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 combinaton of terminals and/or nonterminals, i=1,n

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

[redacted]

<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

[redacted]

<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

[redacted]

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

nonzerodigit = "1" | … | "9"

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

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

Sol.: [redacted]

2

S => aCSb => abSbSb => ababbabb
(2) (4) (1)

4

=> S => ababbabb = 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 $L(G)$.

Sol.: [redacted]

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

? $L \ = \ L(G)$

(1) ? $L \subseteq L(G)$ (all sequences of that shape are generated by G)

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

Take P(n): $a^{2n}bc \in L(G)$ and prove P(n) true, $\forall\, n \in \mathbb{N}$

We'll prove by mathematical induction

(a) Verification step: ? P(0): $a^0bc \in L(G)$ is true

$S \underset{(2)}{\Rightarrow} bc = a^0bc$ => P(0) true

(b) Proof step: We suppose P(k) is true and then prove that P(k+1) is also true, where $k \in \mathbb{N}$

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

$S \underset{(1)}{\Rightarrow} a^2S \underset{\text{(ind. hypo.)}}{\overset{*}{\Rightarrow}} a^2a^{2k}bc = a^{2(k+1)}bc$

=> $S \overset{*}{\Rightarrow} a^{2(k+1)}bc$ => P(k+1) is true

(a) + (b) => (1)

(2) ? $L \supseteq L(G)$ (G generates only sequences of that shape)

S => bc = a^0bc

=> a^2S => a^2bc

=> a^4S => a^4bc

=> a^6S => …

We notice that starting from S and using all grammar productions in all possible combinations, we only get, as sequences of terminals, sequences of the shape $a^{2n}bc$ where $n \in \mathbb{N}$. It follows that the grammar doesn't generate anything else.

Obs.: This inclusion may also be discharged by induction.

----

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

Sol.: [redacted]

$G = (N, \Sigma, P, S)$
$N = \{S, V, C\}$
$\Sigma = \{0, 1, 2\}$
P : S -> VC
V -> 0V1 | 01
C -> 2 | 2C

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

Let $n, m \in N^*$

$$S \underset{(1)}{\Rightarrow} VC \overset{n}{\underset{(a)}{\Rightarrow}} 0^n1^n C \overset{m}{\underset{(b)}{\Rightarrow}} 0^n1^n2^m \Rightarrow S \overset{*}{\Rightarrow} 0^n1^n2^m \Rightarrow 0^n1^n2^m \in L(G)$$

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

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

HW: Prove (a) and (b) above by induction
Justify the reverse inclusion

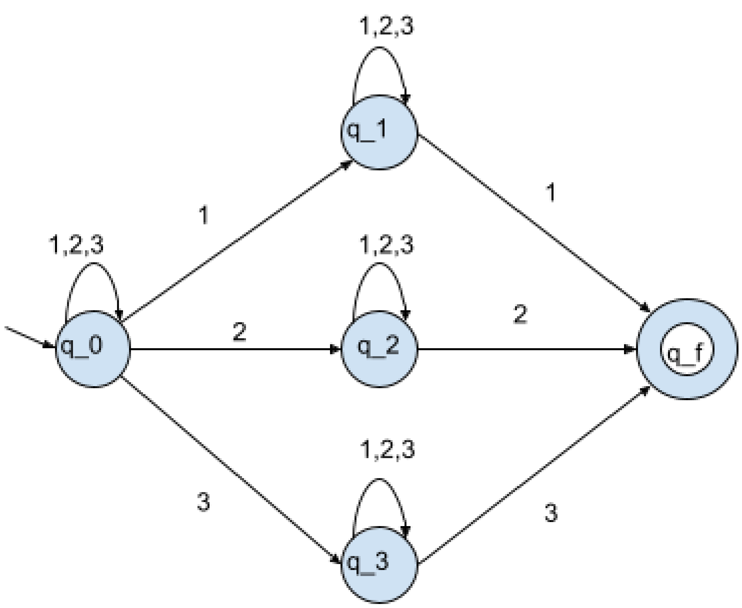
FINITE AUTOMATA (FA)

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

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	$\oslash$	$\oslash$	$\oslash$

