cs2001
Introduction Theory of Computation

ecture 9

Ruth Urner

Febuary 6, 2023

Regular Languages

Reading: ITC Section 1.1

Definition

A language L over some alphabet Σ is called a regular language if there exists a finite automaton M such that $L = L(M)$, that is, if there exists a finite automaton that recognizes it.

Regular language

Σ

Definition

A language L is called a regular language if there exists a finite automaton M such that $L = L(M)$, that is, if there exists a finite automaton that recognizes it.

Examples of regular languages

- $\{ w \in \{0,1\}^* \mid w$ contains 001 as a substring $\}$
- $\{ w \in \{0,1\}^* \mid$ the number of 1 is odd $\}$

Talking: Anmoldeep Gadri

Regular operations on languages

Definition:
Let A and B be languages. Then we define the following operations that each form a new language:

- **Union:** $A \cup B \;=\; \{\mathbf{w} \mid \mathbf{w} \in A \text{ or } \mathbf{w} \in B\}$.

- **Concatenation:** $A \circ B \;=\; \{\mathbf{w}\mathbf{v} \mid \mathbf{w} \in A \text{ and } \mathbf{v} \in B\}$.

- **Star:**
  $A^* \;=\; \{\mathbf{w}_1\mathbf{w}_2 \ldots \mathbf{w}_k \mid k \geq 0 \text{ and } \mathbf{w}_i \in A \;\forall\, i \in \{0, 1, \ldots, k\}\}$

Regular o…

In particular for an alphabet Σ, the set Σ^* is the set of all finite words over Σ.

Ex: $A = \{01, \ldots 01\}$
$B = \{1, 1\ldots$

$A \times B \ni (01, 110)$ not $\in \{0,1\}^*$

Definition:
Let A and B … Then we define the following operations that each form a new language:

- **Union:** $A \cup B = \{w \mid w \in A \text{ or } w \in B\}$.

Ex $A \cup B = \{01, 001, 0001, 1, 11, 110, 10\}$

- **Concatenation:** $A \circ B = \{wv \mid w \in A \text{ and } v \in B\}$

Ex $A \circ B = \{011, 0011, 00011, 0111, 00111, 000111, 01110, 001110, 0001110, 010, 0010, 00010\}$

- **Star:**
$A^* = \{w_1 w_2 \ldots w_k \mid k \geq 0 \text{ and } w_i \in A \; \forall i \in \{0, 1, \ldots, k\}\}$

Ex $A^* = \{\varepsilon, 01, 001, 0001, 0101, 00101, 010001, 00101, 0010001, 0001001, 010101, 0100101, \ldots\}$

infinite set of words…

Talking: Ruth Urner

Regular operations

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

Ex: $A = \{01, \ldots 1\}$

$B = \{1, \ldots\}$

$A \times B \ni (01, 110)$

not $\in \{0,1\}^*$

Definition:
Let A and B ... Then we define the following operations that each form a new language:

- **Union:** $A \cup B = \{w \mid w \in A \text{ or } w \in B\}$.

Ex $A \cup B = \{01, 001, 0001, 1, 11, 110, 10\}$

- **Concatenation:** $A \circ B = \{wv \mid w \in A \text{ and } v \in B\}$

Ex $A \circ B = \{011, 0011, 00011, 0111, 00111, 000111, 0110, 00110, 000110, 010, 0010, 00010\}$

- **Star:**
$A^* = \{w_1 w_2 \ldots w_k \mid k \geq 0 \text{ and } w_i \in A \; \forall i \in \{0, 1, \ldots, k\}\}$

Ex $A^* = \{\varepsilon, 01, 001, 0001, 0101, 00101, 01001, 001001, 0010001, 0001001, 010101, 0100101, \ldots\}$

infinite set of words...

In particular for an alphabet Σ, the set Σ^* is ~~the~~ the set of all finite words over Σ.

