DFA: Exercise

Draw the **transition diagram** of a **DFA** which **accepts/recognizes** the following language:

$\{ w \mid w \neq \varepsilon \wedge w \text{ has equal \# of alternating 0's and 1's} \}$

DFA: Formulation (1)

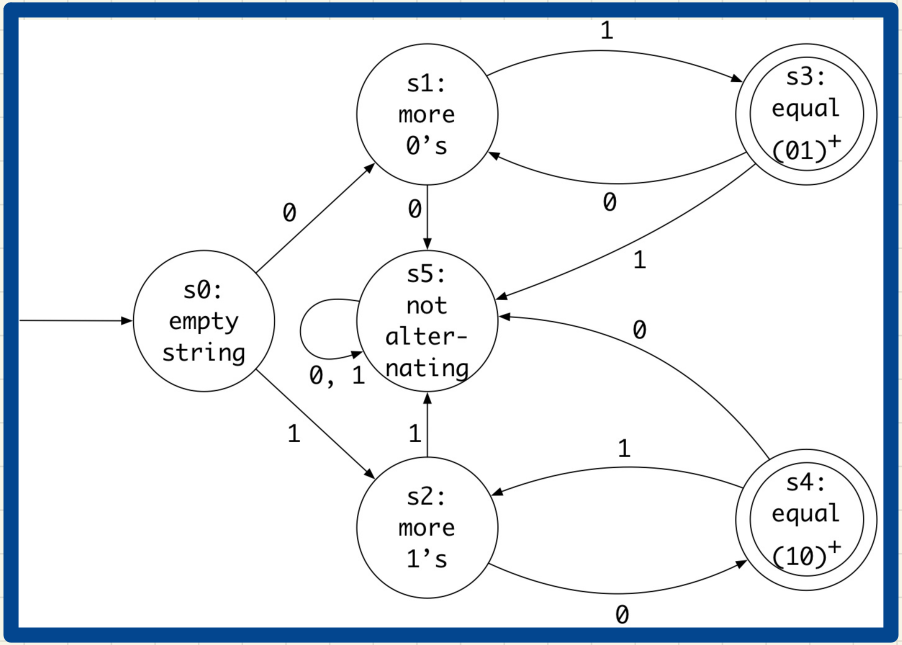
Language of a DFA

A *deterministic finite automata (DFA)* is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

$$L(M) = \left\{ a_1 a_2 \dots a_n \mid \right\}$$

e.g., 0101



DFA: Formulation (2)

Language of a DFA

$$\hat{\delta} : (Q \times \Sigma^*) \rightarrow Q$$

We may define $\hat{\delta}$ recursively, using δ!

$$\hat{\delta}(q, \epsilon) =$$
$$\hat{\delta}(q, xa) =$$

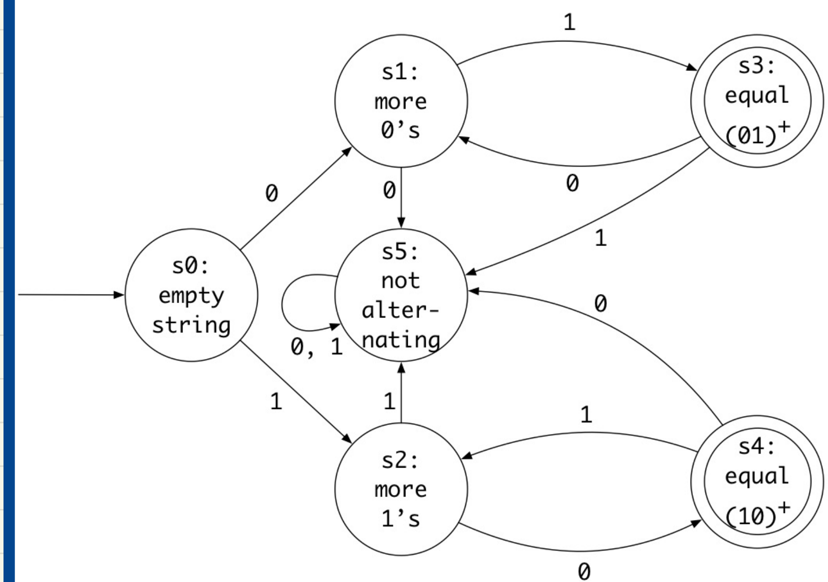
where $q \in Q$, $x \in \Sigma^*$, and $a \in \Sigma$

e.g., 010

$$L(M) = \{ w \mid \quad \}$$

A *deterministic finite automata (DFA)* is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$



DFA vs. NFA

Problem: Design a DFA that accepts the following language:

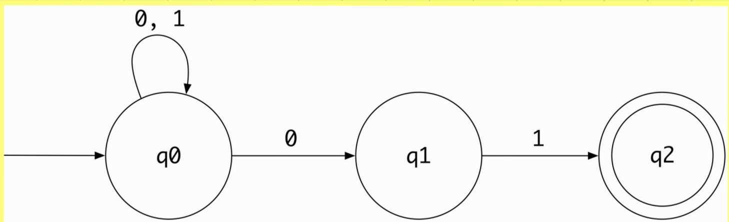
$$L = \{\ x01 \mid x \in \{0, 1\}^*\ \}$$

That is, L is the set of strings of 0s and 1s ending with 01.

