cse581
Computer Science Fundamentals: Theory

Professor Anita Wasilewska
TCB - LECTURE 3

TCB - THEORY OF COMPUTATION BASICS
DM and TCB

PART 3: Special types of Binary Relations
PART 4: Finite and Infinite Sets
PART 5: Some Fundamental Proof Techniques

Theory of Computation BASICS
PART 6: Closures and Algorithms
PART 7: Alphabets and languages
PART 8: Finite Representation of Languages
PART 8: Finite Representation of Languages
Regular Languages
Finite Representation of Languages

Introduction

We can represent a finite language by finite means for example listing all its elements.

Languages are often infinite and so a natural question arises if a finite representation is possible and when it is possible when a language is infinite.

The representation of languages by finite specifications is a central issue of the theory of computation.

Of course we have to define first formally what do we mean by representation by finite specifications, or more precisely by a finite representation.
Idea of Finite Representation

We start with an example: let

\[ L = \{a\}^* \cup (\{b\} \circ \{a\}^*) = \{a\}^* \cup (\{b\}\{a\}^*) \]

Observe that by definition of Kleene’s star

\[ \{a\}^* = \{e, a, aa, aaa \ldots \} \]

and \( L \) is an infinite set

\[ L = \{e, a, aa, aaa \ldots \} \cup \{b\}\{e, a, aa, aaa \ldots \} \]
\[ L = \{e, a, aa, aaa \ldots \} \cup \{b, ba, baa, baaa \ldots \} \]
\[ L = \{e, a, b, aa, ba, aaa baa, \ldots \} \]
Idea of Finite Representation

The expression \( \{a\}^* \cup (\{b\}\{a\}^*) \) is built out of a finite number of symbols:

\(\{,\}, (, ), *, \cup\)

and describe an infinite set

\[ L = \{e, a, b, aa, ba, aaa baa, \ldots\} \]

We write it in a simplified form - we skip the set symbols \(\{, \}\) as we know that languages are sets and we have

\[ a^* \cup (ba^*) \]
Idea of Finite Representation

We will call such expressions as

\[ a^* \cup (ba^*) \]

a finite representation of a language \( L \)

The idea of the finite representation is to use symbols

\((, ), *, \cup, \emptyset,\) and symbols \( \sigma \in \Sigma \)

to write expressions that describe the language \( L \)
Example of a Finite Representation

Let $L$ be a language defined as follows

$L = \{ w \in \{0, 1\}^* : \ w \text{ has two or three occurrences of } 1 \text{, the first and the second of which are not consecutive } \}$

The language $L$ can be expressed as

$L = \{0\}^*\{1\}\{0\}^*\{0\} \circ \{1\}\{0\}^*\{\{1\}\{0\}^* \cup \emptyset^*\}$

We will define and write the **finite representation** of $L$ as

$L = 0^*10^*010^*(10^* \cup \emptyset^*)$

We call expression above (and others alike) a **regular expression**
Problem with Finite Representation

Question
Can we finitely represent all languages over an alphabet $\Sigma \neq \emptyset$?

Observation
O1. Different languages must have different representations
O2. Finite representations are finite strings over a finite set, so we have that

there are $\aleph_0$ possible finite representations
Problem with Finite Representation

O3. There are uncountably many, precisely exactly $C = |\mathbb{R}|$ of possible languages over any alphabet $\Sigma \neq \emptyset$

Proof

For any $\Sigma \neq \emptyset$ we have proved that

$$|\Sigma^*| = \aleph_0$$

By definition of language

$$L \subseteq \Sigma^*$$

so there are as many languages as subsets of $\Sigma^*$ that is as many as

$$|2^{\Sigma^*}| = 2^{\aleph_0} = C$$
Problem with Finite Representation

Question
Can we finitely represent all languages over an alphabet $\Sigma \neq \emptyset$?

Answer
No, we can’t
By $O2$ and $O3$ there are countably many (exactly $\aleph_0$) possible finite representations and there are uncountably many (exactly $C$) possible languages over any $\Sigma \neq \emptyset$

This proves that
NOT ALL LANGUAGES CAN BE FINITELY REPRESENTED
Problem with Finite Representation

Moreover

There are uncountably many and exactly as many as Real numbers, i.e. $C$ languages that can not be finitely represented.

We can finitely represent only a small, countable portion of languages.

