

# FLCD Seminar 1 – Programming Languages' Specification

Monday, October 05, 2020

11:09 AM

## Notations (meta-languages)

### I.BNF (Backus-Naur Form)

Constructs:

1. Meta-linguistic variables (non-terminals) - written between < >
2. Language primitives (terminals) - written as they are, no special delimiters
3. Meta-linguistic connectors
  - a. ::= equals by definition
  - b. | alternative (OR)

General shape of a BNF definition:

<construct> ::= expr\_1 | expr\_2 | ... | expr\_n, where expr\_i is a combination of terminals and/or nonterminals, i=1,n

**Ex.1:** Specify, using BNF, all nonempty sequences of letters



<let\_sequence> ::= <letter> | <letter><let\_sequence>

<letter> ::= a | b | ... | z | A | B | ... | Z

**Ex.2:** Specify, using BNF, both signed and unsigned integers, with the following constraints:

- 0 does not have a sign
- numbers of at least two digits cannot start with 0



<integer> ::= 0 | <sign> <unsigned> | <unsigned>

<sign> ::= - | +

<unsigned> ::= <nonzerodigit> | <nonzerodigit> <digit\_seq>

<digit\_seq> ::= <digit> | <digit> <digit\_seq>

<nonzerodigit> ::= 1 | 2 | 3 .. | 9

<digit> ::= 0 | <nonzerodigit>

### II.EBNF (Extended BNF)

Wirth's dialect

1. Changes to the concrete syntax of standard BNF
  - Nonterminals lose <> => they are written without delimiters
  - Terminals are written between " "
  - ::= becomes =
2. New constructs

- {} - repetition 0 or more times
- [] - optionality (0 or 1)
- () - math grouping
- (\* \*) - comments
- rules end with .

**Ex.3:** Ex. 2 reloaded, in EBNF



integer = "0" | [" + " | " - "] nonzerodigit { "0" | nonzerodigit }

nonzerodigit = "1" | ... | "9"

# FLCD Seminar 2 – Scanning

Monday, October 05, 2020 1:32 PM

Input: source code + lexical tokens

Output: PIF, ST, lex. err

Input source:

```
-----  
Program test;  
Var a : integer;  
Begin  
  a:= b + 1;  
End.
```

Outputs:

PIF

Token ST\_pos

```
-----  
Program -1  
Id      0  
;       -1  
Var     -1  
Id      1  
:       -1  
Integer -1  
;       -1  
Begin   -1  
Id      1  
:=      -1  
Id      2  
+       -1  
Const   3  
;       -1  
End     -1  
.       -1
```

ST (only id & const)

ST\_pos symbol

```
-----  
0      test  
1      a  
2      b  
3      1
```

## GRAMMARS

1. Given the grammar  $G = (N, \Sigma, P, S)$

$$N = \{S, C\}$$

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

$$P : S \rightarrow ab \mid aCSb$$

$$C \rightarrow S \mid bSb$$

$$CS \rightarrow b$$

? Prove that  $w = ab(ab^2)^2 \in L(G)$

$$\text{Obs.: } (ab)^2 = abab \neq a^2b^2 = aabb$$

**Sol.: B:** 

$$S \Rightarrow_2 aCSb \Rightarrow_4 abSbSb \Rightarrow^2_1 ababbabb = w \Rightarrow S \Rightarrow^* w \Rightarrow w \in L(G)$$

2. Given the grammar  $G = (N, \Sigma, P, S)$

$$N = \{S\}$$

$$\Sigma = \{a, b, c\}$$

$$P : S \rightarrow a^2S \mid bc$$

find  $L(G)$ .

**Sol.: B:** 

$$L = \{a^{2n}bc \mid n \in N\}$$

?  $L = L(G)$

$$1. \quad L \subseteq L(G)$$

$$\forall n \in N, a^{2n}bc \in L(G)$$

$$P(n) : a^{2n}bc \in L(G)$$

Prove that  $P(n)$  is true for any natural  $n$ , by math. Induction

i). verification step  $n=0$

$$S \Rightarrow bc \Rightarrow bc \in L(G) \Rightarrow P(0) - \text{true}$$

ii). proof step  $P(k) \Rightarrow P(k+1), k \in N$

$P(k) - \text{true} \Rightarrow S \Rightarrow^* a^{2k}bc$  (induction hypothesis)

$S \Rightarrow a^2S \Rightarrow^* a^2a^{2k}bc = a^{2k+2}bc \Rightarrow P(k+1) - \text{true}$

From i) and ii) we have that  $L \subseteq L(G)$

2. ?  $L(G) \subseteq L$

$S \Rightarrow bc = a^0bc$

$\Rightarrow a^2S \Rightarrow a^2bc$

$\Rightarrow a^4S \Rightarrow a^4bc$

$\Rightarrow a^6S \Rightarrow \dots$

3. Find a grammar that generates  $L = \{0^n1^n2^m \mid n, m \in N^*\}$

Sol.: B: 

$N = \{S, C, D\}$

$\Sigma = \{0, 1, 2\}$

$P : S \rightarrow CD$

$C \rightarrow 0C1 \mid 01$

$D \rightarrow 2D \mid 2$

?  $L(G) = L$

1. ?  $L \subseteq L(G)$

?  $\forall n, m \in N^*, 0^n1^n2^m \in L(G)$

Let  $n, m \in N^*$

$S \Rightarrow CD \Rightarrow \overset{n}{0^n1^n} \overset{m}{D} \Rightarrow 0^n1^n2^m$

(a) (b)

(a):  $\forall n \in N^*, C \Rightarrow \overset{n}{0^n1^n}$

(b):  $\forall m \in N^*, D \Rightarrow \overset{m}{2^m}$

2. ?  $L(G) \subseteq L$

-tree for C

-tree for D

S can only generate concatenations of C and D ...

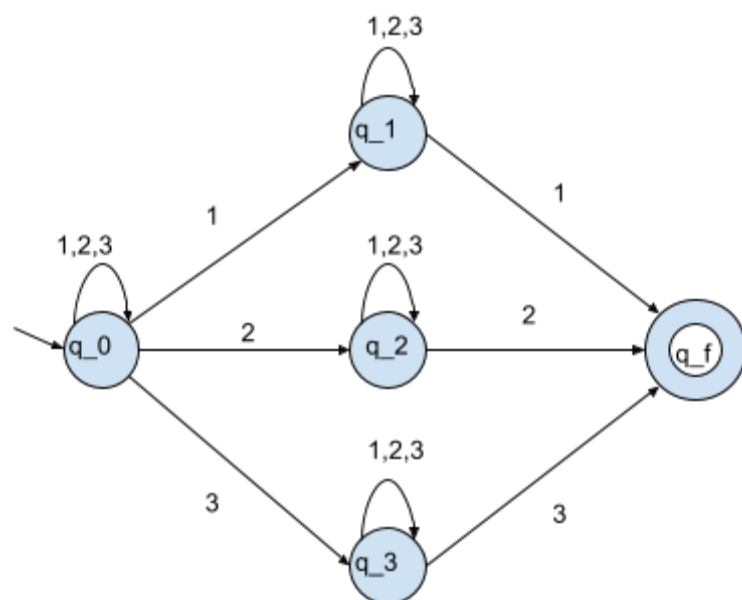
## FINITE AUTOMATA (FA)

1. Given the FA:  $M = (Q, \Sigma, \delta, q_0, F)$ ,  $Q = \{q_0, q_1, q_2, q_3, q_f\}$ ,  $\Sigma = \{1, 2, 3\}$ ,  $F = \{q_f\}$ ,

$\delta$	1	2	3
$q_0$	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_3\}$
$q_1$	$\{q_1, q_f\}$	$\{q_1\}$	$\{q_1\}$
$q_2$	$\{q_2\}$	$\{q_2, q_f\}$	$\{q_2\}$
$q_3$	$\{q_3\}$	$\{q_3\}$	$\{q_3, q_f\}$
$q_f$	$\emptyset$	$\emptyset$	$\emptyset$

Prove that  $w = 12321 \in L(M)$

Sol.: B: XXXXXXXXXXXXXXXXXXXX







$$(p, a^{k+1}) \mid - (p, a^k) \mid - (p, \varepsilon) \Rightarrow (p, a^{k+1}) \mid - (p, \varepsilon) \Rightarrow P(k+1) \text{ -true}$$

Ind. hyp.

