FCS Week 7 Lecture Note

Notebook:	Fundamentals of Computer Science		
Created:	2021-04-13 10:30 AM	**Updated:**	2021-04-28 5:53 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** Automata Theory Part 1	Course: BSc Computer Science
		Class: CM1025 Fundamentals of Computer Science[Lecture]
		Date: April 28, 2021

Essential Question:

What is an automata?

Questions/Cues:

- What is an Alphabet $\sum$?
- What is a string or word?
- What are empty strings?
- How do we denote the length of a string?
- How do we denote the set of all strings composed using letters in sigma?
- How do we denote the set of all non-empty strings composed using letters from sigma?
- What is $\sum^k$?
- What is a language?
- What is a finite automaton?
- What are states?
- What are transitions?
- What are accepting states?
- What is a 5-tuple?
- What is the language of a automaton?

Definitions

- An Alphabet, Σ, is a non-empty set of symbols
- For example: Σ = {0, 1} is a binary alphabet. Σ = {a, b, ..., z} is the collection of lowercase letters
- A string or word is a finite sequence of letters drawn from an alphabet
- For example: x = 01110101 is a binary string. "Life is good" is also a string
- Empty strings denoted by ε are strings with zero occurrences of letters. Empty strings can be from any alphabet.

Definitions

- Length of a string, x, is the sum of occurrences of its symbols, denoted by |x|
- For example: if x = 01110101 then |x| = 8, the length of "Life is good" is 12
- The set of all strings composed from letters in Σ is denoted by Σ^*
- For example: if Σ = { 0, 1} then Σ^* = {ε, 0, 1, 00, 01, 10, 11, ...}
- The set of all non-empty strings composed from letters in Σ is denoted by Σ^+.

Definitions

- The set of all strings of length k composed from letters in Σ is denoted by Σ^k
- For example: $\Sigma = \{0, 1\}$ then $\Sigma^2 = \{00, 01, 10, 11\}$
- $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$
- Size of Σ^k = $|\Sigma|^k$
- A language is a collection of strings over an alphabet
- For example: the language of palindromes over the binary alphabet is {ε, 0, 1, 00, 11, 000, 010, 101, 111, ...}.
    - Note if sigma k is finite, it's size is the size of the alphabet to the power of k.

Examples

- If $\Sigma = \{a, b, c\}$ then what is Σ^2?
- There must be $3^2 = 9$ strings in this set
- $\Sigma^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$
- What about Σ^1?
- Σ^1has three strings. $\Sigma^1 = \{a, b, c\}$
- Is Σ^1 the same as Σ?
- The answer is no! Elements in Σ are called symbols and they are letters, whereas the elements in $\Sigma^1$are strings; they just happen to have length 1
- Do not mix types in programming.

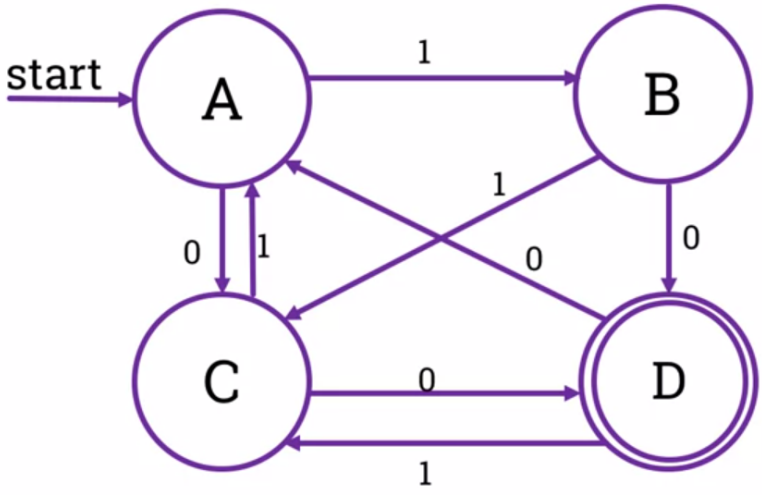
What is a finite automaton?