We define and study here only two classes of languages:

- REGULAR
- CONTEXT FREE
Definition
We define a $\mathcal{R}$ of regular expressions over an alphabet $\Sigma$ as follows

$\mathcal{R} \subseteq (\Sigma \cup \{, , \emptyset, \cup, *\})^*$ and $\mathcal{R}$ is the smallest set such that

1. $\emptyset \in \mathcal{R}$ and $\Sigma \subseteq \mathcal{R}$, i.e. we have that

   $\emptyset \in \mathcal{R}$ and $\forall \sigma \in \Sigma \ (\sigma \in \mathcal{R})$

2. If $\alpha, \beta \in \mathcal{R}$, then

   $(\alpha \beta) \in \mathcal{R}$  concatenation

   $(\alpha \cup \beta) \in \mathcal{R}$  union

   $\alpha^* \in \mathcal{R}$  Kleene’s Star
Regular Expressions Theorem

**Theorem**
The set $\mathcal{R}$ of regular expressions over an alphabet $\Sigma$ is countably infinite

**Proof**
Observe that the set $\Sigma \cup \{(, ), \emptyset, \cup, \ast\}$ is non-empty and finite, so the set $(\Sigma \cup \{(, ), \emptyset, \cup, \ast\})^*$ is countably infinite, and by definition

$$\mathcal{R} \subseteq (\Sigma \cup \{(, ), \emptyset, \cup, \ast\})^*$$

hence we $|\mathcal{R}| \leq \aleph_0$

The set $\mathcal{R}$ obviously includes an infinitely countable set

$$\emptyset, \emptyset\emptyset, \emptyset\emptyset\emptyset, \ldots, \ldots,$$

what proves that $|\mathcal{R}| = \aleph_0$
Regular Expressions

Example

Given \( \Sigma = \{0, 1\} \), we have that

1. \( \emptyset \in \mathcal{R}, \ 1 \in \mathcal{R}, \ 0 \in \mathcal{R} \)

2. \( (01) \in \mathcal{R}, \ 1^* \in \mathcal{R}, \ 0^* \in \mathcal{R}, \ (\emptyset \cup 1) \in \mathcal{R}, \ldots, \ ((0^* \cup 1^*) \cup \emptyset)1^* \in \mathcal{R} \)

Shorthand Notation  

when writing regular expressions we will keep only essential parenthesis

For example, we will write

\[
(0^* \cup 1^* \cup \emptyset)1^* \quad \text{instead of} \quad (((0^* \cup 1^*) \cup \emptyset)1^*)
\]

\[
1^*01^* \cup (01)^* \quad \text{instead of} \quad ((((1^*0)^*1^*) \cup (01)^*))
\]
Regular Expressions and Regular Languages

We use the regular expressions from the set \( R \) as a representation of languages.

Languages represented by regular expressions are called regular languages.
Regular Expressions and Regular Languages

The idea of the representation is explained in the following Example:

The regular expression (written in a shorthand notion)

\[1^*01^* \cup (01)^*\]

represents a language

\[L = \{1\}^*\{0\}\{1\}^* \cup \{01\}^*\]
Definition of Representation

Definition
The representation relation between regular expressions and languages they represent is established by a function $L$ such that if $\alpha \in \mathcal{R}$ is any regular expression, then $L(\alpha)$ is the language represented by $\alpha$. 
Definition of Representation

Formal Definition

The function $\mathcal{L} : \mathcal{R} \longrightarrow 2^{\Sigma^*}$ is defined recursively as follows

1. $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$

2. If $\alpha, \beta \in \mathcal{R}$, then

   $\mathcal{L}(\alpha \beta) = \mathcal{L}(\alpha) \circ \mathcal{L}(\beta)$, concatenation

   $\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$, union

   $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$, Kleene’s Star
Regular Language Definition

Definition
A language $L \subseteq \Sigma^*$ is regular

if and only if

$L$ is represented by a regular expression, i.e.

when there is $\alpha \in \mathcal{R}$ such that $L = \mathcal{L}(\alpha)$

where $\mathcal{L}: \mathcal{R} \rightarrow 2^{\Sigma^*}$ is the representation function

We use a shorthand notation

$$L = \alpha \quad \text{for} \quad L = \mathcal{L}(\alpha)$$
Examples

E1
Given \( \alpha \in \mathcal{R} \), for \( \alpha = ((a \cup b)^* a) \)

Evaluate \( L \) over an alphabet \( \Sigma = \{a, b\} \), such that \( L = \mathcal{L}(\alpha) \)
We write
\[
\alpha = ((a \cup b)^* a)
\]
in the **shorthand** notation as
\[
\alpha = (a \cup b)^* a
\]
Examples