$B \notin A^*$

$A \notin B^*$

$C = \{0, 1\}$

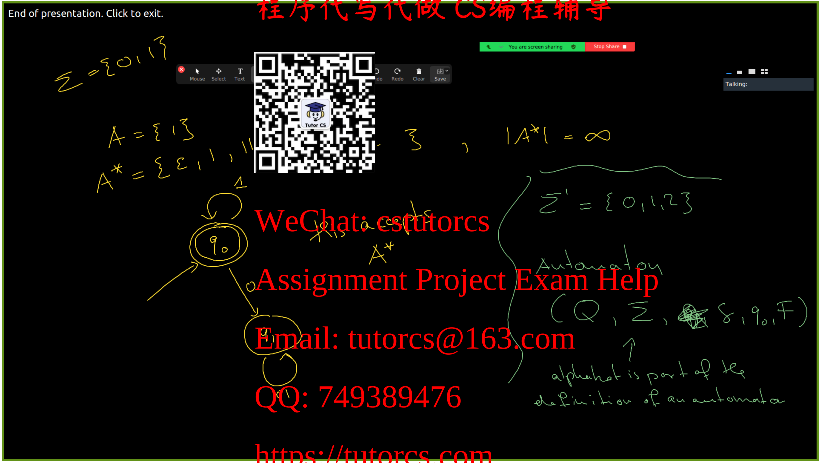
$B \subsetneq C^*$

$A \subsetneq C^*$

$D = \{0, 1, 00, 11\}$

$B \subsetneq D^*$

$A \not\subseteq D^*$


End of presentation. Click to exit.
You are screen sharing
Stop Share
Mouse
Select
Text
Redo
Clear
Save
Talking:
$\Sigma = \{0, 1\}$
$A = \{1\}$
$A^* = \{\varepsilon, 1, 11, \ldots\}$, $|A^*| = \infty$
1
0
q_0
q_1
this accepts A^*
$\Sigma' = \{0, 1, 2\}$
Automaton
$(Q, \Sigma, \delta, q_0, F)$
alphabet is part of the definition of an automaton

Regular operations reproduce regular languages

Theorem
The set of all regular languages is closed under the three regular operations.

That is: if A and B are regular languages, then A^*, $A \cup B$ and $A \circ B$ are also regular languages.

Problem 1.4 from ITC

Here we design an automaton that recognizes the intersection of two languages for which we have simple automata recognizing them.

We will now use a similar "product" for the general construction of an automaton that recognizes the union of two regular languages.

Problem 1.4 from ITC

$\Sigma = \{a, b\}$

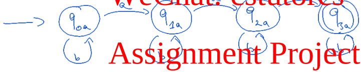
Here we design a … recognizes the intersection of two languages for which we have si… nizing them.

Given two … A and B.

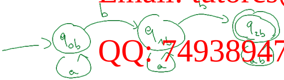
$A = \{ w$ … contains at least 3 a's $\}$

$B = \{ w \in \Sigma^* \mid w$ contains at least 2 b's $\}$

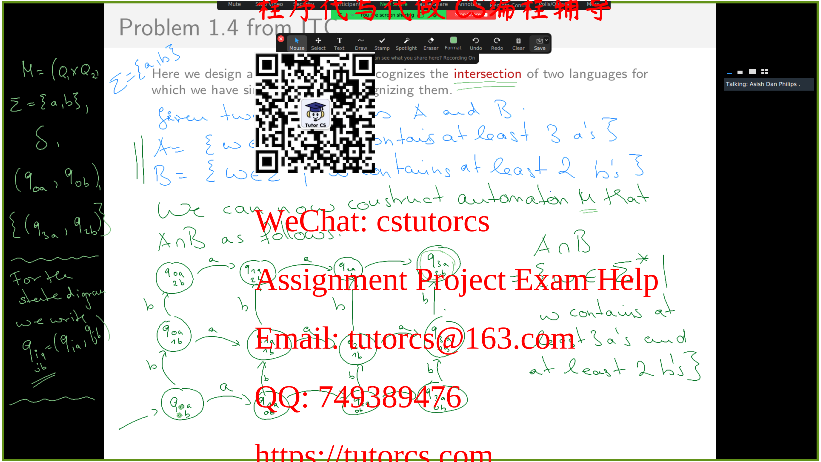
We define automaton M_1 with $L(M_1) = A$