Similarly, we demonstrate b.

2. ?  $L(M) \subseteq L$  (M does not accept anything else but sequences of that shape)

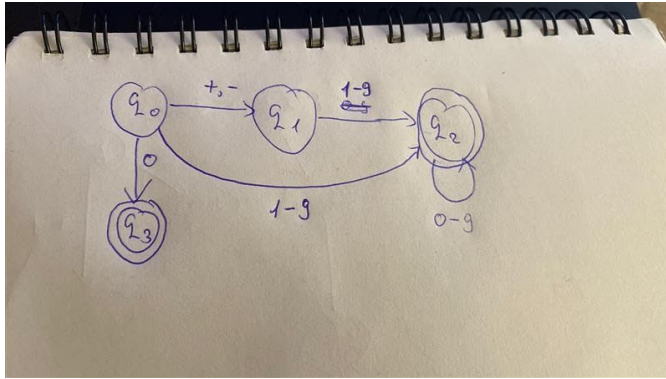
In order to reach the final state  $q$  from the initial state  $p$ , we should read at least one  $b$ . Before the mandatory  $b$ , we can read any natural number of  $a$ 's, while remaining in state  $p$ , and after the mandatory  $b$  we can read any natural number of  $b$ 's, while remaining in state  $q$ . Therefore,  $M$  accepts only sequences of the shape  $a^n b b^k$ ,  $n, k \in N$

**Obs.** In order for such a reasoning to count as proof, you should make sure that you have covered all paths from initial state to final states.

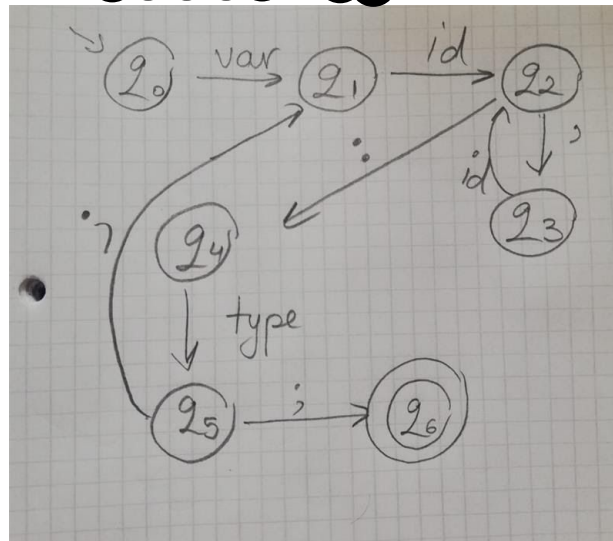
3. Build FAs that accept the following languages

- a. Integer numbers
- b. Variable declarations (Pascal, C, ...)
- c.  $L = \{0^n 1^m 0^q \mid n, m \in N^*, q \in N\}$
- d.  $L = \{0(01)^n \mid n \in N\}$
- e.  $L = \{c^{3n} \mid n \in N^*\}$
- f. The language over  $\Sigma = \{0, 1\}$  having the property that all sequences have at least two consecutive 0's.

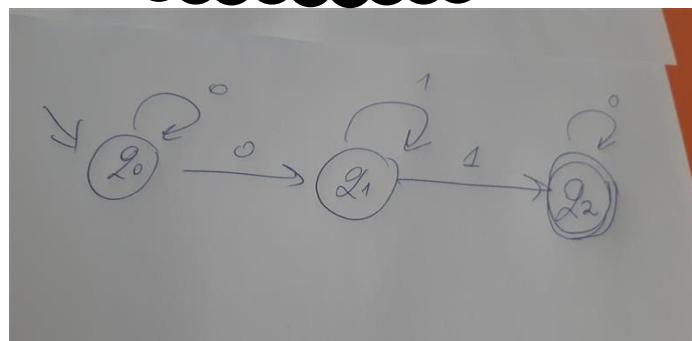
a. IW -> 



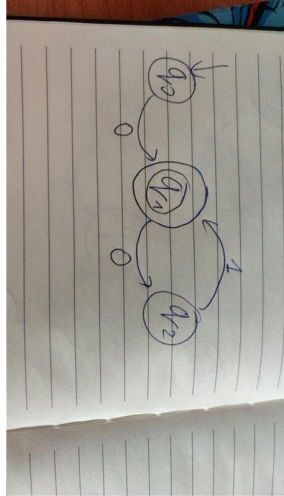
b. IW->



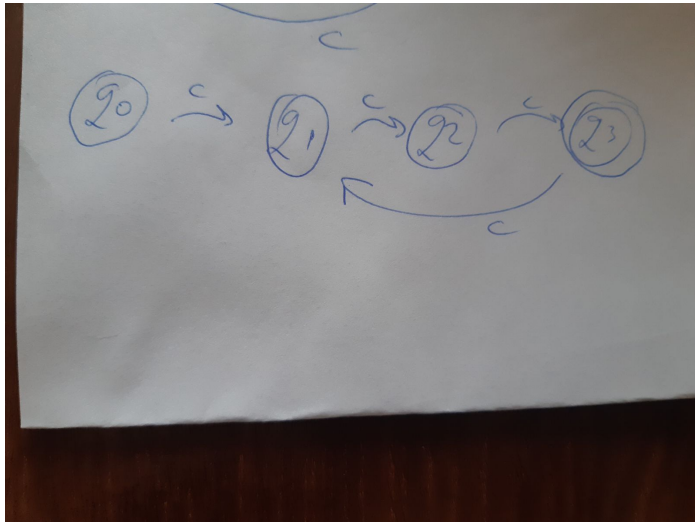
c. IW ->B:



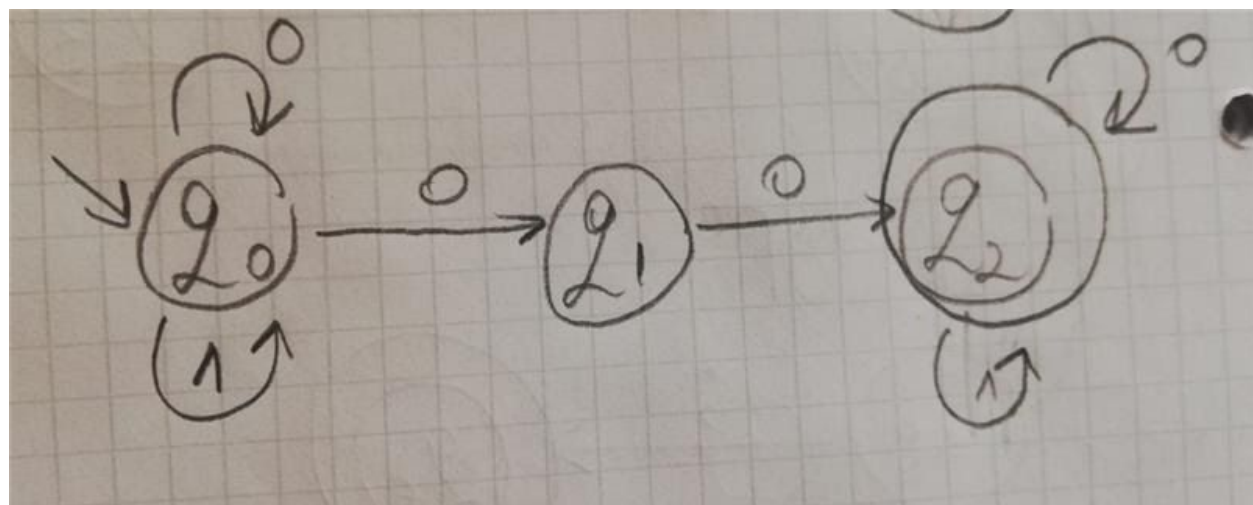
d. IW ->B:



e. IW  $\rightarrow$  B: [REDACTED] (+mark q as initial state)



f. IW  $\rightarrow$  B: [REDACTED]



## FA $\Leftrightarrow$ RG $\Leftrightarrow$ RE

### I) FA $\Leftrightarrow$ RG (team work)

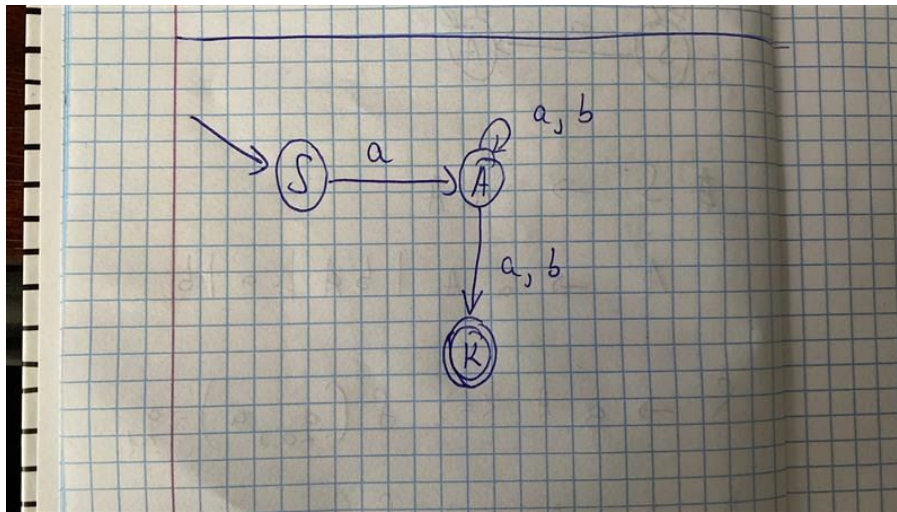
**T1.** Given the regular grammar  $G = (\{S, A\}, \{a, b\}, P, S)$

$$P : S \rightarrow aA$$

$$A \rightarrow aA \mid bA \mid a \mid b,$$

build the equivalent FA.

*Sol.:*



**T2.** Given the regular grammar  $G = (\{S, A\}, \{a, b\}, P, S)$

$$P : S \rightarrow \varepsilon \mid aA$$

$$A \rightarrow aA \mid bA \mid a \mid b,$$

build the equivalent FA.

*Sol.:*

$$M = (Q, \Sigma, \delta, q_0, F)$$

$$Q = \{S, A, K\}, q_0 = S, F = \{K, S\}, \Sigma = \{a, b\}$$

$\delta$	a	b
S	{A}	$\emptyset$
A	{A, K}	{A, K}
K	$\emptyset$	$\emptyset$

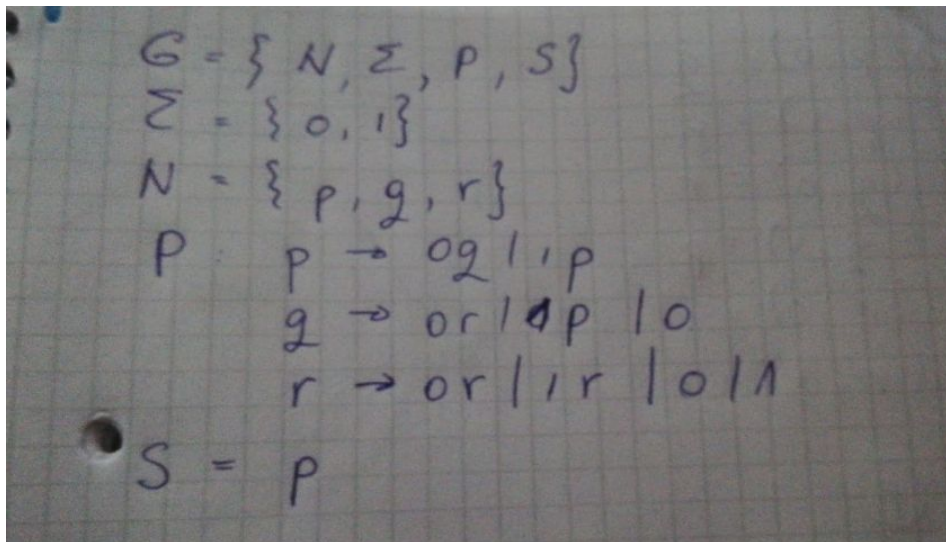
**T3.** Given the following FA  $M = (Q, \Sigma, \delta, q_0, F)$

$$Q = \{p, q, r\}, q_0 = p, F = \{r\}, \Sigma = \{0, 1\}$$

$\delta$	0	1
$p$	$q$	$p$
$q$	$r$	$p$
$r$	$r$	$r$

build the equivalent right linear grammar.

*Sol.:*



**T4.** Given the following FA  $M = (Q, \Sigma, \delta, q_0, F)$

$$Q = \{p, q, r\}, q_0 = p, F = \{p, r\}, \Sigma = \{0, 1\}$$

$\delta$	0	1
$p$	$q$	$p$
$q$	$r$	$p$
$r$	$r$	$r$

build the equivalent right linear grammar.

