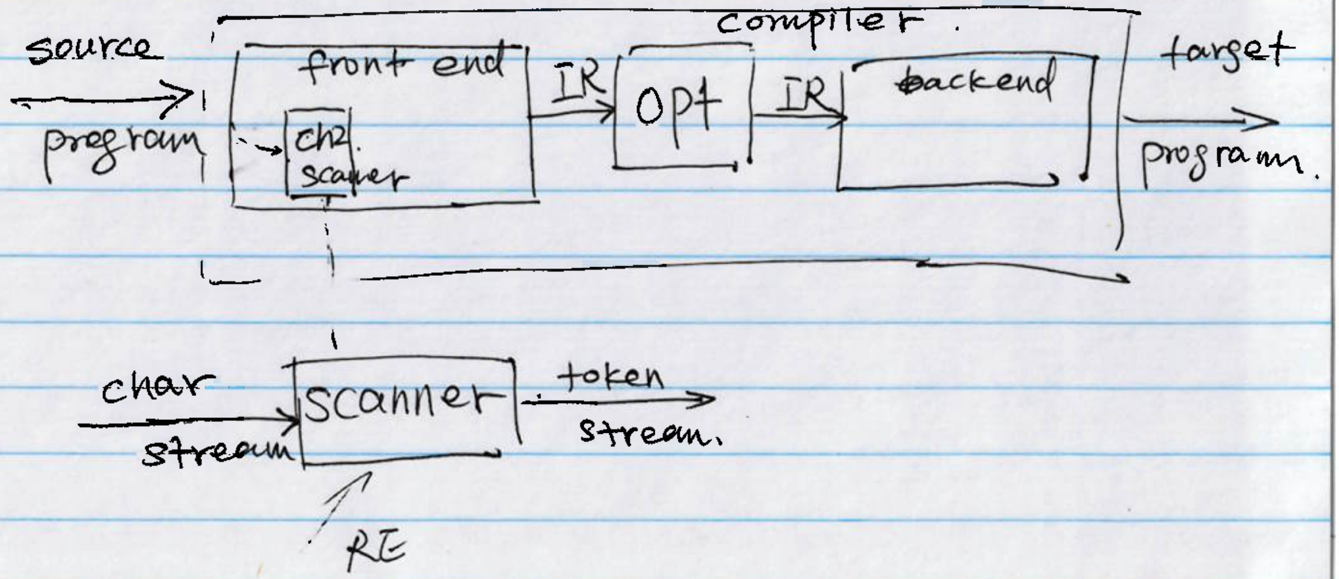
01/15/2025 ch2. scanner.

1.



2. tokens.

token : smallest element of a program which are identified by the compile ~~thatt~~ are meaningful .

different PLs. have different type of tokens. some. are. similar .

3. 2.1 Token. examples.

~~3.1~~. Java :

⊔⊔⊔⊔⊔int a = b + c ;

keywords : int. if else.

identifiers : names. of variable. class. method.

literals: constant.

1   3.14.   true   '1'   "1"

null

operator: +   =   ++   --   *=

separator.

- separate/punctuate other tokens.

  ,   ;   .

- delimiters.

  (   )   {}   [ ]

12   1 2   "1␣2"

x >= y   x >>> 3   unsigned right shift.

f123   f␣123

Note: about tokens in Java.

1. Java always tries to construct the longest token

2. white space: space. tab. newlines.
   - used to separate tokens.
     unless it is inside string literals.

3. comments
   // single line
   /*   multiple line.
   **
   */

/** Java Doc

*/

Scanner recognize comments^whitespace but exclude them from further processing.

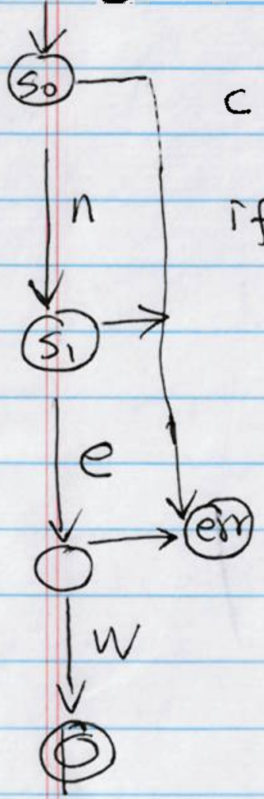
```
 ↓         ↓        ↓       ↓     ↓  ↓  ↓↓    ↓   ~~↓ ↓~~
public   static   void   main  ( String[] args ~~[ ]~~ )
→ {        /* this   */
          system. out. println ( "1+2 =" +(1+2));
              ↑    ↑  ↑ ↑    ↑    ↑       ↑       ↑↑↑↑↑↑↑↑
→ }
```

2.2. tokens in python.

```
 ↓ ↓  ↓ ↓  ↓
 if i > j :  ↓    ↓
  ␣4␣4␣4   print(i)
                ↑  ↑
→else: ←                           |   else:      print(j).
  ␣4␣4␣4   print(j)                |
 print("hello") ↑  ↑↑↑             |
# comment
```

3. from characters to tokens

① Recognize a single word. "new".



```
c ← nextChar( )
if c == 'n'
        c ← next char( )
        if c == 'e'
                c ← nextChar( )
                if c == 'w'
                        return true
                else  } retur false
        else
else.
```

②. recognize several words.

new. not

③. recognize. more complex words.