and automaton M_2 with $L(M_2) = B$



Talking: Ruth Urner


Problem 1.4 from ITC
Here we design a ... cognizes the intersection of two languages for which we have si... gnizing them.
Σ={a,b}
Given two ... A and B.
A = { w ... contains at least 3 a's }
B = { w ∈ Σ* | w contains at least 2 b's }
We can now construct automaton M that
A∩B as follows:
A∩B
{ w | w contains at least 3 a's and at least 2 b's }
M = (Q1×Q2,
Σ = {a,b},
δ,
(q0a, q0b),
{(q3a, q2b)})
For the state diagram we write q_ij = (q_ia, q_jb)
Talking: Asish Dan Philips .

Modification of Pro
~~Problem~~ 1.4 from ITC

$\Sigma = \{a, b\}$

Here we design a ... cognizes the ~~intersection~~ union of two languages for which we have si... gnizing them.

given two ... A and B.

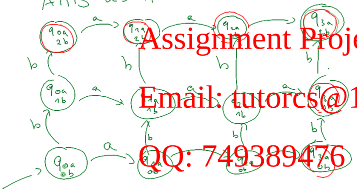
$A = \{ w \ldots$ contains at least 3 a's $\}$

$B = \{ w \in \Sigma^* \mid w$ contains at least 2 b's $\}$

We can now construct automaton M that accepts $A \cap B$ as follows:

$M = (Q_1 \times Q_2,$
$\Sigma = \{a, b\},$
$\delta,$
$(q_{0a}, q_{0b}),$
$\{ \ldots \})$

For the state diagram we write $q_{ij} = (q_{ia}, q_{jb})$



$A \cup B = \{ w \in \Sigma^* \mid w$ contains at least 3 a's or at least 2 b's $\}$

← 6 accept states!

We start by proving that if A and B are regular languages over some alphabet Σ, then so is $A \cup B$.

Proof:

To prove that $A \cup B$ is regular, we need to show that there exists an automaton

$$M = (Q, \Sigma, \delta, q_0, F)$$

with

$$L(M) = A \cup B$$

Since A and B are regular, we know there exist automata $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ with

$$L(M_1) = A \text{ and } L(M_2) = B$$

Regular languages closed under unions

general Σ

We start by p[...] and B are regular languages over some alphabe[...] $A \cup B$.

Given M_1 and M_2 the goal is to construct an automaton that recognizes $A \cup B$ using the components of M_1 and M_2

Goal for[...]

To show that $A \cup B$ is regular, we need show that there exists an automaton that recognizes $A \cup B$.

We know that A and B are regular languages, [...] there exist automata $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ such that $A = L(M_1)$ and $B = L(M_2)$.

Talking: Ruth Urner

Proof continued:
We now define the re[illegible]tomaton M as follows:

1. $Q = Q_1 \times Q_2$
2. We use the same alphabet Σ
3. $\delta((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$
4. $q_0 = (q_1, q_2)$
5. $F = \{(s_1, s_2) \in Q \mid s_1 \in F_1 \text{ or } s_2 \in F_2\}$


Regular languages clo
Proof contin
We now defin
automaton M as follows:
1. Q = Q1 × Q2
2. We use the same alphabet Σ
3. δ((s1, s2), a) = (δ1(s1, a), δ2(s2, a))
4. q0 = (q1, q2)
5. F = {(s1, s2) ∈ Q | s1 ∈ F1 or s2 ∈ F2}
M1 = (Q1, Σ
δ1, q1, F1)
M2 = (Q2, Σ
δ2, q2, F2)
L(M1) = A
L(M2) = B
Talking: Ruth Urner

