Seminar 1 – Programming Languages' Specification

Tuesday, September 29, 2020 2:27 PM

I. BNF (Backus-Naur Form)

Constructs:

1. Meta-linguistic variables (non-terminals) - written between < >
2. Language primitives (terminals) - no special delimiters
3. Meta-linguistic connectors

::= equals by definition

| alternative/OR

<construct> ::= expr_1 | expr_2 | ... | expr_n

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

Onita Andrei

LetterSeq ::= ["Lorem ipsum"]

Petcu Dragos

<LetterSeq> ::= A | B | C | ... | Z

Oana Nourescu

<LetterSeq> ::= <Letter> | <Letter> <LetterSeq>

<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 do not start with 0

Neta Razvan

<Number> ::= 0 | <sign><non zero digit> | <non zero digit><Number>

Onita Andrei

<Number> ::= 0 | <sign> <non zero digit>| <non zero digit> |<sign> <non zero digit><Number> | <non zero digit><Number>

Neta Razvan

<Int> ::= 0 | <sign><DigitSeq>

Onita Andrei

<Int> ::= 0 | <sign><Number> | <Number>

<Number> ::= <Non0digit>| <Number><digit>

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

<Non0digit> ::= 0 | <digit>

Petcu Dragos

<sign> ::= + | -

<finalNumber> ::= 0 | <sign><number> | <number>

<number> ::= <nonZeronumber> | <nonZeronumber><anyDigit> | <number><anyDigit>

<anyDigit> ::= 0 | 1|2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<digitSequence> ::= <anyDigit> | <anyDigit><digitSequence>

<nonZeroNumber> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Miclea George

<Integer> ::= <Digit> | <Minus> <NonZeroDigit> | <Minus> <NonZeroDigit> <DigitSeq> | <NonZeroDigit> <DigitSeq>

<DigitSeq> ::= <Digit> | <Digit> <DigitSeq>

<NonZeroDigit> ::= 1 | 2 | ... | 9

<Digit> ::= 0 | <NonZeroDigit>

Onita Andrei (wip)
<sign> ::= + | -

<finalNumber> ::= 0 | <sign><number> | <number>

<number> ::= <digitSequence>

<anyDigit> ::= 0 | 1|2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<digitSequence> ::= <nonZeroNumber> |<digitSequence><anyDigit>

<nonZeroNumber> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Moldovanu Dragos

<Number> ::= 0 | <non-zero-digit><NumberSeq> | -<non-zero-digit><NumberSeq> | <digit> | -<non-zero-digit>

<NumberSeq> ::= <digit> | <digit><NumberSeq>

<digit> ::= 0|1|...|9

<non-zero-digit> ::=1|2|...|9

Miclea George

<Integer> ::= 0 | <No> | <Sign> <No>

<No> ::= <NonZeroDigit> | <NonZeroDigit> <DigitSeq>

<DigitSeq> ::= <Digit> | <Digit> <DigitSeq>

<NonZeroDigit> ::= 1 | 2 | ... | 9

<Digit> ::= 0 | <NonZeroDigit>

<Sign> ::= + | -

II. EBNF (Extended BNF)

Wirth's dialect

- Nonterminals loose <> => written without delimiters
- Terminals are written between ""
- ::= becomes =
- {} repetition 0 or more times
- [] optionality
- () grouping
- (* *) comments
- rules end with .

Ex.2 reloaded in EBNF

Onita Andrei
Integer = ["+" | "-"] "Non0digit" {Number} | {Number}

Mihalcea Leonard

Integer = '0' | ['+'|'-'] ('1'|'2'|...|'9') {'0'|1'|'2'|...|'9'}

Moldovanu Dragos
Number= |

0" | ["-" | "+"]non-zero-digit{digit}
Non-zero-digit="1"|"2"|…|"9"
Digit="0"|"1"|…|"9"

Petcu Dragos

Integer = "0" | ["+"|"-"] nonZeroDigit {digit}

Pascotescu Iuliana

Number = 0 | [-] nonZeroDigit {digit}

Correct EBNF rules for identifiers and constants (discussed in Seminar 2)

Identifiers:

Moldovanu Dragos

identifier=letter{alphanumeric}

alphanumeric="0"|"1"|...|"9"|"A"|...|"z"

Onita Andrei

identifier ::= letter {seq}

seq ::= letter | digit

letter ::= "A" | "B" | . ..| "Z"|"a"|...|"z"

digit ::= "0" | "1" |...| "9"

Costants (char & string):

Moldovanu Dragos
character= " ' " placeholder " ' "
placeholder="0"|…|"9"|"A"|…|"z"|"!"|…|"#"

Neta Razvan
String = ""{letter|digit|specialchar}""

Seminar 2 - Scanning

Tuesday, October 06, 2020 2:38 PM

Input: source code, token list (res. words, sep., op.)
Output: FIP, ST, lex. err. (if any)

Ex. Oana Nourescu

source.txt

```
----------------
program test;
var a:integer;
begin
  a:= b+1;
  c:= 'c';
  s:= "new message"
end.
```

PIF

token	ST_pos
program	-1
Id	0
;	-1
var	-1
Id	1
:	-1
Id	2
;	-1
begin	-1
Id	1
:=	-1
Id	3
+	-1
const	4
;	-1
id	5
:=	-1
const	6
;	-1
id	7
:=	-1
const	8

end	-1
.	-1

ST (only ids & consts)

ST_pos	symbol
0	test
1	a
2	integer
3	b
4	1
5	c
6	'c'
7	s
8	"new message"

Lexical errors ex:
1a – wrong identifier
"new message – missing "
$ - unknown character
'cc' - wrong char const
05 – wrong int constant

GRAMMARS

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

$N = \{S, C\}$

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

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

$\quad C \ \rightarrow S \mid bSb$

$\quad CS \ \rightarrow b$,

prove that $w \ = \ ab(ab^2)^2 \ \in L(G)$.