- A finite automaton is a simple mathematical machine; it is a representation of how computations are performed with limited memory space

- It is a model of computation, which consists of a set of states that are connected by transitions

- It has input. We will see later on
- It has output: Reject or Accept.

Example



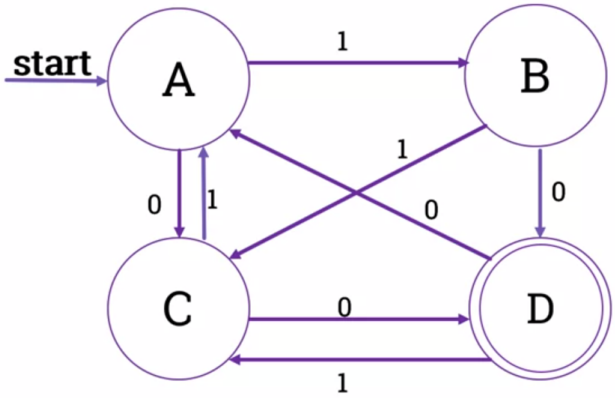
- The circles are the states of the automaton. The one with an arrow coming from nowhere or labeled start is called the initial state or the beginning of the computation.
- The labels or "numbers" on the arrows are called transitions and the labels are the alphabet dictating what to do next
- The arrow labeled "1" or transition "1" from A to B tells us that if we are at state A and we read "1" or one, we go to state B. On the other hand, if we are at A and we read zero or "0", we go the state C
- Double circled states are called accepting states. If the computation ends at any of the accepting states, the machine outputs accept, otherwise it outputs reject.

- Five-tuple = is a sequence of five elements

An automaton M is 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set called the states
- Σ is a finite set called the alphabet
- δ: Q x $\Sigma \rightarrow$ Q is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

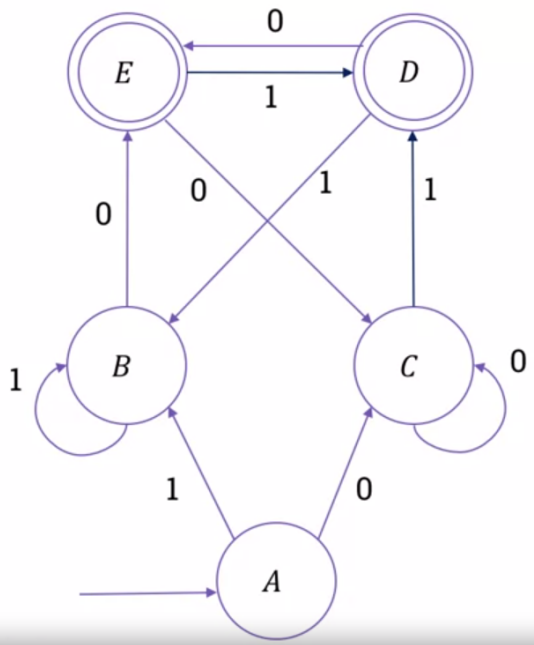
Example



- $Q = \{A, B, C, D\}$
- $\Sigma = \{0, 1\}$
- $q_0 = A$
- $F = \{D\}$
- $\delta = ?$

