Deterministic finite automata

Transition diagrams are useful *graphical* representations of instances of the mathematical concept of **deterministic finite automaton** (**DFA**).

Formally, a DFA $\mathcal{D}$ is a 5-tuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ where

1. a finite set of *states*, often noted Q;
2. an *initial state* $q_0 \in Q$;
3. a set of *final (or accepting) states* $F \subseteq Q$;
4. a finite set of *input symbols*, often noted Σ;
5. a *transition function* δ that takes a state and an input symbol and returns a state: if q is a state with an edge labeled a, the edge leads to state $\delta(q, a)$.

DFA/Recognised words

Independently of the interpretation of the states, we can define how a given word is accepted (or recognised) or rejected by a given DFA.

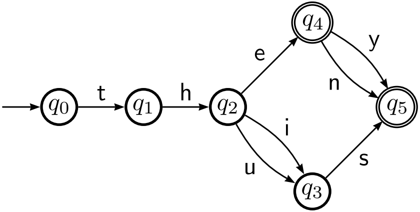
The word $a_1 a_2 \cdots a_n$, with $a_i \in \Sigma$, is recognised by the DFA $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ if

- for all $0 \leqslant i \leqslant n - 1$
- there is a sequence of states $q_i \in Q$ such as
- $\delta(q_i, a_{i+1}) = q_{i+1}$
- and $q_n \in F$.

The language recognised by $\mathcal{D}$, noted $L(\mathcal{D})$ is the set of words recognised by $\mathcal{D}$.

DFA/Recognised words/Example

For example, consider the following DFA:



The word "then" is recognised because there is a sequence of states $(q_0, q_1, q_2, q_4, q_5)$ connected by edges which satisfies

$$\delta(q_0, \mathsf{t}) = q_1$$
$$\delta(q_1, \mathsf{h}) = q_2$$
$$\delta(q_2, \mathsf{e}) = q_4$$
$$\delta(q_4, \mathsf{n}) = q_5$$

and $q_5 \in F$, i.e. q_5 is a final state.

DFA/Recognised language

It is easy to define formally $L(\mathcal{D})$.

Let $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$.

First, let us extend δ to words and let us call this extension $\hat{\delta}$:

- for all state $q \in Q$, let $\hat{\delta}(q, \varepsilon) = q$, where ε is the empty string;
- for all state $q \in Q$, all word $w \in \Sigma^*$, all input $a \in \Sigma$, let $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$.

Then the word w is recognised by $\mathcal{D}$ if $\hat{\delta}(q_0, w) \in F$.
The language $L(\mathcal{D})$ recognised by $\mathcal{D}$ is defined as

$$L(\mathcal{D}) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

DFA/Recognised language/Example

For example, in our last example:

$$
\begin{aligned}
\hat{\delta}(q_0, \epsilon) &= q_0 \\
\hat{\delta}(q_0, \mathsf{t}) &= \delta(\hat{\delta}(q_0, \epsilon), \mathsf{t}) = \delta(q_0, \mathsf{t}) = q_1 \\
\hat{\delta}(q_0, \mathsf{th}) &= \delta(\hat{\delta}(q_0, \mathsf{t}), \mathsf{h}) = \delta(q_1, \mathsf{h}) = q_2 \\
\hat{\delta}(q_0, \mathsf{the}) &= \delta(\hat{\delta}(q_0, \mathsf{th}), \mathsf{e}) = \delta(q_2, \mathsf{e}) = q_4 \\
\hat{\delta}(q_0, \mathsf{then}) &= \delta(\hat{\delta}(q_0, \mathsf{the}), \mathsf{n}) = \delta(q_4, \mathsf{n}) = q_5 \in F
\end{aligned}
$$

DFA/Transition diagrams

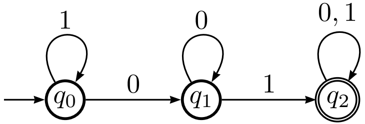
We can also redefine transition diagrams in terms of the concept of DFA.

A transition diagram for a DFA $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ is a graph defined as follows:

1. for each state q in Q there is a **node**, i.e. a single circle with q inside;
2. for each state $q \in Q$ and each input symbol $a \in \Sigma$, if $\delta(q, a)$ exists, then there is an **edge**, i.e. an arrow, from the node denoting q to the node denoting $\delta(q, a)$ labeled by a; multiple edges can be merged into one and the labels are then separated by commas;
3. there is an edge coming to the node denoting q_0 without origin;
4. nodes corresponding to final states (i.e. in F) are double-circled.