Sol.:

$a^2b^2 = aabb \ \neq \ (ab)^2 = \ abab$

//Oana Nourescu

$$S \underset{(2)}{=>} \ aCSb \ \underset{(4)}{=>} abSbSb \ \overset{2}{\underset{(1)}{=>}} ababbabb \ = \ w$$

=> $S \overset{4}{=>} w$ **=>** $w \ \in L(G)$

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

$N = \{S\}$

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

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

find $L(G)$.

// Paun Tudor

L = { $a^{2k}bc$, *where* $k \ \in N$ **}**

L = L(G) ?

1. L ⊆ L(G)
2. L(G) ⊆ L

//Ariana Hategan

1. $\forall \quad k \in N, a^{2k}bc \in L(G)$

   $P(k) : a^{2k}bc \in L(G)$ **and prove that P(k) true for** $\forall$ $k \in N$

   a) Verification step

   $P(0) : a^{0}bc \in L(G) \Leftrightarrow bc \in L(G)$

   S => bc

   So, P(0) is true

b) $P(n)\ true \rightarrow P(n+1)\ true,\ n \in N$

$P(n)\ true \Rightarrow a^{2n}bc \in L(G) \Rightarrow S \overset{*}{\Rightarrow} a^{2n}bc\ (induction\ hypothesis)$

$S \Rightarrow a^{2}S \overset{*}{\Rightarrow} a^{2}a^{2n}bc = a^{2(n+1)}bc$

(1) (ind hypo)

=> P(n+1) true

a) + b) => (1)

//Razvan Neta

$S \Rightarrow bc$

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

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

$\Rightarrow a^{6}S \Rightarrow$ **…**

$$=> ...$$

It may be noticed that starting from S and using **all productions** in **all possible combinations**, we only get, as sequences of terminals, sequences of the shape $a^{2n}bc,\ n \in N$

It follows that the grammar generates nothing else.

3. Find a grammar that generates $L = \{0^n 1^n 2^m \mid n, m \in \mathbb{N}^*\}$

Sol.:

G=({S,A,B}, {0,1,2}, P, S)

P:

S -> AB

A -> 01 | 0A1

B -> 2 | B2

$? \, L(G) = L$

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

$? \, \forall n, m \in N^*,\ 0^n 1^n 2^m \in L(G)$

Let $n, m \in N^*$ **fixed**

$$S \underset{(1)}{=>} AB \overset{n}{\underset{(i)}{=>}} 0^n 1^n B \overset{m}{\underset{(ii)}{=>}} 0^n 1^n 2^m$$

$$=> S \overset{n+m+1}{=>} 0^n 1^n 2^m => 0^n 1^n 2^m \in L(G)$$

n

(i) $A => 0^n 1^n, \ \forall n \ \in N^*$

m

(ii) $B => 2^m, \ \forall m \ \in N^*$

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

Petcu Dragos

A -> 01 | 0A1
B -> 2 | 2B
S -> AB

Oana Nourescu

S -> AB
A -> 01 | 0A1
B -> 2 | B2

Diaconu Bogdan

S->0A1B
A->0A1|ε
B->2|2B

Moldovanu Dragos

A->01|0A1
B->2|2B
S->AB

Moldovan Vasilica

S -> 0S1C | 012 ! 001212

C -> 2C |epsilon

Pascotescu Iuliana

S -> 0K1 | 01 | T | ST   ! 01, 2

T -> 2T | 2

K -> 0K1 | 01

Onita Andrei

P: S= A B

A= 0 S 1 | 01

B= S 2 | 2

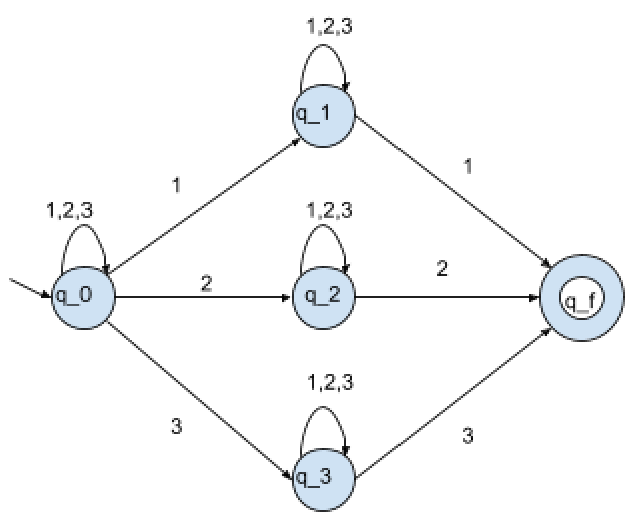
Finite Automata

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

δ	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	$\varnothing$	$\varnothing$	$\varnothing$

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

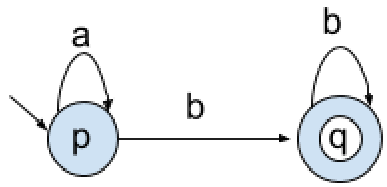
Sol.:
(see chat for graph repres.)



