Sept 17th

Introduction to language theory and compiling

Gilles GEERAERTS
Moosulu BALACHANDER
Nathan SASSOLAS

Université Virtuelle.
↓
Lecture notes
Podcasts.

Evaluation → Escrom January/August /12

→ Project : write a compiler! /8 no second session!

You must achieve at least $\frac{1}{3}$ on test.

Schedule → U.V.

Language Theory

What is a language ?

Word

language = set of words.

Rules

Colorless green ideas sleep furiously

Noam Chomsky.

Grammar: ✓ → Syntax

Meaning: ✗ → semantics

Compiler? → check the syntax.

Semantics ?
= "what the program should do"

Formal definition of language

Alphabet : finite set of symbols

$\{0, 1\}$ ✓

$\{a, b, \ldots, z, A, B, \ldots, Z\}$ ✓

ASCII set ✓

IN ✗

Word : finite sequence of symbols on a given alphabet

alphabet $\Sigma = \{0, 1\}$

0 110 101

⚠ $\varepsilon \notin \Sigma$

Empty word / sequence ε

Language : set of words

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

$L = \{0, 1, 00, 11, 01, 10\}$ finite.

$L = \{\varepsilon, 0, 00, 000, 0000, \dots\}$ infinite

1. The set L_{Cid} of all non-empty words on Σ_C (see example 1.7) that do not begin with a digit, is a language. It contains all valid C identifiers (variable names, function names, etc) and all C keywords (`for`, `while`, etc).

   abi2 ✓
   _c3 ✓
   9ab ✗
   $\Sigma_C = \{a, b, \ldots z, A, B, \ldots Z, 0 \ldots 9, \_\}$

2. The set L_{odd} of all non-empty words on $\{0, 1\}$ that end with a 1 is a language. It contains all the binary encodings of odd numbers.

3. Similarly to the previous example, the set $L_()$ of all words on $\Sigma = \{(,)\}$ which are well-parenthesised, i.e., s.t. each closing parenthesis matches a previously open and still pending parenthesis, and each open parenthesis is eventually closed. For example $(()()) \in L_()$, but neither)((nor (() do.

   This language is also known as the *Dyck language*, named after the German mathematician Walter VON DYCK (1856–† 1934). It is mainly of theoretical interest: we will rely on it several times later to discuss the kind of formalism we need to recognise languages of expressions that contain parenthesis, such as the language L_{alg} defined in the next item:

1. $w \overset{?}{\in} L_{Cid}$ → infinite

No memory needed!

$w = ab12$

↑ $\in \{0, \ldots, 9\}$ → no → ✓

2. $w \overset{?}{\in} L_{odd}$ $\quad w = 01011$

↑ ... ↑ $c_0 = 1$ ✓

$L_{()}$

(→ +1

) → −1

$\neq 0$

(( ) ( )) ✓

1 2 1 2 1 0

$= 0$ ✓

(( ) X

1 2 1

$\neq 0$ X

( )) X

1 0 −1

< 0 X

We need memory: Counter.

More examples...

4. The set L_{alg} of all algebraic expressions that use only the x variable, the $+$ and $*$ operators and parenthesis, and which are well-parenthesised, is a language on the alphabet $\Sigma = \{(,), x, +, *\}$. For instance $((x+x)*x)+x$ belongs to this language, while $)(x+x$ does not, although it is a word on Σ.

5. The set L_C of all syntactically correct C programs is a language.

6. The set L_{Cterm} of syntactically correct C programs that terminate whatever the input given by the user is a language.

5. The set L_C of all syntactically correct C programs is a language.

1 Word = 1 program

Language = all programs whose syntax is correct.

Checking the syntax of a C program $\overset{?}{\Leftrightarrow}$ prog $\in L_C$

Operations on words

Concatenation of words

$$w = w_1 w_2 \dots w_n$$

$$v = v_1 v_2 \dots v_l$$

$$w \cdot v = w_1 \dots w_n v_1 \dots v_l$$

↑ concatenation

$\varepsilon \cdot w = w = w \cdot \varepsilon$

$w^n = \underbrace{w \cdot w \cdot \dots \cdot w}_{n \text{ times}}$

Operations on languages

Concatenation

$$L_1 \cdot L_2 = \{ w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2 \}$$

$L_1 = \{aa, b\}$ $\qquad$ $L_2 = \{c, dd\}$

$$L_1 \cdot L_2 = \{aac, aadd, bc, bdd\}$$

$$L^n = \{ w_1 . w_2 \dots w_n \mid w_1, w_2, \dots, w_n \in L \}$$

$$L = \{a, b\}$$

$$L^2 = \{ aa, ab, ba, bb \}$$

Kleene closure

$$L^* = \{ w_1 w_2 \dots w_n \mid n \neq 0, \; w_1, w_2 \dots w_n \in L \}$$

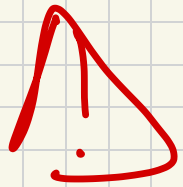
$$L = \{a\} \qquad L^* = \{ \varepsilon, a, aa, aaa, \dots \}$$

$L = \phi$ → empty language

it contains no word.

$L = \{\varepsilon\}$ → non-empty language!

it contains 1 word which is empty.



$$\emptyset \cdot L = \emptyset$$

$$L_1 \cdot L_2 = \{ w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2 \}$$

$L_1 = \emptyset$

$w_1 \in L_1$, $L_1 = \emptyset$

does not exist

$\nexists$

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L$$

Regular languages

Definition 2.1 (Regular languages)**.** Let us fix an alphabet Σ. Then, a language L is regular iff:

1. either $L = \emptyset$;
2. or $L = \{\varepsilon\}$;
3. or $L = \{a\}$ for some $a \in \Sigma$;
4. or $L = L_1 \cup L_2$;
5. or $L = L_1 \cdot L_2$;
6. or $L = L_1^*$

where L_1 and L_2 are regular languages on Σ.

Base cases

Induction