CSC341 – Lab 2A

Anh Vu, Havin Lim, Carter English

January 2024

1 Connect the Dots

1.

- Σ is a finite set. An example would be $\Sigma = \{0, 1\}$.
- L is the language of a machine, which is the set of strings that machine accepts. An example of strings in language L of the machine in Part 2 (Tracing the Lines) is $11, 111, 1100$.

2.

- Q: A finite set of states
- Σ: A finite set
- δ: $Q \times \Sigma \to Q$
- $q0$: A state. $q0 \in Q$.
- F: A set of states. $F \subset Q$.

3.

$L(M)$ is the language accepted by M. A is the language of machine M, and M accepts/recognizes A.

4.

We trace the path created by w through M, and see whether we end up at an accept state of M.

2 Tracing the Lines

1.

- 01100: Accepted. $q_0 \to q_3 \to q_1 \to q_3 \to q_0 \to q_3$.

- 101101: Rejected. $q_0 \to q_2 \to q_1 \to q_3 \to q_1 \to q_2 \to q_0$.
- 11111: Rejected. $q_0 \to q_2 \to q_0 \to q_2 \to q_0 \to q_2$.
- ϵ: Reject. q_0.
- 1010101: Accepted. $q_0 \to q_2 \to q_1 \to q_3 \to q_0 \to q_2 \to q_1 \to q_3$.

2.

- q_0: A even total number of both 0's and 1's.
- q_1: An odd number of both 0's and 1's.
- q_2: An even number of 0's and an odd number of 1's.
- q_3: An odd number of 0's and an even number of 1's.

3.

$L(D)$ is the subset of Σ^* of strings which contain an odd number of 0's and an even number of 1's.

$$L(D) = \{s \in \Sigma^* \mid s \text{ contains an odd number of 0's and an even number of 1's}\}$$

3 DFA Design

1. Emptiness

$$\begin{aligned}
&Q = \{q_0, q_1\} \\
&\Sigma = \{0, 1\} \\
&\delta = f(q, \sigma) = q_1 \quad \forall q \in Q, \sigma \in \Sigma \\
&q_0 = q_0 \\
&F = \{q_0\}
\end{aligned}$$

2. Everything

$$\begin{aligned}
&Q = \{q_0\} \\
&\Sigma = \{0, 1\} \\
&\delta = f(q, \sigma) = q_0 \quad \forall q \in Q, \sigma \in \Sigma \\
&q_0 = q_0 \\
&F = \{q_0\}
\end{aligned}$$

3. Constants

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$
$$\Sigma = \{0, 1\}$$
$$q_0 = q_1$$
$$F = \{q_5\}$$

and δ defined by the following table:

State	Action(s)	Result State
q_1	0	q_2
q_2	1	q_3
q_3	1	q_4
q_4	0	q_5
q_1	1	q_0
q_2	0	q_0
q_3	0	q_0
q_4	1	q_0
q_5	$0, 1$	q_0
q_0	$0, 1$	q_0

As a high-level description of this DFA, the machine only accepts one single sequence of correct instructions. If along the way any "mistake" is made, the machine instantly goes to a "dead" state.

4. Prefixes and Suffixes

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$
$$\Sigma = \{0, 1\}$$
$$q_0 = q_1$$
$$F = \{q_5\}$$

and δ defined by the following table:

State	Action(s)	Result State
q_1	0	q_2
q_2	1	q_3
q_3	1	q_4
q_4	0	q_5
q_3	0	q_3
q_4	1	q_4
q_5	0	q_3
q_5	1	q_4
q_1	1	q_0
q_2	0	q_0
q_0	$0, 1$	q_0

As a high-level description of this DFA, the machine firstly requires the first two instructions to be exactly 01, else it fails instantly. We call the state we are in after the prefix q_3. Then, the machine must accept the simplest path to the accept state (0110, when x is empty). To ensure that the accepted suffix is always 10, any instruction starting from q_3 onwards must take the machine to an appropriate state that readily forms the suffix 10, regardless of the next instruction.

5. Parity

$$
\begin{aligned}
Q &= \{q_0, q_1\} \\
\Sigma &= \{0, 1\} \\
q_0 &= q_0 \\
F &= \{q_0\}
\end{aligned}
$$

and δ defined by the following table:

State	Action(s)	Result State
q_0	0	q_1
q_1	0	q_0
q_0	1	q_0
q_1	1	q_1

This DFA has a fairly simple high-level description: There are two states, even and odd number of 0's. We only need to toggle between these two states when a 0 instruction is received, and maintain the current state when a 1 is received.

6. Counts

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3\} \\
\Sigma &= \{0, 1\} \\
q_0 &= q_0 \\
F &= \{q_3\}
\end{aligned}
$$

and δ defined by the following table:

State	Action(s)	Result State
q_0	1	q_1
q_1	1	q_2
q_2	1	q_3
q_0	0	q_0
q_1	0	q_1
q_2	0	q_2
q_3	$0, 1$	q_3

On a high level, this DFA cares only about getting three 1's and nothing else. Therefore, our possible states must reflect the number of 1's equal or smaller than three. When there are fewer than three 1's, any 0 input does not change the state. When there are already at least three 1's, any further input does not move away from the accept state.

7. Constraints

$$
\begin{aligned}
&Q = \{q_0, q_1, q_2\} \\
&\Sigma = \{0, 1\} \\
&q_0 = q_0 \\
&F = \{q_0, q_1\}
\end{aligned}
$$

and δ defined by the following table:

State	Action(s)	Result State
q_0	0	q_1
q_1	1	q_2
q_0	1	q_0
q_1	0	q_1
q_2	$0, 1$	q_2

On a high level, this DFA keeps track of the current constraints based on what instructions it has seen so far. If it has only seen 1's, there are no constraints. If it has seen one or more 0's, the next instruction cannot be 1. Once a 01 is detected, any further instructions do not change the state since we have already failed the constraints.