//Nenisca Maria

(q_0, 12321) |- (q_1, 2321) $\vdash^{3}$ (q_1, 1) |- (q_f, epsilon) => $w = 12321 \ \ \in L(M)$

2. Find the language accepted by the FA below.



// Moldovan Vasilica

L = $\{ a^n b^m \mid n \in N,\ m \in N^* \}$

? L = L(M)

1. ? L $\subseteq L(M) \iff \forall\ n \in N,\ m \in N^*,\ a^n b^m \in L(M)$

   Let $n \in N,\ m \in N^*$ be fixed.

   $$(p,\ a^n b^m) \underset{(i)}{\overset{n}{|-}} (p,\ b^m) |- (q,\ b^{m-1}) \underset{(ii)}{\overset{m-1}{|-}} (q,\ \varepsilon)$$

   (i) $(p,\ a^n) \overset{n}{|-} (p,\ \varepsilon),\ \forall n \in N$

   (ii) $(q,\ b^k) \overset{k}{|-} (q,\ \varepsilon),\ \forall\ k \in N$

   1.1 For n = 0: $(p,\ \varepsilon) \overset{0}{|-} (p,\ \varepsilon)$ - True

   1.2 P(k) -> P(k + 1)

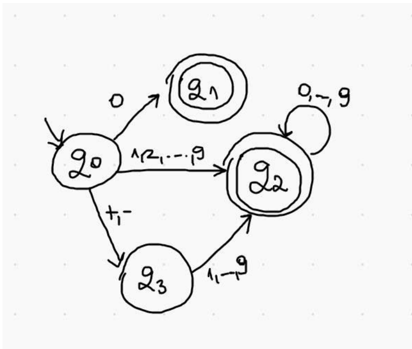
   P(k): $(p,\ a^k) \overset{k}{|-} (p,\ \varepsilon)$ - (induction hypothesis) True

   $(p,\ a^{k+1}) |- (p,\ a^k) \underset{\text{Ind. hyp.}}{\overset{k}{|-}} (p,\ \varepsilon)$ => P(k + 1) is True

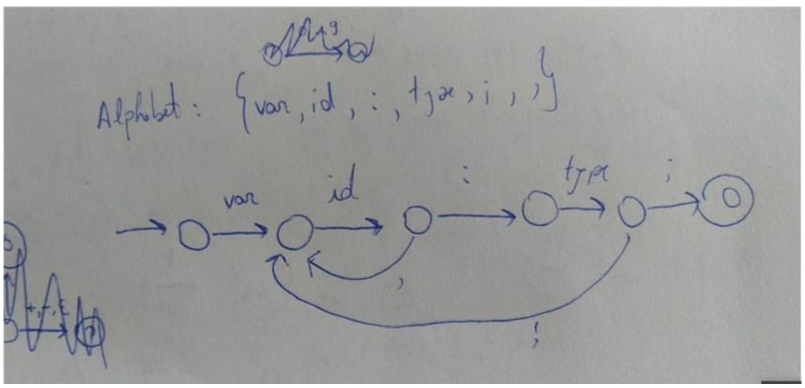
   From 1.1 and 1.2 => (i)

3. Build FAs that accept the following languages (IW activity - Seminar5 chat)

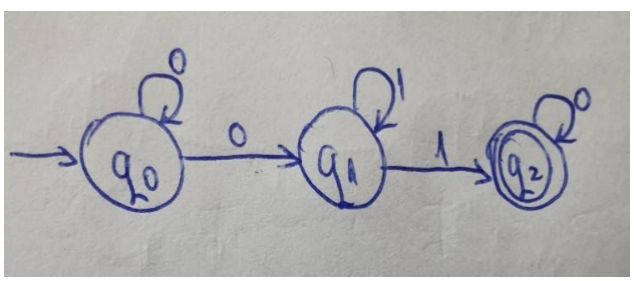
a. Integer numbers



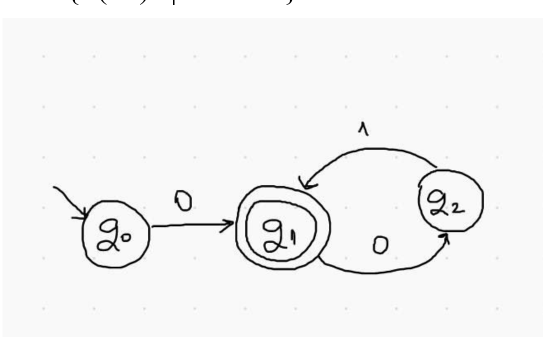
b. Variable declarations (Pascal, C, ...)



c. $L = \{0^n 1^m 0^q \mid n, m \in N^*,\ q \in N\}$



d. $L = \{0(01)^n \mid n \in N\}$



FA ⇔ RG ⇔ RE

I) FA ⇔ RG

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

$$P:\ S \to \varepsilon \mid aA$$
$$A \to aA \mid bA \mid a \mid b,$$

build the equivalent FA.

Sol.:

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

$Q = \{S, A, K\}$

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

$q_0 = S$

$F = \{K, S\}$

$\delta(S, a) = \{A\}$

$\delta(A, a) = \{A, K\}$

$\delta(A, b) = \{A, K\}$

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

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

δ	0	1
p	q	p
q	r	p
r	r	r

build the equivalent right linear grammar.

