Deterministic Finite Automaton (DFA)

machine == automaton

↳ one: automaton (tekil)

↳ two: automata (çoğul)

Example: Did a heatwave occur?

Input: String of weather data

(*) Heatwave: temperature ≥ 45°C for 2 consecutive days

LANGUAGE OF THE MACHINE

L_M = { all strings containing 11 }

M accepts 11.

Also M accepts 110, 0110, 1011, 1010110.

WHY we call deterministic?

↳ Because the next steps are all determined.

Formal Definition of M = (Q, Σ, δ, q_0, F)

- Q is the set of states (q_0, q_1, q_2)
- Σ is the alphabet $\Sigma = \{1, 0\}$
- δ is the transition function

	1	0
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_2	q_2

q_0: Start State

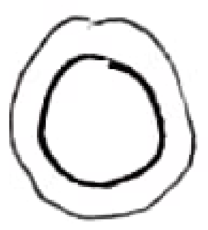
F: set of accept/final states $= \{q_2\}$

Regular Language

A language recognized by some finite automaton

SUMMARY

DFAs are 5-tuples $(Q, \Sigma, \delta, q_0, F)$

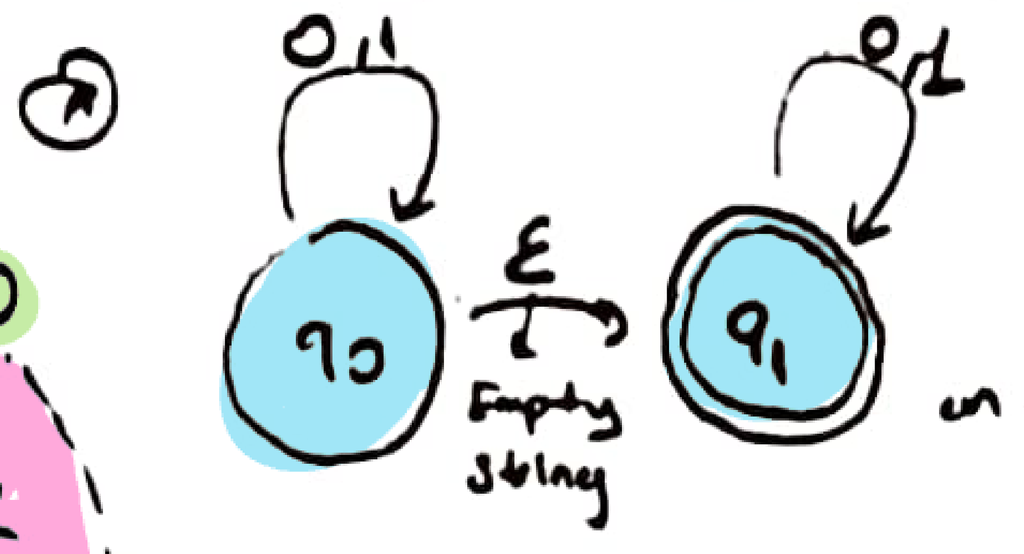
 = This means accept state

L_M is set of all accepted strings

NONDETERMINISTIC FINITE AUTOMATON (NFA)

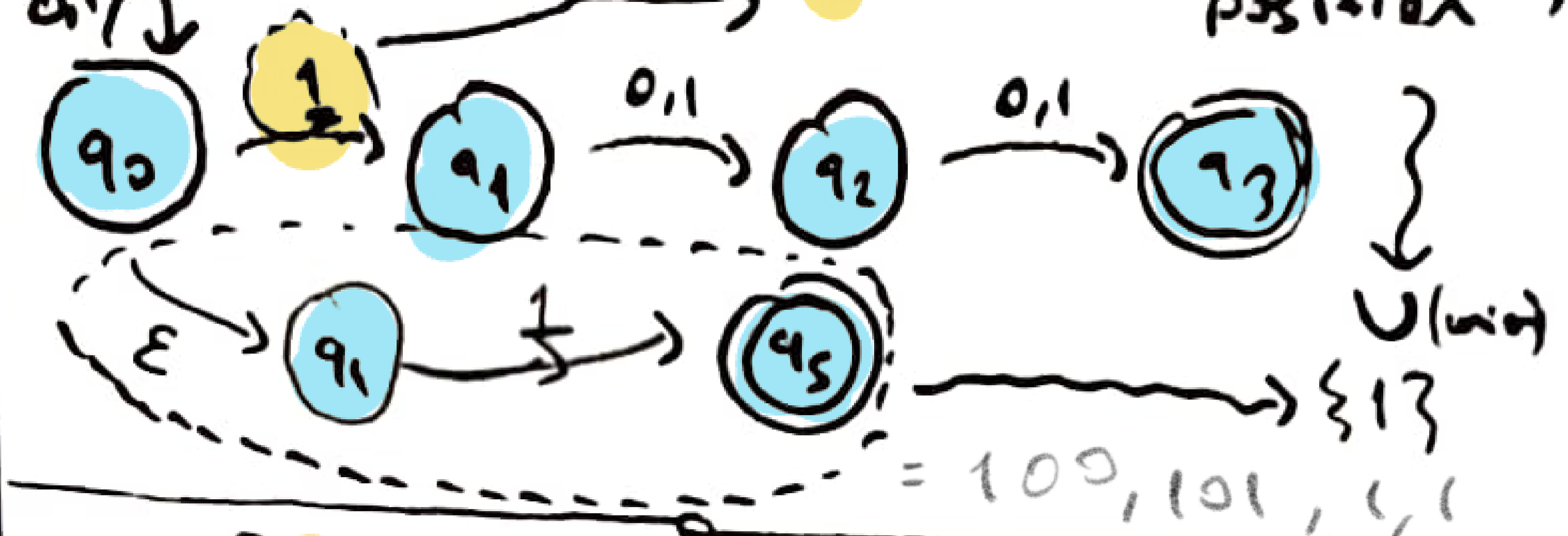
INPUT: 11 = goes two different path.

The NFA still accepts the string, because there is a path to an accept state.



Input: 01

Notice there is an empty string in here



(*) L_M = { x : x contains a 1 in the third final position }

= 100, 101, 11 ↳ ε

M = (Q, Σ, δ, q_0, F)

$\{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{1, 0\}$

q_0: start

F: $\{q_3, q_5\}$

delta: $\delta : Q \times \Sigma_\varepsilon \to P(Q)$

Σ_ε ↳ $\Sigma \cup \{\varepsilon\}$

(*) "δ" gives a set of possible states, instead of just 1

δ:

	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_4\}$

NFA can only do as much as DFA can so DFA is more powerful

SUMMARY

- NFAs are 5-tuples $(Q, \Sigma, \delta, q_0, F)$
- 

- 

That means set of possible states
- Languages recognized by an NFA regular languages