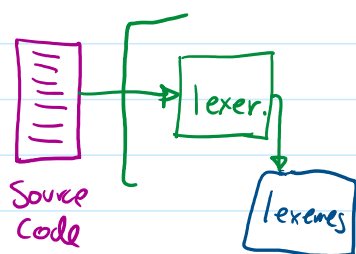


4 Syntax and Regular Expressions

Wednesday, February 5, 2025

12:43 PM



- from Linguistics and Mathematics.

- Orthography.- How to write words.?

apple manzana pomme W Lg ID

that gives a Dictionary.



In a programming language: keywords,

- Syntax.- How to write sentences.

"has jump apple dog green"

Syntax has rules.

In programming languages also

- Semantics: What do sentences mean?

"colorless green ideas sleep furiously"

Adj

Adj

nom

verb

adverb.

N. Chomsky

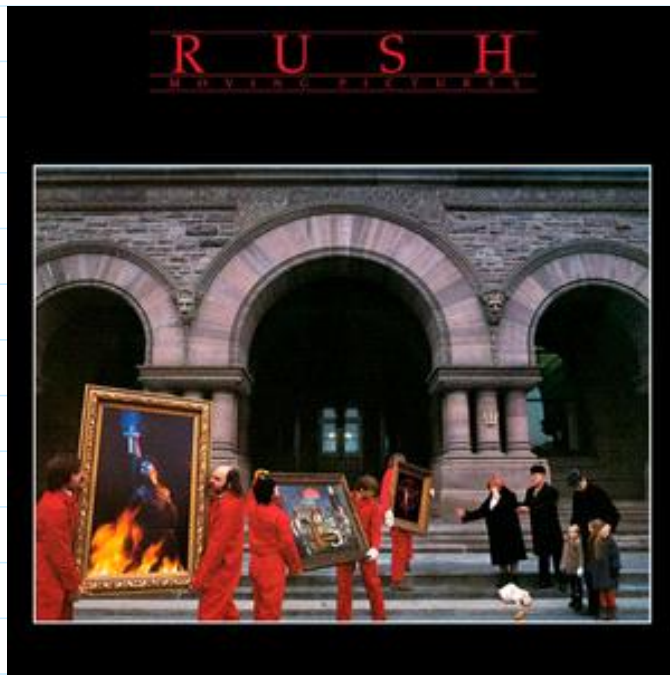
"the chickens are ready to eat"



or



"Moving Pictures"



- Pragmatics

How meaning changes over time

"the container has to handle an unreliable cloud"

• Back to Orthography & Syntax

What are the words of our programming Language?

symbols: ASCII

- keywords

- operators

- Literals - constant values 'apple' 17

- identifiers - named entities - variable, function, class, type, module, namespace, template

• identifiers and literals are potentially an infinite collection.

We borrow from Mathematics
"Mathematical Languages"

S. Kleene
E. Post

DEF: Alphabet:- a finite set of symbols.

e.g. $\Sigma = \{a, b\}$

Language a set of sequences (strings) (words)
made of symbols in the language.

e.g. $L = \{a, aa, aba, bba, bbb, bab\}$

Problem (1) The Specification Problem

How to precisely describe a Language?

e.g. $L = \{a, ab, aba, abb, \dots\}$
 $?, ?, ?$

Problem (2) :- the Recognition Problem

given a word w and a language L

Is w in L ?

$w \in L$?

- Answer to (1) : Specification: Regular Expressions
built from Σ plus $|$ $*$ $($ $)$

DEF: Regular Expressions.

- a word w is a reg. ex.
- if r_1, r_2 are reg. ex. then
 - $r_1 r_2$
 - $r_1 | r_2$
 - r_1^*are also reg. ex.

A regular expression specifies a language

- w $\{w\}$
- $r_1 r_2$ $\{w_1 w_2 \mid w_1 \in r_1 \wedge w_2 \in r_2\}$
- $r_1 | r_2$ $\{w \mid w_1 \in r_1 \vee w_2 \in r_2\}$
- r^* $\{w \mid w \text{ is zero or more repetitions of } r\}$
 $\{ \dots r \dots r \dots r \dots \}$

- r^* Kleene star. $\{w \mid w \text{ is zero or more repetitions of a word in } r\}$

E.g.s $\Sigma = \{a, b\}$

- $(a|b)b = \{ab, bb\}$
 - $a : \{a\}$
 - $b : \{b\}$
 - $a|b : \{a, b\}$
- $(ab)^*$
 - $= \{\epsilon, ab, abab, ababab, \dots\}$

★ an infinite set is ^{clearly} specified by finite means.

