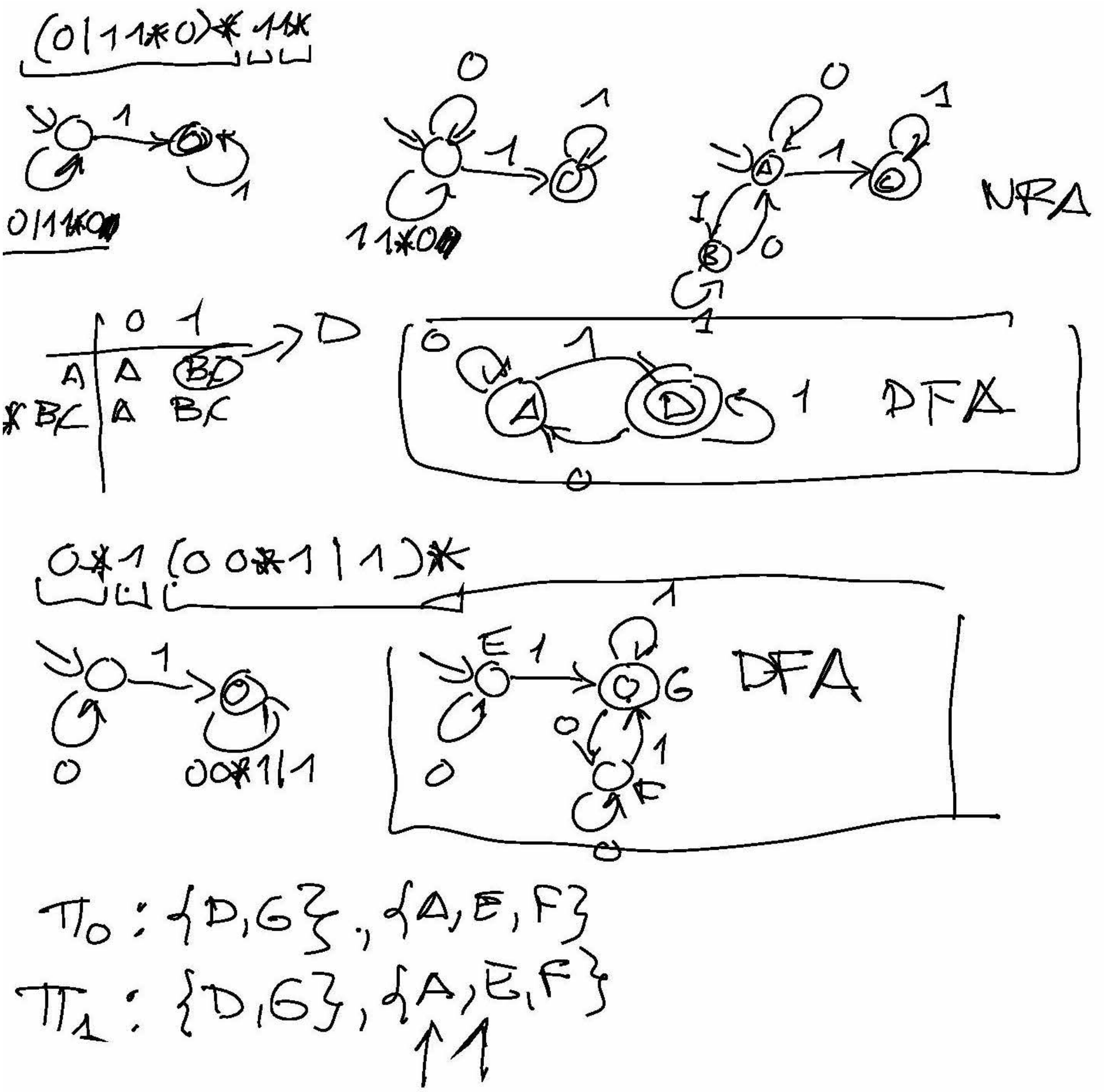
Formal Languages and Compilers
Exercises about theory

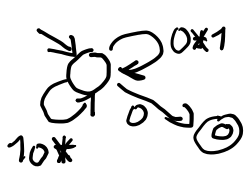
1. Verify the equivalence of the following regular expressions
   (0 | 11*0)* 11*
   0*1 (00*1 | 1)*

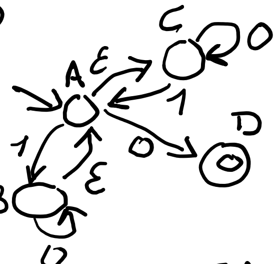


The two regular expressions are equivalent because the initial states of their DFAs (A and E) are in the same equivalence class.

2. Find the minimum state DFA accepting the union of the languages denoted by the following regular expressions
(0*1 | 10*)* 0
(0 | 11)* 01

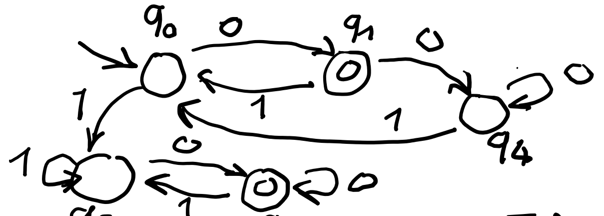
$(0^*1 \mid 10^*)^*0$





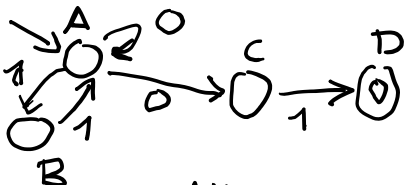
ε-NFA

		0	1
q_0	{A,C}	{C,D}	{A,B,C}
q_1	*{C,D}	{C}	{A,C}
q_2	{A,B,C}	{A,B,C,D}	{A,B,C}
q_3	*{A,B,C,D}	{A,B,C,D}	{A,B,C}
q_4	{C}	{C}	{A,C}



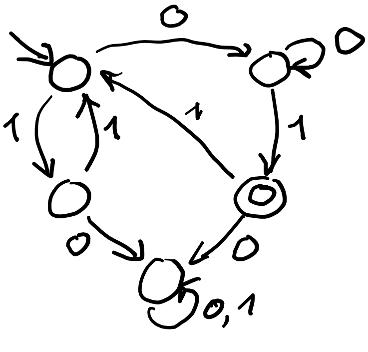
DFA

$(0 \mid 11)^*01$



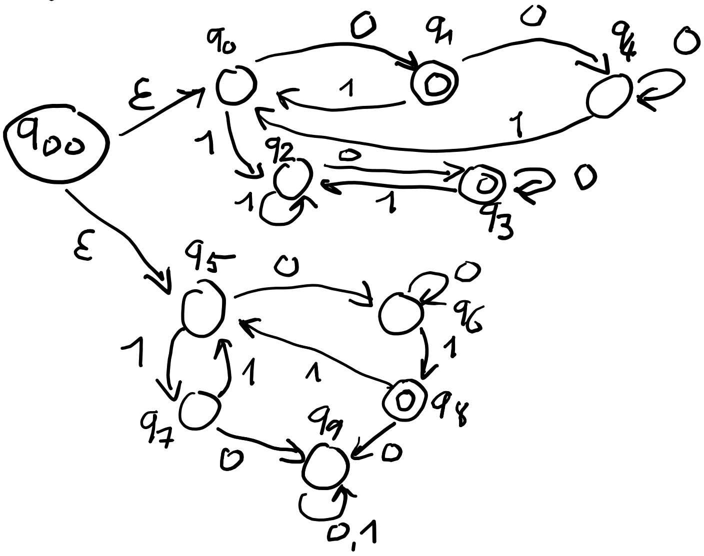
NFA

		0	1
q_5	{A}	{A,C}	{B}
q_6	{A,C}	{A,C}	{B,D}
q_7	{B}	{}	{A}
q_8	*{B,D}	{}	{A}
q_9	{}	{}	{}