Sol.:

$$\begin{aligned}
 G &= \{N, \Sigma, P, S\} \\
 \Sigma &= \{0, 1\} \\
 N &= \{p, q, r\} \\
 S &= p \\
 P: \quad &p \rightarrow 0q \mid 1p \mid \epsilon \\
 &q \rightarrow 0r \mid 1p \mid 0 \mid 1 \\
 &r \rightarrow 0r \mid 1r \mid 0 \mid 1.
 \end{aligned}$$

## II) RG $\Leftrightarrow$ RE

1. Give the RG corresponding to the following RE  $0(0+1)^*1$ .

$$0: G_1 = (\{S_1\}, \{0, 1\}, \{S_1 \rightarrow 0\}, S_1)$$

$$1: G_2 = (\{S_2\}, \{0, 1\}, \{S_2 \rightarrow 1\}, S_2)$$

$$0+1: G_3 = (\{S_1, S_2, S_3\}, \{0, 1\}, \{S_1 \rightarrow 0, S_2 \rightarrow 1, S_3 \rightarrow 0 \mid 1\}, S_3)$$

$$G'_3 = (\{S_3\}, \{0, 1\}, \{S_3 \rightarrow 0 \mid 1\}, S_3)$$

$$(0+1)^*: G_4 = (\{S_3\}, \{0, 1\}, \{S_3 \rightarrow 0 \mid 1, S_3 \rightarrow 0S_3 \mid 1S_3, S_3 \rightarrow \epsilon\})$$

$$G'_4 = (\{S_3\}, \{0, 1\}, \{S_3 \rightarrow 0S_3 \mid 1S_3 \mid \epsilon\}, S_3) \text{ ! not regular}$$

$$0(0+1)^*: G_5 = (\{S_1, S_3\}, \{0, 1\}, \{S_1 \rightarrow 0S_3, S_3 \rightarrow 0S_3 \mid 1S_3 \mid \epsilon\}, S_1)$$

! not regular

$$0(0+1)^*1:$$

$$G_6 = (\{S_1, S_2, S_3\}, \{0, 1\}, \{S_1 \rightarrow 0S_3, S_3 \rightarrow 0S_3 \mid 1S_3 \mid S_2, S_2 \rightarrow 1\}, S_1) \text{ ! not regular}$$

$$G'_6 = (\{S_1, S_3\}, \{0, 1\}, \{S_1 \rightarrow 0S_3, S_3 \rightarrow 0S_3 \mid 1S_3 \mid 1\}, S_1)$$



(TW)

2. Give the RE corresponding to the following grammar

$$G = (\{S, A, B\}, \{a, b\}, P, S)$$

$$P : S \rightarrow aA$$

$$A \rightarrow aA \mid bB \mid b$$

$$B \rightarrow bB \mid b$$

Sol.: T4

Handwritten derivation of the regular expression for the given grammar:

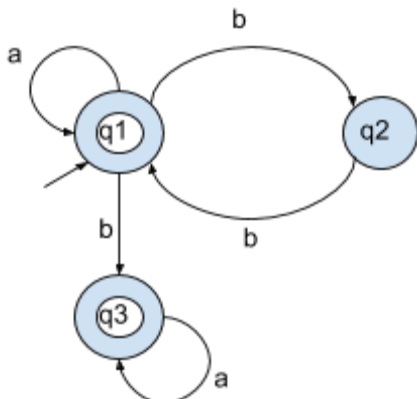
$$\begin{aligned} S &= aA \\ A &= aA + bB + b \\ B &= bB + b \Rightarrow B = b^*b = b^+ \\ \Rightarrow A &= a^* + B \\ A &= a^*B = a^*b^+ \\ S &= aA \Rightarrow S = aa^*b = a^+b^+ \end{aligned}$$

### III) FA $\Leftrightarrow$ RE

1. Give the FA corresponding to the following RE  $01(1+0)^*1^*$ .

#board, pdf attached to Seminar 7 meet in MSTeams

2. Give the regular expression corresponding to the FA below.





//

$$q_1 = \varepsilon + q_1 a + q_2 b$$

$$q_2 = q_1 b$$

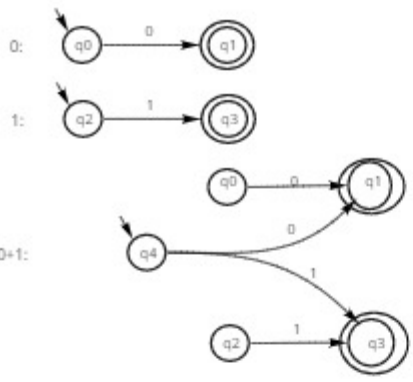
$$q_3 = q_1 b + q_3 a$$

$$X = Xa + b \Rightarrow X = ba^* \text{ solution}$$

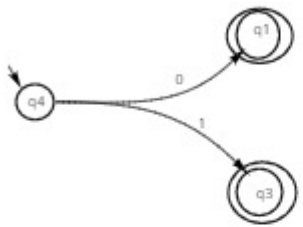
$$q_3 = q_1 ba^*$$

$$q_1 = \varepsilon + q_1 a + q_1 bb = q_1(a + bb) + \varepsilon \Rightarrow q_1 = (a + bb)^* \Rightarrow q_3 = (a + bb)^* ba^*$$

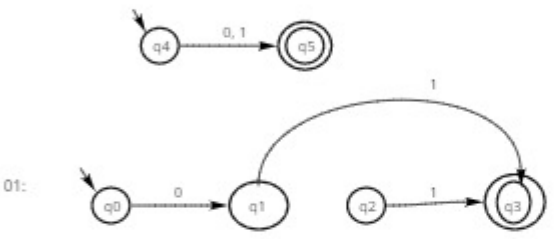
$$RE = q_1 + q_3 = (a + bb)^* + (a + bb)^* ba^* = (a + bb)^* (\varepsilon + ba^*)$$



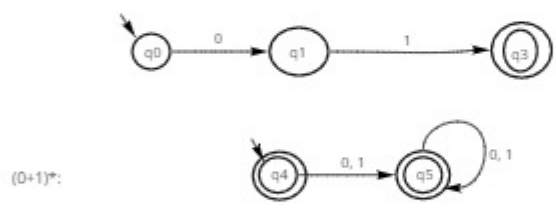
eliminate inaccessible



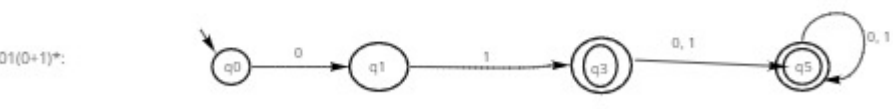
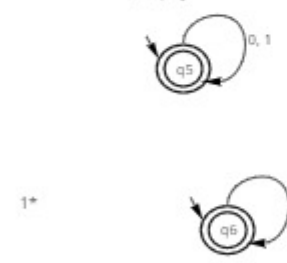
further simplify



