Deterministic FSMs

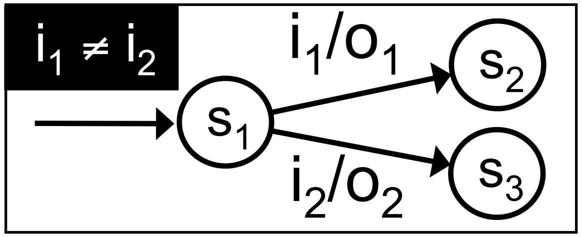
In this lecture we focus on a specific class of finite state system:

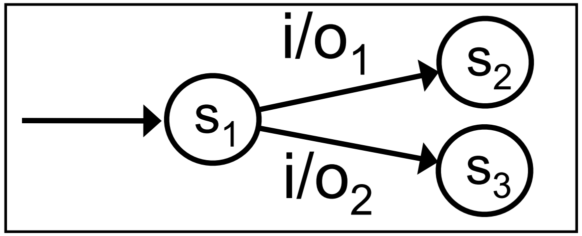
- deterministic FSMs
- that are acceptors

In the process we will show how logic can be used to specify FSMs.

Determinism

- In a deterministic FSM, all states have no more than one transition leaving the state for each input symbol.
- In a non-deterministic FSM, some states have more than one transition leaving to different successor states for the same input symbol.
- Sometimes non-deterministic FSMs are easier to define.
- Can always convert from a non-deterministic to a deterministic FSM.

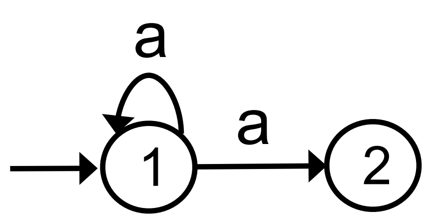




Determinism and Traces

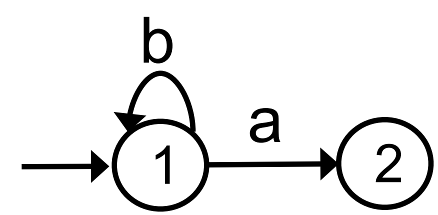
A FSM, M, is deterministic if for every string $x \in \Sigma^*$ there is at most one trace for x in M
(where Σ^* is the set of all strings in alphabet of M)

non-deterministic (choice)



Sequence	Trace
aa	[1,a,1,a,2]
aa	[1,a,1,a,1]

deterministic (no choice)

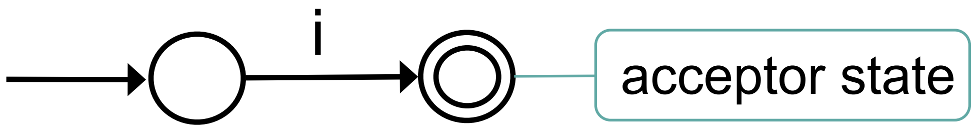


Sequence	Trace
ba	[1,b,1,a,2]
bb	[1,b,1,b,1]

Acceptors

Definition as before but:

- Empty output alphabet (all outputs are ε)
- Some states marked as accepting.



Input sequence is accepted if there is a trace from the initial state to an acceptor state.