Regular languages closed under unions

Proof contin

We now defin automaton M as follows:

1. $Q = Q_1 \times Q_2$
2. We use the same alphabet Σ
3. $\delta((\underbrace{s_1}_{\in Q_1}, \underbrace{s_2}_{\in Q_2}), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$
4. $q_0 = (q_1, q_2)$
5. $F = \{(s_1, s_2) \in Q \mid s_1 \in F_1 \text{ or } s_2 \in F_2\}$

Talking: Ruth Urner

Non-deterministic finite automata

Reading: ITC Section 1.2

To show that regular [illegible]s are also closed under the concatenation and st[illegible]ons, it is convenient to employ the concept of a Non-det[illegible] Finite Automaton (NFA for short).

In the automata we have seen so far, every state has exactly one outgoing edge for every symbol. That is, for every state, the “reaction” to any input symbol is uniquely determined. Such an automaton is also called a Deterministic Finite Automaton (DFA for short).

We will generalize this to also allow non-unique (think randomized or parallelized) computations. NFAs are a model for such computations.

Example: the NFA N_1

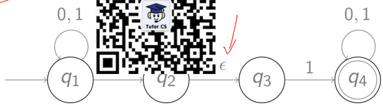
For the machine N_1 we get

$$L(N_1) = \{\mathbf{w} \mid \mathbf{w} \text{ contains 11 or 101 as a substring}\}$$

Example: the NFA N_1

Differences

ϵ can be "read"



some states have several instructions for the same letter

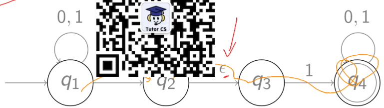
some states [illegible] for all letters


Example: the NFA N1
Differences
ε can be "read"
0, 1
0, 1
q1
q2
q3
q4
ε
1
Examples of words
10110 ✓
accepted
1000 X
not accepted
some states have several instructions for the same letter
some states have no instructions for all letters
A word is accepted by an NFA if there is at least one way of reading the word that leads to an accept state.
Talking: Asish Dan Philips .

Example: the NFA N_1

Differences

ε can be "read"



Examples of words

10110 ✓ accepted

1000 X not accepted

some states have several instructions for the same letter

some states [no instructions] for all letters

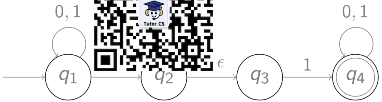
A word is accepted by an NFA if there is at least one way of reading the word that leads to an accept state.

111 = 1ε11

ε

Talking: Asish Dan Philips .

Example: the NFA N

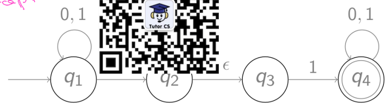


Consider word

the word is accepted since there exist branches leading to accept states

Example: the NFA N_1

10001 not accepted



Talking: Ruth Urner

Reminder: For alphabet Σ, the set of all words over Σ is denoted by Σ^*.

For the machine N_1 we get

$L(N_1) = \{ w \in \{0,1\}^* \mid$ w contains 101 or 11 as a substring $\}$

Formal definition of a non-deterministic finite automaton

For an alphabet Σ, we define $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$.

Definition

A non-deterministic finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set called the set of states,
2. Σ is a finite set called the alphabet,
3. $\delta : Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

Formal description of the NFA N_1

N_1

Formal description of $N_1 = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_1, q_2, q_3, q_4\}$, $q_0 = q_1$, $F = \{q_4\}$.