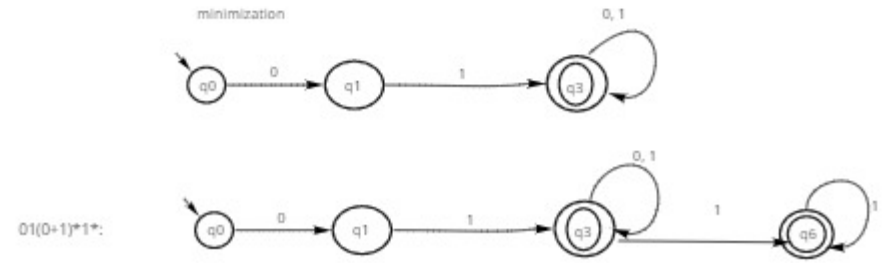
eliminate inaccessible



simplify



minimization



equivalent to the previous one

## CFG

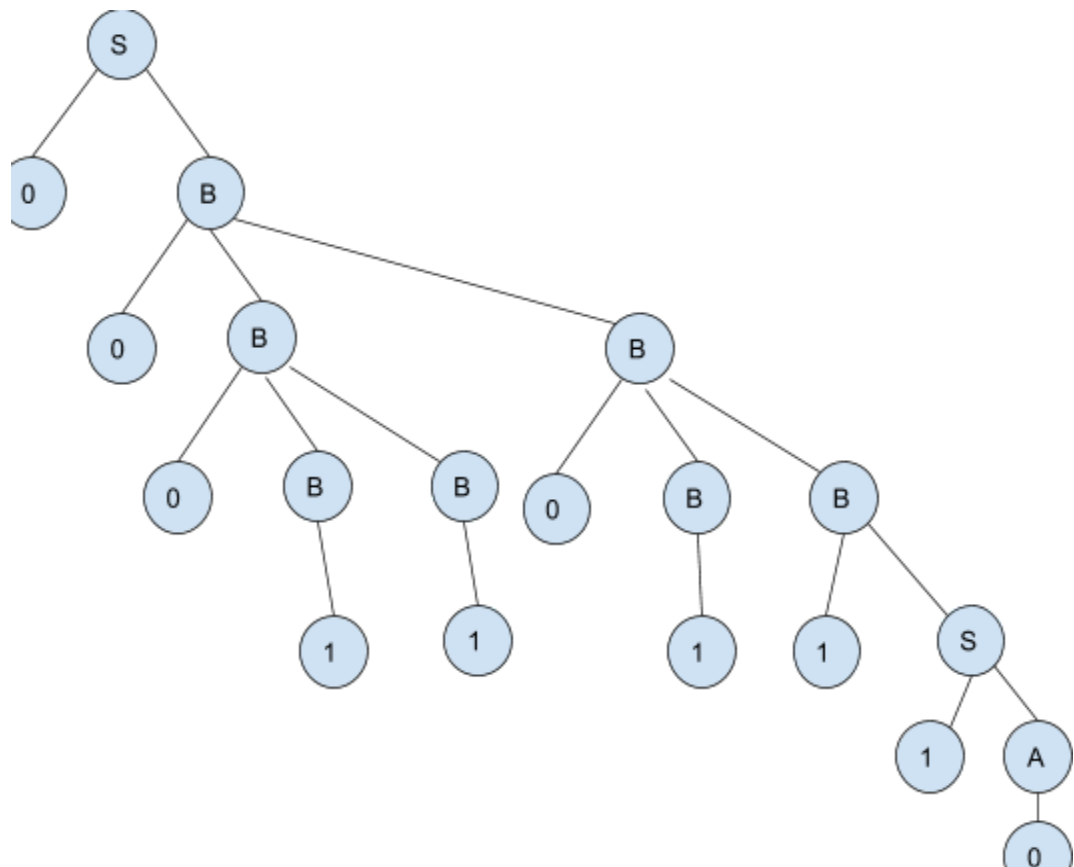
1. Given the CFG grammars below, give a leftmost/rightmost derivation for  $w$ .

- a.  $G = (\{S, A, B\}, \{0, 1\}, \{S \rightarrow 0B \mid 1A, A \rightarrow 0 \mid 0S \mid 1AA, B \rightarrow 1 \mid 1S \mid 0BB\})$ ,  
 $w = 0001101110$

Sol. XXXXXXXXXX

Leftmost: 1886686723

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 000BBB \Rightarrow 0001BB \Rightarrow 00011B \Rightarrow 000110BB \Rightarrow 0001101B$   
 $\Rightarrow 00011011S \Rightarrow 000110111A \Rightarrow 0001101110$



Rightmost: 1887236866

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 00B0BB \Rightarrow 00B0B1S \Rightarrow 00B0B11A \Rightarrow 00B0B110 \Rightarrow 00B01110 \Rightarrow$   
 $00B01110 \Rightarrow 000BB01110 \Rightarrow 000B101110 \Rightarrow 0001101110$

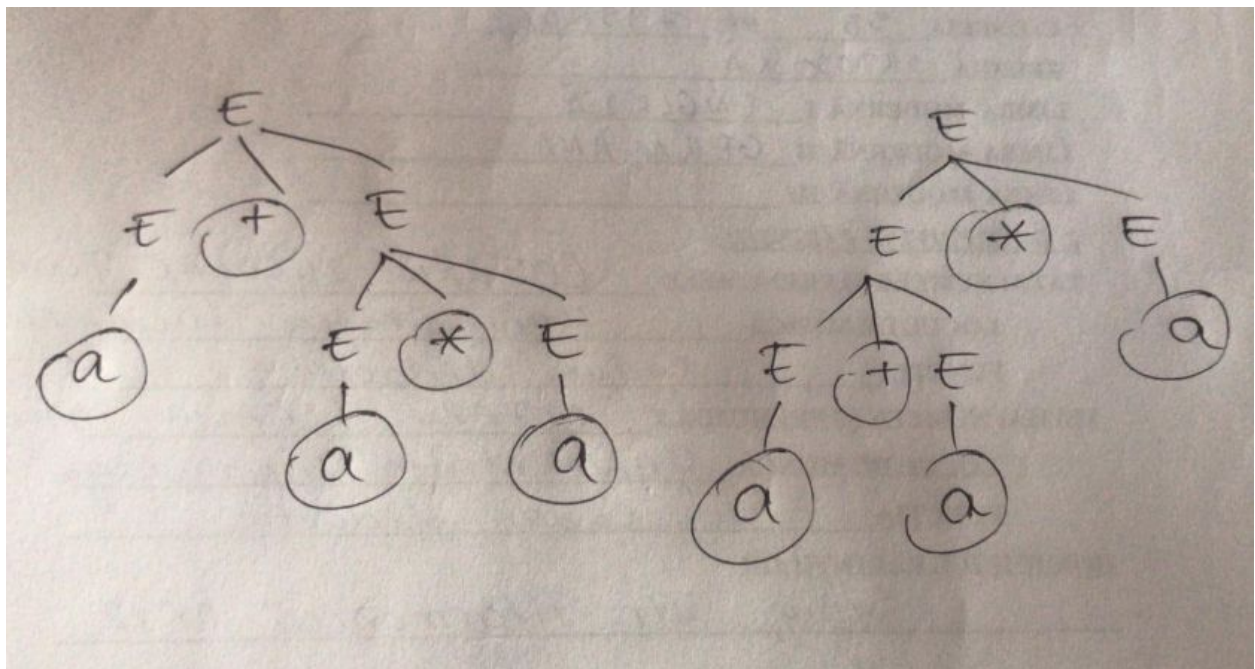
- b.  $G = (\{E, T, F\}, \{a, +, *, (, )\}, \{E \rightarrow E + T \mid T, T \rightarrow T * F \mid F, F \rightarrow (E) \mid a\})$   
 $w = a * (a + a) \rightarrow \text{HW}$

2. Prove that the following grammars are ambiguous

- a.  $G_1 = (\{S, B, C\}, \{a, b, c\}, \{S \rightarrow abC \mid aB, B \rightarrow bC, C \rightarrow c\}, S) \rightarrow \text{HW}$
- b.  $G_2 = (\{E\}, \{a, +, *, (, )\}, \{E \rightarrow E + E \mid E * E \mid (E) \mid a\})$

Sol.:

$w = a * a + a$



- c.  $G_3 = (\{S\}, \{if, then, else, a, b\}, \{S \rightarrow if\ b\ then\ S \mid if\ b\ then\ S\ else\ S \mid a\}, S) \rightarrow \text{HW}$

## Recursive descent parser

1. Given the CFG  $G = (\{S\}, \{a, b, c\}, \{S \rightarrow aSbS \mid aS \mid c\})$ , parse the sequence  $w = aacbc$  using rec. desc. parser.

