Exemplu de limbaj care respectă L.P. $\Rightarrow$ nu este regulat.

$L_{bază} = \{a^n b^n \mid n \geq 0\}$.

$L_1 = \underbrace{ac^+ac^+ \dots ac^+bc^+bc^+ \dots bc^+}_{nr\_a = nr\_b}$

fie n (L.P.) $= 3$ $\Rightarrow$ orice subșir de 3 caract $\rightarrow$ include c^+

$\Rightarrow$ dc elimin c (cf. L.P.) $\Rightarrow$ 2 simb a alăturate, b alăturate $\not\rightarrow \notin L_1$.

$$L_{bază} = \bigcup_{k=0}^{\infty} (a)^k (b)^k$$

$$L_1 = \bigcup_{k=0}^{\infty} (ac^+)^k (bc^+)^k$$

? $L_2 \Rightarrow$ regulat $L_1 \cup L_2$ s. satisfi L.P. dar să nu fie regulat

$\downarrow$

$0^n 1^n \cup \{0,1\}^* \rightarrow \{0,1\}^*$

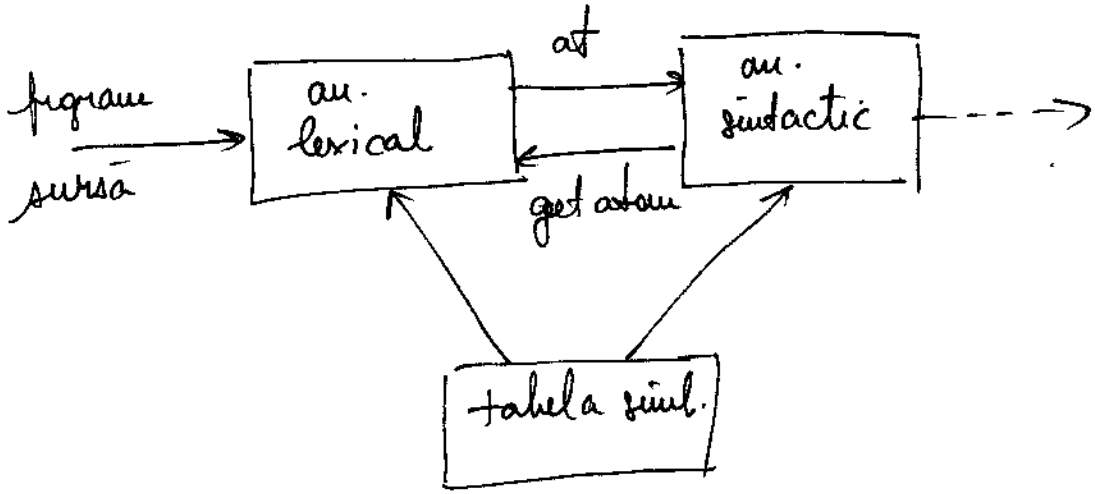
$L_2 \rightarrow$ disjunct L_1

$L_2 = (a^* b^* c)^* (a \cup b)^2 (a \cup b \cup c)^*$. $\rightarrow$ regulat

pt că $L_1, L_2 \rightarrow$ disjuncte $\Rightarrow$ $L_1 = \underbrace{L \cup L_2 - L_2}_{\text{prop incidere}}$.

Analiza lexicală

An. lexicală → prima etapă a proc. de compilare.



At. lexicali → cuv. cheie
operatori
identificatori
constante
siruri de literali
simboluri de punctuație

v = m * 2;

< id, ↑T.S. pt v >

< op- atr >

< id, ↑TS · pt m >

< op- mult >

< nnn, val. 2 >

< ; >

Definiția regulate

Σ → alfabet simboluri de față

$d_1 \to r_1$

$d_2 \to r_2$

- - - - -

$d_n \to r_n$

d_i → nume distinct

r_i → expresie regulată peste $\Sigma \cup \{d_1, \ldots, d_{i-1}\}$.

ex:
identificatori în Pascal → șir de litere și cifre care începe cu o literă

litera → A|B|C...|Z|a|b|...|z
cifra → 0|1|...|9
id → litera (litera|cifra)*

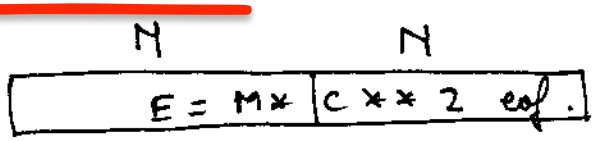
ex:
const. numerică fără semn în Pascal: 5350, 39.34, 6.33E4

cifra → 0|1|...|9
cifre → cifra cifra*
p-fract → .cifre | e
exponent → (E(+|-|e) cifre) | e
numar → cifre p-fract exponent

Convenții de notație

1° opt. postfix + →
2° $r? \equiv r | e$

Input buffer



N	N
E = M*	C ** 2 eof.