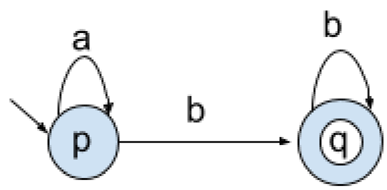
Prove that $w = 12321 \in L(M)$

Sol.: B: [redacted]



$(q_0,\ 12321)\ |-\ (q_1, 2321) \overset{3}{|-} (q_1, 1) \,|-\, (q_f, \varepsilon)$ => $(q_0, w) \overset{*}{|-}\ (q_f, \varepsilon)$ =>
$w\ \in\ L(M)$

2. Find the language accepted by the FA below.



Sol.: B: [redacted]

$L = \{a^n b^m |\ n \in N,\ m \in N^*\}$
? $L = L(M)$

1. $?\, L \subseteq L(M)$ (all sequences of that shape are accepted by M)
   $\forall n \in N,\ \forall m \in N^*\quad a^n b^m\ \in L(M)$
   Let $n \in N,\ m \in N^*$
   $(p,\ a^n b^m) \underset{a}{\overset{n}{|-}}\ (p,\ b^m) \,|-\, (q,\ b^{m-1}) \underset{b}{\overset{m-1}{|-}}\ (q,\ \varepsilon)$ => $a^n b^m\ \in L(M)$

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

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

   $a).\ P(n):\ (p,\ a^n) \overset{n}{|-} (p,\ \varepsilon)$

   $P(0):\quad (p,\ \varepsilon) \overset{0}{|-} (p,\ \varepsilon)$, P(0) - true
   ? $P(k)\ -\ true\ =\ >P(k+1)\ \ -true$

   $P(k) -\ true$ => $(p,\ a^k) \overset{k}{|-} (p,\ \varepsilon)$ (induction hypothesis)

   k k+1

$(p,\ a^{k+1}) \mid\!- \ (p,\ a^k) \mid\!- \ (p,\ \varepsilon)$ => $(p,\ a^{k+1}) \mid\!- (p,\ \varepsilon)$ => P(k+1) -true

Ind. hyp.

Similarly, we demonstrate b.

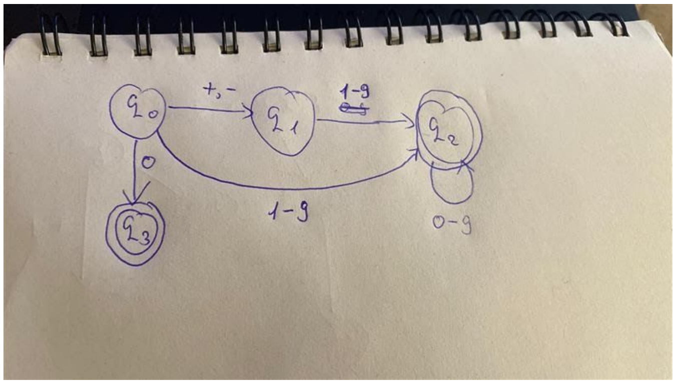
2. ? $L(M) \subseteq L$ (M does not accept anything else but sequences of that shape)

   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 $a^n b b^k,\ n, k \in N$

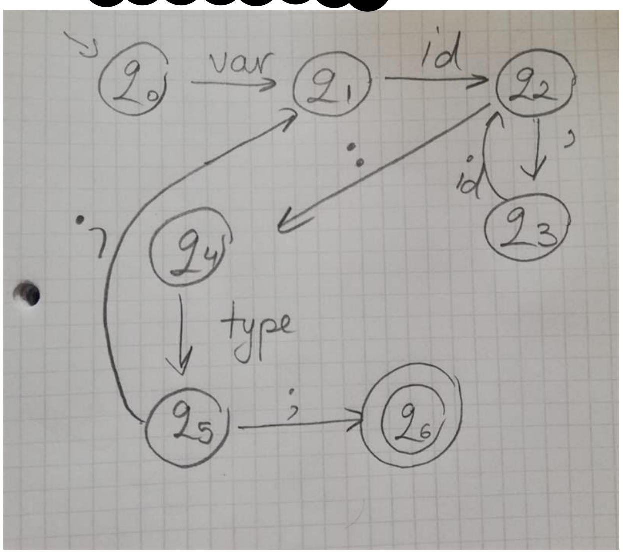
   Obs. In order for such a reasoning to count as proof, you should make sure that you have covered all paths from initial state to final states.

