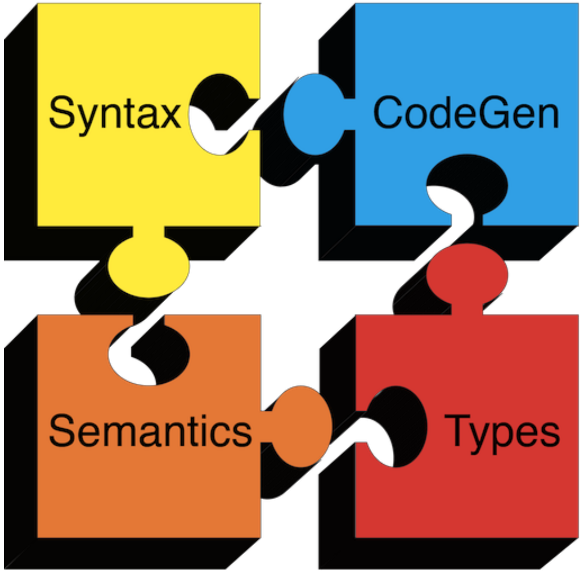


Compilers

Finite Automata

Alex Aiken

Finite Automata

- Regular expressions = specification
- Finite automata = implementation

- A finite automaton consists of
  - An input alphabet Σ
  - A finite set of states S
  - A start state n
  - A set of accepting states $F \subseteq S$
  - A set of transitions state $\rightarrow^{\text{input}}$ state

Finite Automata

- Transition

$$s_1 \xrightarrow{a} s_2$$

- Is read

In state s_1 on input a go to state s_2

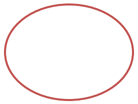
- If end of input and in accepting state => accept

- Otherwise => reject

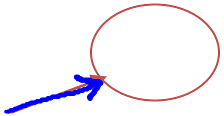
terminates in state $s \notin F$
gets stuck

Finite Automata

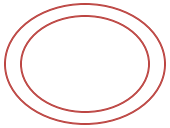
- A state



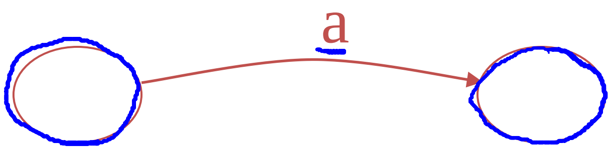
- The start state



- An accepting state

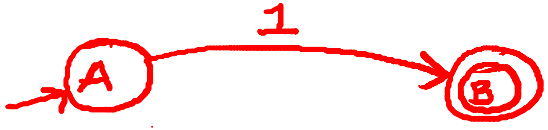


- A transition



Finite Automata

- A finite automaton that accepts only "1"



State	Input	
A	0	Reject

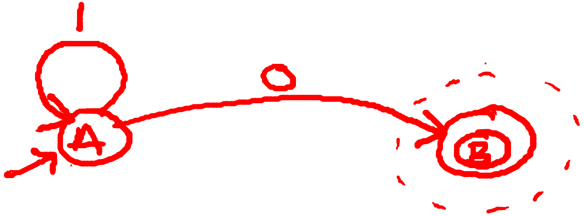
State	Input	
A	1 0	Reject
B	1 0	

	State	Input
	A	1
Accept	B	1

Language of a FA ≡ set of accepted strings

Finite Automata

- A finite automaton accepting any number of 1's followed by a single 0
- Alphabet: {0,1}



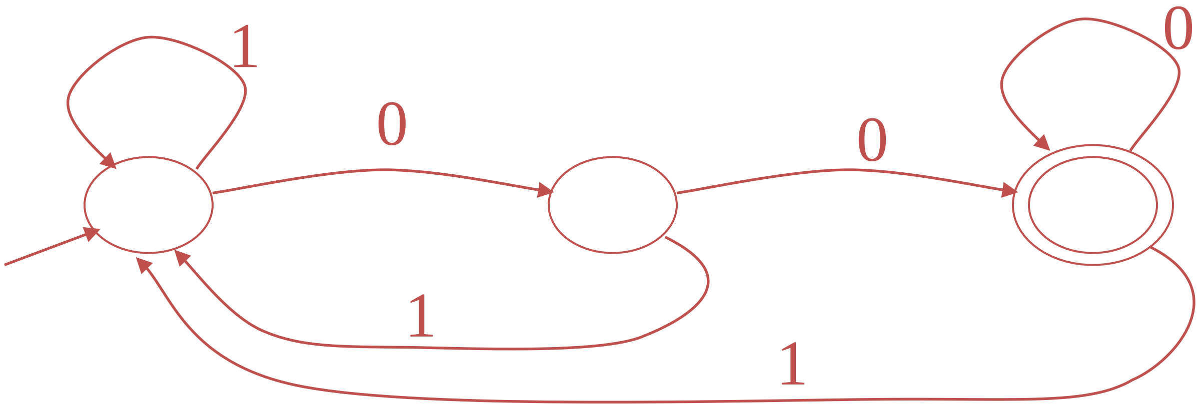
State	Input
A	↑100
A	1↑00
B	10↑0

Reject

State	Input
A	↑110
A	1↑10
A	11↑0
B	110↑

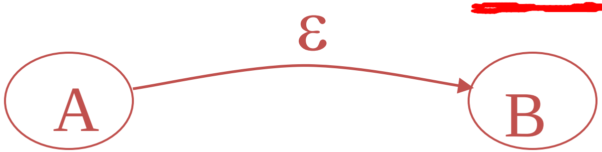
Accepts

Select the regular language that denotes the same language as this finite automaton



