Fundamentele Informatica 1 (I&E)

najaar 2015

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi1ie/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs(dot)nl

college 1, dinsdag 27 oktober 2015

Introduction

1.4 Languages

2.1 Finite Automata: Examples and Definitions

Fundamentele Informatica 1 (I&E) — najaar 2015
`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi1ie/`

- colleges
dinsdag + vrijdag 27 okt - 8/11 dec, zaal 408,
hoorcollege 11.15–13.00, Rudy van Vliet,
werkcollege 13:45–15:30, Jeroen van den Heuvel

- boek: John C. Martin, Introduction to Languages and the Theory of Computation, 4th edition

- tentamens: dinsdag 7 januari 2016, 10:00–13:00
maandag 22 februari 2016, 10:00–13:00

- huiswerkopgave: 0.4 punt bonus voor tentamen

- laatste jaar

Computer receives input, performs ‘computation’, gives output

- Given instance of Nim. Who wins?

- Given sequence of numbers. Sort

- Given edge-weighted graph.
  Give shortest route from A to B

Dealing with languages / sets of instances

1. Abstract machines to accept or to recognize languages

2. Grammars to generate languages

3. Expressions to describe languages

reg. languages	FA	reg. grammar	reg. expression
determ. cf. languages	DPDA		
cf. languages	PDA	cf. grammar	
re. languages	TM	unrestr. grammar	

1.4 Languages

Natural languages, programming languages, formal languages

Alphabet Σ is finite set of symbols: $\{a, b, c\}$, or $\{0, 1\}$

A string x over an alphabet: $ababb$, or 001110, or Λ
finite

Σ^*: all strings over alphabet Σ

$\{a, b\}^*$: all strings over $\{a, b\}$:
$\{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$
infinite set of finite strings
canonical order

A language over an alphabet:

Σ^*: all strings over alphabet Σ
$\{a, b\}^*$: all strings over $\{a, b\}$
$\emptyset$
$\{\Lambda\}$
$\{\Lambda, a, aab\}$
Pal: $\{x \in \{a, b\}^* \mid x$ is palindrome$\}$
$\{x \in \{a, b\}^* \mid n_a(x) > n_b(x)\}$
$\{x \in \{a, b\}^* \mid |x| \geq 2$ and x begins and ends with $b\}$
$\{x \in \{a, b\}^* \mid |x| = 2^k$ for some $k \geq 0\}$

Different 'complexities'

The language of legal algebraic expressions with a, $+$, $*$ and $($, $)$, e.g., a, $a + a * a$, $(a + a * (a + a))$

Balanced: balanced strings of parentheses, e.g., Λ, ()(()), ((((()))))

The language of legal Java programs

Concatenation of strings: xy

Λ, length, associative

prefix, suffix, substring

set operations on languages:

$L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$, complement L'

concatenation of languages:

$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$

$\{a, aa\}\ \{\Lambda, b, ab\} = \ldots$

$L_1\ \{\Lambda\} = \ldots$

$\{\Lambda\}\ L_1 = \ldots$

$L_1\ \emptyset = \ldots$

exponential notation:

$a^k = \underbrace{aa \ldots a}_{k \text{ times}} \qquad x^k = \underbrace{xx \ldots x}_{k \text{ times}}$

$L^k = \ldots$

$\Sigma^k = \ldots$

$k = 2, 3, 4, \ldots, 0$

$L^+ = \bigcup_{k \geq 1} L^k = L \cup L^2 \cup L^3 \cup L^4 \cup \ldots$

$L^* = \bigcup_{k \geq 0} L^k = L^0 \cup L \cup L^2 \cup L^3 \cup L^4 \cup \ldots$

Kleene star

still finite strings

Combination of union, concatenation, Kleene star:

$L_1 \cup L_2 L_3^* = \ldots$

$(L_1 \cup L_2) L_3^*$

$L_1 \cup (L_2 L_3)^*$

$(L_1 \cup L_2 L_3)^*$

Exercise.

Let

$$\begin{aligned} L_1 &= \{\Lambda, a\} \\ L_2 &= \{ab\} \\ L_3 &= \{a\} \end{aligned}$$

Determine

(a) $L_1 \cup L_2 L_3^*$

(b) $(L_1 \cup L_2) L_3^*$

(c) $L_1 \cup (L_2 L_3)^*$

(d) $(L_1 \cup L_2 L_3)^*$

Description of languages:

by formula: $L_1 = \{ab, bab\}^* \cup \{b\}\{ba\}^*\{ab\}^*$

by property: $L_2 = \{x \in \{a,b\}^* \mid n_a(x) > n_b(x)\}$

Notation for generating languages vs.

algorithm/machine for recognizing/accepting language

Exercise.

Let

$$L = \{ab^i a^j b^k \mid j > i + k\}$$

Give the first five elements of L in canonical order.

2.1 Finite Automata: Examples and Definitions

Accepting a language
receives input symbol by symbol

Finite automaton (FA) = finite state machine
finite number of states
initial state
accepting states, nonaccepting states
transition function

Simple machine

Example 2.1.

A finite automaton for accepting

$L_1 = \{x \in \{a,b\}^* \mid\ x \text{ ends with } aa\}$

Example 2.3.

A finite automaton for accepting

$L_2 = \{x \in \{a, b\}^* \mid$ x ends with b
and does not contain the substring $aa\}$

Example 2.5.
A finite automaton for accepting
$L_3 = \{x \in \{a, b\}^* \mid \ x$ contains the substring $abbaab\}$

string search algorithm, like Knuth-Morris-Pratt
(Datastructures course?)