Sol. : //B: 

$(S_1) S \rightarrow aSbS$

$(S_2) S \rightarrow aS$

$(S_3) S \rightarrow c$

$(q, 1, \varepsilon, S) \mid - \exp(q, 1, S_1, aSbS) \mid - \text{adv}(q, 2, S_1a, SbS) \mid - \exp(q, 2, S_1aS_1, aSbSbS) \mid -$   
 $\mid - \text{adv}(q, 3, S_1aS_1a, SbSbS) \mid - \exp(q, 3, S_1aS_1aS_1, aSbSbSbS) \mid -$   
 $\mid - \text{mi}(b, 3, S_1aS_1aS_1, aSbSbSbS) \mid - \text{at}(q, 3, S_1aS_1aS_2, aSbSbS) \mid -$   
 $\mid - \text{mi}(b, 3, S_1aS_1aS_2, aSbSbS) \mid - \text{at}(q, 3, S_1aS_1aS_3, cbSbS) \mid -$   
 $\mid - \text{adv}(q, 4, S_1aS_1aS_3c, bSbS) \mid - \text{adv}(q, 5, S_1aS_1aS_3cb, SbS) \mid -$   
 $\mid - \exp(q, 5, S_1aS_1aS_3cbS_1, aSbSbS) \mid - \text{mi}(b, 5, S_1aS_1aS_3cbS_1, aSbSbS) \mid -$   
 $\mid - \text{at}(q, 5, S_1aS_1aS_3cbS_2, aSbS) \mid - \text{mi}(b, 5, S_1aS_1aS_3cbS_2, aSbS) \mid -$   
 $\mid - \text{at}(q, 5, S_1aS_1aS_3cbS_3, cbS) \mid - \text{adv}(q, 6, S_1aS_1aS_3cbS_3c, bS) \mid -$   
 $\mid - \text{mi}(b, 6, S_1aS_1aS_3cbS_3c, bS) \mid - \text{back}(b, 5, S_1aS_1aS_3cbS_3, cbS) \mid -$   
 $\mid - \text{at}(b, 5, S_1aS_1aS_3cb, SbS) \mid - \text{back}(b, 4, S_1aS_1aS_3c, bSbS) \mid -$   
 $\mid - \text{back}(b, 3, S_1aS_1aS_3, cbSbS) \mid - \text{at}(b, 3, S_1aS_1a, SbSbS) \mid -$   
 $\mid - \text{back}(b, 2, S_1aS_1, aSbSbS) \mid - \text{at}(q, 2, S_1aS_2, aSbS) \mid - \text{adv}(q, 3, S_1aS_2a, SbS) \mid -$   
 $\mid - \exp, \text{mi}, \text{at}, \text{mi}, \text{at}(q, 3, S_1aS_2aS_3, cbS) \mid - \text{adv}(q, 4, S_1aS_2aS_3c, bS) \mid -$   
 $\mid - \text{adv}(q, 5, S_1aS_2aS_3cb, S) \mid - \exp, \text{mi}, \text{at}, \text{mi}, \text{at}(q, 5, S_1aS_2aS_3cbS_3, c) \mid -$   
 $\mid - \text{adv}(q, 6, S_1aS_2aS_3cbS_3c, \varepsilon) \mid - \text{success}(f, 6, S_1aS_2aS_3cbS_3c, \varepsilon)$

$\Rightarrow w$  is syntactically correct

Parse tree:  $S_1 S_2 S_3 S_3$

## LL(1) parser

**Ex.:** Given the CFG  $G = (\{S, A, B, C, D\}, \{+, *, a, (, )\}, P, S)$ ,

- $P$  :
- (1)  $S \rightarrow BA$
  - (2)  $A \rightarrow +BA$
  - (3)  $A \rightarrow \varepsilon$
  - (4)  $B \rightarrow DC$
  - (5)  $C \rightarrow *DC$
  - (6)  $C \rightarrow \varepsilon$
  - (7)  $D \rightarrow (S)$
  - (8)  $D \rightarrow a,$

Parse the sequence  $w = a * (a + a)$  using the LL(1) parser.

1) Compute FIRST 

	$F_0$	$F_1$	F2	F3
$S$	$\emptyset$	$\emptyset$	$(, a$	$(, a$
$A$	$+, \varepsilon$	$+, \varepsilon$	$+, \varepsilon$	$+, \varepsilon$
$B$	$\emptyset$	$(, a$	$(, a$	$(, a$
$C$	$*, \varepsilon$	$*, \varepsilon$	$*, \varepsilon$	$*, \varepsilon$
$D$	$(, a$	$(, a$	$(, a$	$(, a$

$\text{FIRST}(S) = \{ (, a \}$

$\text{FIRST}(A) = \{ +, \varepsilon \}$

$\text{FIRST}(B) = \{ (, a \}$

$\text{FIRST}(C) = \{ *, \varepsilon \}$

$\text{FIRST}(D) = \{ (, a \}$

2) Compute FOLLOW

//B: [REDACTED]

	$L_0$	$L_1$	$L_2$	$L_3$	$L_4$
$S$	$\epsilon$	$\epsilon, )$	$\epsilon, )$	$\epsilon, )$	$\epsilon, )$
$A$	$\emptyset$	$\epsilon$	$\epsilon, )$	$\epsilon, )$	$\epsilon, )$
$B$	$\emptyset$	$+, \epsilon$	$+, \epsilon, )$	$+, \epsilon, )$	$+, \epsilon, )$
$C$	$\emptyset$	$\emptyset$	$+, \epsilon$	$+, \epsilon, )$	$+, \epsilon, )$
$D$	$\emptyset$	$*$	$*, +, \epsilon$	$*, +, \epsilon, )$	$*, +, \epsilon, )$

$\text{FOLLOW}(S) = \{\epsilon, )\}$

$\text{FOLLOW}(A) = \{\epsilon, )\}$

$\text{FOLLOW}(B) = \{+, \epsilon, )\}$

$\text{FOLLOW}(C) = \{+, \epsilon, )\}$

$\text{FOLLOW}(D) = \{*, +, \epsilon, )\}$

3) Fill LL(1) parsing table

//B: [REDACTED]

	$a$	$+$	$*$	$($	$)$	$\$$
$S$	BA, 1			BA, 1		
$A$		+BA, 2			$\epsilon, 3$	$\epsilon, 3$
$B$	DC, 4			DC, 4		
$C$		$\epsilon, 6$	*DC, 5		$\epsilon, 6$	$\epsilon, 6$
$D$	a, 8			(S), 7		
$a$	pop					
$+$		pop				

*			pop			
(				pop		
)					pop	
\$						acc

4) Parse the sequence //B: 