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.

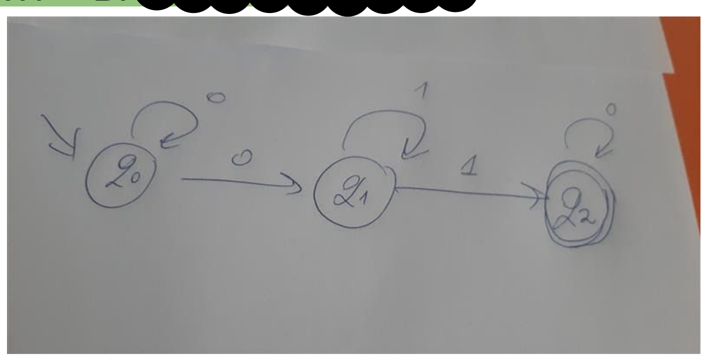
      a. IW -> [redacted]



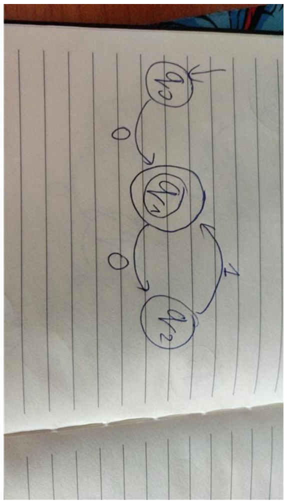
b. IW->[redacted]



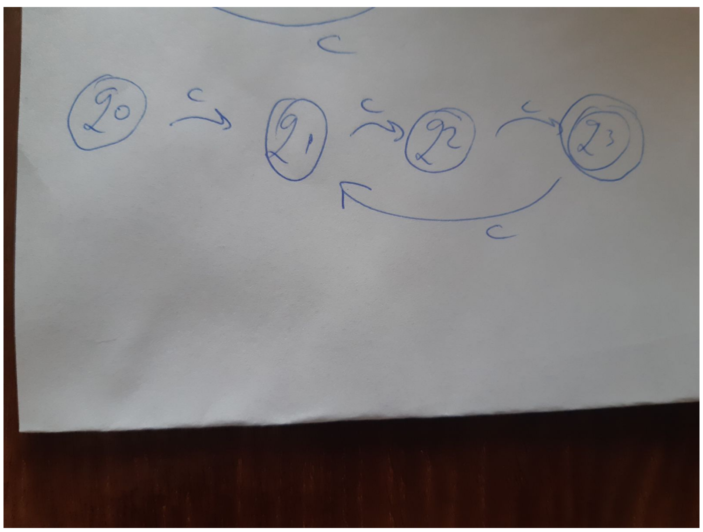
c. IW ->B:[redacted]



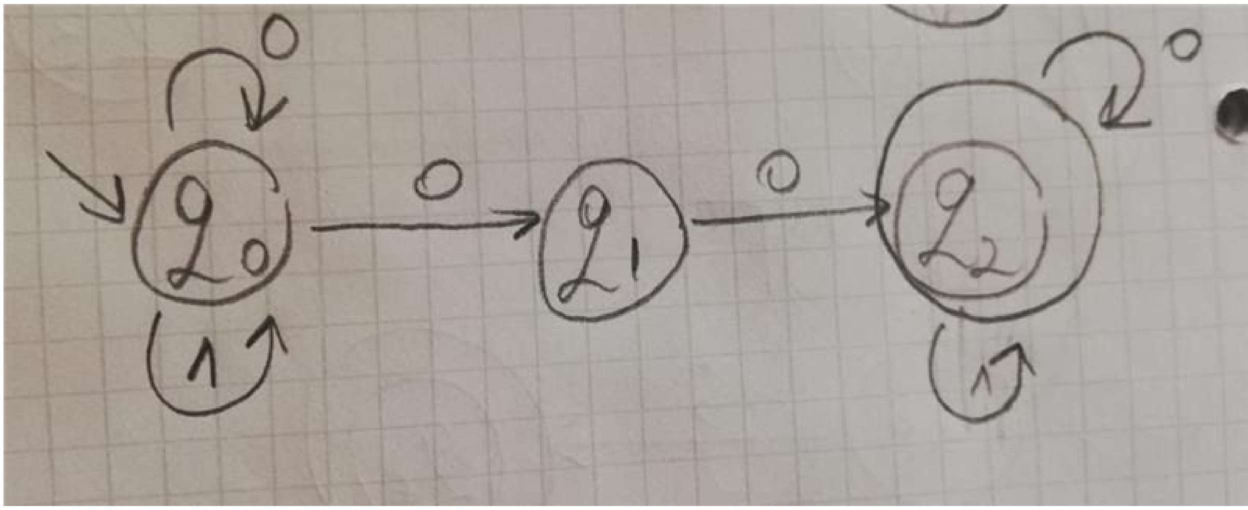
d. IW ->B:[redacted]



