Summary

This document collects the various personal notes from the course "Formal Languages and Compilers" (2012), prof. Silvano Rivoira. The LaTeX source code is available in a dedicated GitHub repository.

Contents

Part I

Formal Languages

Chapter 1

Classification (FLC)

1.1 Grammars

A grammar is a 4-tuple $G = (N, T, P, S)$ where:

N alphabet of non-terminal symbols;

T alphabet of terminal symbols:

- $N \cap T = 0$ (two alphabets are disjoined),
- $V = N \cup T$ (alphabet of the grammar);

P finite set of rules (productions);

S start (non-terminal) symbol.

A language produced by $G = (N, T, P, S)$ is:

$$L(G) = \{w | w \in T^*; S \Rightarrow^* w\}$$

Grammars that produce the same languages are said "equivalent".

1.2 Types of Grammars

Type 0 grammars (phase-structure)

$$P = \left\{\alpha \to \beta \middle| \alpha \in V^+; \alpha \notin T^+; \beta \in V^*\right\}$$

Type 1 grammars (context-sensitive)

$$P = \left\{\alpha \to \beta \middle| \alpha \in V^+; \alpha \notin T^+; \beta \in V^+; |\alpha| \leq |\beta|\right\}$$

Type 2 grammars (context-free)

$$P = \left\{A \to \beta \middle| A \in N; \beta \in V^+\right\}$$

1.3 Linear Grammars

$$P = \left\{ A \to xBy, A \to x \middle| A, B \in N; x, y \in T^{+} \right\}$$

Type 3 grammars (right/left - linear)

- Right-Linear grammars

$$P = \left\{ A \to xB, A \to x \middle| A, B \in N; x \in T^{+} \right\}$$

- Left-Linear grammars

$$P = \left\{ A \to Bx, A \to x \middle| A, B \in N; x \in T^{+} \right\}$$

Type 3 grammars (right/left - regular)

- Right-Regular grammars

$$P = \{A \to aB, A \to a | A, B \in N; a \in T\}$$

- Left-Regular grammars

$$P = \{A \to Ba, A \to a | A, B \in N; a \in T\}$$

Chapter 2

Regular Languages (RL)

Chapter 3

Context-Free Languages (CFL)

Chapter 4

Turing Machines (TM)

Part II

Compilers

Chapter 5

Compiler Structure (CS)

Chapter 6

Lexical Analysis (LA)

Chapter 7

Syntax Analysis (SA)

Chapter 8

Syntax-Directed Translation (SDT)

Chapter 9

Semantic Analysis and Intermediate-Code Generation (SA/ICG)