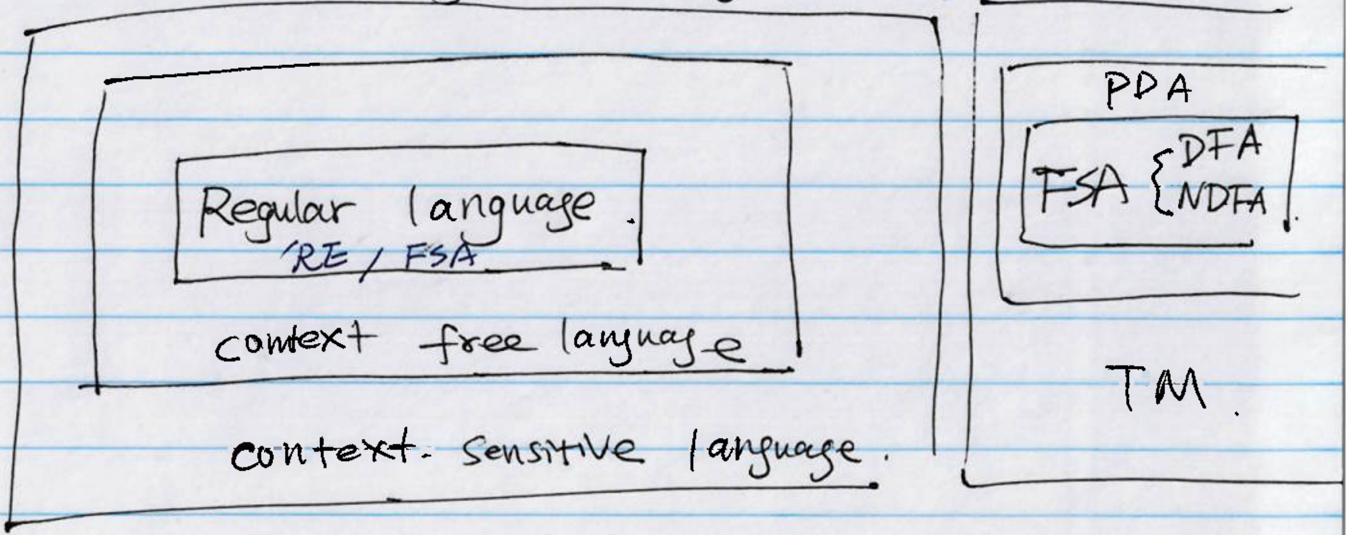
0   001   00   1,000

1_000

Tools from

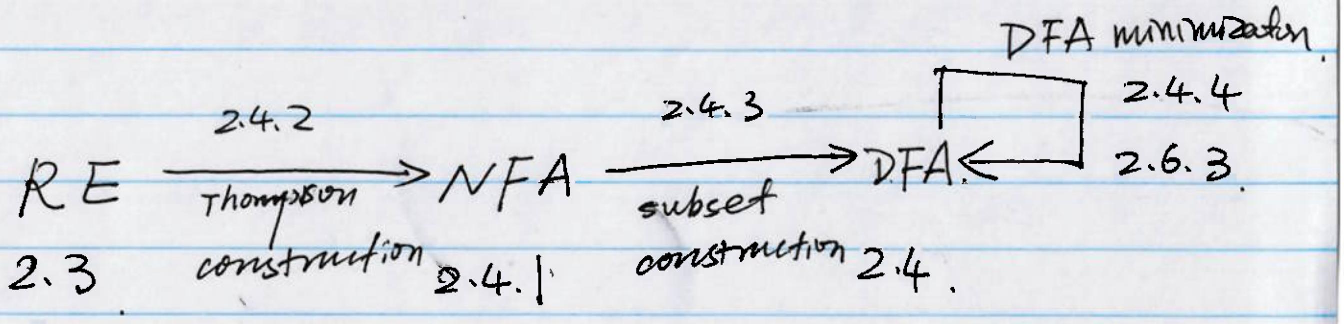
we need formal language theory. automata

4.



specify tokens/words

RL / RE / FSA



01/22/2025 RE

Regular expression (2.3). <u>lexical structure</u>

1. definition.

