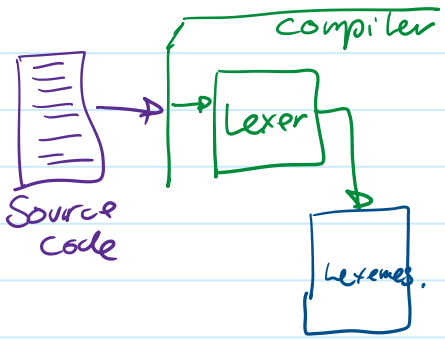
4 Syntax and Regular Expressions

Wednesday, September 4, 2024 12:00 PM



from Linguistics:

- Orthography:- How to write words?

  apple manzana pomme ليمون

  this gives a Dictionary.

  In a programming Language : keywords.

  for while if do else.

- Syntax:- How to write sentences?

  "has jump apple dog green"

  syntax has rules.

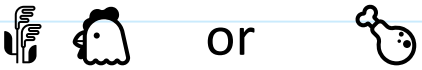
  in programming languages. also

- Semantics:- What do sentences mean?

  "colorless green ideas sleep furiously" N. Chomsky

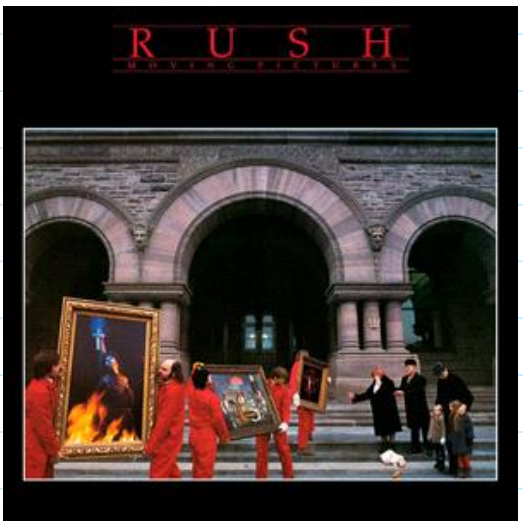
  Adj Adj noun verb adverb.

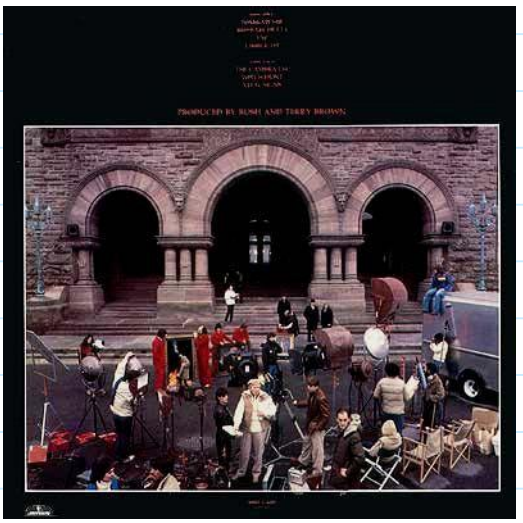
  "The chickens are ready to eat"

   or

  "Moving Pictures"

"Moving Pictures"





- Pragmatics:-
  How meaning changes over time.
  "the program has to handle an unreliable cloud"

• Syntax: for a programming language.
  - How to write words?
    Ascii
    - keywords
    - Identifiers:- names for things
      variable, functions, classes, namespaces ..... templates
    - Literals:-
      constant values: strings
      integer floats

To Answer this question:
"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 alphabet

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

1.- Specification Problem:
How to precisely describe a language?
$L = \{ a, ab, aba, abb, \ldots \}$??

2.- Recognition Problem
given a word w and a Language L
answer: $w \in L$?

- Answer to #1 - specification: Regular expressions
built from: Σ plus | * ( )

DEF Regular Expressions:
- a word w is a regular expression
- if $r1, r2$ are regular expressions
  - $r1\,r2$
  - $r1 \mid r2$
  - $r1^*$

} are also regular expressions.

A regular expression specifies a language.

- w $\{w\}$
- $r1\,r2$ $\{w_1 w_2 \mid w_1 \in r_1 \wedge w_2 \in r_2\}$
- $r1 \mid r2$ $\{w \mid w \in r_1 \vee w \in r_2\}$
- r^* $\{w \mid w$ is zero or more repetitions of a word in $r\}$

Kleene Star.