δ	0	1
A	C	B
B	D	C
C	D	A
*D	A	C

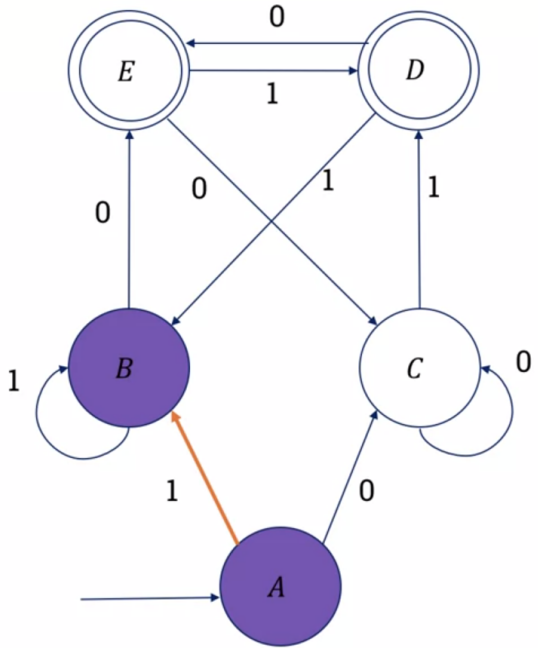
Example



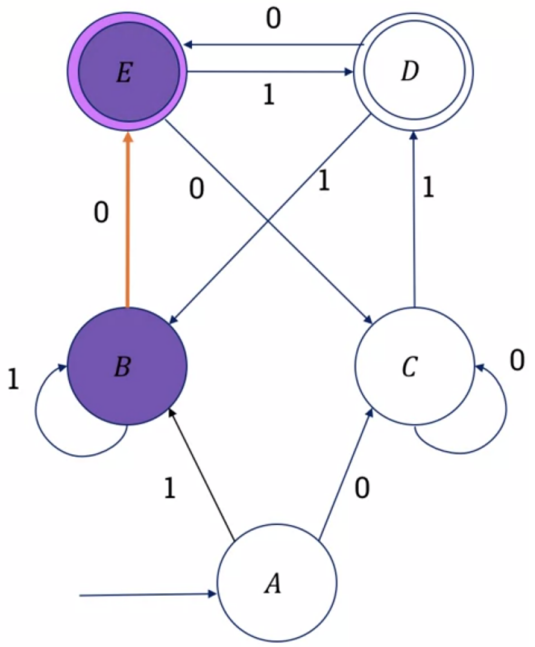
- $Q = \{A, B, C, D, E\}$
- $\Sigma = \{0, 1\}$
- $q_0 = A$
- $F = \{D, E\}$
- $\delta =$

