eecs2001

Introduction to the Theory of Computation

Lecture 12

Ruth Urner

February 15, 2023

eecs2001
Introduction to the Theory of Computation
Lecture 12

Ruth Urner

Reminder: Midterm on March 1!

February 15, 2023

eecs2001

Introduction to the Theory of Computation

Lecture 12

Ruth Urner

Q: show regular languages are closed under OPD(·,·,·)

You can use:

February 15, 2023

Regular languages are closed under

- unions
- concatenation
- star-operation
- intersection
- complements

Regular Expressions

Reading: ITC Section 1.3

A regular expression is a compact way of defining a set of words. It is a sequence of symbols that represent a language over some alphabet Σ.

Inductive definition of sets – general pattern

An inductive definition of a set consists of

1. **A universe set** U
2. **A core set** $C \subseteq U$
3. **A finite set** $O = \{o_1, o_2, \ldots o_n\}$ **of operations from** $o_i : U^{r_i} \to U$ **for some arities** $r_i \in \mathbb{N}$

We define $\mathcal{I}(U, C, O)$ as the set of elements that we obtain by starting with the core set and putting all those elements of U into $\mathcal{I}(U, C, O)$ that one can reach by successively applying the operations in O.

Let Σ be some alphabet. We define the set $\mathcal{R}$ of regular expressions over Σ inductively by setting $\mathcal{R}_\Sigma = \mathcal{I}(U, C, \mathcal{O})$, where

1. The universe U is the set of all strings over $\Sigma \cup \{(,)\} \cup \circ, {}^*, \epsilon, \emptyset\}$

2. The core set C is the set of al symbols in Σ and $\epsilon, \emptyset$ and two additional symbols: $C = \Sigma \cup \{\epsilon, \emptyset\}$.

3. Three operations:
   - $o_\cup(R_1, R_2) = (R_1 \cup R_2)$,
   - $o_\circ(R_1, R_2) = (R_1 \circ R_2)$,
   - $o_*(R) = (R^*)$.

Regular expression–inductive definition

Let Σ be some alphabet. We define the set $\mathcal{R}$ of regular expressions over Σ inductively by setting $\mathcal{R}_\Sigma = \mathcal{I}(U, C, O)$, where

1. The universe U is the set of all strings over $\Sigma \cup \{(,), \cup, \circ, ^*, \epsilon, \emptyset\}$.
2. The core set C is the set of al symbols in Σ and $\epsilon, \emptyset$ and two additional symbols: $C = \Sigma \cup \{\epsilon, \emptyset\}$.
3. Three operations:
   - $o_\cup(R_1, R_2) = (R_1 \cup R_2)$,
   - $o_\circ(R_1, R_2) = (R_1 \circ R_2)$,
   - $o_*(R) = (R^*)$.

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

0

$(1 \cup 0)$ ✓

$(1\ 0$ ✗

ε ✓

$(\varepsilon \circ 1)$ ✓

$\in \mathcal{R}_\Sigma$

member of universe

$)(\circ\emptyset$

these two are not regular expressions

Exercise: Prove (by induction) the number of "(" is always equal to the number of ")" in a regular expression

The language of a regular expression

Each regular expression $R \in \mathcal{R}_\Sigma$ over some alphabet Σ represents a language over Σ. We define the interpretation $L(R)$ of a regular expression R according to the inductive definition:

Members of the core-set:

- The expression a for $a \in \Sigma$ represents the language $\{a\}$, that is $L(a) = \{a\}$.
- The expression ϵ represents the language $\{\epsilon\}$, that is $L(\epsilon) = \{\epsilon\}$.
- The expression $\emptyset$ represents the language $\emptyset$,that is $L(\emptyset) = \emptyset$.

Result of operation: For regular expressions R_1, R_2 and R, we define:

- $L((R_1 \cup R_2)) \;=\; L(R_1) \cup L(R_2)$
- $L((R_1 \circ R_2)) \;=\; L(R_1) \circ L(R_2)$
- $L((R^*)) \;=\; L(R)^*$

We call $L(R)$ the language of R.