$(a * (a + a)\$, S\$, \epsilon) | -$   
 $(a * (a + a)\$, BA\$, 1) | - (a * (a + a)\$, DCA\$, 14) | -$   
 $(a * (a + a)\$, aCA\$, 148) | - (* (a + a)\$, CA\$, 148) | - (* (a + a)\$, * DCA\$, 1485) | -$   
 $((a + a)\$, DCA\$, 1485) | - ((a + a)\$, (S)CA\$, 14857) | - (a + a)\$, S)CA\$, 14857) | -$   
 $(a + a)\$, BA)CA\$, 148571) | - (a + a)\$, DCA)CA\$, 1485714) | -$   
 $(a + a)\$, aCA)CA\$, 14857148) | - (+ a)\$, CA)CA\$, 14857148) | -$   
 $(+ a)\$, A)CA\$, 148571486) | - (+ a)\$, + BA)CA\$, 1485714862) | -$   
 $(a)\$, BA)CA\$, 1485714862) | - (a)\$, DCA)CA\$, 14857148624) | -$   
 $(a)\$, aCA)CA\$, 148571486248) | - ()\$, CA)CA\$, 148571486248) | -$   
 $()\$, A)CA\$, 1485714862486) | - ()\$, )CA\$, 14857148624863) | -$   
 $(\$, CA\$, 14857148624863) | - (\$, A\$, 148571486248636) | -$   
 $(\$, \$, 1485714862486363)$

LL(1) conflict

-----

$A \rightarrow \alpha\beta$

$A \rightarrow \alpha\gamma$

transformed to

$A \rightarrow \alpha B$

$B \rightarrow \beta | \gamma$

,



## LR(0) parser

**Ex.**  $G = (\{S', S, A\}, \{a, b, c\}, P, S')$

P:  $S' \rightarrow S$

(1)  $S \rightarrow aA$

(2)  $A \rightarrow bA$

(3)  $A \rightarrow c$

$w = abbc$

### 1. Compute the canonical collection of states //B: ●●●●●●●●

$$s_0 = \text{closure}(\{[S' \rightarrow \cdot S]\}) = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot aA]\}$$

$$s_1 = \text{goto}(s_0, S) = \text{closure}(\{[S' \rightarrow S \cdot]\}) = \{[S' \rightarrow S \cdot]\}$$

$$\text{goto}(s_0, A) = \{\dots\}$$

$$s_2 = \text{goto}(s_0, a) = \text{closure}(\{[S \rightarrow a \cdot A]\}) = \{[S \rightarrow a \cdot A], [A \rightarrow \cdot bA], [A \rightarrow \cdot c]\}$$

$$s_3 = \text{goto}(s_2, A) = \text{closure}(\{[S \rightarrow aA \cdot]\}) = \{[S \rightarrow aA \cdot]\}$$

$$s_4 = \text{goto}(s_2, b) = \text{closure}(\{[A \rightarrow b \cdot A]\}) = \{[A \rightarrow b \cdot A], [A \rightarrow \cdot bA], [A \rightarrow \cdot c]\}$$

$$s_5 = \text{goto}(s_2, c) = \text{closure}(\{[A \rightarrow c \cdot]\}) = \{[A \rightarrow c \cdot]\}$$

$$s_6 = \text{goto}(s_4, A) = \text{closure}(\{[A \rightarrow bA \cdot]\}) = \{[A \rightarrow bA \cdot]\}$$

$$\text{goto}(s_4, b) = \text{closure}(\{[A \rightarrow b \cdot A]\}) = s_4$$

$$\text{goto}(s_4, c) = \text{closure}(\{[A \rightarrow c \cdot]\}) = s_5$$

### 2. Fill in LR(0) parsing table //B: ●●●●●●●●

	ACTION	GOTO				
		a	b	c	S	A
0	shift	2			1	
1	accept					

2	shift		4	5		3
3	r1					
4	shift		4	5		6
5	r3					
6	r2					

3. Parse the input sequence // B: ●●●●●●●●

work stack	input stack	output band
\$0	abbc\$	$\epsilon$
\$0a2	bbc\$	$\epsilon$
\$0a2b4	bc\$	$\epsilon$
\$0a2b4b4	c\$	$\epsilon$
\$0a2b4b4c5	\$	$\epsilon$
\$0a2b4b4A6	\$	3
\$0a2b4A6	\$	23
\$0a2A3	\$	223
\$0S1	\$	1223
accept	\$	1223

## SLR parser

**Ex.**  $G = (\{S', E, T\}, \{+, id, const, (, )\}, P, S')$

P:  $S' \rightarrow E$

(1)  $E \rightarrow T$

(2)  $E \rightarrow E + T$

(3)  $T \rightarrow (E)$

(4)  $T \rightarrow id$

(5)  $T \rightarrow const$

$w = id + const$

1. Compute the canonical collection

// 

$S_0 = \text{closure}(\{[S' \rightarrow \cdot E]\}) = \{[S' \rightarrow \cdot E], [E \rightarrow \cdot T], [E \rightarrow \cdot E + T], [T \rightarrow \cdot (E)], [T \rightarrow \cdot id], [T \rightarrow \cdot const]\}$

$S_1 = \text{goto}(s_0, E) = \text{closure}(\{[S' \rightarrow E \cdot], [E \rightarrow E \cdot + T]\}) = \{[S' \rightarrow E \cdot], [E \rightarrow E \cdot + T]\}$

$S_2 = \text{goto}(s_0, T) = \text{closure}(\{[E \rightarrow T \cdot]\}) = \{[E \rightarrow T \cdot]\}$

$S_3 = \text{goto}(s_0, () = \text{closure}(\{[T \rightarrow (\cdot E)]\}) = \{[T \rightarrow (\cdot E)], [E \rightarrow \cdot T], [E \rightarrow \cdot E + T], [T \rightarrow (\cdot (E)], [T \rightarrow (\cdot id], [T \rightarrow (\cdot const]\}$

$S_4 = \text{goto}(s_0, id) = \text{closure}(\{[T \rightarrow id \cdot]\}) = \{[T \rightarrow id \cdot]\}$

$S_5 = \text{goto}(s_0, const) = \text{closure}(\{[T \rightarrow const \cdot]\}) = \{[T \rightarrow const \cdot]\}$

$S_6 = \text{goto}(s_1, +) = \text{closure}(\{[E \rightarrow E + \cdot T]\}) = \{[E \rightarrow E + \cdot T], [T \rightarrow \cdot (E)], [T \rightarrow \cdot id], [T \rightarrow \cdot const]\}$

$S_7 = \text{goto}(s_3, E) = \text{closure}(\{[T \rightarrow (E \cdot)], [E \rightarrow E \cdot + T]\}) = \{[T \rightarrow (E \cdot)], [E \rightarrow E \cdot + T]\}$

$\text{goto}(s_3, T) = \text{closure}(\{[E \rightarrow T \cdot]\}) = S_2$

$\text{goto}(s_3, id) = \text{closure}(\{[T \rightarrow id \cdot]\}) = S_4$

$\text{goto}(s_3, const) = \text{closure}(\{[T \rightarrow const \cdot]\}) = S_5$

$\text{goto}(s_3, () = \text{closure}(\{[T \rightarrow (\cdot E)]\}) = S_3$

$S_8 = \text{goto}(s_6, T) = \text{closure}(\{[E \rightarrow E + T \cdot]\})$

$\text{goto}(s_6, () = \text{closure}(\{[T \rightarrow (\cdot E)]\}) = S_3$

$\text{goto}(s_6, id) = \text{closure}(\{[T \rightarrow id \cdot]\}) = S_4$

$\text{goto}(s_6, const) = \text{closure}(\{[T \rightarrow const \cdot]\}) = S_5$

$S_9 = \text{goto}(s_7, ) = \text{closure}(\{[T \rightarrow (E) \cdot]\}) = \{[T \rightarrow (E) \cdot]\}$