	0	1
A	C	B
B	E	B
C	C	D
$*D$	E	B
$*E$	C	D

- 10011



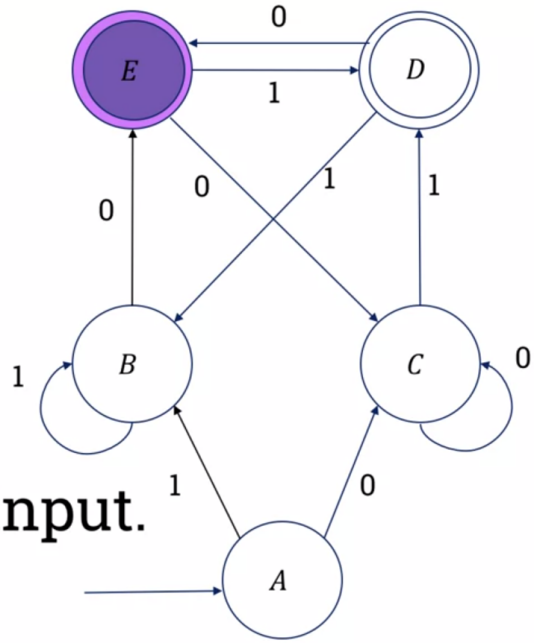
•10011



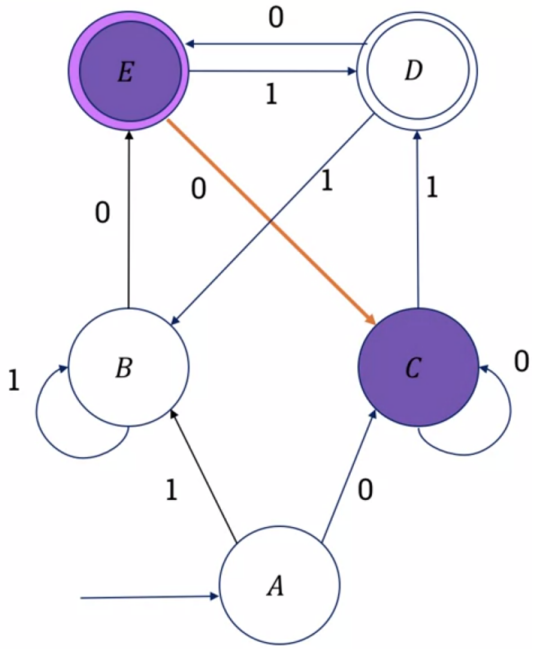
•10011

•Do we accept at E?

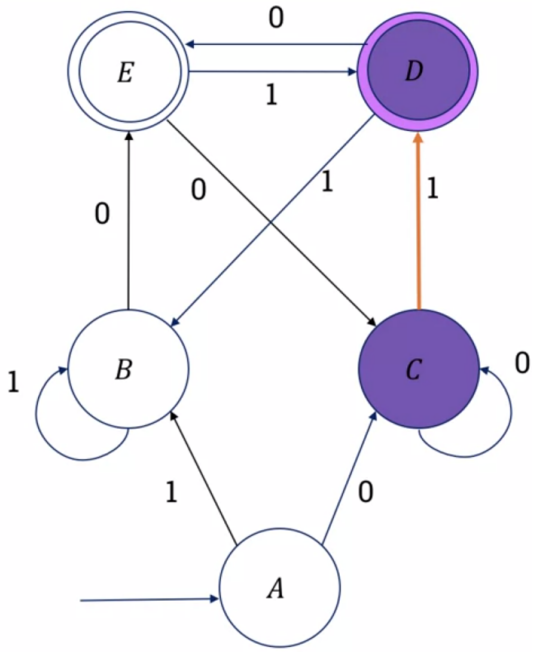
•No, not the end of the input.