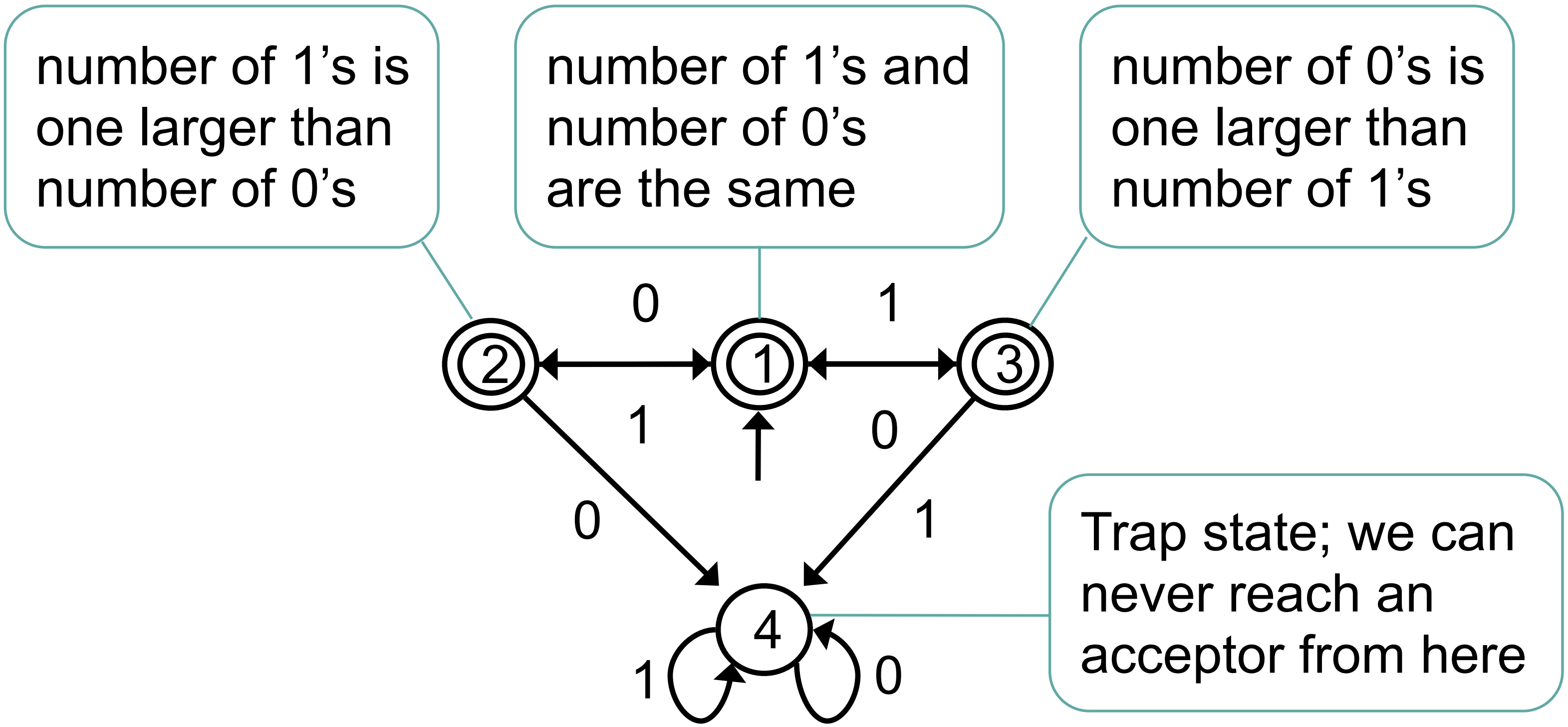
Language of the FSM is the set of sequences it accepts.

Formal Definition

Deterministic FSM acceptor 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 function, δ, that produces as output the successor state, given the current state and the set of transitions, T, each of the form (s_{i-1}, i_i, s_i).

Acceptor Example

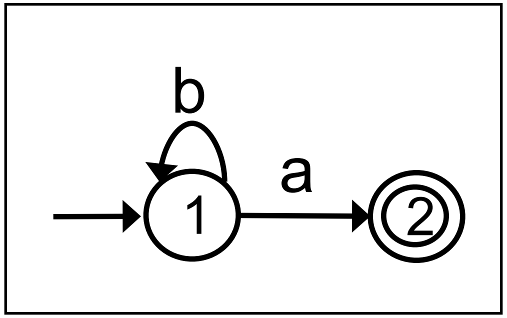


Accepts strings of 0's and 1's for which we never get more than two 0's or 1's consecutively.

Language of a Deterministic FSM Acceptor

Set of strings whose (unique) traces end in an accepting state ($s_a \in F$)

Accepts: ba
bba
bbba
…*etc*



Rejects: aba (no trace)
bbb (trace but not ending in accepting state)

Example: Number Systems

Let n(B,S) be a function giving the number represented by the string S in base B.
S is of the form $d_{k-1}\ldots d_2 d_1 d_0$ where each d_i is a digit.

$$n(B, d_{k-1}\ldots d_2 d_1 d_0) = d_0 + B\times d_1 + B^2\times d_2 + \ldots + B^{k-1}\times d_{k-1}$$

$$= \sum_{i=0}^{k-1} B^i d_i$$

Decimal

$$n(10, 123) = 3 + 10\times 2 + 10^2\times 1 = 123$$

Binary

$$n(2, 1111011) = 1 + 2\times 1 + 2^2\times 0 + 2^3\times 1 + 2^4\times 1 + 2^5\times 1 + 2^6\times 1 = 123$$

