Formal Languages And Compilers
Exercises

Christian Rossi

Academic Year 2023-2024

Abstract

The lectures are about those topics:

- Definition of language, theory of formal languages, language operations, regular expressions, regular languages, finite deterministic and non-deterministic automata, BMC and Berry-Sethi algorithms, properties of the families of regular languages, nested lists and regular languages.
- Context-free grammars, context-free languages, syntax trees, grammar ambiguity, grammars of regular languages, properties of the families of context-free languages, main syntactic structures and limitations of the context-free languages.
- Analysis and recognition (parsing) of phrases, parsing algorithms and automata, push down automata, deterministic languages, bottom-up and recursive top-down syntactic analysis, complexity of recognition.
- Syntax-driven translation, direct and inverse translation, syntactic translation schemata, transducer automata, and syntactic analysis and translation. Definition of semantics and semantic properties. Static flow analysis of programs. Semantic translation driven by syntax, semantic functions and attribute grammars, one-pass and multiple-pass computation of the attributes.

The laboratory sessions are about those topics:

- Modellization of the lexicon and the syntax of a simple programming language (C-like).
- Design of a compiler for translation into an intermediate executable machine language (for a register-based processor).
- Use of the automated programming tools Flex and Bison for the construction of syntax-driven lexical and syntactic analyzers and translators.

Contents

Chapter 1

Exercise session I

Exercise 1

Given two regular expression:

$$R_1 = ((2b)^* c)^* \qquad R_2 = (c^*(2b)^*)^*$$

Check if they are equal. If they are not give a counterexample.

Answer of exercise 1

It is possible to see that R_1 and R_2 are not equivalent because the character c is in a different position and can be found multiple times in R_2, while in R_1 is found exactly one time. An example can be ab. This string is included in the second language, but not in the first one.

Exercise 2

Given the regular expression:

$$R_1 = (a|\varepsilon)^+(ba|bab)^*$$

check if it is ambiguous.

Answer of exercise 2

First, we enumerate all the characters in the regular expression, obtaining:

$$R_1 = (a_1|\varepsilon)^+(b_2a_3|b_4a_5b_6)^*$$

and now we try to come up with an ambiguous string to prove that the regular expression is ambiguous. A regular expression is considered ambiguous if there is a string which can be matched by more than one way from the regular expression. For instance, we can have the string a_1 can be generated multiple times selecting the ε $n-1$ times. This proves that the regular expression is ambiguous.

Exercise 3

Given two regular expressions:

$$R_1 = a((b|bb)a)^+ \qquad R_2 = (ab)^*ba$$

Define the quotient language $L = R_1 - R_2$.

1. Write the three shortest strings of the language L.
2. Write a regular expression that defines the language.

Answer of exercise 3

First, we enumerate all the characters in the regular expressions, obtaining:

$$R_1 = a_1((b_2|b_3b_4)a_5)^+$$
$$R_2 = (a_1b_2)^*b_3a_4$$

The a_1 is surely a prefix for every string in the language, and all the strings have a_5 as a suffix. The shortest strings of this language are: *aba*, *ababa* and *abababa*.

For the second regular expression we have that every string generated starts with *ab* and have a single *ba* as a suffix. The shortest strings of this language are: *ba*, *abba* and *abababa*.

We can see that all the strings that have the suffix *aba* or have at least two *bb* are certainly in L.

1. Now, we can see that the three shortest strings are: *aba*, *ababa*, and *abbaba*.
2. The regular expression is:

$$L = \{(a(b|bb))^*aba\} \cup \{(a(b|bb))^*abba(a(b|bb))^+abba\}$$