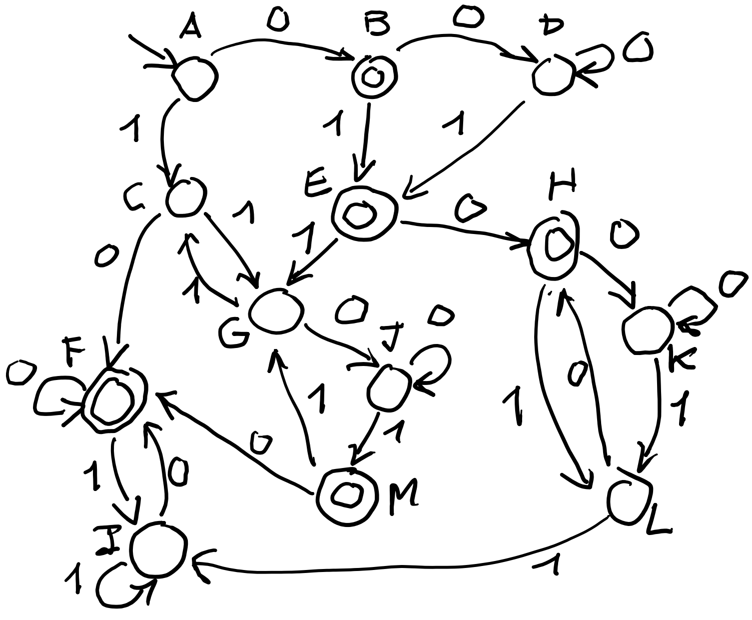
DFA

Union DFA:



ε-NFA

		0	1
A	$q_0 q_0 q_5$	$q_1 q_6$ (B)	$q_2 q_7$ (C)
B *	$q_1 q_6$	$q_4 q_6$ (D)	$q_0 q_8$ (E)
C	$q_2 q_7$	$q_3 q_9$ (F)	$q_2 q_5$ (G)
D	$q_4 q_6$	$q_4 q_6$ (D)	$q_0 q_8$ (E)
E *	$q_0 q_8$	$q_1 q_9$ (H)	$q_2 q_5$ (G)
F *	$q_3 q_9$	$q_3 q_9$ (F)	$q_2 q_9$ (I)
G	$q_2 q_5$	$q_3 q_6$ (J)	$q_2 q_7$ (C)
H *	$q_1 q_9$	$q_4 q_9$ (K)	$q_0 q_9$ (L)
I	$q_2 q_9$	$q_3 q_9$ (F)	$q_2 q_9$ (I)
J *	$q_3 q_6$	$q_3 q_6$ (J)	$q_2 q_8$ (M)
K	$q_4 q_9$	$q_4 q_9$ (K)	$q_0 q_9$ (L)
L	$q_0 q_9$	$q_1 q_9$ (H)	$q_2 q_9$ (I)
M	$q_2 q_8$	$q_3 q_9$ (F)	$q_2 q_5$ (G)



DFA

Minimization

π_0 : {A, C, D, G, I, J, K, L}, {B, E, F, H, M}

π_1 : {A, C, I, L}, {D, J}, {G, K}, {B}, {E, F, M}, {H}

π_2 : {A}, {C, I}, {L}, {D, J}, {G}, {K}, {B}, {E}, {F}, {H}

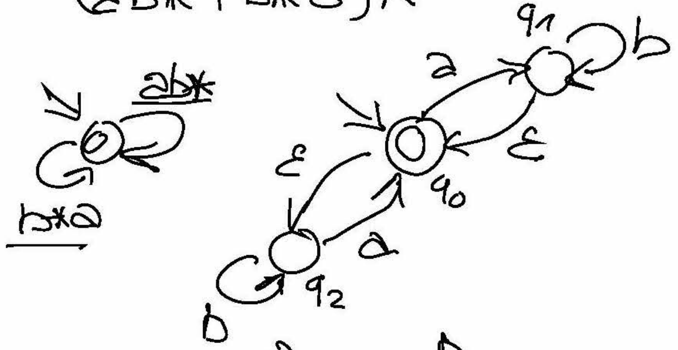
π_3 : {A}, {C}, {I}, {L}, {D}, {J}, {G}, {K}, {B}, {E}, {F}, {H}

The DFA is already minimum

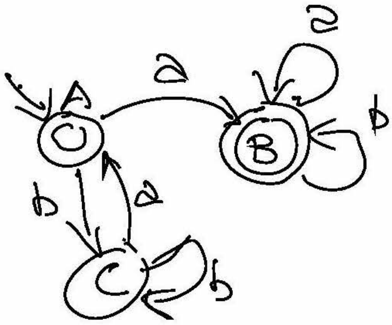
3. Find the minimum state DFA accepting the language denoted by the regular expression
(ab* | b*a)*

3

(ab* | b*a)*



ε-NFA



DFA

	a	b	
* $\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_2\}$	A
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_2, q_1, q_0\}$	B
$\{q_2\}$	$\{q_0, q_2\}$	$\{q_2\}$	C

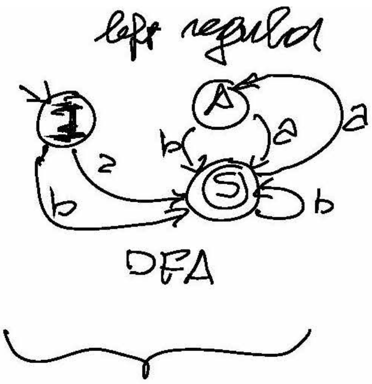
$\Pi_0 : \{C\}, \{A, B\}$
$\Pi_1 : \{C\}, \{A\}, \{B\}$

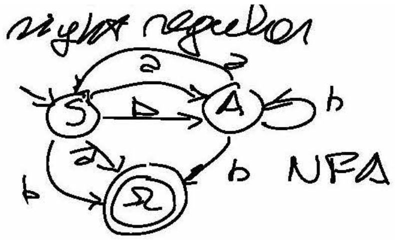
The DFA is already minimum-state

4. Verify the equivalence of the grammars G1=({S,A},{a,b},P1,S), G2=({S,A},{a,b},P2,S) with

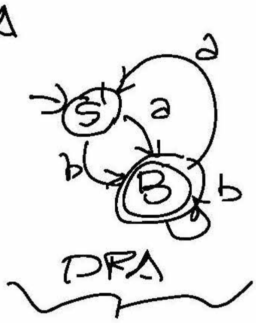
```
P1={    S -> A a | A b | S b | a | b
        A -> S a
}
P2={    S -> a A | b A | a | b
        A -> a S | b A | b
}
```

We build DFAs equivalent to the grammars





	a	b
S'	A,Ω	A,Ω
*A,Ω (B)	S'	A,Ω



Π_0: {S,B}, {I,A,S'}

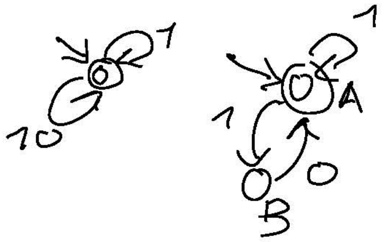
Π_1: {S,B}, {I,A,S'}

the two grammars are equivalent because the start states of their equivalent DFAs (I and S') are in the same equivalence class