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

$0 \in \mathcal{R}_\Sigma$

$L(0) = \{0\}$

$0, \varepsilon \in \mathcal{R}_\Sigma$

$(0 \cup \varepsilon) \in \mathcal{R}_\Sigma$

$L((0 \cup \varepsilon)) = \{0, \varepsilon\}$

$(0 \cup \emptyset) \in \mathcal{R}_\Sigma$

$L((0 \cup \emptyset)) = \{0\}$

The language of a regular expression

Each regular expression $R \in \mathcal{R}_\Sigma$ over some alphabet Σ represents a language over Σ. We define the interpretation $L(R)$ of a regular expression R according to the inductive definition:

Members of the core-set:

- The expression a for $a \in \Sigma$ represents the language $\{a\}$, that is $L(a) = \{a\}$.
- The expression ϵ represents the language $\{\epsilon\}$, that is $L(\epsilon) = \{\epsilon\}$.
- The expression $\emptyset$ represents the language $\emptyset$, that is $L(\emptyset) = \emptyset$.

Result of operation: For regular expressions R_1, R_2 and R, we define:

- $L(\boxed{(R_1 \cup R_2)}) = L(R_1) \cup L(R_2)$

Regular expression–additional notation

For convenience and readability, we use the following notational conventions:

1. For an alphabet Σ, we use Σ as a regular expression representing all words of length 1 over Σ.And then Σ^* is a regular expression for the set of all words over Σ.

2. We often omit brackets. The order of precedence then is: $*$, $\circ$, $\cup$.
3. The $\circ$-symbol is typically omitted: we use R_1R_2 as shorthand for $R_1 \circ R_2$.
4. We let R^+ be shorthand for RR^*.

5. We let R^k be the k times repeated concatenation of R with itself: $R^k = R \circ R \circ R \circ \ldots \circ R$.

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$
- $(\Sigma\Sigma)^*$
- $(\Sigma\Sigma\Sigma)^*$

Regular expressions–examples

precedence: $*$ $\circ$ $\cup$

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $0^*10^* = ((0^*)\circ 1 \circ (0^*))$

$= \{w \in \Sigma^* \mid w \text{ contains exactly one } 1\}$

$1 \in L(0^*10^*)$

$0010 \in L(0^*1\,0^*)$

$10000 \in L(0^*10^*)$

$L((0^*))$ = {set of all words that have only letter 0}

ε is included

$L((0^*)) = \{\varepsilon, 0, 00, 000, \dots\}$

$L(1) = \{1\}$

Regular expressions–examples

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- 0^*10^*
- $\Sigma^*1\Sigma^* = (\Sigma^*)1(\Sigma^*) = (\Sigma^*) \circ 1 \circ (\Sigma^*)$

$L(\Sigma^*1\Sigma^*) = \{w \in \Sigma^* \mid w$ contains at least one $1\}$

$1 \in L(\Sigma^* 1 \Sigma^*)$

$\underbrace{010110}_{\in\Sigma^*}1\underbrace{010}_{\in\Sigma^*} \in L(\Sigma^* 1 \Sigma^*)$

Regular expressions–examples

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$

$L(\Sigma^*001\Sigma^*) = \{w \in \Sigma^* \mid w$ contains 001 as a substring$\}$

Regular expressions–examples

$* \circ \cup$

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$

$L(1^*(01^+)^*) = \{w \in \Sigma^* \mid$ every 0 is followed by a $1\}$

$1\,1\,1\,0\,1\,0\,1\,1\,1\,0\,1\,1$

$\epsilon\ 1^*\quad \epsilon 1^+\quad 1^+\quad 1^+$

Regular expr

precedence $* \circ \cup$

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ.

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$

$L(1^*(01\,1^*)^*) = \{w \in \Sigma^* \mid \text{every } 0 \text{ is followed by a } 1\}$

Regular expr...

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ.