DFA/Transition diagram/Example

Here is a transition diagram for the language over alphabet $\{0, 1\}$, called **binary alphabet**, which contains the string 01:



DFA/Transition table

There is a compact textual way to represent the transition function of a DFA: a **transition table**.

The rows of the table correspond to the states and the columns correspond to the inputs (symbols). In other words, the entry for the row corresponding to state q and the column corresponding to input a is the state $\delta(q, a)$:

δ	$\ldots$	a	$\ldots$
$\vdots$			
q		$\delta(q, a)$	
$\vdots$			

DFA/Transition table/Example

The transition table corresponding to the function δ of our last example is

$\mathcal{D}$	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$\# q_2$	q_2	q_2

Actually, we added some extra information in the table: the initial state is marked with $\rightarrow$ and the final states are marked with $\#$.

Therefore, it is not only δ which is defined by means of the transition table here, but the whole DFA $\mathcal{D}$.

DFA/Example

We want to define formally a DFA which recognises the language L whose words contain an even number of 0's and an even number of 1's (the alphabet is binary).

We should understand that the role of the states here is to **not** to count the exact number of 0's and 1's that have been recognised before but this number **modulo 2**.

Therefore, there are four states because there are four cases:

1. there has been an even number of 0's and 1's (state q_0);
2. there has been an even number of 0's and an odd number of 1's (state q_1);
3. there has been an odd number of 0's and an even number of 1's (state q_2);
4. there has been an odd number of 0's and 1's (state q_3).

DFA/Example (cont)

What about the initial and final states?

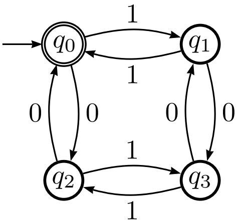
- State q_0 is the initial state because before considering any input, the number of 0's and 1's is zero and zero is even.
- State q_0 is the lone final state because its definition matches exactly the characteristic of L and no other state matches.

We know now almost how to specify the DFA for language L. It is

$$\mathcal{D} = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

where the transition function δ is described by the following transition diagram.

DFA/Example (cont)



Notice how each input 0 causes the state to cross the horizontal line.

Thus, after seeing an even number of 0's we are always above the horizontal line, in state q_0 or q_1, and after seeing an odd number of 0's we are always below this line, in state q_2 or q_3.

There is a vertically symmetric situation for transitions on 1.

DFA/Example (cont)

We can also represent this DFA by a transition table:

$\mathcal{D}$	0	1
$\# \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

We can use this table to illustrate the construction of $\hat{\delta}$ from *delta*. Suppose the input is 110101. Since this string has even numbers of 0's and 1's, it belongs to L, i.e. we expect $\hat{\delta}(q_0, 110101) = q_0$, since q_0 is the sole final state.

DFA/Example (cont)

We can check this by computing step by step $\hat{\delta}(q_0, \mathtt{110101})$, from the shortest prefix to the longest (which is the word $\mathtt{110101}$ itself):

$$
\begin{aligned}
\hat{\delta}(q_0, \varepsilon) &= q_0 \\
\hat{\delta}(q_0, \mathtt{1}) &= \delta(\hat{\delta}(q_0, \varepsilon), \mathtt{1}) && = \delta(q_0, \mathtt{1}) = q_1 \\
\hat{\delta}(q_0, \mathtt{11}) &= \delta(\hat{\delta}(q_0, \mathtt{1}), \mathtt{1}) && = \delta(q_1, \mathtt{1}) = q_0 \\
\hat{\delta}(q_0, \mathtt{110}) &= \delta(\hat{\delta}(q_0, \mathtt{11}), \mathtt{0}) && = \delta(q_0, \mathtt{0}) = q_2 \\
\hat{\delta}(q_0, \mathtt{1101}) &= \delta(\hat{\delta}(q_0, \mathtt{110}), \mathtt{1}) && = \delta(q_2, \mathtt{1}) = q_3 \\
\hat{\delta}(q_0, \mathtt{11010}) &= \delta(\hat{\delta}(q_0, \mathtt{1101}), \mathtt{0}) && = \delta(q_3, \mathtt{0}) = q_1 \\
\hat{\delta}(q_0, \mathtt{110101}) &= \delta(\hat{\delta}(q_0, \mathtt{11010}), \mathtt{1}) && = \delta(q_1, \mathtt{1}) = q_0 \in F
\end{aligned}
$$