Non-deterministic FSMs

In this lecture we expand our scope to include non-deterministic finite state systems.

In the process we will discuss structured design in system specification, and how modularity can be used to control the design process.

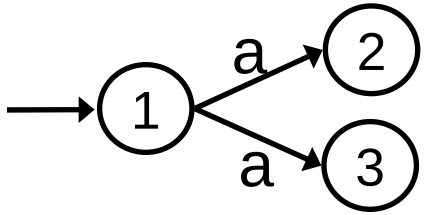
Formal Definition

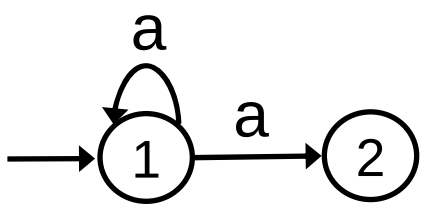
Non-deterministic FSM model, $M = (Q, \Sigma, s_0, F, \Delta)$

- Set of states, Q (one identified as initial)
- Set of input symbols, Σ ("input alphabet")
- Initial state, $s_0 \in Q$
- Set of accepting states, $F \subseteq Q$
- Transition relation, Δ, that can generate the **set** of successor states, given the current state and the set of transitions, T, each of the form (s_{i-1}, i_i, s_i).

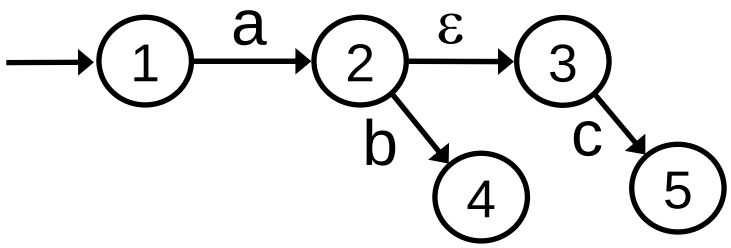
Two Types of Non-determinism

$s \in Q$ and $i \in \Sigma$ and $(s, i, s_1) \in T$ and $(s, i, s_2) \in T$ and $s_1 \neq s_2$





$(s, \varepsilon, s_1) \in T$



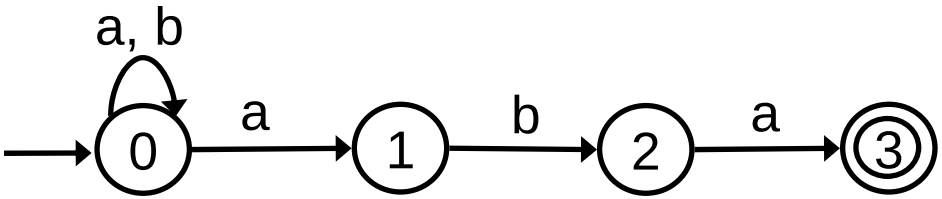
Language of a Non-deterministic FSM

Given: a FSM model, $M = (Q,\Sigma,s_0,F,\Delta)$
a string $x \in \Sigma^*$
accepts(x, M) succeeds when there is a trace for x in M ending in an accepting state ($s_a \in F$)

Language of M: $L(M) = \{x \in \Sigma^* \mid \text{accepts}(x, M)\}$

For some language, $L \subseteq \Sigma^*$
L is a regular language if, for some M, $L(M) = L$