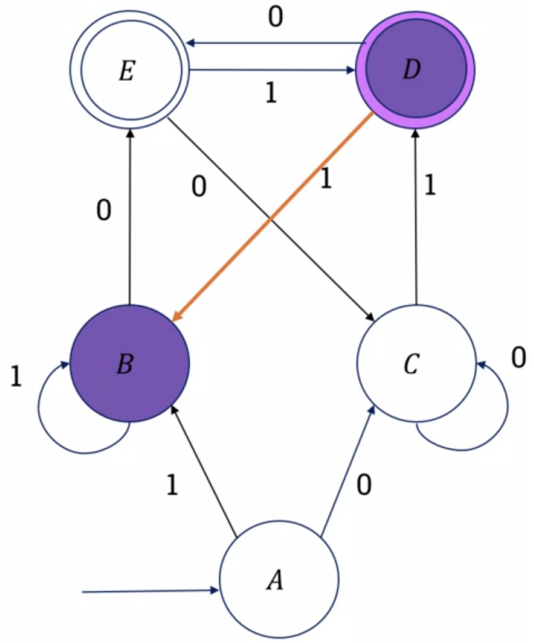
•10011



•10011



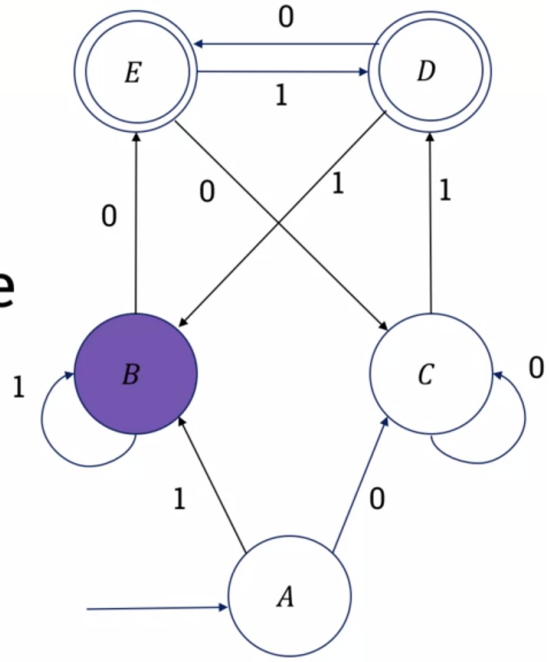
•10011



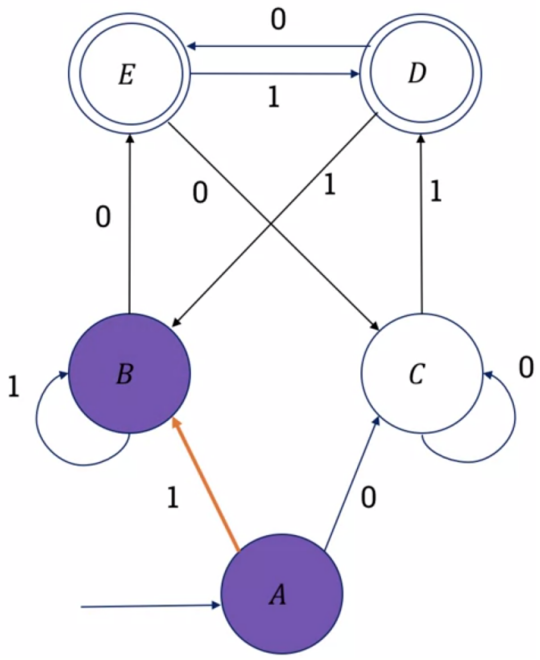
•10011

•B is not an accept state

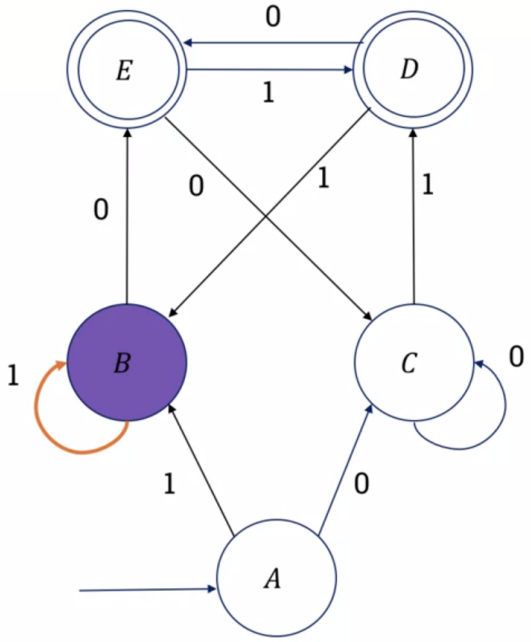
•Reject 10011



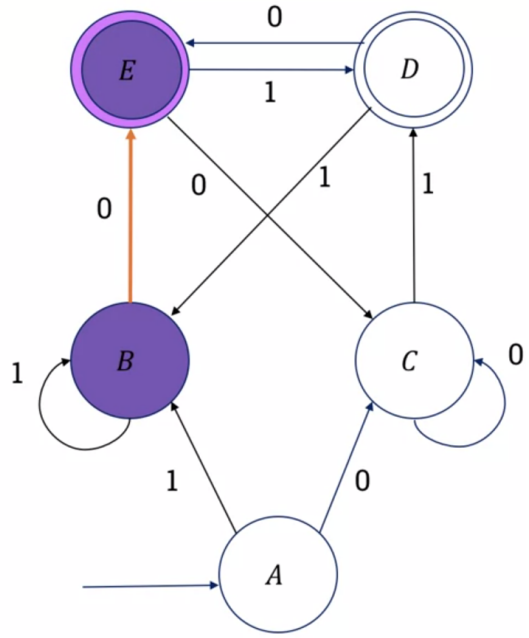
