Regular Languages

H. Geuvers and J. Rot

Institute for Computing and Information Sciences
Radboud University Nijmegen

Version: fall 2016

Outline

About this course I

Lectures

- Teachers: Herman Geuvers and Jurriaan Rot
- Weekly, 2 hours, on Tuesdays 13:45 – 15:30
- Presence not compulsory ...
  - but active, polite attitude expected, when present
- The lectures follow:
  - these slides, available via the web
  - *Languages and Automata* by Alexandra Silva (LnA)
- Course URL:

  http://www.cs.ru.nl/~herman/onderwijs/2016TnA/

Check there first, before you dare to ask/mail a question!

About this course II

Exercises

- There are weekly exercises; the ones marked with **points** are to be handed in.
- Handing in is compulsory: To receive a grade for the course, you have to hand in every week.
- Exercises must be done individually
- Weekly exercise classes, on Fridays, 10:45 – 12:30 (and one class on Friday 13:45 – 15:30)
  - Presence not compulsory
  - Answers (for old exercises) & Questions (for new ones)
- Schedule:
  - New exercises on the web: Tuesday afternoon
  - Next exercise meeting (Friday) you can ask questions
  - Hand-in: **Tuesday before 13:45** in the delivery boxes.

About this course III

Exercise Classes

- 7 Assistants:

Michiel de Bondt	HG00.058	10:45 – 12:30
Bas Steeg	HG02.032	10:45 – 12:30
Demian Janssen	HG01.058	10:45 – 12:30
Jan Martens	HG01.139	10:45 – 12:30
Ties Robroek	HG02.028	10:45 – 12:30
Sjoerd Hemels	HG03.632	10:45 – 12:30
Rick Erkens	HG02.028	**afternoon 13:45 – 15:30**

- Please register for an exercise group via blackboard by **Tuesday Novemer 8, 17:00**, depending on your own assessment of your skiils.
- You will be assigned to an exercise class by me
- Each assistant has a blue delivery box on the ground floor of the **Mercator 1 building**

About this course IV

Examination

- There is a half-way test and a final test.
- The final grade is composed of
  - the grade of your half-way test, **h**,
  - the grade of your final test, **f**,
  - the average grade of your exercises, **a**,
- Your final grade is $\min(10, \frac{\mathbf{f}+\mathbf{h}}{2} + \frac{\mathbf{a}}{10})$
  - The re-exam is a full 3hrs exam about the whole course. You keep the (average) grade of the exercises.
- If you fail again, you must start all over next year (including re-doing new exercises, and additional requirements)

About this course V

If you fail more than twice ...

- Additional requirements are imposed
- you will have to talk to the study advisor
  - if you have not done so yet, make an appointment
  - compulsory: presence at lectures and exercise classes (and of course handing in of exercises)
  - go see Herman Geuvers (M1 0.05) to sign the form

Overview

Topics

Languages:	Automata:	Grammars:
regular	finite	regular
context-free	push-down	context-free
[natural languages]	[bounded Turing machine]	[context-sensitive]
[enumerable]	[Turing machine]	[unrestricted]

Automata: accept words of a language
given a word, compute if it is in the language

Grammars: generate words of a language
produce all correct words in the language

Languages

An alphabet A is a (finite) set of symbols

Examples

$A_1 = \{a\}$

$A_2 = \{0, 1\}$

$A_3 = \{A, C, G, T\}$

$A_4 = \{a, b, c, d, \ldots, x, y, z\}$

$A_5 = \{s \mid s \text{ is an ascii symbol}\}$

$A_6 =$ { あ、い、う、え、お、か、き、く、け、こ、. . . }
Japanese alphabet: 2×52 signs

$A_7 =$ { 山 (mountain) 川 (river) 日 (sun) 雨 (rain) 水 (water) 火 (fire) 田 (rice field), . . . }
Chinese alphabet: 40.000 signs

$A_8 = \{0, 1, +, \times, x_0, x_1, x_2, \ldots\}$
mathematical alphabet, countably infinite

$A_9 = \{0, 1, +, \times, x_0, x_1, x_2, \ldots\} \cup \{c_r \mid r \in \mathbb{R}\}$
mathematical alphabet, uncountably infinite

Words

A word (string) over A is a finite sequence of elements from A
The set A^* consists of all words over A

Inductive definition of the set of words, A^*

1. $\lambda \in A^*$ (λ denotes the empty word).
2. If $a \in A$ and $v \in A^*$, then $a\,v \in A^*$.

Note that $a\,\lambda$ is just a
Note the difference between $a \in A$ and $a \in A^*$
Think of a word as a chain of letters on a necklace:

$$\lambda = \text{——}$$
$$Eva = \text{–}E\text{—}v\text{—}a\text{–}$$

The difference between a and $\text{–}a\text{–}$ is clear

Operation on words

Inductive definition of the set of words, A^*

1. $\lambda \in A^*$ (λ denotes the empty word).
2. If $a \in A$ and $v \in A^*$, then $a\, v \in A^*$.

Operations on words

$$
\begin{array}{lll}
v \in A^*,\ u \in A^* & \Rightarrow & v \cdot u \in A^*, \text{ concatenation} \\
v \in A^*,\ n \in \mathbb{N} & \Rightarrow & v^n \in A^*, \text{ repetition} \\
v \in A^* & \Rightarrow & v^R \in A^*, \text{ reverse}
\end{array}
$$

Inductive definitions of concatenation, repetition and reverse

$$
\begin{array}{rcl}
\lambda \cdot u & = & u \\
(av) \cdot u & = & a(v \cdot u)
\end{array}
\qquad
\begin{array}{rcl}
v^0 & = & \lambda \\
v^{k+1} & = & v \cdot v^k
\end{array}
\qquad
\begin{array}{rcl}
\lambda^R & = & \lambda \\
(av)^R & = & (v^R) \cdot a
\end{array}
$$

We write concatenation $v \cdot u$ as $v\, u$

Operation on words; Language

A language over A is a subset of A^*, notation $L \subseteq A^*$

Examples (with $A = \{a, b\}$)

$L_1 = \{w \in \{a, b\}^* \mid abba \text{ is a substring of } w\}$
$L_2 = \{w \in \{a, b\}^* \mid w = w^R\}$

Examples of languages

Let $A = \{a, b, c\}$.

