Regular Languages

H. Geuvers and T. van Laarhoven

Institute for Computing and Information Sciences – Intelligent Systems
Radboud University Nijmegen

Version: fall 2014

Outline

About this course I

Lectures

- Teachers: Herman Geuvers and Twan van Laarhoven
- Weekly, 2 hours, on Wednesdays 08:45
- 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/flga2014/

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

About this course II

Exercises

- Handing in is compulsory in the sense that
  - To receive a grade for the course, the average exercise grade a must be ≥ 6.
  - $\frac{a}{10}$ is added to your exam grade as a bonus.
- The exercises marked with **points** are to be handed in.
- Exercises must be done individually
- Weekly exercise classes, on Wednesdays, 10:45.
  - Presence not compulsory
  - Answers (for old exercises) & Questions (for new ones)
- Schedule:
  - New exercises on the web: Tuesday evening
  - Next exercise meeting (Wednesday) you can ask questions
  - Hand-in: **Monday before 16:00** in the delivery boxes (or via other means in agreement with your assistant).

About this course III

Exercise Classes

- 2 Assistants:
  - Bastiaan Cijsouw, HG00.310
    If your surname starts with A...K
  - Maaike Zwart, HG00.065
    If your surname starts with L...Z
- Each assistant has a delivery box on the ground floor of the **Mercator 1 building**

About this course IV

Examination

- There is a half-way test-exam and a final exam.
- The final grade is composed of
  - the average grade of your exercises, **a** (must be ≥ 6),
  - the grade of your half-way test-exam, **h**,
  - the grade of your final exam, **f**, (must be ≥ 5).
- Your final grade is $\max(\mathbf{f}, \frac{\mathbf{f}+\mathbf{h}}{2}) + \frac{\mathbf{a}}{10}$ (with a max of 10).
  - Final exam: Monday, January 26, 8:30–11:30, LIN 6
  - Re-exam of written exam on ... *not yet known*.
  - You keep the (average) grade of the exercises and the grade of the half-way test-exam.
- 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 01.07A) 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 Σ is a (finite) set of symbols

Examples

$\Sigma_1 = \{a\}$

$\Sigma_2 = \{0, 1\}$

$\Sigma_3 = \{A, C, G, T\}$

$\Sigma_4 = \{a, b, c, d, \ldots, x, y, z\}$

$\Sigma_5 = \{s \mid s \text{ is an ascii symbol}\}$

$\Sigma_6 =$ { あ、い、う、え、お、か、き、く、け、こ、... }
Japanese alphabet: 2×52 signs

$\Sigma_7 =$ { 山 mountain 川 river 日 sun 雨 rain 水 water 火 fire 田 rice field ,... }
Chinese alphabet: 40.000 signs

$\Sigma_8 = \{0, 1, +, \times, x_0, x_1, x_2, \ldots\}$
mathematical alphabet, countably infinite

$\Sigma_9 = \{0, 1, +, \times, x_0, x_1, x_2, \ldots\} \cup \{c_r \mid r \in \mathbb{R}\}$
mathematical alphabet, uncountably infinite

Words

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

$$\lambda \in \Sigma^*, \quad \lambda \text{ denotes the empty word}$$
$$w \in \Sigma^* \text{ and } s \in \Sigma \quad \Rightarrow \quad ws \in \Sigma^*$$

Note that λw is just w
Note the difference between $a \in \Sigma$ and $a \in \Sigma^*$
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 –a– is clear

Operation on words; Language

Operations on words

$$u \in \Sigma^*,\ v \in \Sigma^* \Rightarrow u \cdot v \in \Sigma^*,$$ concatenation

$$u \in \Sigma^*,\ n \in \mathbb{N} \Rightarrow u^n \in \Sigma^*,$$ repetition

$$u \in \Sigma^* \Rightarrow u^R \in \Sigma^*,$$ reverse

Inductive definitions of concatenation, repetition and reverse

$$\begin{array}{rcl} u \cdot \lambda & = & u \\ u \cdot (vs) & = & (u \cdot v)s \end{array} \qquad \begin{array}{rcl} u^0 & = & \lambda \\ u^{k+1} & = & u^k \cdot u \end{array} \qquad \begin{array}{rcl} \lambda^R & = & \lambda \\ (ws)^R & = & s(w^R) \end{array}$$

We write concatenation $u \cdot v$ as uv

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

Examples

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

$L_2 = \{w \in \{a, b\}^* \mid w = w^R\}$

Examples of languages

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

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. $L_4 = \{a^n \mid n \in \mathbb{N} \text{ is prime}\}$
5. $L_5 = \{n \mid n \text{ denotes an integer number}\}$
6. $L_6 = \{e \mid e \text{ is a well-formed arithmetical expression}\}$
7. $L_7 = \{P \mid P \text{ is a syntactically correct Java program}\}$
8. $L_8 = \{S \mid S \text{ is a grammatically correct English sentence}\}$

Operations on languages

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

$$L_1 \cup L_2 \quad L_1 L_2 \quad L^*$$

$$\begin{array}{rcl}
L_1 \cup L_2 & = & \{w \mid w \in L_1 \text{ or } w \in L_2\} \\
L_1 L_2 & = & \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\} \\
L^0 & = & \{\lambda\} \\
L^{n+1} & = & L^n L \\
L^* & = & \bigcup_{n \in \mathbb{N}} L^n = L^0 \cup L^1 \cup L^2 \cup \dots \\
 & \neq & \{w^n \mid w \in L,\ n \in \mathbb{N}\}
\end{array}$$

Regular expressions and languages over Σ

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

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

For general Σ the regular expressions over Σ are generated by

$$\boxed{\mathtt{rexp}_\Sigma ::= \emptyset \mid \lambda \mid \mathtt{s} \mid (\mathtt{rexp}_\Sigma\ \mathtt{rexp}_\Sigma) \mid (\mathtt{rexp}_\Sigma + \mathtt{rexp}_\Sigma) \mid (\mathtt{rexp}_\Sigma^*)}$$

with $s \in \Sigma$
This means $\emptyset \in \mathtt{rexp}_\Sigma$, $\lambda \in \mathtt{rexp}_\Sigma$, and $s \in \mathtt{rexp}_\Sigma$ for $s \in \Sigma$and

$$e_1, e_2 \in \mathtt{rexp}_\Sigma \Rightarrow (e_1 + e_2) \in \mathtt{rexp}_\Sigma$$
$$e_1, e_2 \in \mathtt{rexp}_\Sigma \Rightarrow (e_1 e_2) \in \mathtt{rexp}_\Sigma$$
$$e \in \mathtt{rexp}_\Sigma \Rightarrow (e^*) \in \mathtt{rexp}_\Sigma$$

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

We economize on brackets

$$\boxed{\texttt{rexp}_\Sigma ::= \emptyset \mid \lambda \mid \texttt{s} \mid (\texttt{rexp}_\Sigma\ \texttt{rexp}_\Sigma) \mid (\texttt{rexp}_\Sigma + \texttt{rexp}_\Sigma) \mid (\texttt{rexp}_\Sigma^*)}$$

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

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

Regular languages

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

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

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

Examples

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

- 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

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