5. Find a regular expression denoting the complement of the language denoted by (1 | 10)*

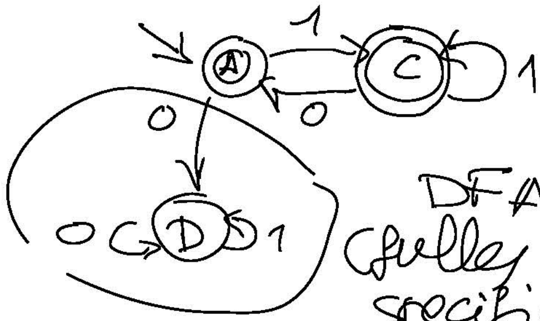
(1|10)*

5



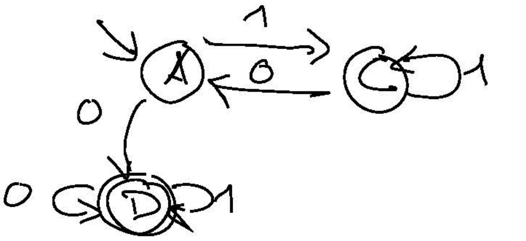
NFA

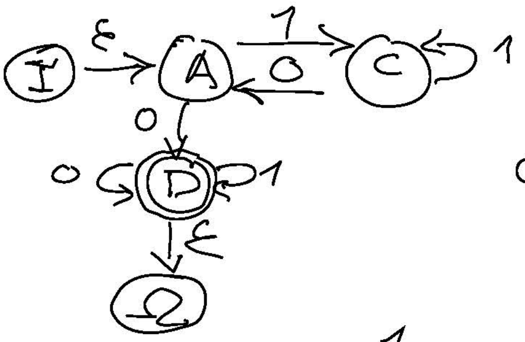
	0	1
* A	–	AB → C
* AB	A	AB

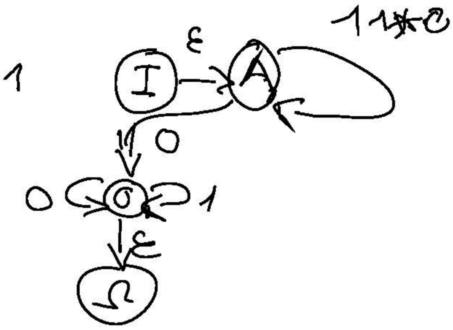


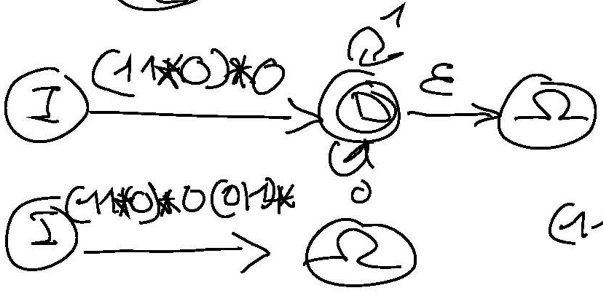
DFA (fully specified)

complement DFA









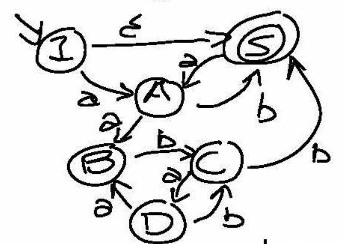
(11*0)*0(0|1)*

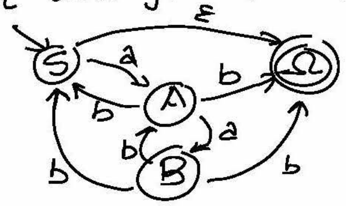
6. Verify the equivalence of the grammars G1=({S,A,B,C,D},{a,b},P1,S), G2=({S,A,B},{a,b},P2,S) with

```
P1={   S -> A b | C b | ε
       A -> S a | a
       B -> A a | D a
       C -> B b | D b
       D -> C a

}
P2={   S -> a A | ε
       A -> a B | b S | b
       B -> b A | b S | b
}
```

Build DFAs for the two grammars :





		a	b
Q_0	$\rightarrow * \{I,S\}$	$\{A\}$	Φ
Q_1	$\{A\}$	$\{B\}$	$\{S\}$
Q_2	$\{B\}$	Φ	$\{C\}$
Q_3	$* \{S\}$	$\{A\}$	Φ
Q_4	$\{C\}$	$\{D\}$	$\{S\}$
Q_5	$\{D\}$	$\{B\}$	$\{C\}$
Q_6	Φ	Φ	Φ

		a	b
Q_0'	$\rightarrow * \{S,\Omega\}$	$\{A\}$	Φ
Q_1'	$\{A\}$	$\{B\}$	$\{S,\Omega\}$
Q_2'	$\{B\}$	Φ	$\{A,S,\Omega\}$
Q_3'	$* \{A,S,\Omega\}$	$\{A,B\}$	$\{S,\Omega\}$
Q_4'	$\{A,B\}$	$\{B\}$	$\{A,S,\Omega\}$
Q_5'	Φ	Φ	Φ

check equivalence of the two DFAs

$\Pi_0 : \{Q_0, Q_3, Q_0', Q_3'\}, \{Q_1, Q_2, Q_4, Q_5, Q_6, Q_1', Q_2', Q_4', Q_5'\}$

$\Pi_1 : \{Q_0, Q_3, Q_0'\}, \{Q_3'\}, \{Q_1, Q_4, Q_1', Q_2', Q_4'\}, \{Q_2, Q_5, Q_6, Q_5'\}$

$\Pi_2 : \{Q_0, Q_3, Q_0'\}, \{Q_3'\}, \{Q_1, Q_4\}, \{Q_1'\}, \{Q_2'\}, \{Q_4'\}, \{Q_2, Q_5, Q_6, Q_5'\}$

$\Pi_3 : \{Q_0, Q_3\}, \{Q_0'\}, \ldots$

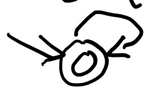
The two DFAs are NOT equivalent because Q_0 and Q_0' belong to different equivalence classes
$\Rightarrow$ the two grammars are not equivalent.
(the counter-example is aab, which is accepted only by the second grammar).