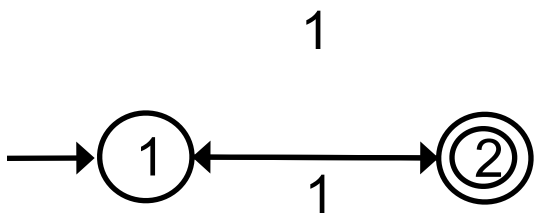
Unary

$$n(1, 1111111) = 1 + 1\times 1 + 1^2\times 1 + 1^3\times 1 + 1^4\times 1 + 1^5\times 1 + 1^6\times 1 = 7$$

FSM for Odd Unary Numbers

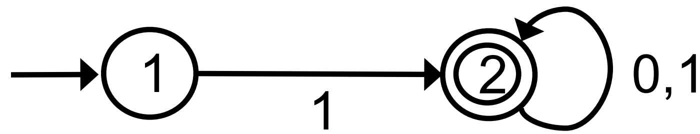
$Q = \{1,2\}$
$\Sigma = \{1\}$
$s0 = 1$
$F = \{2\}$
$\delta =$

Current state	Next state on
	1
1	2
2	1



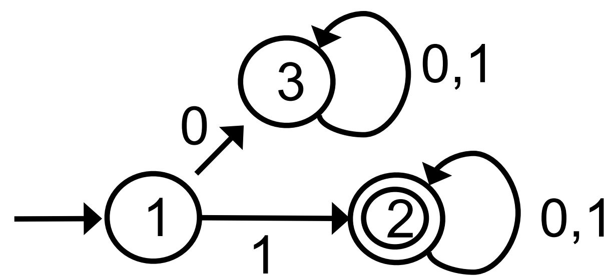
FSM for Odd Binary Numbers

$Q = \{1,2\}$
$\Sigma = \{1,0\}$
$s0 = 1$
$F = \{2\}$
$\delta =$



Current state	Next state on input	
	0	1
1		2
2	2	2

Undefined in table so assumed to be to a non-accepting state



Defining FSM Accept in Logic

FSM: model(Q,Σ,S_0,F,δ)

String: σ

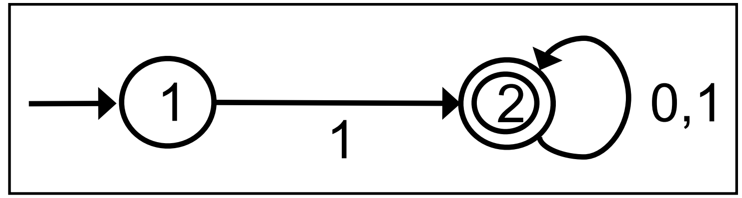
$$\text{accept}(\sigma) \leftarrow \\ \quad \text{model}(Q, \Sigma, S_0, F, \delta) \text{ and} \\ \quad \text{trace}(S_0, \sigma, F, \delta)$$

$$\text{trace}(S, [], F, \delta) \leftarrow \; S \in F$$

$$\text{trace}(S, [X|R], F, \delta) \leftarrow \\ \quad (S,X,S1) \in \delta \text{ and} \\ \quad \text{trace}(S1,R, F, \delta)$$

[] is the empty sequence
[X|R] separates first element, X, from rest of sequence, R.

Binary Odd



model({1,2}, {1,0}, 1, {2}, δ) $\delta = \{(1,1,2), (2,0,2), (2,1,2)\}$

Is the sequence [1,0,1] accepted?

trace(1, [1,0,1], {2}, δ)

$(1,1,2) \in \delta$ trace(2, [0,1], {2}, δ)

$(2,0,2) \in \delta$ trace(2, [1], {2}, δ)

$(2,1,2) \in \delta$ trace(2, [], {2}, δ)

$2 \in \{2\}$

trace(S, [], F, δ) ← $S \in F$

trace(S, [X|R], F, δ) ←
  (S,X,S1) $\in \delta$ and
  trace(S1,R, F, δ)