•11001



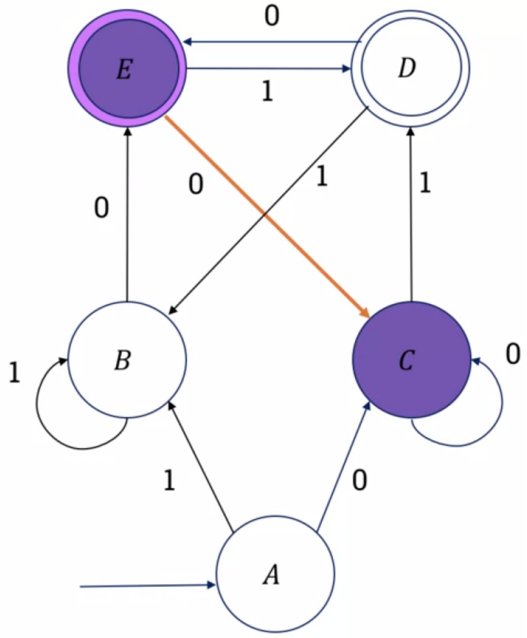
•11001



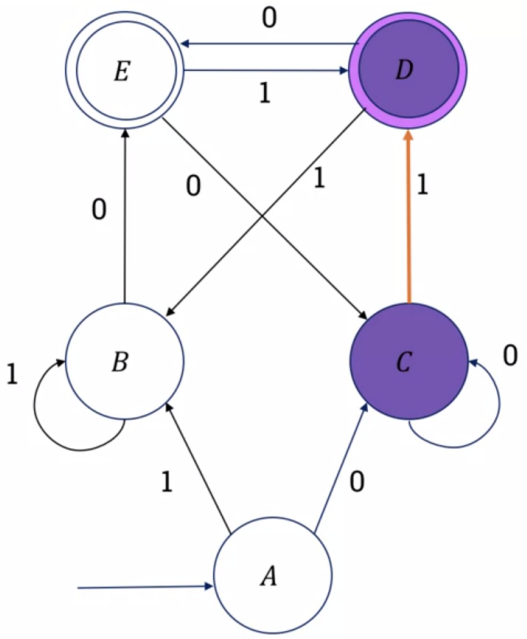
•11001



•11001



•11001

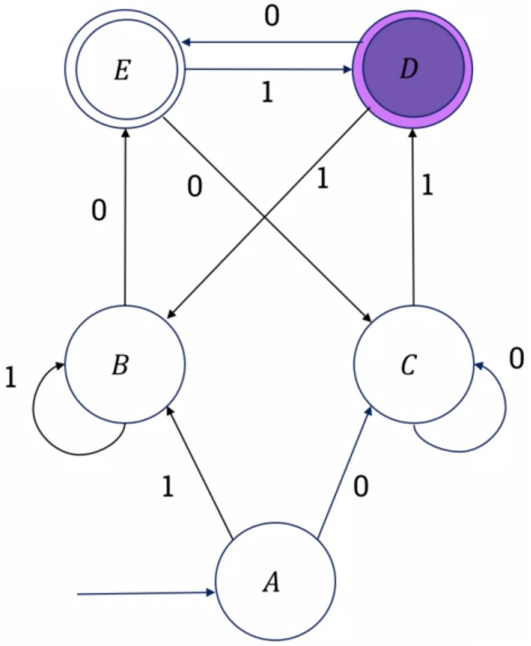


•11001

•Do we accept?

•Yes!

• End of the input at an accepting state.