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$
- $(\Sigma\Sigma)^*$

$L(\Sigma) = \{w \in \Sigma^* \mid w \text{ has only one letter}\}$

$= \{0, 1\}$

$L((\Sigma \circ \Sigma)) = \{00, 01, 10, 11\}$

$L((\Sigma\Sigma)^*) = \{w \in \Sigma^* \mid w \text{ has an even length}\}$

$A^+ = \{w \in \Sigma^* \mid w$ is the concatenation of one or more words from $A\}$

A^* always includes ε

Regular exp

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$
- $(\Sigma\Sigma)^*$
- $(\Sigma\Sigma\Sigma)^*$

$L((\Sigma\Sigma\Sigma)^*) = \{w \in \Sigma^* \mid$ the length of w is divisible by $3\}$

Regular expressions–examples

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10$
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
- $(0 \cup \epsilon)1^*$
- $(0 \cup \epsilon)(1 \cup \epsilon)$
- $1^*\emptyset$
- $\emptyset^*$

Regular expr

$* \quad \circ \quad \cup$

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10 = ((0 \circ 1) \cup (1 \circ 0))$

Construction sequence

1. (core set)
2. 0 (core - set)
3. $(0 \circ 1)$ (applying $O_\circ$ to 2+1)
4. $(1 \circ 0)$ (applying $O_\circ$ to 1+2)
5. $((0 \circ 1) \cup (1 \circ 0))$ (applying $O_\cup$ to 3,4)

$L(1) = \{1\}$

$L(0) = \{0\}$

$L((0 \circ 1)) = \{01\}$

$L((1 \circ 0)) = \{10\}$

$L(((0 \circ 1) \cup (1 \circ 0))) = \{01, 10\}$

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Talking:

Regular expr[...]

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10$
- $(0\Sigma^*0) \cup (1\Sigma^*1) \cup 0 \cup 1$

$L(0) = \{0\}$

$L(1) = \{1\}$

$L(0\Sigma^*0) = \{w \in \Sigma^* \mid w$ starts and ends with 0 and has length at least 2$\}$

$L(1\Sigma^*1) = \{$ ... 1 ... $\}$

$L(0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1) = \{w \in \Sigma^* \mid w \neq \varepsilon$ and w starts and ends on the same letter$\}$

Regular expr

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10$
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
- $(0 \cup \epsilon)1^*$

$L(0 \cup \varepsilon) = \{0, \varepsilon\}$

$L((0 \cup \varepsilon)1^*) = \{w \in \Sigma^* \mid w$ has only 1 or contains a single 0 at the start$\}$

Regular expr

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10$
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
- $(0 \cup \epsilon)1^*$
- $(0 \cup \epsilon)(1 \cup \epsilon)$
- $1^*\emptyset$
- $\emptyset^*$

$L(1^*\emptyset) = \emptyset$

(concatenating any language with $\emptyset$ results in $\emptyset$)

Regular expr

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ:

- $01 \cup 10$
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
- $(0 \cup \epsilon)1^*$
- $(0 \cup \epsilon)(1 \cup \epsilon)$
- $1^*\emptyset$ $L(1^*\emptyset) = \emptyset$
- $\emptyset^*$

$L(\emptyset^*) = \{\varepsilon\}$

Regular expressions–more examples

Let R be some regular expression. Then we have:

- $L(R \cup \emptyset) = L(R)$
- $L(R \circ \epsilon) = L(R)$
- $L(R \cup \epsilon)$ is not necessarily equal to $L(R)$
- $L(R \circ \emptyset)$ is not necessarily equal to $L(R)$

Regular expr

Let R be some regular expression. Then we have:

- $L(R \cup \emptyset) = L(R)$
- $L(R \circ \epsilon) = L(R)$
- $L(R \cup \epsilon)$ is not necessarily equal to $L(R)$

Example: $L(R) = \{1, 10\}$, $L(R \cup \varepsilon) = \{\varepsilon, 1, 10\}$

- $L(R \circ \emptyset)$ is not necessarily equal to $L(R)$

Example: $L(R) = \{1, 10\}$, $L(R \circ \emptyset) = \emptyset$

Theorem
A language is regular if and only if some regular expression describes it.

Lemma 1
If a language is described by a regular expression, then it is regular.

Lemma 2
If a language is regular, then there is regular expression that describes it.

Expressive p

Theorem
A language is regular if and only if some regular expression describes it.