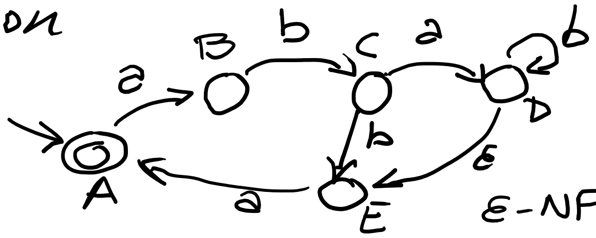
7. Find a left-regular grammar that generates the complement of the language denoted by the following expression

   (a b (a b* | b) a)*

DFA construction

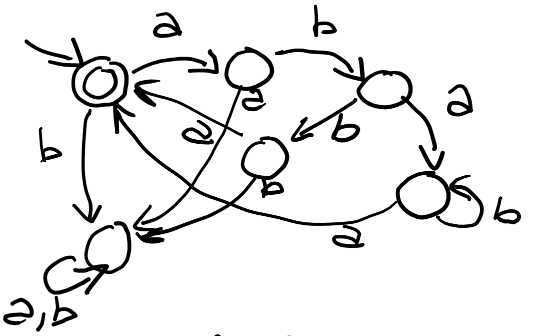
ab(ab*|b)a





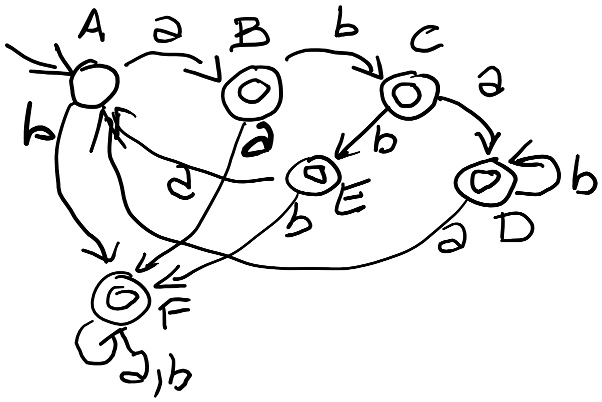
ε-NFA

	a	b
* A	B	—
B	—	C
C	{D,E}	E
{D,E}	A	{D,E}
E	A	—
{ }	—	—



completely specified DFA

complement DFA:



$S \to B | C | D | E | F$

$A \to Ea | Da | \varepsilon$

$B \to Aa$

$C \to Bb$

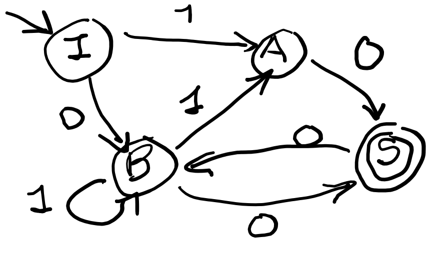
$D \to Ca | Da | a$

$E \to Cb | a$

$F \to Ab | Ba | Eb | Fa | Fb$

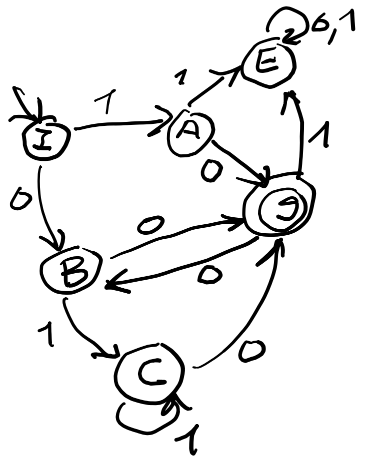
8. Find a minimum-state DFA equivalent to the following grammar (over alphabet {0,1}, with start symbol S), and then the regular expression for the accepted language

   S -> A 0 | B 0
   A -> B 1 | 1
   B -> B 1 | S 0 | 0



NFA

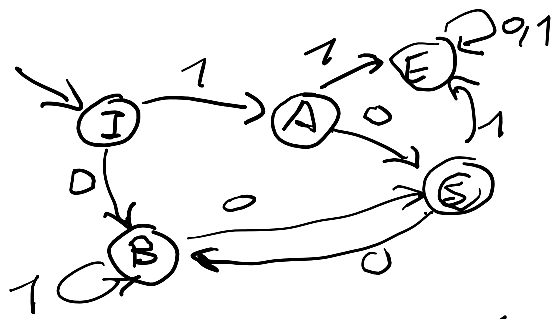
	0	1
{I}	{B}	{A}
{B}	{S}	{A,B}
{A}	{S}	{ }
{S}	{B}	{ }
{A,B}	{S}	{A,B}
{ }	{ }	{ }

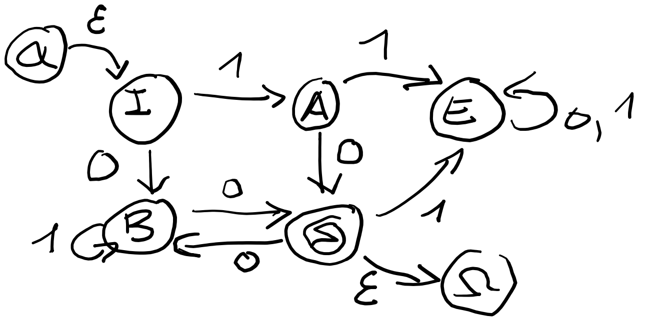


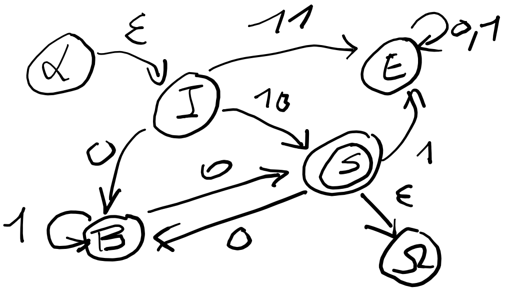
DFA

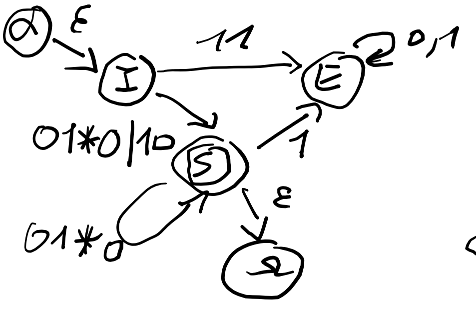
π_0 : {S}, {I, A, B, C, E}
π_1 : {S}, {I, E}, {A, B, C}
π_2 : {S}, {I}, {E}, {A}, {B, C}
π_3 : {S}, {I}, {E}, {A}, {B, C}   states B, C are equivalent.

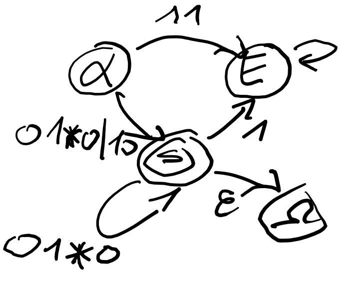
Min-state DFA :

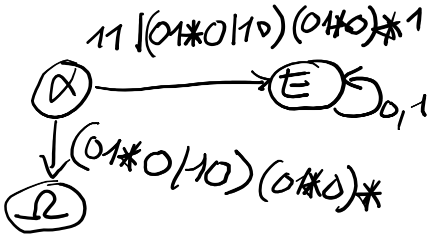












$\Rightarrow$ (01*0 | 10)(01*0)*

9. Find a pushdown automaton accepting the language: { $a^m b^{m+n} a^n$ | m >= 0 , n >= 0 }.

$\{a^m b^{m+n} a^n \mid m \geq 0, n \geq 0\}$ 9

$$a^m b^{m+n} a^n = \underbrace{a^m b^m}_{A} \underbrace{b^n a^n}_{B}$$

$$\begin{cases} S \to AB \\ A \to aAb \mid \varepsilon \\ B \to bBa \mid \varepsilon \end{cases}$$

$$PDA = (\{q\}, \{a,b\}, \{a,b,S,A,B\}, \delta, q, S, \phi)$$

$$\begin{cases} \delta(q,\varepsilon,S) = \{(q, AB)\} \\ \delta(q,\varepsilon,A) = \{(q, aAb), (q,\varepsilon)\} \\ \delta(q,\varepsilon,B) = \{(q, bBa), (q,\varepsilon)\} \\ \delta(q,a,a) = \delta(q,b,b) = \{(q,\varepsilon)\} \end{cases}$$

10. Find a pushdown automaton accepting the language: { b^{2m} a^{n+m+1} b^{2n} | m >= 0 , n >= 0 }.

$$\underbrace{b^{2m}\, a^{m}}_{A}\, a\, \underbrace{a^{n}\, b^{2n}}_{B}$$

$S \rightarrow AaB$

$A \rightarrow bbAa \mid \varepsilon$

$B \rightarrow aBbb \mid \varepsilon$

alternative solution

$$\underbrace{b^{2m}\, a^{m}\, a}_{A}\, \underbrace{a^{n}\, b^{2n}}_{B}$$

$S \rightarrow AB$

$A \rightarrow bbAa \mid \cancel{bb}a$

$B \rightarrow aBbb \mid \varepsilon$

11. Find a pushdown automaton accepting the language: $\{ (a^n b^n)^m c^m \mid m >= 0 , n >= 0 \}$.

The pushdown automaton cannot be built for this language because it is not context-free.

