Formal Defn of a DFA - and an "OOP oriented" version

deterministic finite automaton

A \( (M) \) is a 5-tuple \( M = (Q, \Sigma, \delta, s, F) \) class DFA if

- \( Q \) is a finite set of states
- \( \Sigma \) is the input alphabet
- \( S \), a member of \( Q \), is the start state
- \( F \), a subset of \( Q \), is the set of (desired) final states - also called accepting states, and
- \( \delta \) : \( Q \times \Sigma \to Q \) is the transition function.

Insofar as a function \( f : A \to B \) can be listed out as the set of ordered pairs \( \{ (a, b) \mid b = f(a) \} \), we can also list out \( \delta \) as a set of ordered triples called instructions - what you really want is for the class to have a member function, aka delegate (in C#), aka pointer - to - function pointer:

\[ \text{State} \ (\ast \delta) \ (\text{State}, \text{char}) \]

set \( \langle \text{State}, \text{char}, \text{State} \rangle \to \delta \);
Example

Abstract picture:
\[ \delta(q, c) = r \]

instructions
\[(q, c, r)\]

usually

\[ \text{L}(x) = \text{the number of blue discs} \]

\[ \text{c} \]

\[ \text{r} \]

\[ \text{F} \]

\[ \text{Even} \]

\[ \text{Odd} \]

\[ \text{Q} = \{ \text{Even, Odd} \} \]

\[ \Sigma = \{ 0, 1 \} \]

\[ s = \text{Even} \]

\[ F = \{ \text{Even} \} \]

\[ \delta(\text{Even}, 0) = \text{Even} \]

\[ \delta(\text{Odd}, 0) = \text{Odd} \]

\[ \delta(\text{Even}, 1) = \text{Odd} \]

\[ \delta(\text{Odd}, 1) = \text{Even} \]

\[ \delta = \{ (\text{Even, 0, Even}), (\text{Even, 1, Odd}), (\text{Odd, 0, Odd}), (\text{Odd, 1, Even}) \} \]

as a set of instructions

OK not to write out whole tables and just pool the diagram
So long as every state has a comment saying its purpose or meaning.

Given a DFA \( M \), \[ \text{L}(M) = \{ x \in \Sigma^* : M \text{ when run on input } x \text{ ends up in a state in } F \} \]

Language of \( M \). Here, \[ \text{L}(M) = \{ x : \#1(x) \text{ is even} \} \]
Dragon Slayer DFA on Winn; $\Sigma = \{0, \$, \$, \$\}

A DFA must define an action for every pair $(q, c)$ even if $q = " \text{dead}"$ (An NFA will be able to leave some configurations)

Text: Animated Door Example

$Q = \{\text{Open, Closed}\}$

$\Sigma = \{N, F, R, B\}$

Similar but different to prior clear example

L(M) = $\{x : x \text{ has at least one } 1 \text{ and does not end with } 10\}$

Regular expression: $(0 \cup 1)^{*}1$

Text: door begins closed and we desire that it ends closed.
Lecture page 4, Exercise 11, now we see an example.

\[ L(M) = \{ x : \exists y \frac{1}{y} \text{ is congruent to } 1 \mod 3 \} \]