$\Longleftrightarrow$

$\Longleftarrow$

Lemma 1
If a language is described by a regular expression, then it is regular.

$\Longrightarrow$

Lemma 2
If a language is regular, then there is regular expression that describes it.

you can find the proof in the textbook

Expressive power of regular expressions–proof of Lemma 1

Let Σ be some alphabet. We prove by induction (according to the inductive definition of regular expressions) that for every regular expression R there exists and NFA that recognizes the language $L(R)$ (and this implies that $L(R)$ is a regular language).

Base case

We prove that the claim is true for members of the core-set.

1. If $R = a$ for some member of the alphabet Σ then we have $L(R) = \{a\}$. We can construct an NFA recognizing $L(R)$ as follows:

2. If $R = \epsilon$, then we have $L(R) = \{\epsilon\}$. We can construct an NFA recognizing $L(R)$ as follows:

3. If $R = \emptyset$, then we have $L(R) = \emptyset$. We can construct an NFA recognizing $L(R)$ as follows:

Expressive p... proof of Lemma 1

Let Σ be some alphabet. We prove by induction (according to the inductive definition of regular expressions) that for every regular expression R there exists an *NFA* that recognizes the language $L(R)$ (and this implies that $L(R)$ is a regular language).

Base case

We prove that the claim is true for members of the core-set.

1. If $R = a$ for some member of the alphabet Σ then we have $L(R) = \{a\}$. We can construct an *NFA* recognizing $L(R)$ as follows:

2. If $R = \epsilon$, then we have $L(R) = \{\epsilon\}$. We can construct an *NFA* recognizing $L(R)$ as follows:

3. If $R = \emptyset$, then we have $L(R) = \emptyset$. We can construct an *NFA* recognizing $L(R)$ as follows:

Expressive power of regular expressions–proof of Lemma 1

Induction Hypothesis
We assume that for two regular expressions R_1 and R_2 there exists NFAs N_1 and N_2 such that

$$L(R_1) = L(N_1) \quad \text{and} \quad L(R_2) = L(N_2)$$

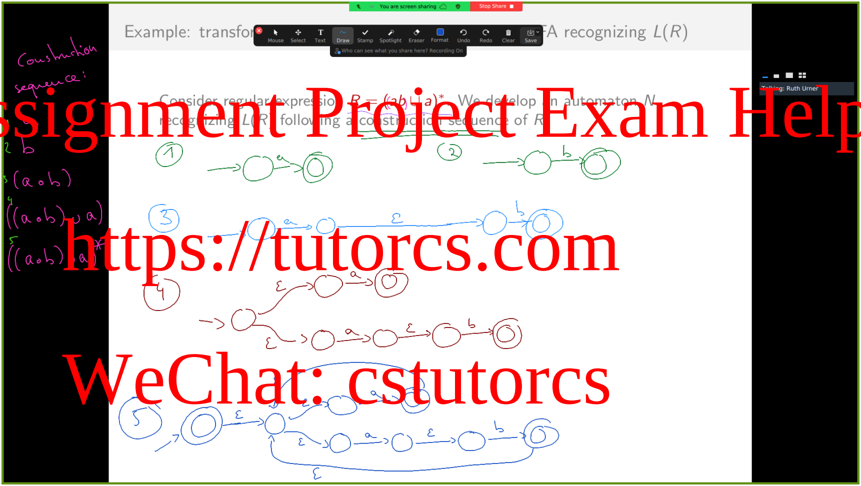
Induction Step
We need to show that, given the induction hypothesis, there exist NFAs that recognize the languages obtained by applying the three operations to the expressions R_1 and R_2.

1. We have $o_\cup(R_1, R_2) = (R_1 \cup R_2)$ and $L((R_1 \cup R_2)) = L(R_1) \cup L(R_2)$. By the induction hypothesis, there exist NFAs N_1 and N_2 recognizing $L(R_1)$ and $L(R_2)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1) \cup L(R_2)$ (regular languages are closed under unions).