We evaluate \( L = (a \cup b)^* a \) as follows

\[
L((a \cup b)^* a) = L((a \cup b)^*) \circ L(a) = L((a \cup b)^*) \circ \{a\} =
\]

\[
(L(a \cup b))^* \{a\} = (L(a) \cup L(b))^* \{a\} = (\{a\} \cup \{b\})^* \{a\}
\]

Observe that

\[
(\{a\} \cup \{b\})^* \{a\} = \{a, b\}^* \{a\} = \Sigma^* \{a\}
\]

so we get

\[
L = L((a \cup b)^* a) = \Sigma^* \{a\}
\]

\[
L = \{w \in \{a, b\}^* : w \text{ ends with } a\}
\]
Examples

E2  Given $\alpha \in \mathbb{R}$, for $\alpha = (\{(c^*a) \cup (bc^*)\}^*)$

Evaluate $L = \mathcal{L}(\alpha)$, i.e. describe $L = \alpha$

We write $\alpha$ in the shorthand notation as

$$\alpha = c^*a \cup (bc^*)^*$$

and evaluate $L = c^*a \cup (bc^*)^*$ as follows

$$\mathcal{L}((c^*a \cup (bc^*)^*)) = \mathcal{L}(c^*a) \cup (\mathcal{L}(bc^*))^* = \{c\}^*\{a\} \cup (\{b\}\{c\}^*)^*$$

and we get that

$$L = \{c\}^*\{a\} \cup (\{b\}\{c\}^*)^*$$
Examples

**E3** Given \( \alpha \in \mathcal{R} \), for

\[
\alpha = (0^* \cup (((0^*(1 \cup (11))))((00^*)(1 \cup (11)))^*)0^*)
\]

Evaluate \( L = \mathcal{L}(\alpha) \), i.e **describe** the language \( L = \alpha \)

We write \( \alpha \) in the **shorthand** notation as

\[
\alpha = 0^* \cup 0^*(1 \cup 11)((00^*(1 \cup 11))^*)0^*
\]

and evaluate

\[
L = \mathcal{L}(\alpha) = 0^* \cup 0^*\{1, 11\}(00^*\{1, 11\})^*0^*
\]

**Observe** that \( 00^* \) contains at least one 0 that separates \( 0^*\{1, 11\} \) on the left with \( (00^*\{1, 11\})^* \) that follows it, so we get that

\[
L = \{w \in \{0, 1\}^* : w \text{ does not contain a substring } 111\}
Class $\textbf{RL}$ of Regular Languages

Definition

Class $\textbf{RL}$ of regular languages over an alphabet $\Sigma$ contains all $L$ such that $L = \mathcal{L}(\alpha)$ for certain $\alpha \in R$, i.e.

$$\textbf{RL} = \{ L \subseteq \Sigma^* : L = \mathcal{L}(\alpha) \text{ for certain } \alpha \in R \}$$

Theorem

There $\aleph_0$ regular languages over $\Sigma \neq \emptyset$ i.e.

$$|\textbf{RL}| = \aleph_0$$

Proof

By definition that each regular language is $L = \mathcal{L}(\alpha)$ for certain $\alpha \in R$ and the interpretation function $\mathcal{L} : R \rightarrow 2^{\Sigma^*}$ has an infinitely countable domain, hence $|\textbf{RL}| = \aleph_0$
Class **RL** of Regular Languages

We can also think about languages in terms of **closure** and get immediately from definitions the following

**Theorem**

Class **RL** of regular languages is the **closure** of the set of languages

$$\{{\{\sigma\} : \sigma \in \Sigma}\} \cup \{\emptyset\}$$

with respect to **union**, **concatenation** and **Kleene Star**
Languages that are NOT Regular

Given an alphabet $\Sigma \neq \emptyset$
We have just proved that there are $\aleph_0$ regular languages, and we have also there are $C$ of all languages over $\Sigma \neq \emptyset$, so we have the following

Fact
There is $C$ languages that are not regular

Natural Questions
Q1 How to prove that a given language is regular?
A1 Find a regular expression $\alpha$, such that $L = \alpha$, i.e. $L = \mathcal{L}(\alpha)$
Languages that are NOT Regular

Q2 How to prove that a given language is not regular?
A2 Not easy!