N = 1024 sau 4096.

Inițial → cei 2 ptr indică începutul lexemei

Deplasez dr fwd-ptr => potrivire => poz. fwd-ptr pe 1 nul caracter din dr. lexemei orte și adac beg-ptr în poz. fwd-ptr.

=> buff circular

Recunoașterea atomilor lexicali

instr → if expr then instr
| if expr then instr else instr
| e

expr → termen opral termen
| termen

termen → id | num

if → if

then → then

else → else

oprel → < | <= | = | <> | > | >=

id → litera (litera | cifra) *

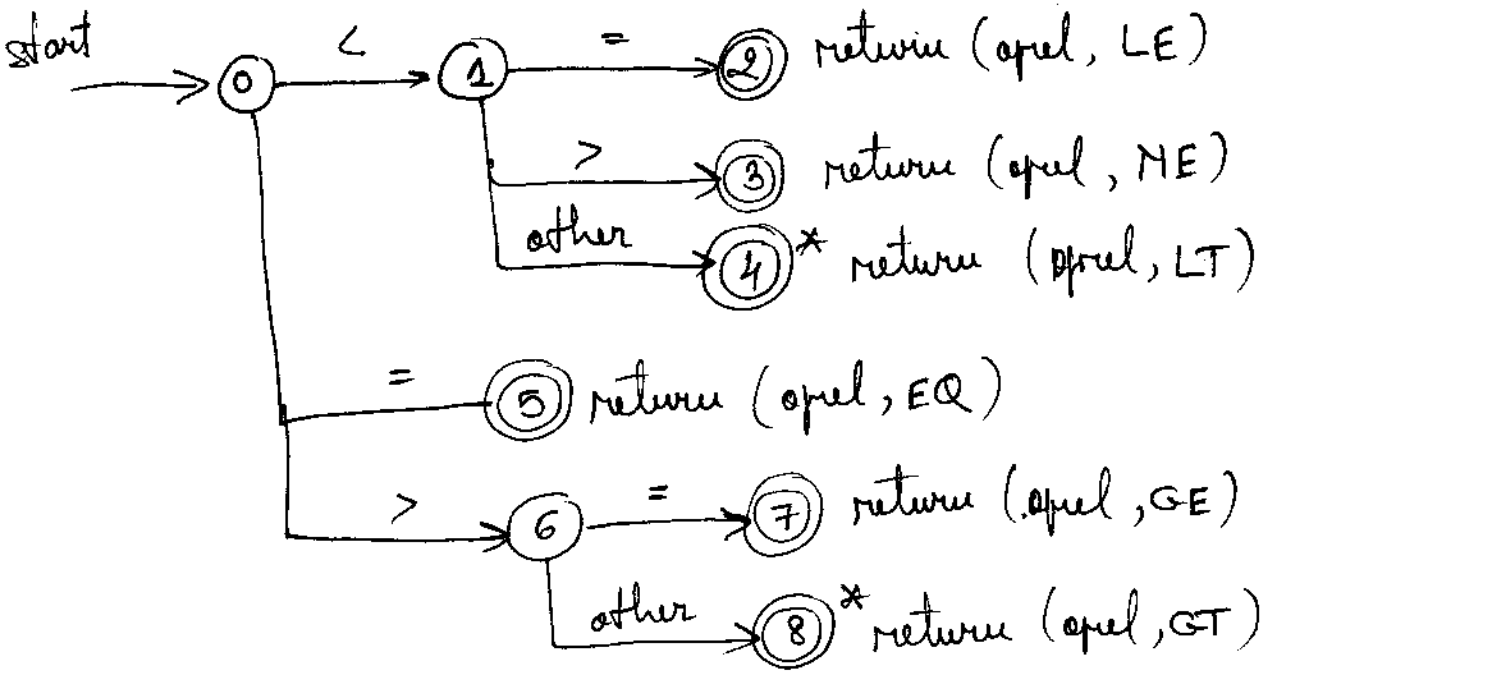
num → cifra$^+$ (. cifra$^+$)? (E (+|-)? cifra$^+$)?