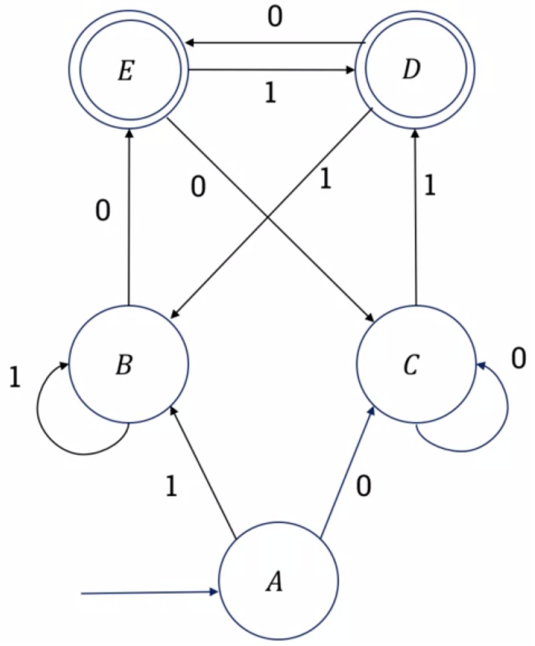


- We do ***not*** accept when passing through an accepting state.
- Only accept if the input ***ends*** in an accepting state.

What else is accepted by this automaton?

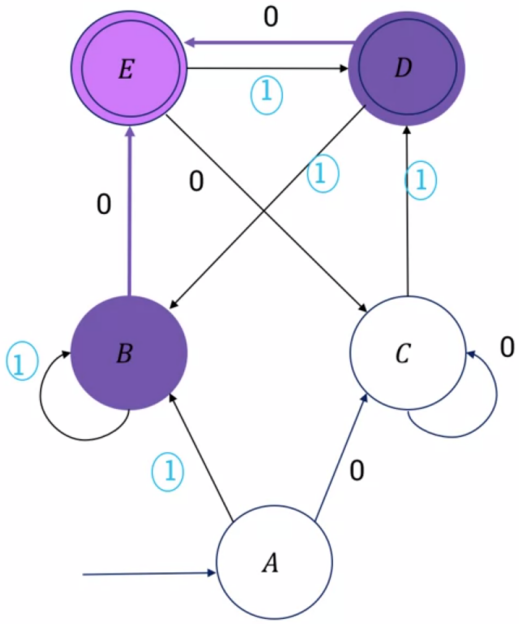
- 10
- 110
- 1110
- 1010
- 10110

- Do you see a pattern?



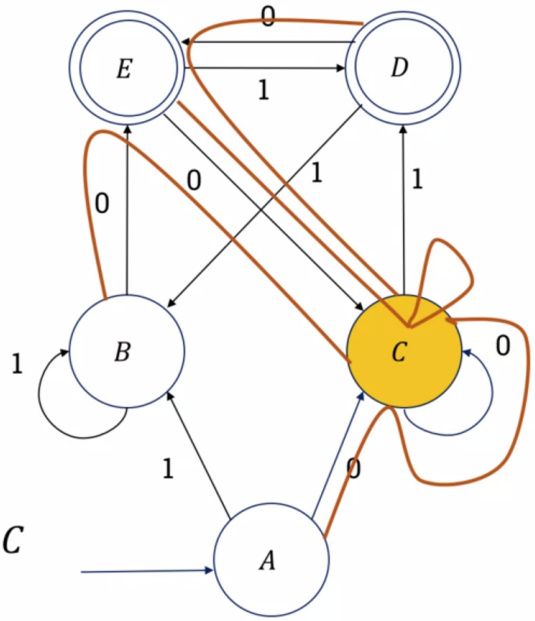
Backward computation

- Choose an accept state, E
- Incoming transitions to E
- Input ends with 0
- Penultimate states: B, D
- Incoming transitions to B, D
- Input ends with 10