E.g: $(a \mid b)\,b = \{ab, bb\}$

a : $\{a\}$
b : $\{b\}$

E.g.

a : {a}

b : {b}

(a|b) : {a, b}

Eg: $(ab)^*$ = {Λ, ab, abab, ababab, ... }

Λ ↔ empty string.

ab : {ab}

⭐ an infinite language is specified by finite means.

Eg: $(a|b)^*$ = {Λ, a, b, aa, ab, aabba, ... }

a : {a}

b : {b}

a|b : {a, b}

Eg: $(aa|b^*)a$ = {a, ba, aaa, bba, bbba, bbbba, ... }

aa : {aa}

b^* : {Λ, b, bb, bbb, bbbb, ...}

- Shorthands.

  $r? \equiv (r|\Lambda)$ "to r, or not to r"

  $r+ \equiv r(r^*)$ "one or more r's"

* Regular expressions & programming Languages.

  Σ = ASCII keyboard characters.

Note: you cannot use regular expressions to describe a programming language.

L_{C++} = the set of all valid C++ programs.

Why?

consider simple example:

$\{a^n b^n \mid n \in \mathbb{Z}^+\} = \{ab, aabb, aaabbb, aaaabbbb, \ldots\}$

consider matching brackets.

{ ( ) [ ] } { ([ )] }

Famous stack overflow Question:

What Regular expression to validate HTML?

closed.

<a> .....

<h3> ..... </h3>

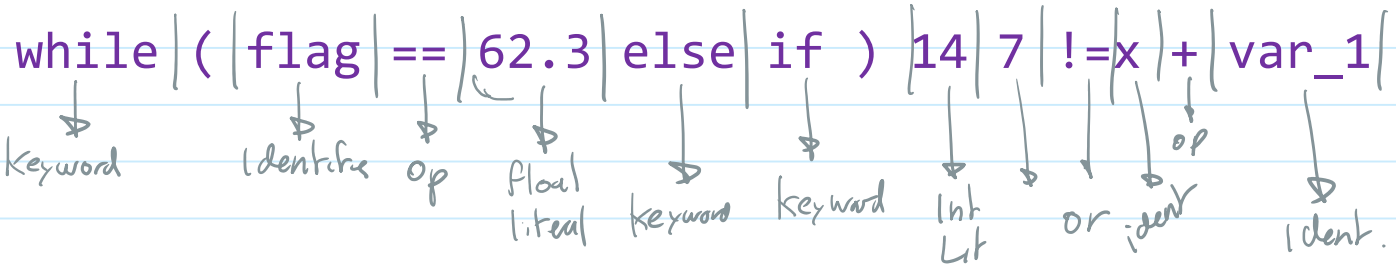
- RegEx are good enough to describe "tokens"

  token .- a category of atomic strings.

  lexeme.- an instance of a token

e.g. token → identifiers, integers, decimals.

e.g. lexemes foo 5 0.1



- Describing tokens using RegEx

  Σ = keyboard symbols.

- Shorthands

  [0-9] ≡ (0|1|2|3|4|5|6|7|8|9)

  [A-Z] ≡ (A|B|C|D|.....|Y|Z)

  [a-z] ≡ (a|b|c|d|....|y|z)

  [A-Za-z] ≡ ([A-Z]|[a-z])

- Regex for keywords

  e.g C++ (if | while | for | do | else | .....)

- Regex for operators

  (+ | * | - | / | % | = | == | <= | >= | .....)

- Regex for integer literals.

  (+|-)?([0-9])$^{+}$ = { 1 2 0 00 001, 27 -5 +10 +00, 235 -12 -001 -00 }

- RegEx for identifiers.

Fortran ([A-Z])([A-Z]|[0-9])*

A A2 K7F APPLE3

Pascal ([A-Za-z])([A-Za-z]|[0-9]|'_')*

var1 point_2D initX PascalCase

Python ([A-Za-z]|'_')([A-Za-z]|[0-9]|'_')*

__init__ __main__ this_is_snake

var_1 ____ _____

- RegEx for scientific notation

- RegEx for scientific notation
  eg. $3.7e^{-5}$ $-8.9e7$ $123e^{3}$ $+12.7e-8$

  `(+|-)?[0-9]+('.'[0-9]+)?'e'(+|-)?[0-9]+`

- RegEx can also be used to describe other string formats
  - phone numbers
  - SSN
  - URL's
  - Serial numbers
  - e-mail addresses
  - Licence plates
  - etc..

EOF