Sol.:

$G = (N,\ \Sigma,\ P,\ S)$

$N = \{p, q, r\}$

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

$S = p$

$P:\ p \text{ -> } 0q \mid 1p \mid 1 \mid \varepsilon$

$q \text{ -> } 0r \mid 0 \mid 1p \mid 1$
$r \text{ -> } 0r \mid 0 \mid 1r \mid 1$

II) RG ⇔ RE

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

$0: \ G_1 = (\{S_1\},\ \{0,1\},\ \{S_1 \text{ -> } 0\},\ S_1)$
$1: \ G_2 = (\{S_2\},\ \{0,1\},\ \{S_2 \text{ -> } 1\},\ S_2)$
$0+1 \quad G_3 = (\{S_1,\ S_2,\ S_3\},\ \{0,1\},\ \{S_1 \text{ -> } 0,\ S_2 \text{ -> } 1,\ S_3 \text{ -> } 0 \mid 1\},\ S_3)$
$\quad G'_3 = (\{S_3\},\ \{0,1\},\ \{S_3 \text{ -> } 0 \mid 1\},\ S_3)$
$(0+1)^* \quad G_4 = (\{S_3\},\ \{0,1\},\ \{S_3 \text{ -> } 0 \mid 1 \mid \varepsilon \mid 0S_3 \mid 1\ S_3\},\ S_3)$
$\quad G'_4 = (\{S_3\},\ \{0,\ 1\},\ \{S_3 \text{ -> } 0S_3 \mid 1\ S_3 \mid \varepsilon\},\ S_3)$! not regular
$0(0+1)^* \quad G_5 = (\{S_1,\ S_3\},\ \{0,1\},\ \{S_3 \text{ -> } 0S_3 \mid 1\ S_3 \mid \varepsilon,\ S_1 \text{ -> } 0S_3\},\ S_1)$! not regular
$0(0+1)^*1 \quad G_6 = (\{S_1,\ S_2,\ S_3\},\ \{0,1\},\ \{S_2 \text{ -> } 1,\ S_1 \text{ -> } 0S_3,\ S_3 \text{ -> } 0S_3 \mid 1S_3,\ S_3 \text{ -> } S_2\},\ S_1)$
G_6 not regular

Sol.: $G'_6 = (\{S_1,\ S_3\},\ \{0,1\},\ \{S_1 \text{ -> } 0S_3,\ S_3 \text{ -> } 0S_3 \mid 1S_3 \mid 1\},\ S_1)$

4. Give the RE corresponding to the following grammar
$G = (\{S, A, B\},\ \{a, b\},\ P,\ S)$
$P: \ S \rightarrow aA$
$\quad A \rightarrow aA \mid bB \mid b$
$\quad B \rightarrow bB \mid b$

Sol.: //???

S = aA

A = aA + bB + b

B = bB + b

We know that <u>rule 1</u>

X = aX+b

X = a*b

$B = b^*b$ by rule 1

$B = b^+$

$A = aA + B$

$A = aA + b^+$

$A = a^*b^+$ by rule 1

$S = aa^*b^+ = a^+ b^+$

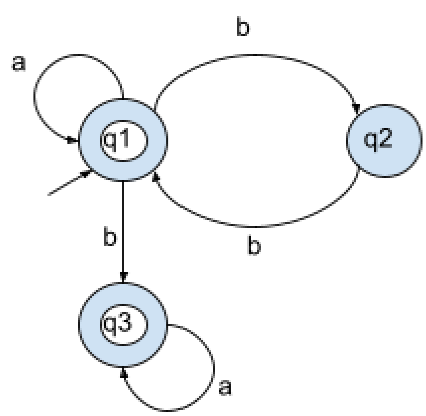
$\Rightarrow S = a^+b^+$

III) FA ⇔ RE

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

Sol: on pdf board attached to MSTeams Seminar7 meet

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



//Hategan Ariana

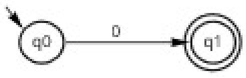
q1 = epsilon + q1 a + q2 b
q2= q1 b
q3=q1 b + q3 a

X = Xa + b => sol: X=ba*

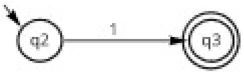
$q3=q1ba^*$
$q1=\epsilon + q1a + q1bb = \epsilon + q1(a+bb) \Rightarrow q1 = (a+bb)^*$

Regular expression: $q1 + q3= (a+bb)^* + (a+bb)^*ba^* = (a+bb)^*(\epsilon+ba^*)$

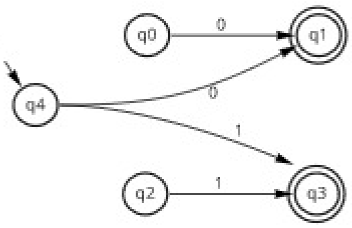
0:



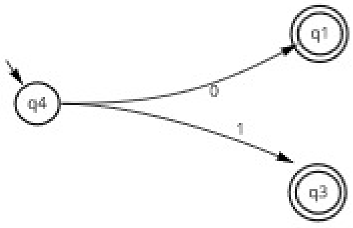
1:



0+1:

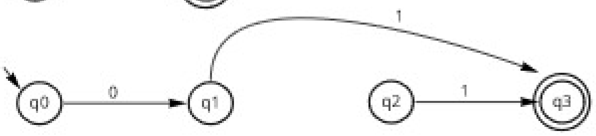


eliminating inaccessible states

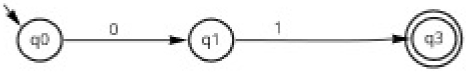


minimize

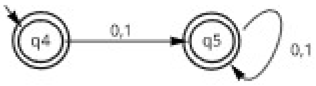
01:



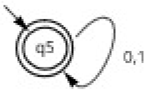
eliminating inaccessible states



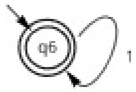
(0+1)*:



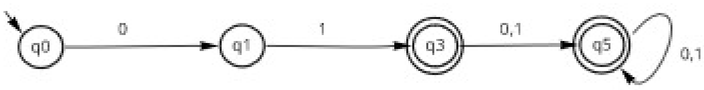
minimize



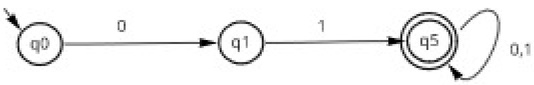
1*:



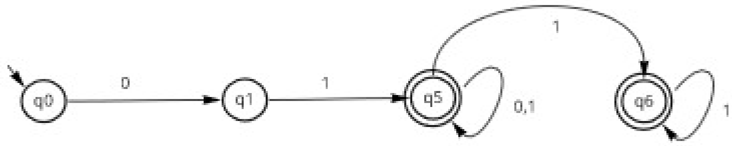
01(0+1)*:



minimize



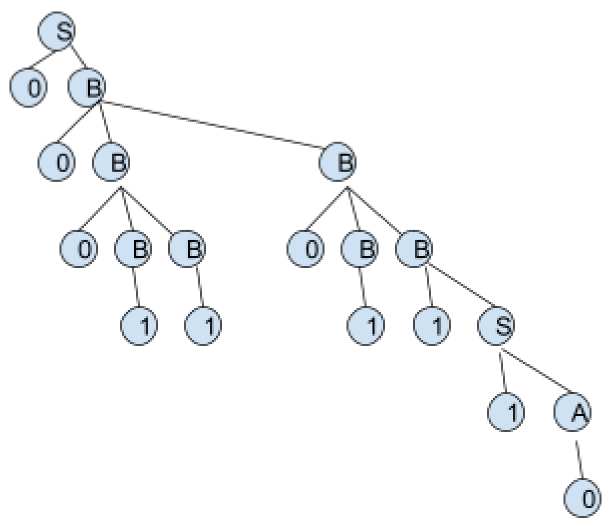
01(0+1)*1*:



CFG

1. Given the CFG grammars below, give a leftmost/rightmost derivation for w.
   a. $G = (\{S, A, B\},\ \{0,1\},\ \{S \to 0B \mid 1A,\ A \to 0 \mid 0S \mid 1AA,\ B \to 1 \mid 1S \mid 0BB\})$,
   $w\ =\ 0001101110$
   Sol.

   @B: Moca David
   Leftmost: 1 8 8 6 6 8 6 7 2 3
   $S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 000BBB \Rightarrow 0001BB \Rightarrow 00011B \Rightarrow 000110BB \Rightarrow 0001101B$
   $\Rightarrow 00011011S\ \Rightarrow 000110111A \Rightarrow 0001101110$

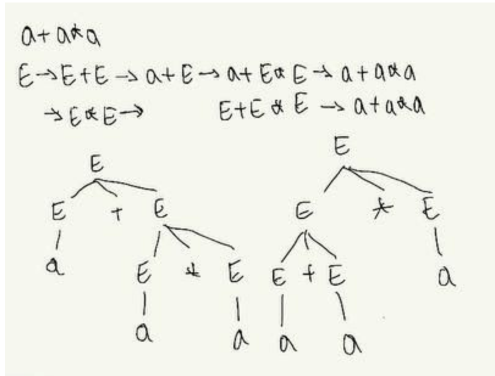


   @B: Neta Razvan
   Rightmost: 1 8 7 2 3 8 6 7 2 3
   S $\Rightarrow$0B $\Rightarrow$ 00BB $\Rightarrow$00B1S $\Rightarrow$00B11A $\Rightarrow$ 00B110 $\Rightarrow$ 000BB110
   $\Rightarrow$000B1110 $\Rightarrow$0001S1110 $\Rightarrow$00011A1110 $\Rightarrow$ 0001101110

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

2. Prove that the following grammars are ambiguous
   a. $G_1 = (\{S, B, C\},\ \{a, b, c\},\ \{S \to abC \mid aB,\ B \to bC,\ C \to c\},\ S)$ HW
   b. $G_2 = (\{E\},\ \{a, +, *, (,)\},\ \{E \to E + E \mid E * E \mid (E) \mid a\},\ E)$

   Sol. IW #inmeet



w=a+a*a

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

Recursive descendent parser

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

Sol.: VA+B: Dragos P., Andrei O.