There is a Theorem, called Pumping Lemma which provides a criterium for proving that a given language is not regular

E1 A language

$$L = 0^*1^*$$

is is regular as it is given by a regular expression $$\alpha = 0^*1^*$$

E2 We will prove, using the Pumping Lemma that the language

$$L = \{0^n1^n : \ n \geq 1, \ n \in N\}$$

is not regular
PROBLEMS

Some REGULAR LANGUAGES Problems
Problem 1

Consider the following languages over $\Sigma = \{a, b\}$

$$L_1 = \{w \in \Sigma^* : \exists u \in \Sigma\Sigma (w = uu^Ru)\}$$

$$L_2 = \{w \in \Sigma^*: \text{ww = www}\}$$

Part 1: Prove that $L_1$ is a finite set

Give example of 3 words $w \in L_1$

Solution

We evaluate first the set $\Sigma\Sigma = \{a, b\}\{a, b\} = \{aa, bb, ab, ba\}$

$\Sigma\Sigma$ is a finite set, hence the set $B = \{uyu : u, y \in \Sigma\Sigma\}$ is also a finite set and by definition $L_1 \subseteq B$

This proves that $L_1$ must be a finite set
Problem 1

We evaluated that $\Sigma \Sigma = \{a, b\}\{a, b\} = \{aa, bb, ab, ba\}$

We defined $L_1 = \{w \in \Sigma^* : \exists u \in \Sigma \Sigma (w = uu^R u)\}$

By evaluation we have that

$L_1 = \{aaaaaa, abbaab, baabba, bbbbbbb\}$

Part 2: Give examples of 2 words over $\Sigma$ such that $w \notin L_1$

Solution $a \notin L_1, \ bba \notin L_1$

There are countably infinitely many words that are not in $L_1$
Problem 1

Part 3  Consider now the following language

\[ L_2 = \{ w \in \{a, b\}^* : \; ww = www \} \]

Show that \( L_2 \neq \emptyset \)

**Solution**  \( e \in L_2 \), as \( ee = eee \)

In fact, \( e \) is the only word with this property, hence

\[ L_2 = \{ e \} \]

Part 4  Show that the set \( (\Sigma^* - L_2) \) is infinite

**Solution**  \( \Sigma^* \) is countably infinite, \( L_2 \) is finite, so (basic theorem) \( (\Sigma^* - L_2) \) is countably infinite

Any \( w \in \Sigma^* \), such that \( w \neq e \) is in \( (\Sigma^* - L_2) \)
Problem 2

Given expressions (written in a short hand notation)

\[ \alpha_1 = \emptyset^* \cup a^* \cup b^* \cup a \cup b \cup (a \cup b)^* \]

\[ \alpha_2 = (a \cup b)^* b (a \cup b)^* \]

Part 1 Re-write \( \alpha_1 \) as a \textbf{simpler} expression representing the same language

List \textbf{properties} you used in your solution

Describe the language \( L = \mathcal{L}(\alpha_1) \)
Problem 2

Solution  We first evaluate

\[ L(\alpha_1) = L(\emptyset^* \cup a^* \cup b^* \cup a \cup b \cup (a \cup b)^*) \]

\[ = e \cup \{a\}^* \cup \{b\}^* \cup \{a\} \cup \{b\} \cup (\{a\} \cup \{b\})^* = \Sigma^* \]

This is true because of the properties:

\[ (\{a\} \cup \{b\})^* = \{a, b\}^* = \Sigma^* \]

and

\[ \{a\} \subseteq \{a\}^*, \ {b\} \subseteq \{b\}^*, \ \{a\}^* \subseteq \Sigma^*, \ \{b\}^* \subseteq \Sigma^* \]

and we know that for any sets \( A, B \), if \( A \subseteq B \), then \( A \cup B = B \)

\[ L(\alpha_1) = \Sigma^* = (\{a\} \cup \{b\})^* = L((a \cup b)^*) \]

We hence simplify \( \alpha_1 \) as follows

\[ \alpha_1 = \emptyset^* \cup a^* \cup b^* \cup a \cup b \cup (a \cup b)^* = (a \cup b)^* \]
Problem 3

Part 2  Given

\[ \alpha_2 = (a \cup b)^* b (a \cup b)^* \]

Re-write \( \alpha_2 \) as a simpler expression representing the same language

Describe the language \( L = \mathcal{L}(\alpha_2) \)

Solution  \( \alpha_2 \) can not be simplified, but we can use property \( (\{a\} \cup \{b\})^* = \Sigma^* \) to describe informally the language determined by \( \alpha_2 \) as

\[ L = \mathcal{L}(\alpha_2) = \Sigma^* b \Sigma^* \]

Remember that informal description \( \Sigma^* b \Sigma^* \) is not a regular expression - but just an useful notation
Problem 3

Let $\Sigma = \{a, b\}$ and a language $L \subseteq \Sigma^*$ be defined as follows:

$$L = \{w \in \Sigma^* : w \text{ contains no more then two } a's\}$$

Write a regular expression $\alpha$, such that $L(\alpha) = L$. Use shorthand notation. **Explain** shortly your answer.