- $(0 + 1)^*$
- $(1^* + 0)(1 + 0)$
- $1^* + (01)^* + (001)^* + (000^*1)^*$
- $(0 + 1)^*00$

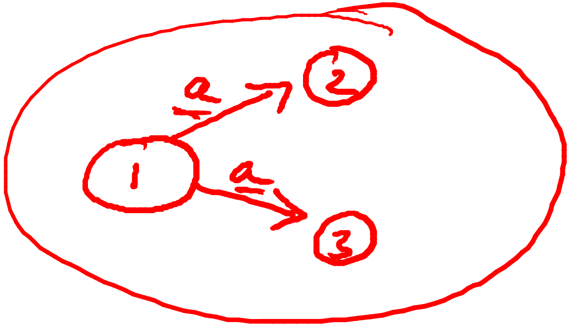
- Another kind of transition: ε-moves



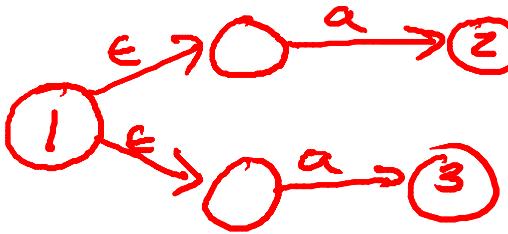
State	Input
A	$x_1 \; x_2 \; x_3$
B	$x_1 \; x_2 \; x_3$

Finite Automata

- Deterministic Finite Automata (DFA)
  – One transition per input per state
  – No ε-moves



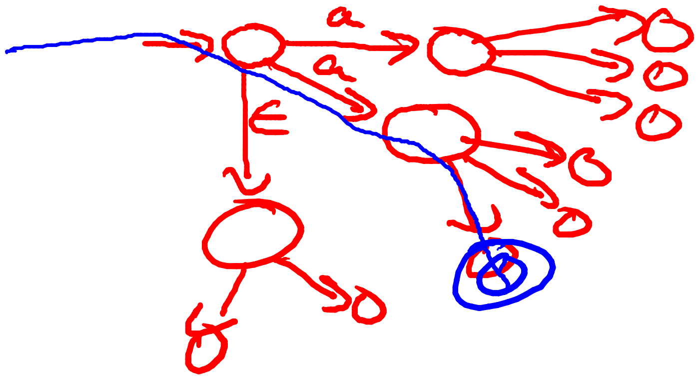
- Nondeterministic Finite Automata (NFA)
  – Can have multiple transitions for one input in a given state
  – Can have ε-moves



- A DFA takes only one path through the state graph per input

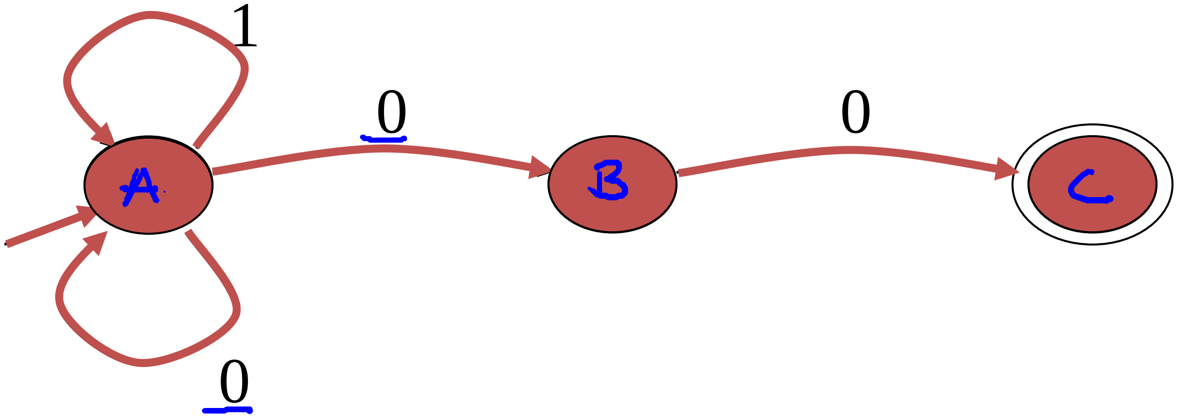
abc

- An NFA can choose



An NFA accepts if <u>some</u> choices lead to an accepting state.

- An NFA can get into multiple states



- Input: 1 0 0
- States: {A} {A,B} {A,B,C}

accepts

- NFAs and DFAs recognize the same set of languages
  - regular languages
- DFAs are faster to execute
  - There are no choices to consider
- NFAs are, in general, smaller

exponentially smaller