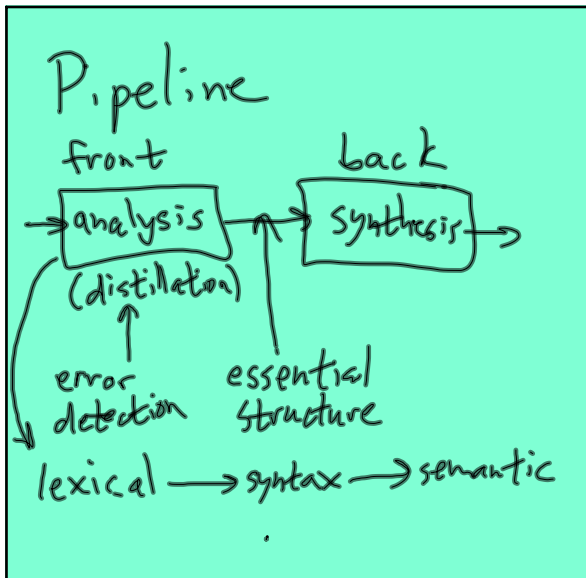
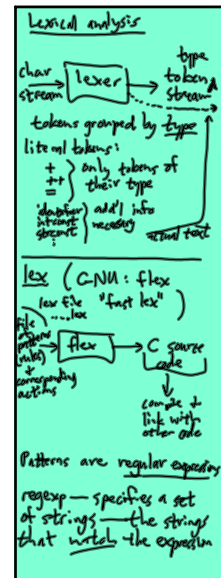




Jan 19-11:02 AM



Jan 19-11:07 AM

Fix an alphabet
(finite set of symbols)
ASCII char set, e.g.
Alphabet Σ fixed.

A string over Σ is any finite seq of symbols from Σ . A language over Σ is a set of strings over Σ .

Ex: language of identifiers

A regex is an expression built from atomic regexes via certain operators.

Atomic regexes:

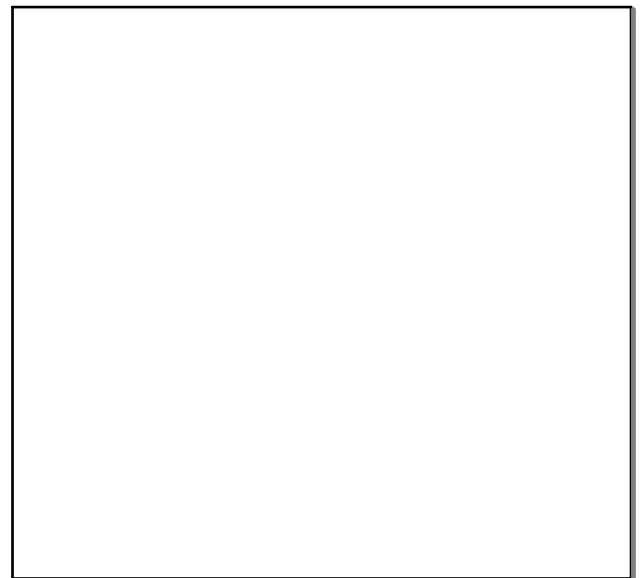
char constants: $a, \dots, z$
specifies the language $\{a\}$

string constants: "main"
lang: $\{main\}$

operators:

concatenation: RS
matches any string that concatenates strings matching R with strings matching S

Jan 19-11:17 AM



Jan 19-11:24 AM

RS

lang: $\{xy : x \text{ matches } R, y \text{ matches } S\}$

| ("or")

R|S matches anything matching R or matching S (or both)

R* ("Kleene closure")

match the concatenation of one or more strings each matching R.

E.g. $(0|1)(0|1)(0|1)$

000, 001, 010, 011, 100, 101, 110, 111

$(0|1)(0|1)^*$

the empty string (null string)

ϵ denotes the null string

(ϵ is not a literal symbol — not part of alphabet)

$\epsilon x = x = x \epsilon$ for any string x.

Jan 19-11:24 AM

$(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)^*$

[0123456789]

shorthand for 0|1|2|...|9

character class

[...] match any single char from between brackets

IMPORTANT: don't confuse (...) with [...]

group / regex char class

(a|abc) [a|abc]

matches either 'a' or 'abc' matches 'a' or 'abc' or 'a|abc' on 'a|abc' or 'abc|a|abc'

[0-9] same as [0123456789]

Jan 19-11:36 AM

$[0-9a-z]$
 $[0-9]$
 $\uparrow$
 $[0-9 \setminus a-z]$
 capital 6
 $[0-9]$
 A first in class means
 complement: match any
 single char not in the
 class.
 $[^]$?
 • (period) matches
 any single char
 (by default)
 $[0-9][0-9]^*$
 $[0-9]^*[0-9]$
 $[0-9]^+$ integer constant
 "one or more"
 $R^+ \equiv RR^*$

Jan 19-11:45 AM

precedence of operators:
 highest: $[], ()$
 $[()]$
 matches either "(" or ")"
 $*, +, ?$
 unary, postfix, zero or one
 postfix
 $\downarrow$
 concatenation
 RS^* means
 $R(S^*)$
 $|$ (or)
 $RS|T$
 grouped as
 $(RS)|T$
 identifier:
 $[A-Za-z_][A-Za-z_0-9]^*$
 optionally signed int const
 $[-+]?[0-9]^+$
 optional sign
 Literal $\setminus ?, \setminus *, \setminus []$

Jan 19-11:54 AM

$[[]]$
 $[\\]$ matches
 $\\$ " "
 flex compiles a
 series of regexps into
 C code that detects
 matching parts of the
 input stream
 Lex program:
 preamble
 %%
 rules
 %%
 utilities
 C code
 lex prog $\rightarrow$ flex $\rightarrow$ C file
 C file contains: preamble, rules, utilities
 A rule looks like
 regexp $\rightarrow$ action
 exp. action C code
 exp. action { block of }
 exp. action code

Jan 19-12:04 PM