e. IW ->B: [redacted] (+mark q as initial state)



f. IW->B: [redacted]



0
q0
0
q1
0
q2
0
1
1

FA ⇔ RG ⇔ RE

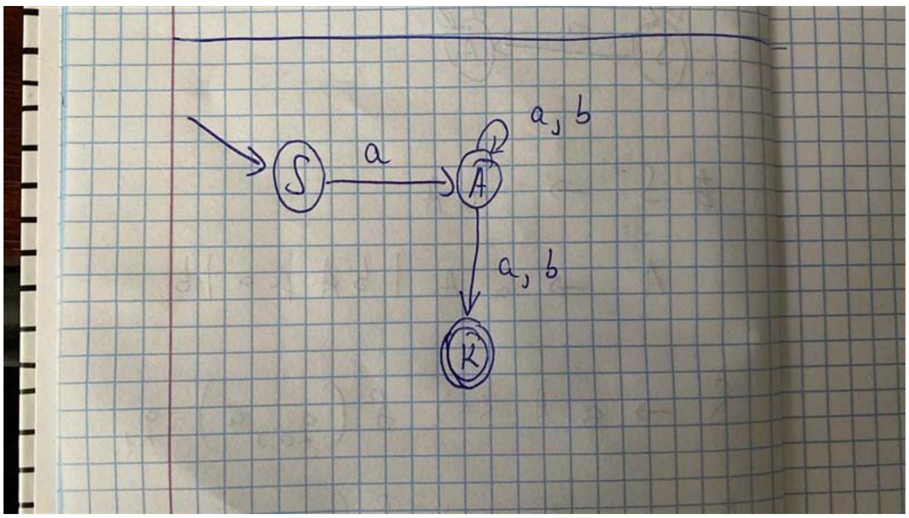
I) FA ⇔ RG (team work)

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

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

build the equivalent FA.

Sol.:



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

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

build the equivalent FA.

Sol.:

$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$
$Q = \{S,\ A,\ K\},\ q_0 = S,\ F = \{K,\ S\},\ \Sigma = \{a,\ b\}$

δ	a	b
S	$\{A\}$	$\varnothing$
A	$\{A,\ K\}$	$\{A,\ K\}$
K	$\varnothing$	$\varnothing$