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

Parse tree repr. As seq of production nos: $S_1S_2S_3S_3$

LL(1) parser

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

$P:$ (1) $S \to BA$

(2) $A \to +BA$

(3) $A \to \varepsilon$

(4) $B \to DC$

(5) $C \to *DC$

(6) $C \to \varepsilon$

(7) $D \to (S)$

(8) $D \to a$,

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

1) Compute FIRST @B Iuliana Pascotescu

	F_0	F_1	F_2	F_3 = F_2 = FIRST
S	{}	{}	{(, a}	{(, a}
A	{+, ε}	{+, ε}	{+, ε}	{+, ε}
B	{}	{(, a}	{(, a}	{(, a}
C	{*, ε}	{*, ε}	{*, ε}	{*, ε}
D	{(, a}	{(, a}	{(,a}	{(, a}

FIRST(S) = {(, a}
FIRST(A) ={+, ε}
FIRST(B) ={(, a}
FIRST(C) ={*, ε}
FIRST(D) ={(, a}

2) Compute FOLLOW @B Moca David

	L_0	L_1	L_2	L_3	L_4 = L_3 = FOLLOW
S	{ε}	{ε ,)}	{ε ,)}	{ε ,)}	{ε ,)}
A	{}	{ε}	{ε,)}	{ε,)}	{ε,)}
B	{}	{+, ε}	{+, ε,)}	{+, ε,)}	{+, ε,)}
C	{}	{}	{+, ε}	{+, ε,)}	{+, ε,)}
D	{}	{*}	{*,+, ε}	{*,+, ε,)}	{*,+, ε,)}

FOLLOW(S) = {ε ,)}
FOLLOW(A) ={ε,)}
FOLLOW(B) ={+, ε,)}
FOLLOW(C) ={+, ε,)}
FOLLOW(D) ={*,+, ε,)}

3) Fill LL(1) parsing table @B Iuliana Pascotescu

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

+		pop				
*			pop			
(				pop		
)					pop	
\$						acc

4) Parse the sequence @B Dragos P.

$(a*(a+a)\$,\ S\$,\ \varepsilon) \vdash (a*(a+a)\$,\ BA\$,\ 1) \vdash$
$(a*(a+a)\$,\ DCA\$,\ 1\ 4) \vdash (a*(a+a)\$,\ aCA\$,\ 1\ 4\ 8) \vdash$
$(\ *(a+a)\$,\ CA\$,\ 1\ 4\ 8) \vdash (\ *(a+a)\$,\ *DCA\$,\ 1\ 4\ 8\ 5) \vdash$
$((a+a)\$,\ DCA\$,\ 1\ 4\ 8\ 5) \vdash ((a+a)\$,\ (S)CA\$,\ 1\ 4\ 8\ 5\ 7) \vdash$
$(a+a)\$,\ S)CA\$,\ 1\ 4\ 8\ 5\ 7) \vdash (a+a)\$,\ BA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1) \vdash$
$(a+a)\$,\ DCA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4) \vdash (a+a)\$,\ aCA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8) \vdash$
$(\ +a)\$,\ CA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8) \vdash (\ +a)\$,\ A)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6) \vdash$
$(\ +a)\$,\ +BA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2) \vdash (a)\$,\ BA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2) \vdash$
$(a)\$,\ DCA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4) \vdash$
$(a)\$,\ aCA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8) \vdash$
$()\$,\ CA)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8) \vdash$
$()\$,\ A)CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8\ 6) \vdash$
$()\$,\ )CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8\ 6\ 3) \vdash$
$(\$,\ CA\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8\ 6\ 3) \vdash$
$(\$,\ A\$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8\ 6\ 3\ 6) \vdash$
$(\$,\ \$,\ 1\ 4\ 8\ 5\ 7\ 1\ 4\ 8\ 6\ 2\ 4\ 8\ 6\ 3\ 6\ 3) \vdash$
acc

LR(0) parser

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

P: $S' \to S$

(1) $S \to aA$

(2) $A \to bA$

(3) $A \to c$

$w = abbc$

Obs.: LR(0) item $[A \to \alpha.\beta]$

1. Compute the canonical collection of states @B Razvan Neta

$s_0 = closure(\{[S' \to .S]\}) = \{[S' \to .S], [S \to .aA]\}$
$s_1 = goto(s_0, S) = closure(\{[S' \to S.]\}) = \{[S' \to S.]\}$
$s_2 = goto(s_0,\ a) = closure(\{[S \to a.A]\}) = \{[S \to a.A],\ [A \to .bA],\ [A \to .c]\}$
$s_3 = goto(s_2,\ A) = closure(\{[S \to aA.]\}) = \{[S \to aA.]\}$
$s_4 = goto(s_2, b) = closure(\{[A \to b.A]\}) = \{[A \to b.A],\ [A \to .bA], [A \to .c]\}$
$s_5 = goto(s_2, c) = closure(\{[A \to c.]\}) = \{[A \to c.]\}$
$s_6 = goto(s_4, A) = closure(\{[A \to bA.]\}) = \{[A \to bA.]\}$
$goto(s_4,\ b) = closure(\{[A \to b.A]\}) = s_4$
$goto(s_4,\ c) = closure(\{[A \to c.]\}) = s_5$

2. Fill in LR(0) parsing table @B Iuliana Pascotescu

	ACTION	**GOTO**				
		a	**b**	**c**	**S**	**A**
0	shift	2			1	
1	acc					
2	shift		4	5		3

3	reduce 1					
4	shift		4	5		6
5	reduce 3					
6	reduce 2					

3. Parse the input sequence @B Nenisca Maria

work stack	**input stack**	**output band**
\$0	abbc\$	ε
\$0a2	bbc\$	ε
\$0a2b4	bc\$	ε
\$0a2b4b4	c\$	ε
\$0a2b4b4c5	\$	ε
\$0a2b4b4A6	\$	3
\$0a2b4A6	\$	2,3
\$0a2A3	\$	2,2,3
\$0S1	\$	1,2,2,3
acc	\$	1,2,2,3