- $(a|b)^* = \{\epsilon, a, b, aa, ab, bb, ba, \dots\}$
 - $a : \{a\}$
 - $b : \{b\}$
 - $a|b : \{a, b\}$
- $(aa|b^*)a = \{aaa, a, ba, bba, bbba, bbbba, \dots\}$
 - $aa : \{aa\}$
 - $b^* : \{\epsilon, b, bb, bbb, \dots\}$
 - $(aa|b^*) : \{aa, \epsilon, b, bb, bbb, \dots\}$

- $r? \equiv (r|\epsilon)$ "to r or not to r "
- $r+ \equiv r(r^*)$ "one or more repetitions of r "

- Regular Expressions and Programming languages

$\Sigma = \text{ASCII keyboard characters.}$

$L_{c++} = \text{the set of all c++ programs.}$ ⊗

Note: we cannot use reg.ex. to specify a programming language.

Note: we cannot use reg.ex. to specify a programming language.

Why? example language.

$\{a^n b^n \mid n \in \mathbb{Z}^+\} = \{ab, aabb, aaabbb, aaaa bbbb, \dots\}$

consider matching brackets

$\{() [] \}$

$\{ ([]) \}$

- Famous Stack Overflow Question:

What is a regular expression to validate HTML?

$\langle p \rangle \dots \langle /p \rangle$
 $\langle h3 \rangle \dots \langle /h3 \rangle$

closed

- RegEx are good enough to describe "tokens"

token :- a category of atomic strings

lexeme :- an instance of a token.

tokens :- $\begin{cases} \text{keywords} \\ \text{identifiers} \\ \text{literals:} \begin{cases} \text{integers} \\ \text{decimals.} \end{cases} \end{cases}$

lexemes :- while foo 3.7

- Describing tokens using RegEx.

$\Sigma = \text{ASCII}$

- Shorthands:

$[0-9] \equiv (0|1|2|3|4|5|6|7|8|9)$

$[A-Z] \equiv (A|B|C| \dots |Y|Z)$

$[a-z] \equiv (a|b|c| \dots |y|z)$

$[A-Za-z] \equiv ([A-Z]| [a-z])$

- RegEx for keywords

- Regex for keywords

C++ (if | while | for | do |)

- Regex for operators

C++ (+ | * | - | % | / | == | | | ...)

- Regex for integer Literals.

$$(+|-)?([0-9])^+ = \left\{ \begin{array}{cccccc} 1 & 2 & 3 & 0 & 00 & 003 \\ 27 & -5 & +7 & & +101 & \dots \\ 235 & & -001 & & -00 & \dots \end{array} \right\}$$

- Regex for decimal Literals.

$$(+|-)?([0-9])^+ \cdot ([0-9])^+ = \left\{ \begin{array}{ccc} 1.5 & 3.7 & 1.0 \\ 111.33 & & \end{array} \right\}$$

~~1.~~ ~~.03~~

- Regex for identifiers.-

Forhan $([A-Z])([A-Z] | [0-9])^*$
A B2 K7F APPLE3

Pascal $([A-Za-z])([A-Za-z] | [0-9] | -)^*$
var1 point_2D init_X Pascal Case
camel Case

Python $([A-Za-z] | -)([A-Za-z] | [0-9] | -)^*$
__init__ __main__ this_is_snake_case
var1_ _ _ _ _

- RegEx for scientific notation

e.g. 3.7e-5 -8.9e7 123e3 +12.7e-8 -5e+7

$$(+|-)?([0-9])^+ (\cdot [0-9]^+)? e (+|-)?([0-9])^+$$

- RegEx can also be used to describe other string formats.
 - phone numbers
 - e-mails
 - URL
 - Serial numbers
 - Credit Card numbers
 - Licence plates
 - etc



— • —
EOF