12. Find a context-free grammar for the language: { $a^n b^m$ | $2n <= m <= 3n$ }.

Each string of the language contains, for each a, at least 2 b and at most 3 b

Hence, the grammar can be written as

$$S \rightarrow aSbb \mid aSbbb \mid \varepsilon$$

(assuming $n \geq 0$)

13. Find a context-free grammar for the language: $\{ (10)^n\ 11\ (01)^n \mid n >= 0 \}$.

$S \to 10\,S\,01 \mid 11$

14. Find a context-free grammar for the language made of all the strings in {a,b,c}* with just one 'c' and an equal number of 'a' and 'b'.

CFG for $L = \{ s \in \{a,b\}^* \mid s$ contains an equal number of a and b $\}$

$S \to aSb \mid bSa \mid SS \mid \varepsilon$

CFG for the required language:

$S' \to aS'b \mid bS'a \mid S'S \mid SS' \mid c$

15. Find a context-free grammar for the language: $\{ a^n b^m c^k \mid n = 2m \text{ OR } m = 2k \}$.

We assume $n \geq 0, m \geq 0, k \geq 0$

A) $n = 2m \Rightarrow \underbrace{a^{2m} b^{m}}_{A1} \underbrace{c^{k}}_{A2}$

B) $m = 2k \Rightarrow \underbrace{a^{n}}_{B1} \underbrace{b^{2k} c^{k}}_{B2}$

$S \rightarrow A \mid B$

$A \rightarrow A1\ A2$

$A1 \rightarrow aa A1 b \mid \varepsilon$

$A2 \rightarrow c A2 \mid \varepsilon$

$B \rightarrow B1\ B2$

$B1 \rightarrow a B1 \mid \varepsilon$

$B2 \rightarrow bb B2 c \mid \varepsilon$

16. Find a grammar equivalent to the following one (over alphabet {a,b} with start symbol S), having only useful symbols:
S -> a A a | b C B b
A -> a S a | b A | b
B -> b S B b | a B C b
C -> a B C a | a A S | a

symbols that generate non-empty languages:

{a}, {b}, {A, C}, {S}

$\Rightarrow$ B generates the empty language

S $\rightarrow$ aAa
A $\rightarrow$ aSa | bA | b
C $\rightarrow$ aAS | a

Reachable symbols:

{S}, {A}

$\Rightarrow$ C is not reachable

S $\rightarrow$ aAa
A $\rightarrow$ aSa | bA | b

17. From the following grammar (over alphabet {0,1}, with start symbol S):

S -> 0 S 0 | 1C1 | BB
A -> 0 C 1 | 0
B -> 1 B | B A C
C -> 0 B | 1 S | 1

eliminate useless symbols

symbols that generate a non-empty long.
{0,1}, {A, C}, {S}
symbols that generate the empty long:
{B}

$S \to 0S0 \mid 1C1$
$A \to 0C1 \mid 0$
$C \to 1S \mid 1$

Reachable symbols:
{S}, {C}
Unreachable symbols: {A}

$S \to 0S0 \mid 1C1$
$C \to 1S \mid 1$

18. From the following grammar (over alphabet {0,1}, with start symbol S):

S -> 0 S 0 | 1B1 | BB
A -> 0 C 1 | 0
B -> 1 B | S 0 | ε
C -> 0 A | 1

eliminate useless symbols and ε-productions.

symbols that can produce the empty string:

$\{\varepsilon\}, \{B\}, \{S\}$

$S \to 0S0 \mid 1B1 \mid BB \mid 00 \mid 11 \mid B$
$A \to 0C1 \mid 0$
$B \to 1B \mid S0 \mid 1 \mid 0$
$C \to 0A \mid 1$

Elimination of useless symbols

Symbols that generate a non-empty lang.:

$\{0,1\} \cup \{S, A, B, C\}$

Reachable symbols: $\{S\} \cup \{B\}$

Non-reachable symbols: $\{A, C\}$

$\Rightarrow$ $S \to 0S0 \mid 1B1 \mid BB \mid 00 \mid 11 \mid B$
$B \to 1B \mid S0 \mid 1 \mid 0$