SLR parser

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

P: $S' \to E$

(1) $E \to T$

(2) $E \to E + T$

(3) $T \to (E)$

(4) $T \to id$

(5) $T \to const$

$w = id + const$

FOLLOW(E) $=\{\varepsilon,+,)\}$

FOLLOW(T) = $\{\varepsilon,+,)\}$

1. Canonical collection Onita Andrei @B: Bogdan Diaconu

$s_0 = closure(\{[S' \to .E]\}) = \{[S' \to .E], [E \to .T], [E \to .E + T], [T \to .(E)],$
$[T \to .id], [T \to .const]\}$

$s_1 = goto(s_0, E) = closure(\{[S' \to E.], [E \to E. + T]\}) = \{[S' \to E.], [E \to E. + T]\}$

$s_2 = goto(s_0, T) = closure(\{[E \to T.]\}) = \{[E \to T.]\}$

$s_3 = goto(s_0, () = closure(\{[T \to (.E)]\}) = \{[T \to (.E)], [E \to .T], [E \to .E + T], [T \to .(E)],$
$[T \to .id], [T \to .const]\}$

$s_4 = goto(s_0, id) = closure(\{[T \to id.]\}) = \{[T \to id.]\}$

$s_5 = goto(s_0, const) = closure(\{[T \to const.]\}) = \{[T \to const.]\}$

$s_6 = goto(s_1, +) = closure(\{[E \to E + .T]\}) = \{[E \to E + .T], [T \to .(E)], [T \to .id]$
$[T \to .const]\}$

$s_7 = goto(s_3, E) = closure(\{[T \to (E.)], [E \to E. + T]\}) = \{[T \to (E.)], [E \to E. + T]\}$

$goto(s_3, T) = closure(\{[E \to T.]\}) = \{[E \to T.]\} = s_2$

$goto(s_3, () = closure(\{[T \to (.E)]\}) = s_3$

$goto(s_3, id) = closure(\{[T \to id.]\}) = s_4$

$goto(s_3, const) = closure(\{[T \rightarrow const.]\}) = s_5$

$s_8 = goto(s_6, T) = closure(\{[E \rightarrow E + T.]\}) = \{[E \rightarrow E + T.]\}$

$goto(s_6, () = closure(\{[T \rightarrow (.E)]\}) = s_3$

$goto(s_6, id) = closure(\{[T \rightarrow id.]\}) = s_4$

$goto(s_6, const) = closure(\{[T \rightarrow const.]\}) = s_5$

$s_9 = goto(s_7,)\,) = closure(\{[T \rightarrow (E).]\}) = \{[T \rightarrow (E).]\}$

$goto(s_7, +) = closure(\{[E \rightarrow E + .T]\}) = s_6$

2. SLR table **@B: Iuliana Pascotescu**

	ACTION						GOTO	
	+	(	)	id	const	$	E	T
0		Shift 3		Shift 4	Shift 5		1	2
1	Shift 6					acc		
2	Reduce 1		Reduce 1			Reduce 1		
3		Shift 3		Shift 4	Shift 5		7	2
4	Reduce 4		Reduce 4			Reduce 4		
5	Reduce 5		Reduce 5			Reduce 5		
6		Shift 3		Shift 4	Shift 5			8
7	Shift 6		Shift 9					
8	Reduce 2		Reduce 2			Reduce 2		
9	Reduce 3		Reduce 3			Reduce 3		

3. Parse the sequence @B: Petcu Dragos

Work stack	**Input stack**	**Output band**
\$0 **\$0id4** **\$0T2** **\$0E1** **\$0E1+6** **\$0E1+6const5** **\$0E1+6T8** **\$0E1** **acc**	**id+const\$** **+const\$** **+const\$** **+const\$** **const\$** **\$** **\$** **\$**	ε **4** **1,4** **1,4** **1,4** **5,1,4** **2,5,1,4**

$$E \underset{2}{=>} E + T \underset{5}{=>} E + const \underset{1}{=>} T + const \underset{4}{=>} id + const$$

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 = abab

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

FIRST(A) = {a,b}

FIRST(S) ={a,b}

1. Canonical collection
//Onita Andrei
//Nourescu Oana

$s_0 = closure(\{[S' \to .S,\ \$]\}) = \{[S' \to .S,\ \$],\ [S \to .AA,\ \$], [A \to .aA, a], [A \to .aA, b],$
$, [A \to .b,\ a],\ [A \to .b,\ b]\}$
$s_1 = goto(s_0,\ S) = closure(\{[S' \to S.,\ \$]\}) = \{[S' \to S.,\ \$]\}$
$s_2 = goto(s_0,\ A) = closure($
$\{[S \to A.A,\ \$]\}) = \{[S \to A.A,\ \$],\ [A \to .aA, \$], [A \to .b, \$]\}$
$s_3 = goto(s_0,\ a) = closure(\{[A \to a.A, a], [A \to a.A, b]\}) = \{[A \to a.A, a], [A \to a.A, b]$
$, [A \to .aA, a], [A \to .b, a],\ [A \to .aA, b], [A \to .b, b]\}$
$s_4 = goto(s_0,\ b) = closure(\{[A \to b.,\ a],\ [A \to b.,\ b]\}) = \{[A \to b.,\ a],\ [A \to b.,\ b]\}$
$s_5 = goto(s2,\ A) = closure(\{[S \to AA.,\ \$]\}) = \{[S \to AA.,\ \$]\}$
s6 = goto(s2, a) = closure({[A -> a.A, $]}) = {[A -> a.A, $], [A -> .aA, $], [A -> .b, $]}
s7 = goto(s2, b) = closure({ $[A \to b., \$]\}$) = {[A -> b., $]}
s8 = goto(s3, A) = closure({[A -> aA., a], [A -> aA., b]}) = {[A -> aA., a], [A -> aA., (Eb]}
goto(s3, a) = closure({[A -> a.A, a], [A -> a.A, b]}) = s3
goto(s3, b) = closure({[A -> b., a], [A -> b., b]}) = s4
s9 = goto(s6, A) = closure({[A -> aA., $]}) = {[A -> aA., $]}
goto(s6, a) = closure({[A -> a.A, $]}) = s6
goto(s6, b) = closure({[A -> b., $]}) = s7