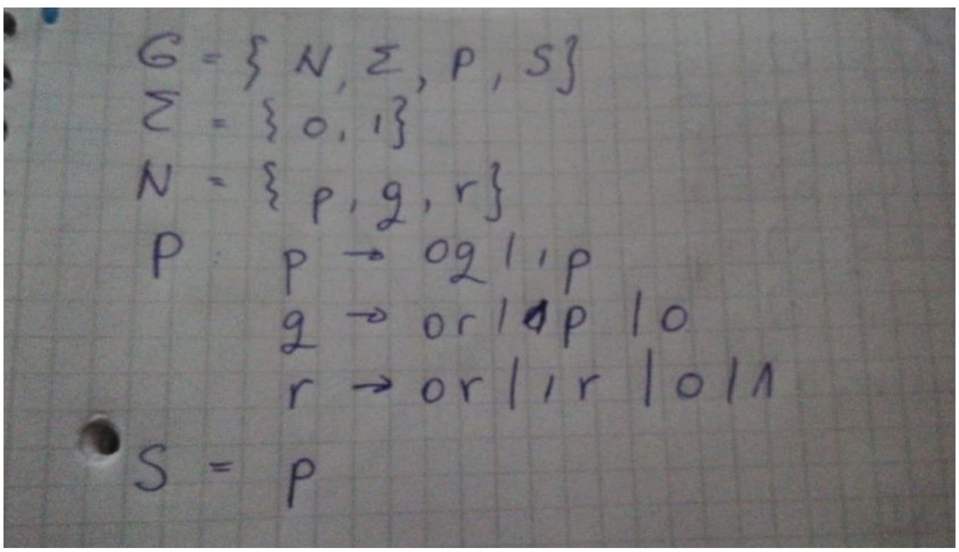
T3. Given the following FA $M = (Q, \Sigma, \delta, q_0, F)$

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

δ	0	1
p	q	p
q	r	p
r	r	r

build the equivalent right linear grammar.

Sol.:



$G = \{N, \Sigma, P, S\}$

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

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

P: $p \to 0q \mid 1p$

$q \to 0r \mid 1p \mid 0$

$r \to 0r \mid 1r \mid 0 \mid 1$

$S = p$

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

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

δ	0	1
p	q	p
q	r	p
r	r	r

build the equivalent right linear grammar.

Sol.:

$G = \{N, \Sigma, P, S\}$

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

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

$S = p$

$P: \quad p \to 0q \mid 1p \mid 1 \mid \varepsilon$

$\qquad q \to 0r \mid 1p \mid 0 \mid 1$

$\qquad r \to 0r \mid 1r \mid 0 \mid 1.$

II) RG ⇔ RE

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

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

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

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

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

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

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

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

! not regular

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

$G_6 = (\{S_1, S_2, S_3\}, \{0,1\}, \{S_1 \rightarrow 0S_3, S_3 \rightarrow 0S_3 \mid 1S_3 \mid S_2, S_2 \rightarrow 1\}, S_1)$! 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
$G = (\{S, A, B\},\ \{a, b\},\ P,\ S)$
$P:\ S \to aA$
$\quad A \to aA \mid bB \mid b$
$\quad B \to bB \mid b$

Sol.: T4

$S = aA$
$A = aA + bB + b$
$B = bB + b \quad \Rightarrow \quad B = b^*b = b^+$

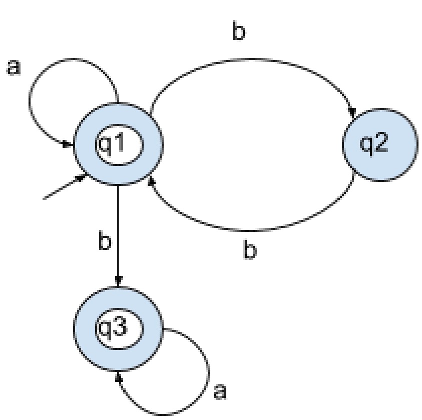
$\Rightarrow A = aA + B$
$A = a^*B = a^*b^+$
$S = aA \quad \Rightarrow \quad S = aa^*b = a^+b^+$

III) FA ⇔ 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.



//[redacted]

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

$q_2 = q_1 b$

$q_3 = q_1 b + q_3 a$

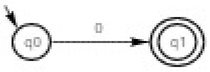
$X = Xa + b \Rightarrow X = ba^*$ *solution* [redacted]

$q_3 = q_1 ba^*$

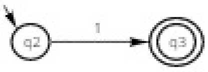
$q_1 = \varepsilon + q_1 a + q_1 bb = q_1(a + bb) + \varepsilon \Rightarrow q_1 = (a + bb)^* \Rightarrow q_3 = (a + bb)^* ba^*$

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

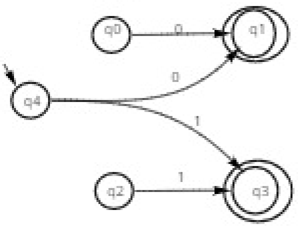
0:



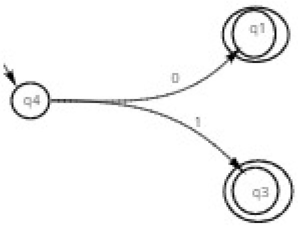
1:



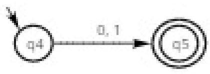
0+1:



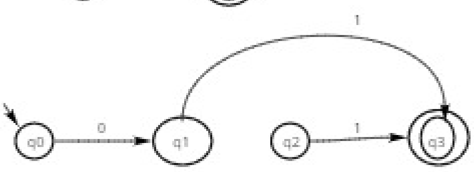
eliminate innaccessible



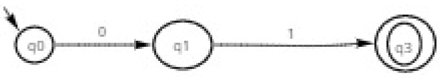
further simplify



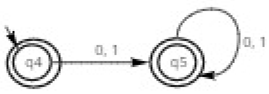
01:



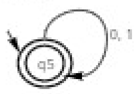
eliminate innaccessible



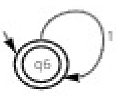
(0+1)*:



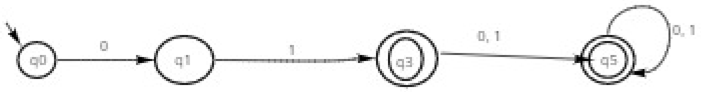
simplify



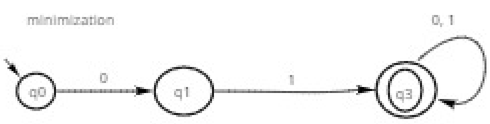
1*



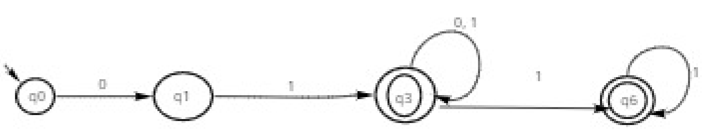
01(0+1)*:



minimization



01(0+1)*1*:



equivalent to the previous one

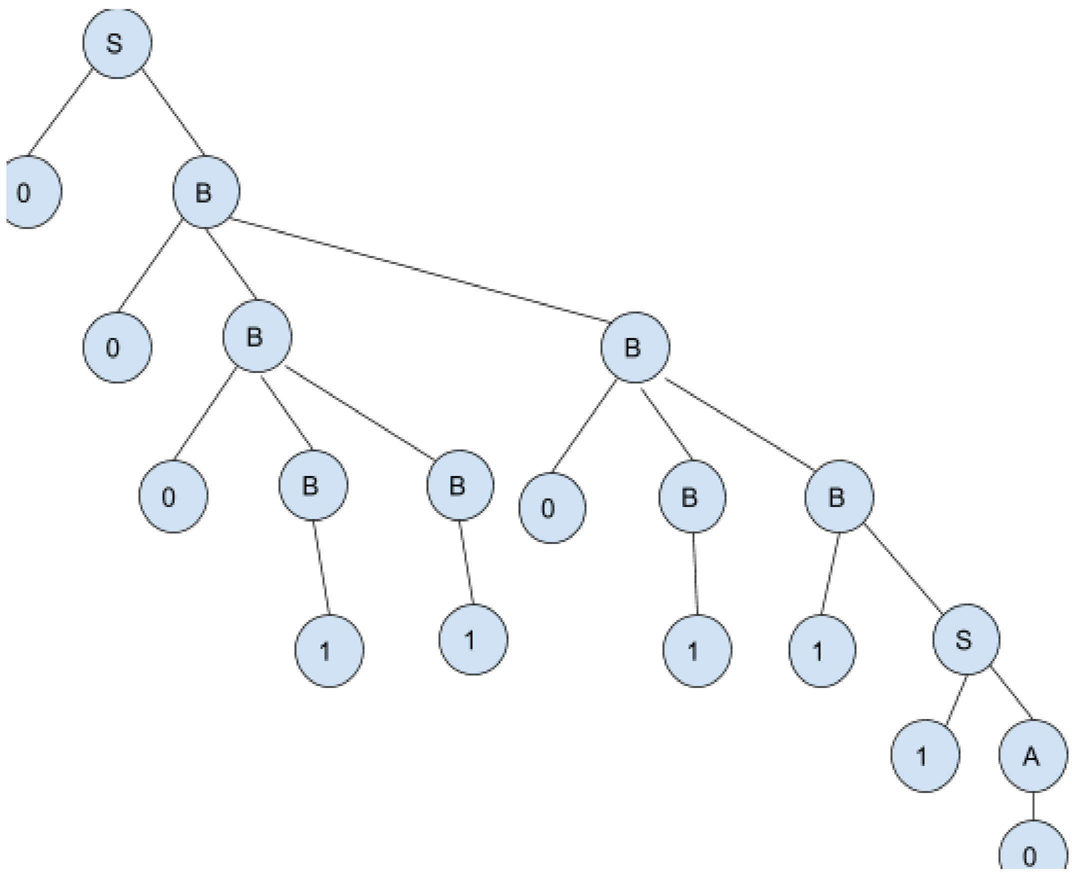
CFG

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

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

   Sol. [redacted]

   Leftmost: 1886686723
   $S => 0B => 00BB \Rightarrow 000BBB \Rightarrow 0001BB \Rightarrow 00011B \Rightarrow 000110BB \Rightarrow 0001101B$
   $\Rightarrow 00011011S \Rightarrow 000110111A \Rightarrow 0001101110$



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

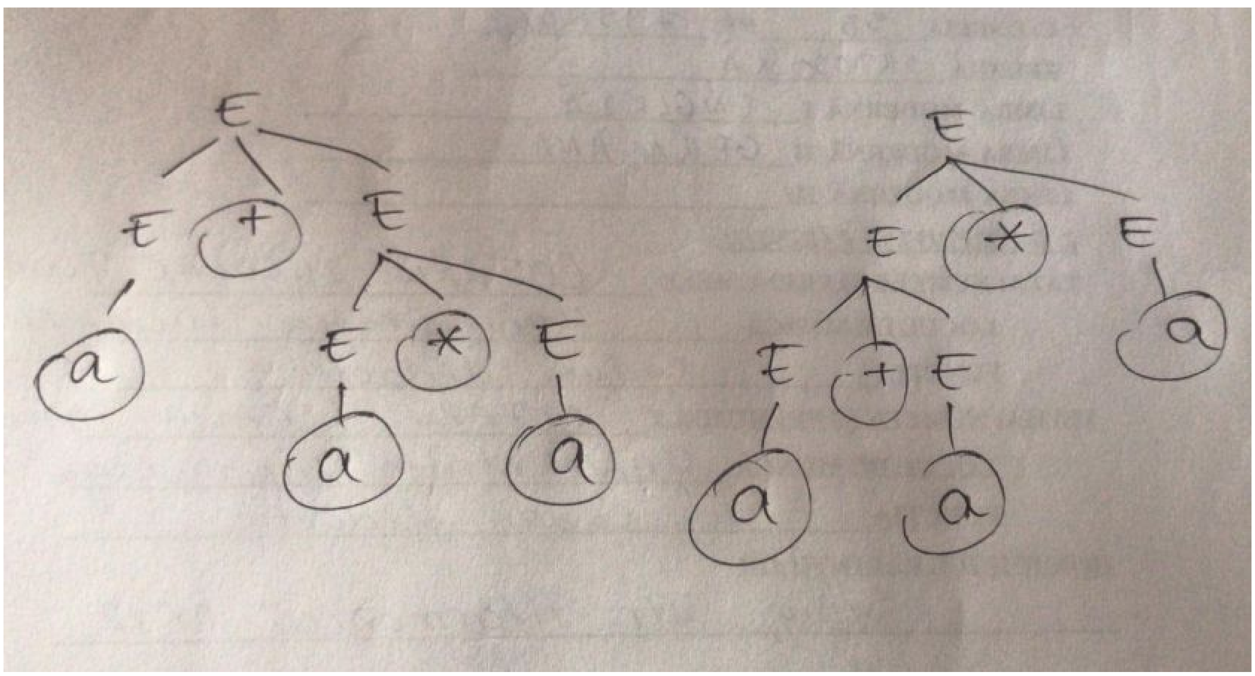
2. Prove that the following grammars are ambiguous

a. $G_1 = (\{S, B, C\},\ \{a, b, c\},\ \{S \to abC \mid aB,\ B \to bC,\ C \to c\},\ S)$ -> HW

b. $G_2 = (\{E\},\ \{a, +, *, (,)\},\ \{E \to E + E \mid E * E \mid (E) \mid a\})$