$\text{goto}(s_7, +) = \text{closure}(\{[E \rightarrow E + \cdot T]\}) = S_6$

$\text{FOLLOW}(E) = \{\epsilon, +, )\}$

$\text{FOLLOW}(T) = \{\epsilon, +, )\}$

2. Fill the SLR table

//

	ACTION						GOTO	
	+	(	)	id	const	\$	E	T
0		Shift 3		Shift 4	Shift 5		1	2
1	Shift 6					acc		
2	Reduce1		Reduce1			Reduce1		
3		Shift 3		Shift 4	Shift 5		7	2
4	Reduce4		Reduce4			Reduce4		
5	Reduce 5		Reduce 5			Reduce 5		
6		Shift3		Shift4	Shift5			8
7	Shift6		Shift9					
8	Reduce 2		Reduce 2			Reduce 2		
9	Reduce 3		Reduce 3			Reduce 3		

3. Parse the sequence

//

Work stack	Input stack	Output band
\$0	id+const\$	ε
\$0id4	+const\$	ε
\$0T2	+const\$	4
\$0E1	+const\$	14
\$0E1+6	const\$	14
\$0E1+6const5	\$	14
\$0E1+6T8	\$	514
\$0E1	\$	2514
accept		

E => E + T => E + const => T + const => id + const

2

5

1

4

## LR(1) parser

**Ex.**  $G = (\{S', S, A\}, \{a, b\}, P, S')$

P:  $S' \rightarrow S$

(1)  $S \rightarrow AA$

(2)  $A \rightarrow aA$

(3)  $A \rightarrow b$

W =

LR(1) item  $[A \rightarrow \alpha.\beta, a]$

FIRST(S) = {a,b}

FIRST(A) = {a,b}

### 1. Canonical collection

// 

$S_0 = \text{closure}(\{[S' \rightarrow .S, \$]\}) = \{[S' \rightarrow .S, \$], [S \rightarrow .AA, \$], [A \rightarrow .aA, a], [A \rightarrow .aA, b], [A \rightarrow .b, a], [A \rightarrow .b, b]\}$

$S_1 = \text{goto}(S_0, S) = \text{closure}(\{[S' \rightarrow S., \$]\}) = \{[S' \rightarrow S., \$]\}$

$S_2 = \text{goto}(S_0, A) = \text{closure}(\{[S \rightarrow A.A, \$]\}) = \{[S \rightarrow A.A, \$], [A \rightarrow .aA, \$], [A \rightarrow .b, \$]\}$

$S_3 = \text{goto}(S_0, a) = \text{closure}(\{[A \rightarrow a.A, a], [A \rightarrow a.A, b]\}) = \{[A \rightarrow a.A, a], [A \rightarrow a.A, b], [A \rightarrow .aA, a], [A \rightarrow .b, a], [A \rightarrow .aA, b], [A \rightarrow .b, b]\}$

$S_4 = \text{goto}(S_0, b) = \text{closure}(\{[A \rightarrow b., a], [A \rightarrow b., b]\}) = \{[A \rightarrow b., a], [A \rightarrow b., b]\}$

$S_5 = \text{goto}(S_2, A) = \text{closure}(\{[S \rightarrow AA., \$]\}) = \{[S \rightarrow AA., \$]\}$

$S_6 = \text{goto}(S_2, a) = \text{closure}(\{[A \rightarrow a.A, \$]\}) = \{[A \rightarrow a.A, \$], [A \rightarrow .aA, \$], [A \rightarrow .b, \$]\}$

$S_7 = \text{goto}(S_2, b) = \text{closure}(\{[A \rightarrow b., \$]\}) = \{[A \rightarrow b., \$]\}$

$S_8 = \text{goto}(S_3, A) = \text{closure}(\{[A \rightarrow aA., a], [A \rightarrow aA., b]\}) = \{[A \rightarrow aA., a], [A \rightarrow aA., b]\}$

$\text{goto}(S_3, a) = \text{closure}(\{[A \rightarrow a.A, a], [A \rightarrow a.A, b]\}) = S_3$

$\text{goto}(S_3, b) = \text{closure}(\{[A \rightarrow b., a], [A \rightarrow b., b]\}) = S_4$

$S_9 = \text{goto}(S_6, A) = \text{closure}(\{[A \rightarrow aA., \$]\}) = \{[A \rightarrow aA., \$]\}$

$\text{goto}(S_6, a) = \text{closure}(\{[A \rightarrow a.A, \$]\}) = S_6$

$\text{goto}(S_6, b) = \text{closure}(\{[A \rightarrow b., \$]\}) = S_7$

### 2. Fill the LR(1) table

// 

	ACTION			GOTO	
	a	b	\$	S	A
0	Shift 3	shift4		1	2
1			accept		
2	shift6	shift7			5
3	shift3	shift4			8
4	reduce3	reduce3			
5			reduce1		
6	shift6	shift7			9
7			reduce3		
8	reduce2	reduce2			
9			reduce2		

### 3. Syntactical Analysis

W = abab



Work stack	Input stack	Output band
\$0	abab\$	-
\$0a3	bab\$	-
\$0a3b4	ab\$	-
\$0a3A8	ab\$	3
\$0A2	ab\$	23
\$0A2a6	b\$	23
\$0A2a6b7	\$	23
\$0A2a6A9	\$	23
\$0A2A5	\$	323
\$0S1	\$	2323
AC		x

## LALR(1) parser

**Ex.**  $G = (\{S', S, A\}, \{a, b\}, P, S')$

P:  $S' \rightarrow S$   
 (1)  $S \rightarrow AA$   
 (2)  $A \rightarrow aA$   
 (3)  $A \rightarrow b$

W = aaab

## 1. Canonical collection

$s_0 = \{[S' \rightarrow .S, \$], [S \rightarrow .AA, \$], [A \rightarrow .aA, a], [A \rightarrow .aA, b], [A \rightarrow .b, a], [A \rightarrow .b, b]\}$   
 $s_1 = \{[S' \rightarrow S., \$]\}$   
 $s_2 = \{[S \rightarrow A.A, \$], [A \rightarrow .aA, \$], [A \rightarrow .b, \$]\}$   
 $s_{36} = \{[A \rightarrow a.A, a/b/\$], [A \rightarrow .aA, a/b/\$], [A \rightarrow .b, a/b/\$]\}$   
 $s_{47} = \{[A \rightarrow b., a/b/\$]\}$   
 $s_5 = \{[S \rightarrow AA., \$]\}$   
 $s_{89} = \{[A \rightarrow aA., a/b/\$]\}$

## 2. LALR(1) table

	ACTION			GOTO	
	a	b	\$	S	A
s0	Shift s36	Shift s47		s1	s2
s1			accept		
s2	Shift s36	Shift s47			s5
s36	Shift s36	Shift s47			s89
s47	Reduce 3	Reduce 3	Reduce 3		
s5			Reduce 1		
s89	Reduce 2	Reduce 2	Reduce 2		

### 3. Parse the sequence

Work stack	Input stack	Output band
\$ s0	a a a b \$	Eps
\$ s0 a s36	a a b \$	Eps
\$ s0 a s36 a s36	a b \$	Eps
\$ s0 a s36 a s36 a s36	b \$	Eps
\$ s0 a s36 a s36 a s36 b s47	\$	Eps
\$ s0 a s36 a s36 a s36 A s89	\$	3
\$ s0 a s36 a s36 A s89	\$	23
\$ s0 a s36 A s89	\$	223
\$ s0 A s2	\$	2223