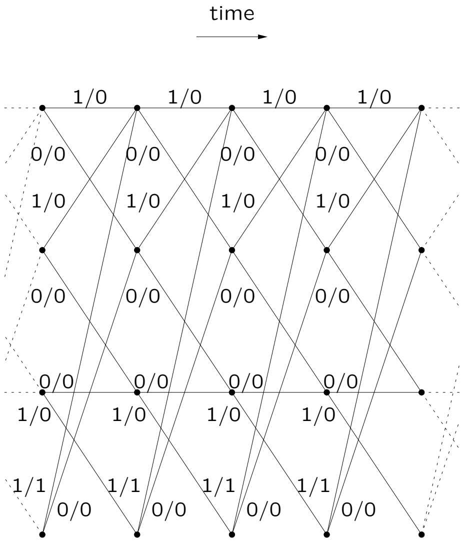
- $\mathcal{I}$, (input set)
- $\mathcal{S}$, (state set)
- $\mathcal{Z}$, (output set)
- δ, $\mathcal{S} \times \mathcal{I} \to \mathcal{S}$ (*state transition* function)
- λ, $\mathcal{S} \times \mathcal{I} \to \mathcal{Z}$ (*output* function)

Detector (Graph)



Detector (Trellis)





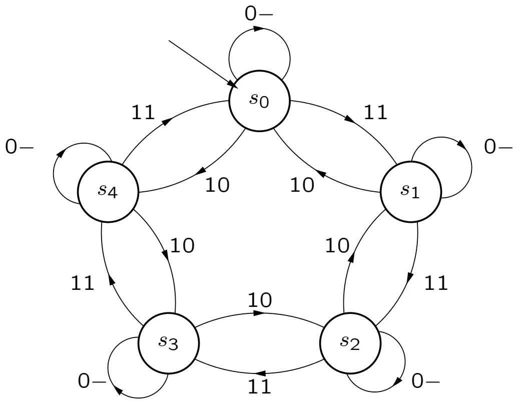
Modulo 5 counter

Example

Construct a system that works as a modulo 5 counter. It has two inputs, `Count` and `U/D`. If `Count` is 0 it does not count and if it is 1 it counts. If `U/D` is 1 it adds one to the result and if 0 it subtracts one. (There is no explicit output here.)

Count	U/D	state	next state
0	0	Q	Q
0	1	Q	Q
1	0	Q	Q-1 mod 5
1	1	Q	Q+1 mod 5

Modulo 5 counter (Graph)



The lion threat (two lions)

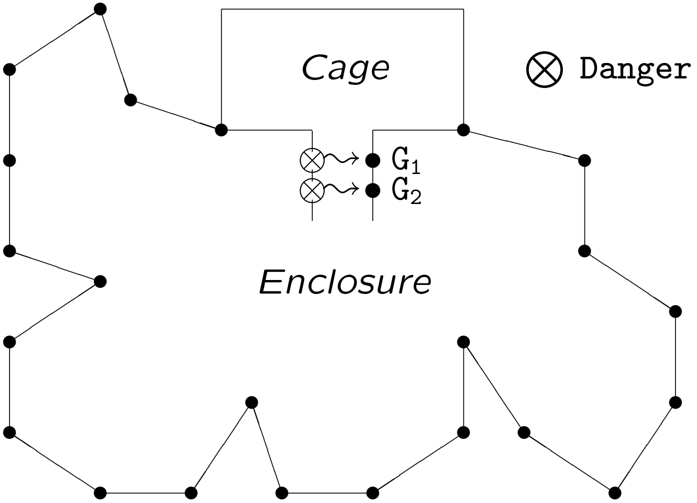
Example 2.4

In a zoo there is an enclosure for lions. It consists of two parts, a cage where the lions can rest and the enclosure, which is a big open field where the lions can stroll around. The enclosure is hilly and, therefore, difficult to overview. Since the keeper wants to clean the enclosure while the lions are in the cage he needs a system where a lamp, `Danger`, tells if there is any lion in the enclosure.

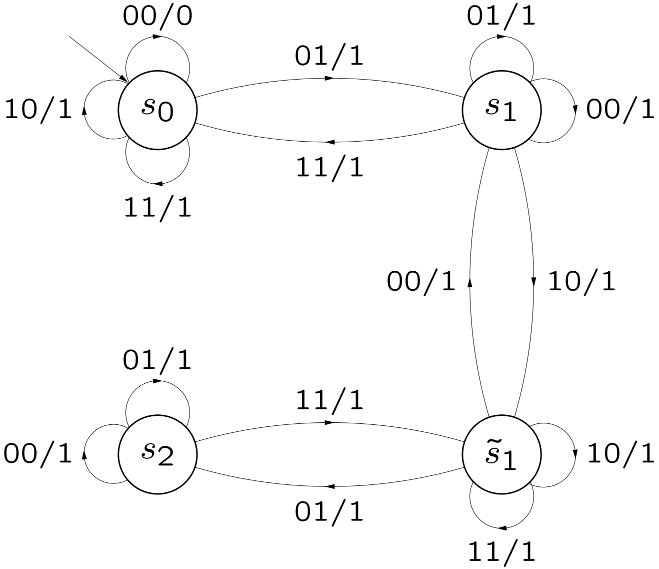
At the passage between the cage and the enclosure there are two photo detectors, $\mathtt{G}_1$ and $\mathtt{G}_2$ that are parted by half a meter. The lions are about two meters long and cannot pass the passage at the same time.

Construct a graph that solves the problem for two lions.

The lion threat (figure)



The lion threat (Graph)

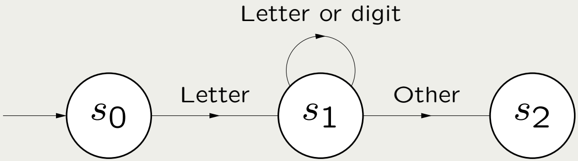


Variable name

Example 2.8

In programming languages the variable names often consist of a letter followed by letters or digits.

The name ends when the sequence of letters and digits ends, i.e., when some other character is written.



Event controlled or clock controlled

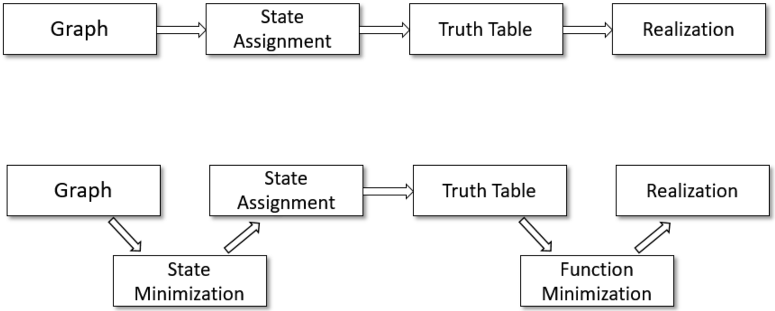
When do we change state, i.e., follow an edge?

- **New event** can cause a state transition. On each new input, look at input and state, and make a state transition. *Example*: Variable name.
- **External clock signal** can cause state transition. On each clock signal, look at input and state, and make a state transition. *Example*: Modulo 5 counter.

Clock controlled situations often require additional states since the clock can be very fast!

- Modulo 5 counter would require extra states
- Lion threat would not require extra states

Design flow


Graph
State Assignment
Truth Table
Realization
Graph
State Assignment
Truth Table
Realization
State Minimization
Function Minimization