19. Find the LR(0) parsing table for the following grammar (over alphabet {a,b}, with start symbol S):

S -> Ab | aBa   1,2

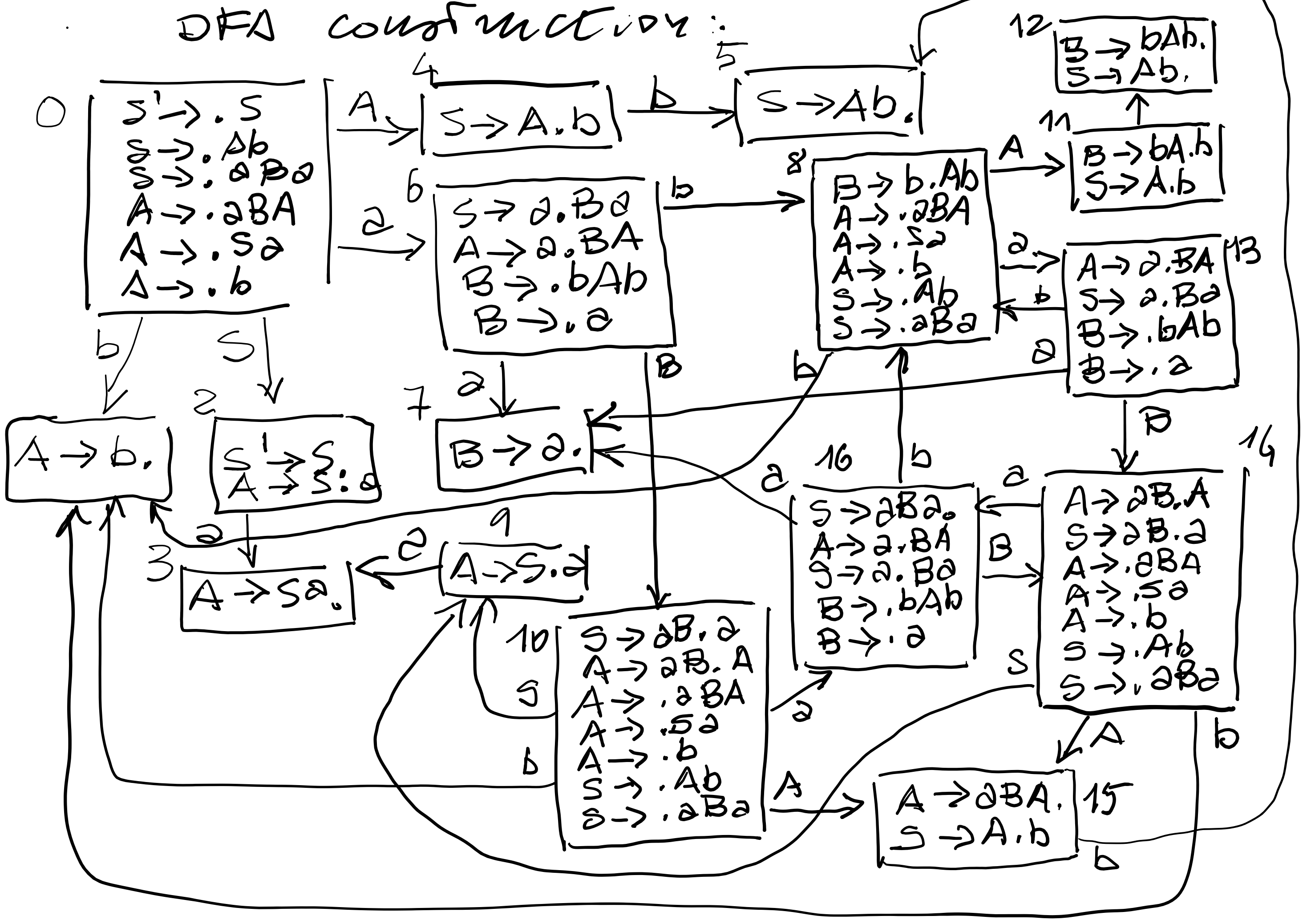
A -> aBA | Sa | b   3,4,5

B -> bAb | a   6,7

We add new start symbol S'

0 S' -> S

DFA construction:

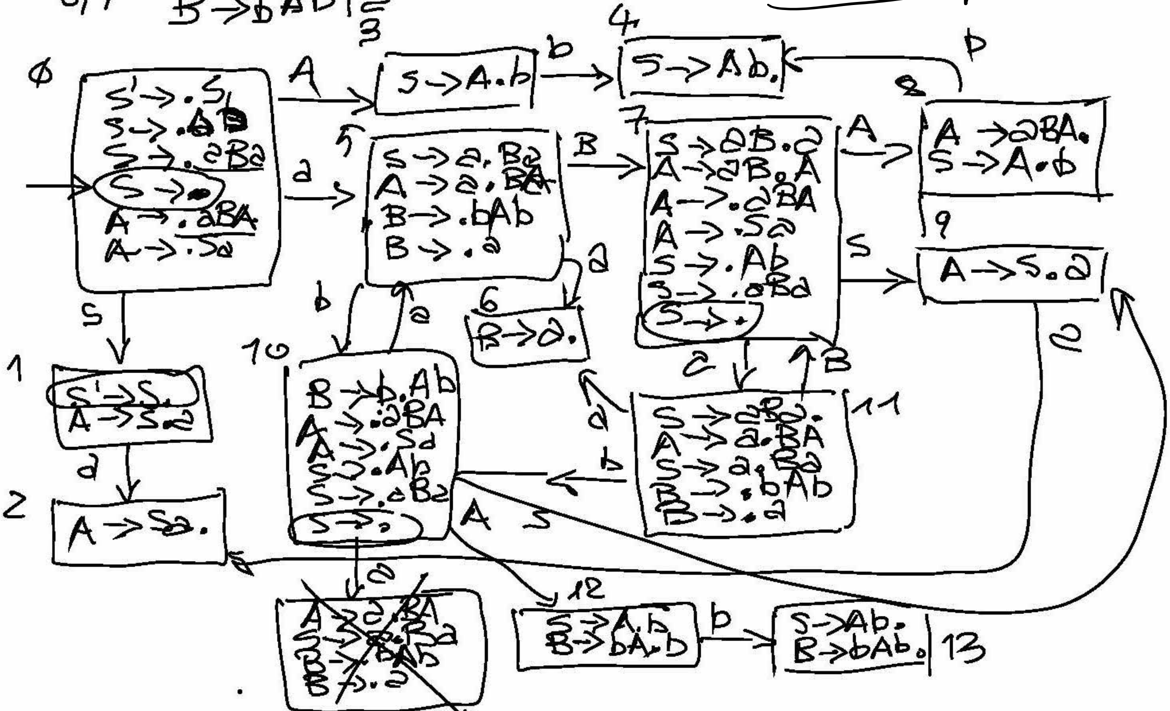


	a	b	$	S	A	B
0	s6	s1		2	4	
1	r5	r5	r5			
2	s3		acc			
3	r4	r4	r4			
4		s5				
5	r1	r1	r1			
6	s7	s8				10
7	r7	r7	r7			
8	s13	s1		9	11	
9	s3					
10	s16	s1		9	15	
11		s12				
12	r6/r1	r6/r1	r6/r1			
13	s7	s8				14
14	s16	s1		9	15	
15	r3	s5/r3	r3			
16	s7/r2	s8/r2	r2			14

20. Find the SLR parsing table for the following grammar (over alphabet {a,b}, with start symbol S):
S → Ab | aBa | ε
A → aBA | Sa
B → bAb | a

	nullable	FIRST	FOLLOW
S	T	a	a, $
A	F	a	b
B	F	a, b	a
S'	T	a	$

0 S' → S
1,2,3 S → ~~Ab~~ | aBa | ε
4,5 A → aBA | Sa
6,7 B → bAb | a

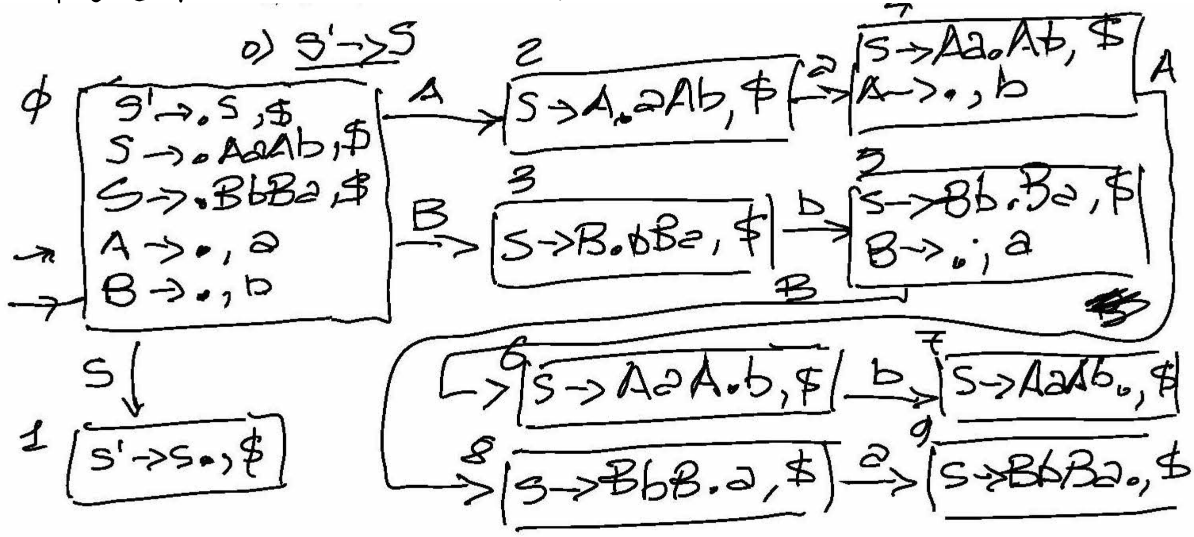


ACTION GOTO

	a	b	$	S	A	B
0	r3/s5		r3	1	3	
1	s2		acc			
3		s4				
4	r1		r1			
5	s6	s10				7
6	r7					
7	r3/s11		r3	9	8	
2		r5				
8		r4/s4				
9	s2					
10	r3/s5		r3	9	12	
11	r2/s6	s10	r2			
12		s13				
13	r~~1~~/r6		r1			