Sol.: [redacted]

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. : //B: [redacted]

$(S_1)\ S \to aSbS$
$(S_2)\ S \to aS$
$(S_3)\ S \to c$

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

=> 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) $S \to BA$
(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[redacted]

	F_0	F_1	F2	F3
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: [redacted]

	L_0	L_1	L2	L3	L4
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: [redacted]

	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: [redacted]

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

LL(1) conflict

$A \to \alpha\beta$
$A \to \alpha\gamma$
transformed to
$A \to \alpha B$
$B \to \beta \mid \gamma$

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: [redacted]

$s_0 = closure(\{[S' -> .S]\}) = \{[S' -> .S], [S \rightarrow .aA]\}$

$s_1 = goto(s_0, S) = closure(\{[S' -> S.]\}) = \{[S' -> S.]\}$

$goto(s_0, A) = \{\}$**...**

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

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

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

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

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

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

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

2. Fill in LR(0) parsing table //B: [redacted]

	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: [redacted]

work stack	**input stack**	**output band**
$0	**abbc$**	ε
$0a2	**bbc$**	ε
$0a2b4	**bc$**	ε
$0a2b4b4	**c$**	ε
$0a2b4b4c5	**$**	ε
$0a2b4b4A6	**$**	**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

   // [redacted]

   S0 = closure({[S’ -> .E]}) = {[S’ -> .E], [E->.T], [E-> .E + T], [T -> .(E)], [T -> .id], [T -> .const]}
   S1 = goto(s0, E) = closure({[S’ -> E.],[E -> E.+T]}) = {[S’ -> E.],[E -> E.+T]}
   S2 = goto(s0, T) = closure({[E -> T.]}) = {[E -> T.]}
   S3 = goto(s0, () = closure({[T -> (.E)]}) = {[T -> (.E)], [E -> .T], [E->.E+T], [T->.(E)], [T->.id], [T->.const]}
   S4 = goto(s0, id) = closure({[T -> id.]}) = {[T -> id.]}
   S5 = goto(s0, const) = closure({[T -> const.]}) = {[T -> const.]}
   S6 = goto(s1, +) = closure({[E -> E+.T]}) = {[E -> E+.T], [T -> .(E)], [T -> .id], [T -> .const]}
   S7 = goto(s3, E) = closure({[T -> (E.)], [E -> E.+T]}) = {[T -> (E.)], [E -> E.+T]}
   goto(s3, T) = closure({[E-> T.]}) = S2
   goto(s3, id) = closure({[T -> id.]}) = S4
   goto(s3, const) = closure({[T -> const.]}) = S5
   goto(s3, () = closure({[T -> (.E)]}) = S3
   S8 = goto(s6, T) = closure({[E -> E+T.]})
   goto(s6, () = closure({[T->(.E)]}) = s3
   goto(s6, id) = closure({[T -> id.]}) = s4
   goto(s6, const) = closure({[T -> const.]}) = s5
   S9 = goto(s7,)) = closure({[T -> (E).]}) = {[T -> (E).]}
   goto(s7, +) = closure({[E -> E+.T]}) = s6

FOLLOW(E) = {ε, +,)}
FOLLOW(T) = {ε, +,)}

2. Fill the SLR table

//[redacted]

	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

//[redacted]

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

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

2 5 1 4

PDA

Design PDA for accepting:

1. $L=\{0^n1^{2n} \mid n >= 0\}$.
2. $L=\{0^{2n}1^n \mid n >= 0\}$.
3. $L=\{0^n1^m2^n \mid m, n>=1\}$.
4. $L=\{0^n1^m \mid m, n>=0, m> n\}$.
5. $L=\{ww^R \mid w \in \{a,b\}^+\}$.
6. All sequences of matching parantheses, ex (), ()(), (()), ((()())), etc .