2. LR(1) table
// Nourescu Oana

	ACTION			**GOTO**	
	a	**b**	**$**	**S**	**A**
0	shift 3	shift 4		1	2
1			accept		
2	Shift 6	Shift 7			5
3	Shift 3	Shift 4			8

4	Reduce 3	Reduce 3			
5			Reduce 1		
6	Shift 6	Shift 7			9
7			Reduce 3		
8	Reduce 2	Reduce 2			
9			Reduce 2		

3. Parse the sequence

// Nourescu Oana

Work stack	Input stack	Output band
\$0 \$0a3 \$0a3b4 \$0a3A8 \$0A2 \$0A2a6 \$0A2a6b7 \$0A2a6A9 \$0A2A5 \$0S1 accept	abab\$ bab\$ ab\$ ab\$ ab\$ b\$ \$ \$ \$ \$	 3 23 23 23 323 2323 12323 **12323**

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

// Nourescu Oana

Work stack	Input stack	Output band

\$ s0
\$ s0 a s36
\$ s0 a s36 a s36
\$ s0 a s36 a s36 a s36
\$ s0 a s36 a s36 a s36 b s47
\$ s0 a s36 a s36 a s36 A s89
\$ s0 a s36 a s36 A s89
\$ s0 a s36 A s89
\$ s0 A s2
	a a a b \$
	a a b \$
	a b \$
	b \$
	\$
	\$
	\$
	\$
	\$
		Eps
		Eps
		Eps
		Eps
		Eps
		3
		23
		223
		2223

The sequence is syntactically incorrect

PDA’s

Ex.: Find PDA’s that accept the following languages:

1. $L_1 = \{ww^R \mid w \in \{a, b\}^+\}$
2. $L_2 = \{a^n b^{2n} \mid n \in N^*\}$
3. $L_3 = \{a^n b^{2n} \mid n \in N\}$
4. $L_4 = \{a^{2n} b^n \mid n \in N^*\}$

Sol.:

2,3 - see the test for Seminar 13

1. M=({q0, q1, q2}, {a,b}, {A, B, Z}, d, q0, Z, {q2}) // Ariana Hategan

d(q0,a,Z) = {(q0, AZ)}
d(q0,b,Z) = {(q0, BZ)}

d(q0,a,A) = {(q0, AA), (q1, epsilon)}
d(q0,a,B) = {(q0, AB)}
d(q0,b,A) = {(q0, BA)}
d(q0,b,B) = {(q0, BB), (q1, epsilon)}

d(q1,a,A) = {(q1, epsilon)}
d(q1,b,B) = {(q1, epsilon)}

d(q1,epsilon,Z) = {(q2, Z)}

(q0,abba,Z) |- (q0, bba, AZ) |- (q0, ba, BAZ) |- (q1, a, AZ) |- (q1, epsilon, Z) |- (q2, epsilon, Z)
=> accepted
(q0,abaa,Z) |- (q0, baa, AZ) |- (q0, aa, BAZ) |- (q0, a, ABAZ) |- (q0, epsilon, AABAZ)
|- (q1, epsilon, BAZ)
=> not accepted

- d(q0, epsilon, Z) = {(q2, Z)} - transition to add in order to also accept the empty sequence ($L'_1 = \{ww^R \mid w \in \{a, b\}^*\}$)

4. M=({q0, q1, q2, q3}, {a,b}, {A, Z}, d, q0, Z, {q3}) // Ariana Hategan

d(q0,a,Z) = {(q1, AZ)}
d(q1,a,A) - {(q0, AZ)}
d(q0,a,A) = {(q1, AA)}
d(q1,a,A) = {(q0, AA)}
d(q0,b,A) = {(q2, epsilon)}
d(q2,b,A) = {(q2, epsilon)}
d(q2,epsilon,Z) = {(q3,Z)}

(q0, aaaabb, Z) |- (q1, aaabb, AZ) |- (q0, aabb, AZ) |- (q1, abb, AAZ) |- (q0, bb, AAZ) |- (q2, b, AZ) |- (q2, epsilon, Z) |- (q3, epsilon, Z) => aaaabb is accepted

Attribute Grammars

Ex.: Give an attribute grammar for evaluating simple arithmetic expressions with id, (,), +, *

Sol.:

S - attributed grammar

E -> E + T	{E1.val = E2.val + T.val}
E -> T	{E..val = T.val}
T - > T * F	{T1.val = T2.val * F.val}
T -> F	{T.val = F.val}
F -> (E)	{F.val = E.val}
F - > id	{F. val = id.val}

Obs.: The green arrows from the syntax tree below indicate how evaluation is performed (bottom-up - value of attribute in parent is computed based on values of attributes in descendants, attributes are synthesized) w = 2 + 3 * 4