2. We have $o_\circ(R_1, R_2) = (R_1 \circ R_2)$ and $L((R_1 \circ R_2)) = L(R_1) \circ L(R_2)$. By the induction hypothesis, there exist NFAs N_1 and N_2 recognizing $L(R_1)$ and $L(R_2)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1) \circ L(R_2)$ (regular languages are closed under concatenation).

3. We have $o_*(R_1) = (R_1^*)$ and $L((R_1^*)) = L(R_1)^*$. By the induction hypothesis, there exists an NFA N_1 recognizing $L(R_1)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1)^*$ (regular languages are closed under the star-operation).

Consider regular expression $R = (ab \cup a)^*$. We develop an automaton N recognizing $L(R)$ following a construction sequence of R:


Example: transfor
FA recognizing L(R)
You are screen sharing
Stop Share
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Talking: Ruth Urner
Consider regular expression R = (ab ∪ a)*. We develop an automaton N
recognizing L(R) following a construction sequence of R
Construction sequence:
2 b
3 (a∘b)
4 ((a∘b)∪a)
5 ((a∘b)∪a)*
1
2
3
4
5
a
b
ε

Non-regular Languages

Reading: ITC Section 1.4

Proving a language is not regular

We have seen multiple techniques to prove that a language L over some alphabet Σ is regular by now:

- It suffices to prove that there exists a DFA that recognizes (accepts) L.

- It suffices to prove that there exists an NFA that recognizes (accepts) L.

- It suffices to prove that there exists a regular expression that is interpreted as L.

To prove that some language L is not regular, we need to prove that none of these is possible.

Here is a general technique for showing that some object t is not a member of some set S:

- Prove that all members of S have some property P.
- Prove that t doesn't have property P.
- Conclude that $t \notin S$.

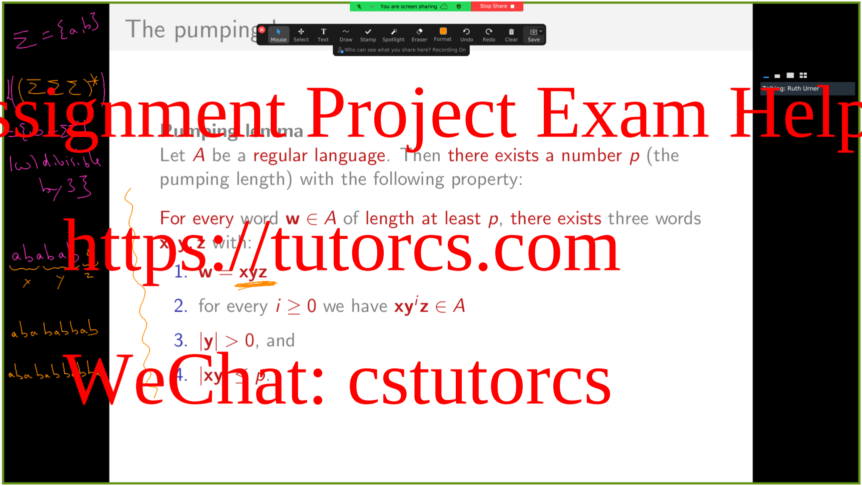
We will next state (and prove) a property that all regular languages have.

Pumping lemma

Let A be a regular language. Then there exists a number p (the pumping length) with the following property:

For every word $\mathbf{w} \in A$ of length at least p, there exists three words $\mathbf{x}, \mathbf{y}, \mathbf{z}$ with:

1. $\mathbf{w} = \mathbf{xyz}$
2. for every $i \geq 0$ we have $\mathbf{xy}^i\mathbf{z} \in A$
3. $|\mathbf{y}| > 0$, and
4. $|\mathbf{xy}| \leq p$.


You are screen sharing
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Talking: Ruth Urner
The pumping l
Pumping lemma
Let A be a regular language. Then there exists a number p (the pumping length) with the following property:
For every word w ∈ A of length at least p, there exists three words x, y, z with:
1. w = xyz
2. for every i ≥ 0 we have xy^i z ∈ A
3. |y| > 0, and
4. |xy| ≤ p.
Σ = {a, b}
|w| divisible by 3}
ababab
x y z
ababbabbab