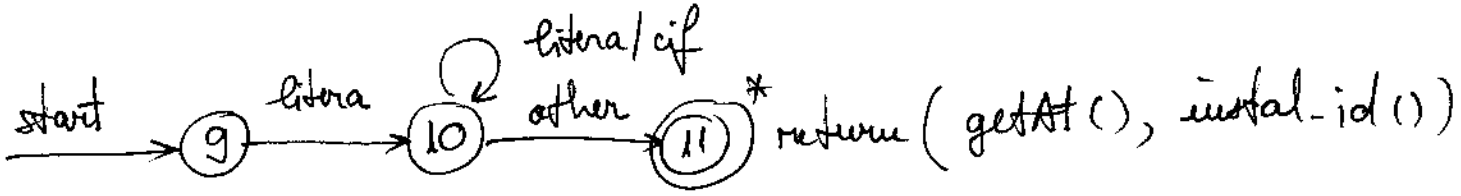
delim → blanks | tab | newline

ws → delim$^+$

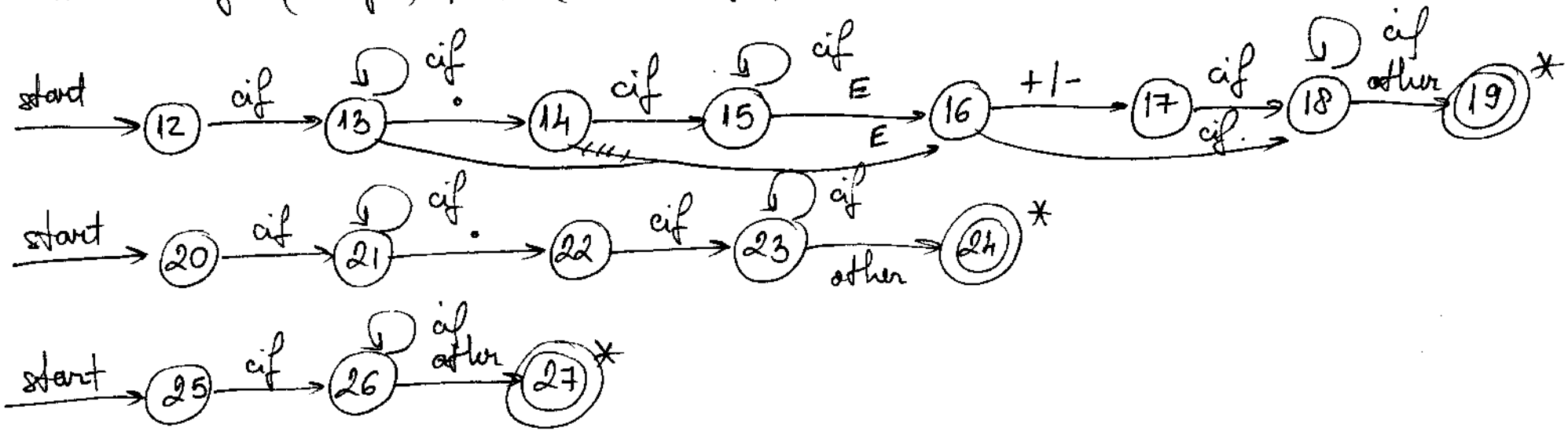
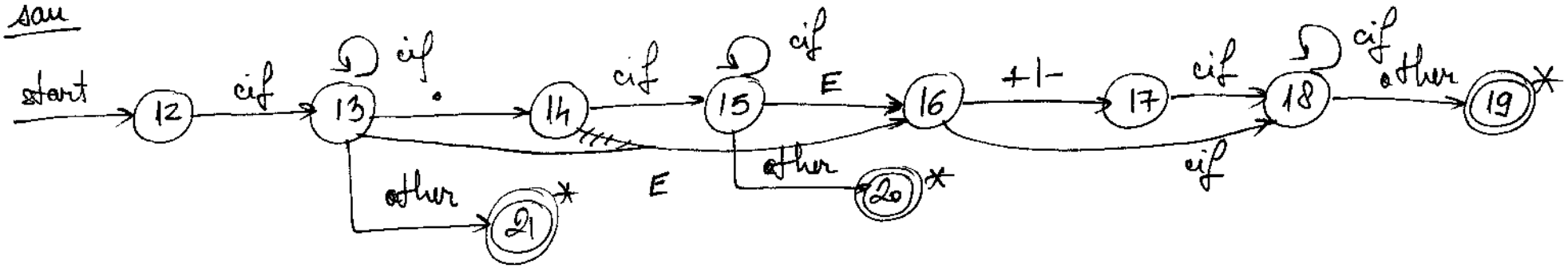
Expresii regulate	At. lexical	Atribut
ws	—	—
if	if	—
then	then	—
else	else	—
id	ID	↑ T.S.
num	NUM	↑ T.S.
<	oprel	LT
<=	oprel	LE

Diagrame de tranziție





num → dif+ (. dif+)? (E(+|-)? dif+)?



sau