**Solution**

$$\alpha = b^* \cup b^* ab^* \cup b^* ab^* ab^*$$

**Explanation**

$b^*$ contains 0 of $a$'s (case n=0)

$b^* ab^*$ contains 1 occurrence of $a$ (case n=1)

$b^* ab^* ab^*$ contains 2 occurrence of $a$ (case n=2)
Let \( \Sigma = \{a, b\} \)

The language \( L \subseteq \Sigma^* \) is defined as follows:
\[
L = \{ w \in \Sigma^* : \text{the number of } b \text{'s in } w \text{ is divisible by } 4 \}
\]

Write a regular expression \( \alpha \), such that \( L(\alpha) = L \)

You can use **shorthand notation**. Explain shortly your answer.

**Solution**
\[
\alpha = a^* (a^* ba^* ba^* ba^* ba^*)^*
\]

Observe that the regular expression \( a^* ba^* ba^* ba^* ba^* \) describes a string \( w \in \Sigma^* \) with **exactly four** \( b \)’s.
Problem 4

The regular expression

\[(a^{*}ba^{*}ba^{*}ba^{*}ba^{*})^{*}\]

represents multiples of \(w \in \Sigma^{*}\) with **exactly four** \(b\)'s and hence words in which a number of \(b\)'s is **divisible by** 4.

**Observe** that 0 is divisible by 4, so we need to add the case of 0 number of \(b\)'s, i.e. we need to include words \(e, a, aa, aaa, \ldots\).

We do so by concatenating \((a^{*}ba^{*}ba^{*}ba^{*}ba^{*})^{*}\) with \(a^{*}\) and get

\[L = a^{*}(a^{*}ba^{*}ba^{*}ba^{*}ba^{*})^{*}\]
Problem 5

Let $L$ be a language defined as follows

$$L = \{ w \in \{a, b\}^*: \ P(w) \}$$

for the property $P(w)$ defined as follows

$P(w)$: between any two $a$’s in $w \in \{a, b\}^*$ there is an even number of consecutive $b$’s

1. Describe a regular expression $r$ such that $L(r) = L$

Remark that 0 is an even number, hence $a^* \in L$ and

$$r = b^* \cup b^*a^*b^* \cup b^*(a(bb)^*a)^*b^* = b^*a^*b^* \cup b^*(a(bb)^*a)^*b^*$$
Problem 6

Let $\Sigma$ be any alphabet, $L_1, L_2$ two languages over $\Sigma$ such that $e \in L_1$ and $e \in L_2$

Show that

$$(L_1 \Sigma^* L_2)^* = \Sigma^*$$

Solution

By definition, $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$ and $\Sigma^* \subseteq \Sigma^*$

Hence

$$(L_1 \Sigma^* L_2) \subseteq \Sigma^*$$
Problem 6

Now we use the following property:

Property
For any languages $L_1, L_2$, if $L_1 \subseteq L_2$, then $L_1^* \subseteq L_2^*$
and obtain that $(L_1 \Sigma^* L_2)^* \subseteq \Sigma^* = \Sigma^*$, i.e. we proved that

$$(L_1 \Sigma^* L_2)^* \subseteq \Sigma^*$$

We have to show now that also

$$\Sigma^* \subseteq (L_1 \Sigma^* L_2)^*$$

Let $w \in \Sigma^*$, we have that also $w \in (L_1 \Sigma^* L_2)^*$ because $w = ewe$ and $e \in L_1$ and $e \in L_2$. We have hence proved that

$$(L_1 \Sigma^* L_2)^* = \Sigma^*$$
Problem 7

Let $L$ be a function that associates with any regular expression $\alpha$ the regular language $L = L(\alpha)$

1. Evaluate $L = L(\alpha)$ for $\alpha = (a \cup b)^*a$

Solution

$L = L((a \cup b)^*a) = L((a \cup b)^*)L(a) = (L(a \cup b))^*\{a\} = (L(a) \cup L(b))^*\{a\} = (\{a\} \cup \{b\})^*\{a\} = \{a, b\}^*\{a\}$

2. Describe a property that defines the language $L = L((a \cup b)^*a)$

Solution

$L = \{a, b\}^*\{a\} = \Sigma^*\{a\} = \{w \in \{a, b\}^* : w \text{ ends with } a \}$
General Problem

Given a language \( L \) over \( \Sigma \) and a word \( w \in \Sigma^* \),
HOW to RECOGNIZE whether

\[ w \in L \quad \text{or} \quad w \notin L \]

Next SUBJECT

Automata - LANGUAGE RECOGNITION devices