A *non-deterministic finite automata (NFA)* that accepts the same language:

NFA Behaviour ≈ Alternative Universe

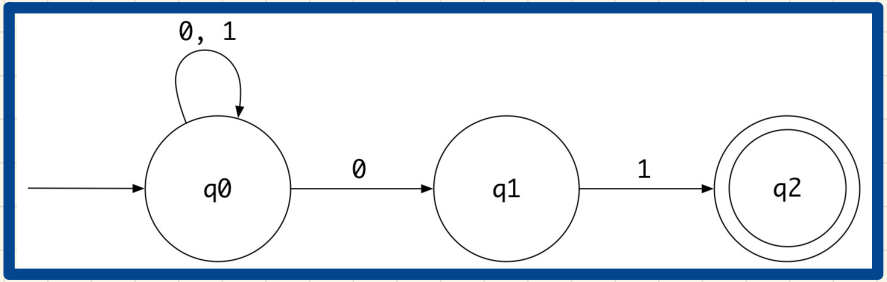
Obviously the **time continuum** has been disrupted, creating this new temporal event sequence resulting in this **alternate reality**. [redacted]



Trace: 00101

NFA: Processing Strings

How an NFA determines if an input 00101 should be accepted:



Read 0:

Read 0:

Read 0:

Read 0:

Read 0:

NFA: Formulation

Language of a NFA

A *nondeterministic finite automata (NFA)* is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

$$\hat{\delta} : (Q \times \Sigma^*) \to \mathbb{P}(Q)$$

We may define $\hat{\delta}$ recursively, using δ!

$$\hat{\delta}(q, \epsilon) =$$
$$\hat{\delta}(q, xa) =$$

where $q \in Q$, $x \in \Sigma^*$, and $a \in \Sigma$

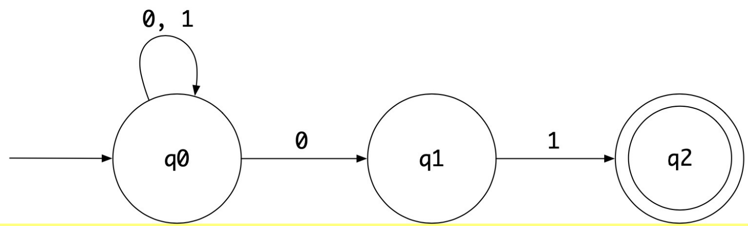
Given an input string 00101:

- **Read 0**: $\delta(q_0, 0) = \{ q_0, q_1 \}$
- **Read 0**: $\delta(q_0, 0) \cup \delta(q_1, 0) = \{ q_0, q_1 \} \cup \varnothing = \{ q_0, q_1 \}$
- **Read 1**: $\delta(q_0, 1) \cup \delta(q_1, 1) = \{ q_0 \} \cup \{ q_2 \} = \{ q_0, q_2 \}$
- **Read 0**: $\delta(q_0, 0) \cup \delta(q_2, 0) = \{ q_0, q_1 \} \cup \varnothing = \{ q_0, q_1 \}$
- **Read 1**: $\delta(q_0, 1) \cup \delta(q_1, 1) = \{ q_0, q_1 \} \cup \{ q_2 \} = \{ q_0, q_1, q_2 \}$

$$L(M) = \{ w \mid \quad \}$$

NFA to DFA: Subset Construction (Lazy Evaluation)

Given an ***NFA***:



Subset construction (with ***lazy evaluation***) produces a ***DFA*** with δ as:

state \ input	0	1

Subset Construction: Algorithmic Specification

Given an *NFA* $N = (Q_N, \Sigma_N, \delta_N, q_0, F_N)$:

ALGORITHM: *ReachableSubsetStates*
  INPUT: $q_0 : Q_N$; **OUTPUT:** $Reachable \subseteq \mathbb{P}(Q_N)$
PROCEDURE:
  $Reachable := \{ \{q_0\} \}$
  $ToDiscover := \{ \{q_0\} \}$
  while $(ToDiscover \neq \emptyset)$ {
    choose $S : \mathbb{P}(Q_N)$ such that $S \in ToDiscover$
    remove S from $ToDiscover$
    $NotYetDiscovered :=$
      $(\{ \{\delta_N(s,0) \mid s \in S\} \} \cup \{ \{\delta_N(s,1) \mid s \in S\} \}) \setminus Reachable$
    $Reachable := Reachable \cup NotYetDiscovered$
    $ToDiscover := ToDiscover \cup NotYetDiscovered$
  }
  return $Reachable$

state \ input	0	1
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$

epsilon-NFA: Motivation

Draw NFA

$$\left\{ xy \;\middle|\; \begin{array}{ll} & x \in \{0,1\}^* \\ \wedge & y \in \{0,1\}^* \\ \wedge & x \texttt{ has alternating } 0'\texttt{s and } 1'\texttt{s} \\ \wedge & y \texttt{ has an odd \# } 0'\texttt{s and an odd \# } 1'\texttt{s} \end{array} \right\}$$

$$\left\{ w : \{0,1\}^* \;\middle|\; \begin{array}{ll} & w \texttt{ has alternating } 0'\texttt{s and } 1'\texttt{s} \\ \vee & w \texttt{ has an odd \# } 0'\texttt{s and an odd \# } 1'\texttt{s} \end{array} \right\}$$