Notation for specifying microsyntax such as the spelling of a positive integer.

RE over an alpabet Σ. each RE is associated with $\underbrace{\text{a set of strings over } \Sigma}_{\text{language}}$. e.g $\Sigma = \{a, b\}$

- ε is a regular expression. empty string.
- if $x \in \Sigma$, then x is a RE

  e.g. a, b. $L(a) = \{a\}$. $L(b) = \{b\}$.

- if X and y are REs. then

  - concatination xy is a RE.

  eg. ab, aa, aab $L(ab) = L(a)L(b) = \{a\}\{b\}. = \{ab\}$

$$L(xy) = L(x)\,L(y) = \{\underline{pq} : p \in L(x), q \in L(y)\}$$

$$L(ba) = \{ba\}.$$

- union / alternation.

$x|y$ is a RE

$$L(x|y) = L(x) \cup L(y)$$

$$= \{ p : p \in L(x) \text{ or } p \in L(y) \}.$$

$$L(a|ab) = \{ a, ab, \cancel{aab} \}.$$

$$L(a) = \{a\}$$

$$L(ab) = \{ab\}.$$

- klenee closure. X^*

$$L(x^*) = L(x)^0 \cup L(x)^1 \cup L(x)^2 \cup \cdots$$

////

$$L(x)^0 = \{ \varepsilon \}$$

$$L(x)^i = L(x)^{i-1} L(x)$$

$$L(a^*) = \{ \varepsilon, a, aa, aaa, aaaa, \ldots \}$$

$$L(ab)^* = \{ \varepsilon, ab, abab, ababab, \ldots \}.$$

$$L(a|b)^* = \{ \varepsilon, a, b, ab, aa, bb, ba, \ldots$$

$$L(a|b)^2 = L(a|b)\, L(a|b)$$

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

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

precedence ( ) highest

$*$ $\downarrow$

concatination

alteration. lowest.

$$L(\underline{a\underline{b^*}}) = \{a, ab, abb, abbb, \cdots\}.$$

$$= L(a)\,L(b^*)$$

$$= \{a\}\ \{\varepsilon, b, bb, bbb, \cdots\}.$$

$$= \{\underline{a}\ \underline{\varepsilon},\ \underline{a}\ \underline{b},\ \underline{a}\ \underline{bb},\ \ldots\ldots\},$$

$$x\varepsilon = \varepsilon x = x$$

2. closure properties of REs / RLs.

REs/RLs are closed under

concatination

alternation

kleene clousure*

Note: X^+ : positive clousure.

$$L(x^+) = L(x) \cup L(x^2) \cdots .$$

$$ab^+ \equiv abb^*$$

not recommend.

X^i : finite closure.

$L(X^0) \cup L(X^1) \cup L(X^2) \cdot L(X^i)$

two possible meaning:

$L(X^i)$

1^2 $\{\varepsilon, 1, 11\}$

$\{11\}$

3. more examples.

①. RE → RL. Languages.

$(0|1)(0|1)$ $\{00, 01, 10, 11\}$.

$0(0|1)^*$ $\{0, 00, 01, 000, 001, \cdots\}$.

~~10~~ ~~110~~

1ε $\{1\}$

$(1|\underline{01})^*$ $\{\varepsilon, 1, 01, 11, \underline{111}, 1\underline{01}, \underline{01}1\}$

every "0" has to be followed by a '1'

$\underline{1}$ $\underline{01}$

$\underline{1}\,\underline{1}$ $\underline{01}\,1$

$1\,\underline{01}$ $\underline{01}\,01$

$\underline{111}$

$0|01$

$(1|01)^*(\varepsilon|0)$ ε, 1, 0, 10, 01, 11, ~~00~~

010 011. 101 110 111

⋮

ε or any binary strng w/o two consecutive "0's.

219989 $(0|1|2|3|4|5|6|7|8|9)(\quad)(\quad)(\quad)$

$[0\cdots9]$ $[0\cdots9]\ [0\cdots9]\ [0\cdots9]$

② RL ⟹ RE.

all binary strgs ending with "1" $(0|1)^*1$

{public, static, if}.   ~~{~~public| static|, if~~}~~.

unsigned integers w/o leading 0-s.

$[0\cdots9]^*$ ~~ε~~

$[0\cdots9][0\cdots9]^*$ ~~00~~ ~~01~~

$[0\cdots9][2\cdots9]^*$ ~~11~~ ~~02~~

$[1\cdots9][0\cdots9]^*$ ~~0~~

$0\,|\,([1\cdots9][0\cdots9]^*)$