21. Find the LR(1) parsing table for the following grammar (over alphabet {a,b}, with start symbol S):
    1) S → AaAb
    2) S → BbBa
    3) A → ε
    4) B → ε

We add new start symbol S' and new rule:



state	a	b	$	A	B	S
0	r3	r4		2	3	1
1			acc			
2	s4					
3		s5				
4		r3		6		
5	r4				8	
6		s7				
7			r1			
8	s9					
9			r2			

22. Find the predictive parsing table for the following grammar (over alphabet {a,b}, with start symbol S):

$S \rightarrow AB$
$S \rightarrow B$
$A \rightarrow aA$
$A \rightarrow \varepsilon$
$B \rightarrow bB$
$B \rightarrow \varepsilon$

The grammar is NOT left-recursive

	nullable	FIRST	FOLLOW
S	T	a, b	$
A	T	a	b, $
B	T	b	$

	a	b	$
S	S → AB	S → AB S → B	S → AB S → B
A	A → aA	A → ε	A → ε
B		B → bB	B → ε

23. Find the LL(1) parsing table for the following grammar (over alphabet {(,),a}, with start symbol S):

S → (L) | a

L → L , S | S

Grammar is left recursive. First, we eliminate left recursion:

S → (L) | a

L → S R

R → , S R | ε

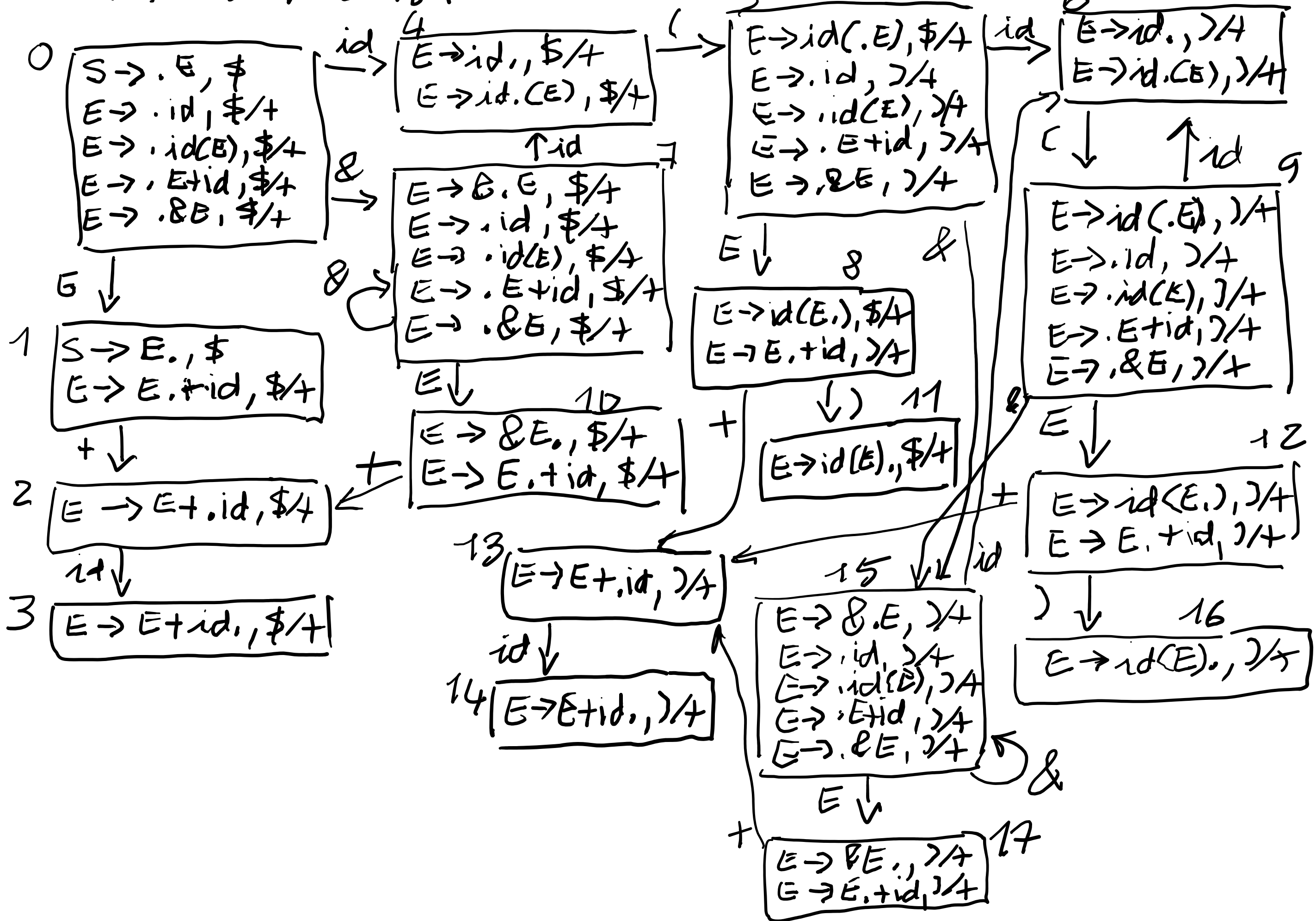
	nullable	FIRST	FOLLOW
S	F	(a	$
L	F	(a	)
R	T	,	)

	(	)	a	,	$
S	S→(L)		S→a		
L	L→SR		L→SR		
R		R→ε		R→,SR	

24. Produce the LR(1) parsing table for the following grammar (over alphabet {id,(,),+,&} and with start symbol E:
E → id | id (E) | E + id | & E
Is the language generated by the grammar a deterministic CFL?

We add start symbol S and rule S → E (0)

DFA construction



	id	(	)	+	&	$	S	E
0	s4				s7			1
1				s2		acc		
2	s3							
3				r3		r3		
4		s5		r1		r1		
5	s6				s15			8
6		s9	r1	r1				
7	s4				s7			10
8			s11	s13				
9	s6				s15			12
10				r4/s2		r4		
11				r2		r2		
12			s16	s13				
13	s14							
14			r3	r3				
15	s6				s15			17
16			r2	r2				
17			r4	r4/s13				

The language generated by the grammar is not a deterministic CFL because the grammar is not LR(1), as the LR(1) parsing table has conflicts.

25. Eliminate left recursion from the following grammar and produce its predictive parsing table:

S → Aa | Bc
A → aB | Bb
B → Sc | a

Let us define the ordering S, A, B.