Example Non-deterministic FSM

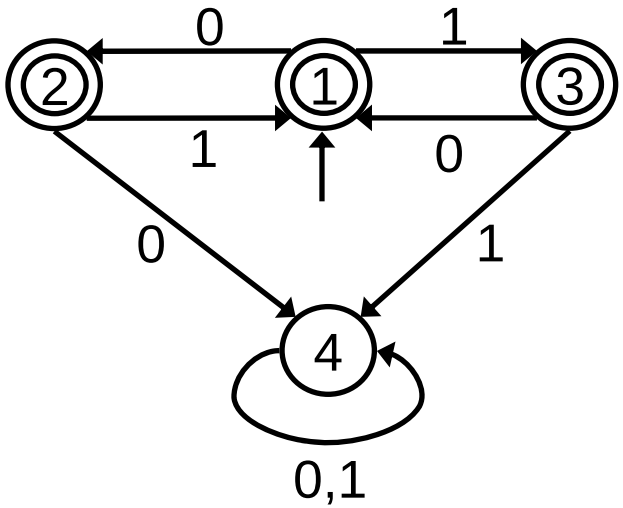


Accepts strings over Σ = {a, b} ending aba

$M = (\{0,1,2,3\}, \{a,b\}, 0, 3, \Delta)$

Δ	a	b
0	{0,1}	{0}
1		{2}
2	{3}	
3		

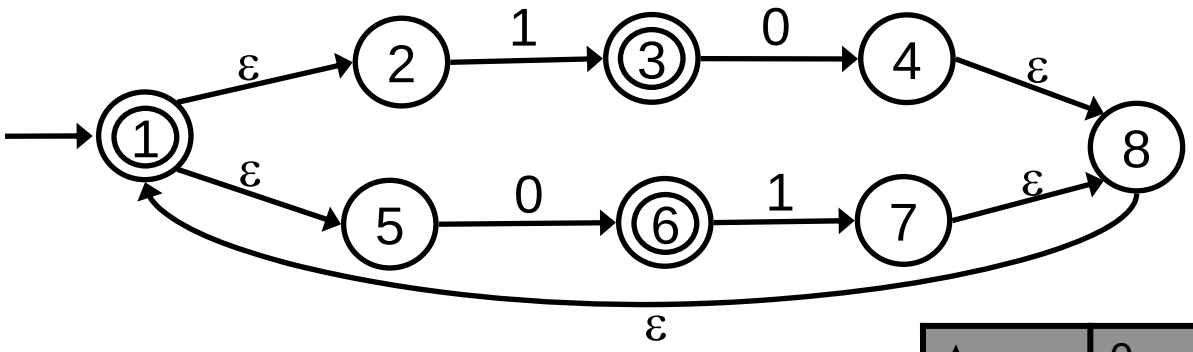
Return to Earlier Example



Accepts strings of 0's and 1's for which the number of 0's so far never exceeds number of 1's so far and *vice versa*.

What is this like as a non-deterministic FSM?

N-FSM for Earlier Example

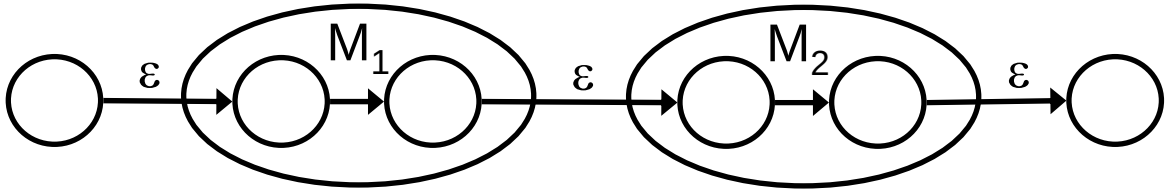


$M = (\{1,2,3,4,5,6,7,8\}, \{0,1\}, 0, \{1,3,6\}, \Delta)$

Δ	0	1	ε
1			{2, 5}
2		{3}	
3	{4}		
4			{8}
5	{6}		
6		{7}	
7			{8}
8			{1}

Structured Design: Sequence

M_1M_2 M_1 followed by M_2

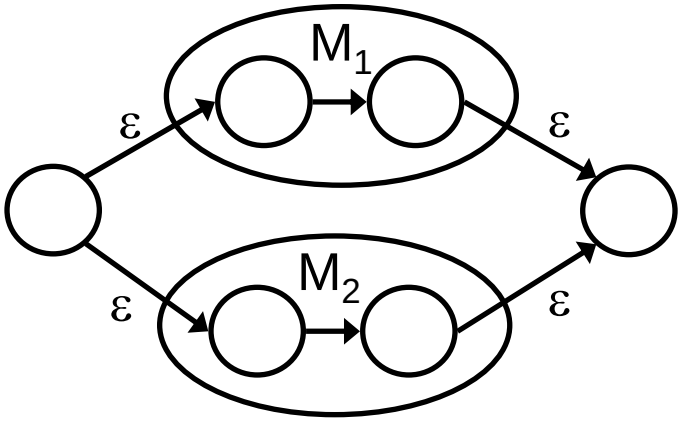


Accepts the language

$L(M_1)L(M_2) = \{ XY \mid X \in L(M_1) \text{ and } Y \in L(M_2) \}$

Structured Design: Choice

$M_1|M_2$ M_1 or M_2

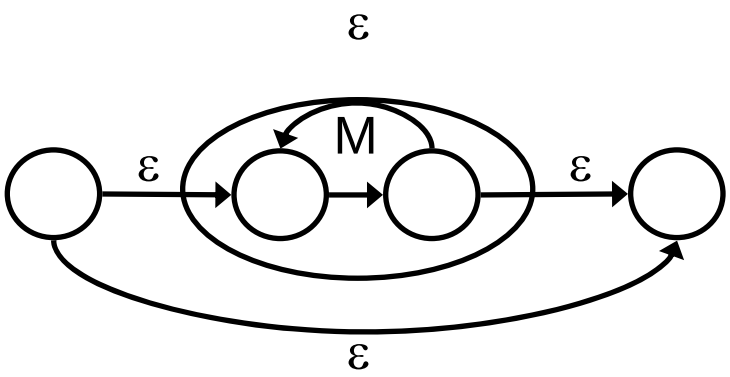


Accepts the language

$L(M_1) \cup L(M_2) = \{ X \mid X \in L(M_1) \text{ or } X \in L(M_2) \}$

Structured Design: Repeat

M* M repeated zero or more times



Accepts the language

$L(M)^* = \varepsilon \cup L(M) \cup L(M)^2 \cup L(M)^3$