$\Sigma_\varepsilon = \{0, 1, \varepsilon\}$

δ	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

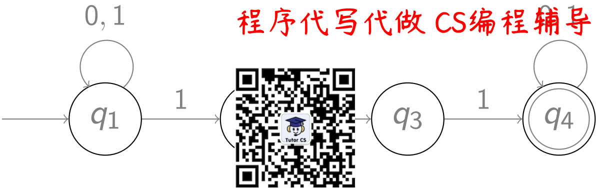
$|\varepsilon|$

$\varepsilon|\varepsilon|\varepsilon\varepsilon$

$|\varepsilon\varepsilon\varepsilon|\varepsilon$

these are the same word !!

Formal description of the NFA N_1



For the machine N_1 we get

$$L(N_1) = \{\mathbf{w} \mid \mathbf{w} \text{ contains 11 or 101 as a substring}\}$$

Formal definition of acceptance for NFAs

For a string $\mathbf{w} = w_1 w_2 \ldots w_n$ where $w_i \in \Sigma$ for some alphabet Σ, we let strings $\mathbf{z}_0 w_1 \mathbf{z}_1 w_2 \mathbf{z}_3 \ldots \mathbf{z}_{n-1} w_n \mathbf{z}_n$ [illegible]ere each $\mathbf{z}_i$ is a sequence of 0 or more symbols ϵ represent the [illegible]$\mathbf{w}$ as a string over Σ.

Definition

Let $N = (Q, \Sigma, \delta, q_0, F)$ a non-deterministic finite automaton (NFA) and let $\mathbf{w}$ be a string over Σ. We say that N accepts $\mathbf{w}$ if we can write $\mathbf{w} = y_1 y_2 \ldots y_m$ with $y_i \in \Sigma_\epsilon$ and there exists a sequence $s_0 s_1 s_2 \ldots s_m$ of states such that

1. $s_0 = q_0$,
2. $s_{i+1} \in \delta(s_i, y_{i+1})$ for $i = 0, 1, \ldots, m$, and
3. $s_m \in F$.

We then define the language $L(N)$ recognized by NFA N in the same way as for DFAs, namely as the set of all words that are accepted by N.

$|| = \omega$

$|\varepsilon|\varepsilon$

$|\varepsilon\varepsilon\varepsilon\varepsilon|$

$\varepsilon|\varepsilon|$

all the same word w!!

Formal definition of acceptance for NFAs

For a string $\mathbf{w}$ = … e $w_i \in \Sigma$ for some alphabet Σ, we let strings $\mathbf{z}_0 w_1 \mathbf{z}_1 w_2 \mathbf{z}_3 \ldots \mathbf{z}$ … where each $\mathbf{z}_i$ is a sequence of 0 or more symbols ϵ repres… $\mathbf{w}$ as a string over Σ.

Definition

Let $N = (Q, \Sigma, \delta, q_0, F)$ a non-deterministic finite automaton (NFA) and let $\mathbf{w}$ be a string over Σ. We say that N accepts $\mathbf{w}$ if we can write $\mathbf{w} = y_1 y_2 \ldots y_m$ with $y_i \in \Sigma_\epsilon$ and there exists a sequence $s_0 s_1 s_2 \ldots s_m$ of states such that

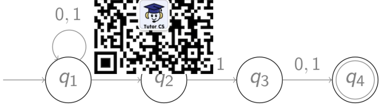
1. $s_0 = q_0$,
2. $s_{i+1} \in \delta(s_i, y_{i+1})$ for $i = 0, 1, \ldots, m$, and
3. $s_m \in F$.

Example: the NFA N_2

For the machine N_2 we get

$$L(N_2) = \{\mathbf{w} \mid \mathbf{w} \text{ contains 1 at the third position from the end}\}$$

Example: the NFA N_2



0101 ✓ 0001 ✗

1111 ✓ 00001 ✗

For the machine N_2 we get

$L(N_2) = \{ w \mid w$ has a 1 at the third position from the end $\}$

$= \{ w = w_1 w_2 \dots w_n \in \Sigma^* \mid w_{n-2} = 1 \}$

Talking: Ruth Urner