1. $L_1 = \{a^n \mid n \in \mathbb{N} \text{ is even}\}$
2. $L_2 = \{a^n b^n \mid n \in \mathbb{N}\}$
3. $L_3 = \{a^n b^n c^n \mid n \geq 2\}$
4. $L_4 = \{a^n \mid n \in \mathbb{N} \text{ is prime}\}$

Over other alfabets:

1. $L_5 = \{n \mid n \text{ denotes an integer number}\}$
2. $L_6 = \{e \mid e \text{ is a well-formed arithmetical expression}\}$
3. $L_7 = \{P \mid P \text{ is a syntactically correct Java program}\}$
4. $L_8 = \{S \mid S \text{ is a grammatically correct English sentence}\}$

Operations on languages

Given languages $L_1, L_2, L \subseteq A^*$ we can define new languages:

$$L_1 \cup L_2 \quad L_1 \cap L_2 \quad \overline{L} \quad L_1 L_2 \quad L^*$$

$$\begin{aligned}
L_1 \cup L_2 &= \{w \mid w \in L_1 \text{ or } w \in L_2\} \\
L_1 \cap L_2 &= \{w \mid w \in L_1 \text{ and } w \in L_2\} \\
\overline{L} &= \{w \in A^* \mid w \notin L\} \\
L_1\, L_2 &= \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\} \\
L^0 &= \{\lambda\} \\
L^{n+1} &= L\, L^n \\
L^* &= \bigcup_{n \in \mathbb{N}} L^n = L^0 \cup L^1 \cup L^2 \cup \ldots \\
&\neq \{w^n \mid w \in L,\ n \in \mathbb{N}\}
\end{aligned}$$

Regular expressions and languages over A

Let $A = \{a, b\}$. Then $a(ba)^* bb$ is a *regular expression* denoting

$$
\begin{aligned}
L &= \{a(ba)^n bb \mid n \in \mathbb{N}\} \\
&= \{abb, ababb, abababb, ababababb, \ldots, a(ba)^n bb, \ldots\}
\end{aligned}
$$

For general A the regular expressions over A are generated by

$$\boxed{\mathtt{rexp}_\mathtt{A} ::= 0 \mid 1 \mid \mathtt{s} \mid (\mathtt{rexp}_\mathtt{A}\ \mathtt{rexp}_\mathtt{A}) \mid (\mathtt{rexp}_\mathtt{A} + \mathtt{rexp}_\mathtt{A}) \mid (\mathtt{rexp}_\mathtt{A})^*}$$

with $s \in A$

This means $0 \in \mathtt{rexp}_A$, $1 \in \mathtt{rexp}_A$, and $s \in \mathtt{rexp}_A$ for $s \in A$ and

$$
\begin{aligned}
e_1, e_2 \in \mathtt{rexp}_A &\Rightarrow (e_1 + e_2) \in \mathtt{rexp}_A \\
e_1, e_2 \in \mathtt{rexp}_A &\Rightarrow (e_1 e_2) \in \mathtt{rexp}_A \\
e \in \mathtt{rexp}_A &\Rightarrow (e)^* \in \mathtt{rexp}_A
\end{aligned}
$$

For example $(abb)^*(a + 1)$ is a regular expression

We economize on brackets

$$\texttt{rexp}_\texttt{A} ::= 0 \mid 1 \mid \texttt{s} \mid (\texttt{rexp}_\texttt{A}\ \texttt{rexp}_\texttt{A}) \mid (\texttt{rexp}_\texttt{A} + \texttt{rexp}_\texttt{A}) \mid (\texttt{rexp}_\texttt{A})^*$$

- We omit the outermost brackets,
- $*$ binds strongest,
- $+$ binds weakest.

So $a + ba^*$ denotes $((a + (b(a)^*)))$.
This denotes the language of either just a or b followed by a finite (possibly 0) number of a's.

Regular languages

For a regular expression e over A we define the language $\mathcal{L}(e)$:

$$
\begin{aligned}
\mathcal{L}(0) &= \emptyset \\
\mathcal{L}(1) &= \{\lambda\} \\
\mathcal{L}(s) &= \{s\} \\
\mathcal{L}(e_1 e_2) &= \mathcal{L}(e_1)\mathcal{L}(e_2) \\
\mathcal{L}(e_1 + e_2) &= \mathcal{L}(e_1) \cup \mathcal{L}(e_2) \\
\mathcal{L}(e^*) &= (\mathcal{L}(e))^*
\end{aligned}
$$

A language L is called regular if $L = \mathcal{L}(e)$ for some $e \in \texttt{rexp}$

Examples

Let $A = \{a, b\}$.

- Also $L = \{w \mid w \text{ begins with } bb\}$ is regular

$$L = \mathcal{L}(bb(a + b)^*)$$

- $L = \{w \mid bb \text{ occurs in } w\}$ is regular

$$L = \mathcal{L}((a + b)^* bb(a + b)^*)$$

- $L = \{w \mid |w|_b \leq 2\}$ is regular
  NB. $|w|$ denotes the length of w,
  $|w|_b$ denotes the number of b's in w

$$L = \mathcal{L}(a^*(ba^*b + b + 1)a^*)$$