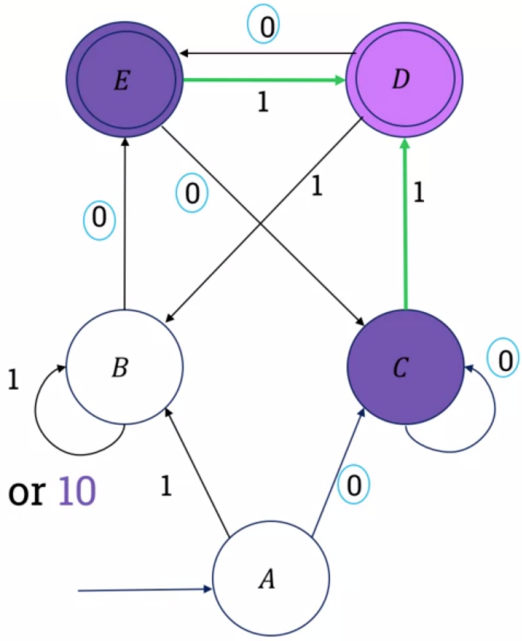
Backward computation

- Choose an accept state, E
- Incoming transitions to E
- Input ends with 0
- Penultimate states: B, D
- Incoming transitions to B, D
- Input ends with 10
- Check states with incoming 0, C
- Paths to C: ACC, ECC, DEC, BEC
- No input ending with 10 ends at C



Backward computation

- Other accepting state: D
- This automaton is symmetrical!
- Input ends with 1
- Penultimate states: C, E
- Incoming transitions to C, E
- Input ends with 01



- Any string ending with either 01 or 10

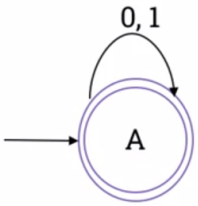
- The set of all strings accepted by an automaton is called the *language* of that automaton

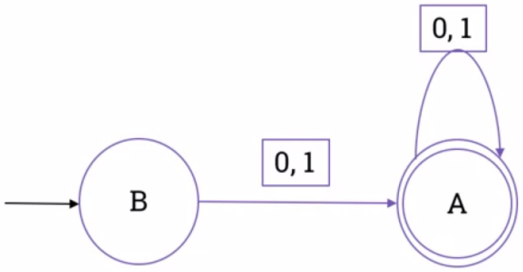
- If M is an automaton on alphabet Σ, then $\mathcal{L}(M)$ is the language of M:

$$\mathcal{L}(M) = \{x \in \Sigma^* | M \text{ accept } x\}$$

Automaton accepting every binary

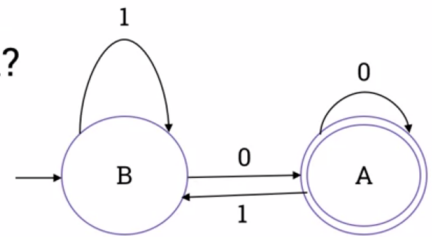
- There is a final state, A
- The incoming arrows have no restriction
- The outgoing arrows from A end at A
- Initial state, B
- 0 and 1 must be accepted





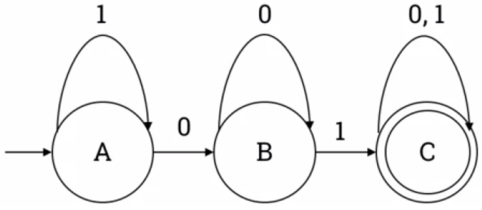
Automaton accepting strings ending in 0

- There is a final state, A
- The incoming arrow is labelled 0
- The outgoing arrow labelled 0 ends at A
- String 0 is accepted, initial state B
- What happens to arrows labelled 1?
- They cannot end at A
- End at B



Automaton accepting strings with 01

- String 01 is accepted, so we have at least 3 states
- If we read 01, anything after that is accepted
- Loop over state C with 0, 1
- If we read 0 at state B, we stay put
- What if we read 1 at state A?
- We stay put.



Summary

In this week, we learned about finite state automatons and automata theory.