$S \to Aa \mid Bc$
$A \to aB \mid Bb$
$B \to (Aa \mid Bc)c \mid a \Rightarrow B \to Aac \mid Bcc \mid a$

$B \to (aB \mid Bb)ac \mid Bcc \mid a$
$B \to aBac \mid Bbac \mid Bcc \mid a$
$B \to B\underbrace{(bac \mid cc)}_{\alpha} \mid \underbrace{aBac \mid a}_{\beta}$

$\begin{cases} B \to aBacR \mid aR \\ R \to bacR \mid ccR \end{cases}$

$\Rightarrow$
$S \to Aa \mid Bc$
$A \to aB \mid Bb$
$B \to aBacR \mid aR$
$R \to bacR \mid ccR$

	nullable	FIRST
S	F	a
A	F	a
B	F	a
R	F	b, c

	a	b	c	$
S	S → Aa S → Bc			
A	A → aB A → Bb			
B	B → aBacR B → aR			
R		R → bacR	R → ccR	

26. Tell if the following grammar (over alphabet {‘a’ , ‘(‘ , ‘)’ , ‘,’} and with start symbol S) is LL(1):
S → (L) | a
L → L , S | S

The grammar is LL(1) because its parsing table (see exercise 23) has no conflicts.

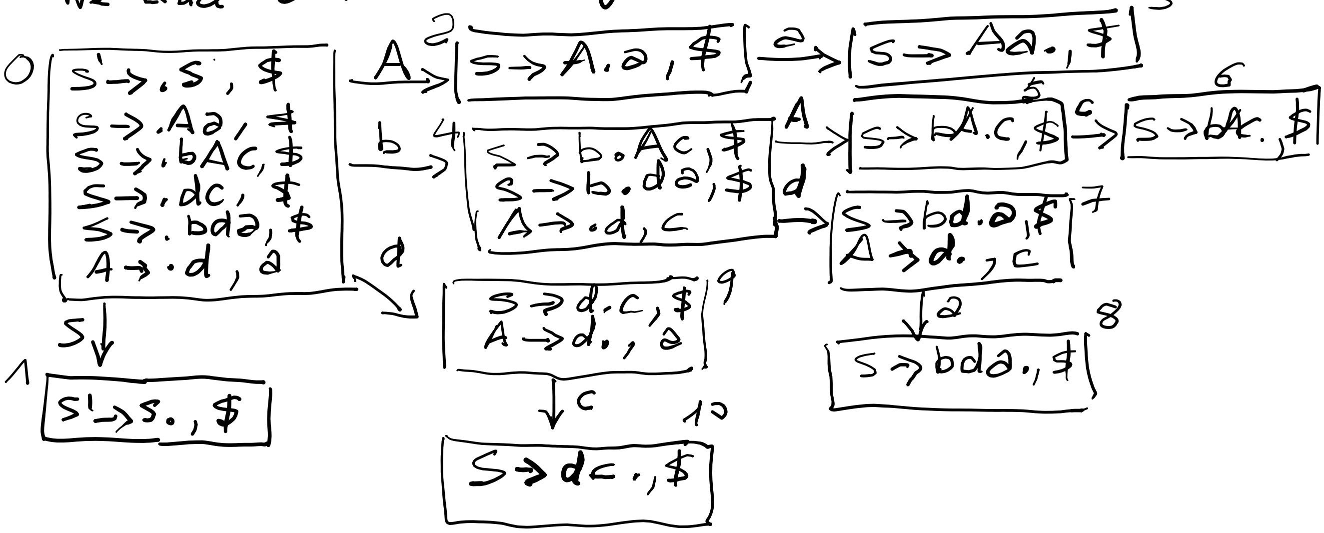
27. Tell if the following grammar (over alphabet {a,b,c,d}, with start symbol S) is LALR(1):

$S \rightarrow A\,a \mid b\,A\,c \mid d\,c \mid b\,d\,a$ 1,2,3,4

$A \rightarrow d$ 5

DFA construction (LR(1))

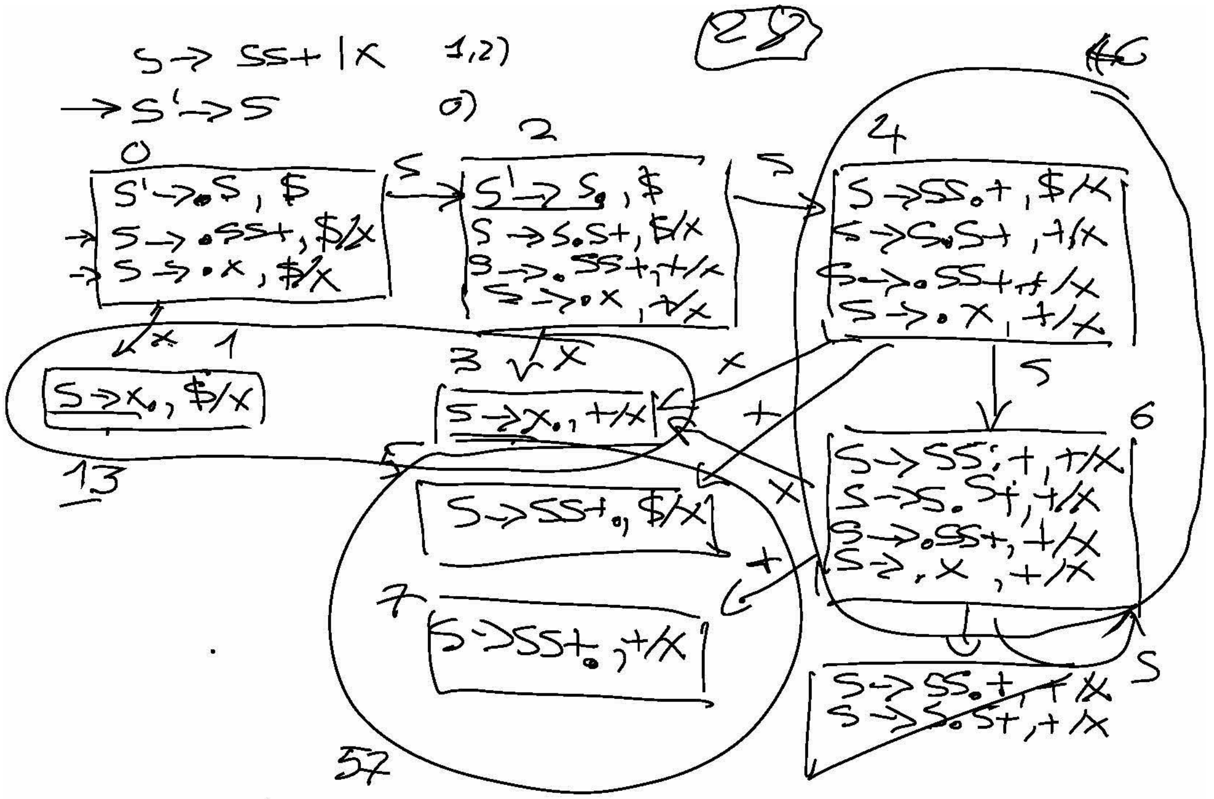
We add a new start symbol S' and rule S'→S (0)



There are no states to be merged

By analyzing the states with complete items it is possible to see that there are no conflicts: In states 1, 3, 6, 8, 10 there is only the complete item. In state 7 we have s8 for a and r5 for c. In state 9 we have s10 for c and r5 for a.

28. Find the LALR(1) parsing table for the following grammar (over alphabet {+,x}, with start symbol S):
S → S S + | x
Is the grammar LALR(1)?



	x	+	$	S
0	s13			2
13	r2	r2	r2	
2	s13		acc	46
46	s13	s57		46
57	r1	r1	r1	

The grammar is LALR(1) because the LALR(1) parsing table has no conflicts.