Automata Theory and Formal Language
Lab Session 4

Carlos Gallardo Polanco `XXXXXXXX-X`
mail@mail.com

November 26, 2017

1 Exercise

Given the RE $E = (((00)*+(00)*0)10 + ((11)*+(11)*1)10)*$

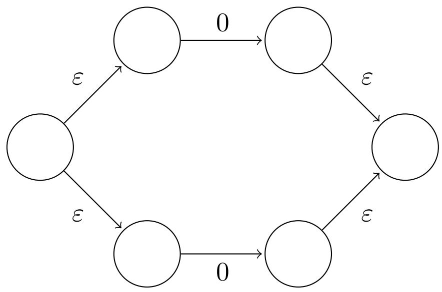
1. Build a $\mathcal{E}-NFA$ thats accepts exactly the same language, following the method explained in class to convert from RE to $\mathcal{E}-NFA$.
2. Generate the equivalent DFA
3. Implement it in a programming language (`Python`, `C/C++`, `Java`) following the table method.

2 Solution

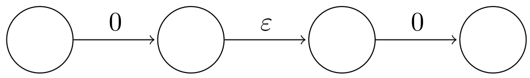
1. Build a $\mathcal{E}-NFA$ thats accepts exactly the same language, following the method explained in class to convert from RE to $\mathcal{E}-NFA$.

   To convert a RE to an $\mathcal{E}-NFA$ we'll follow the following equivalences:

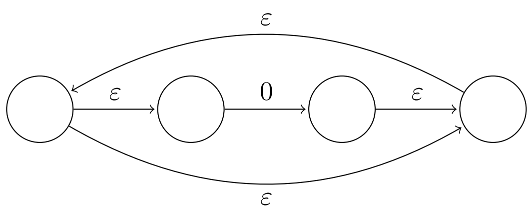
   - Union.
     If $E1 = 0$ and $E2 = 0$ then $E1 \cup E2$ $(E1 + E2)$:



   - Concatenation.
     If $E1 = 0$ and $E2 = 0$ then $E1E2$:



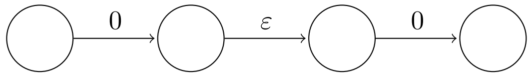
   - Concatenation.
     If $E = 0$ then $E*$:



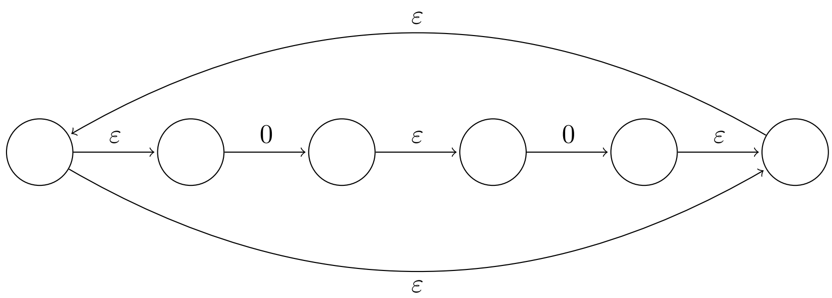
We'll divide the main RE into smaller REs:

$$E = \underbrace{(\underbrace{(\underbrace{(\underbrace{\underbrace{\underbrace{(00)}_{E1}*}_{E1*}+\underbrace{\underbrace{\underbrace{(00)}_{E1}*}_{E1*}0}_{E3}}_{E5})10}_{E7}+\underbrace{(\underbrace{\underbrace{\underbrace{(11)}_{E2}*}_{E2*}+\underbrace{\underbrace{\underbrace{(11)}_{E2}*}_{E2*}1}_{E4}}_{E6})10}_{E8})}_{E9}*}_{E9*=E}$$

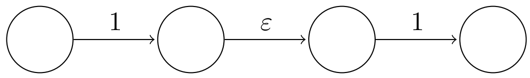
- $E1 = 00$



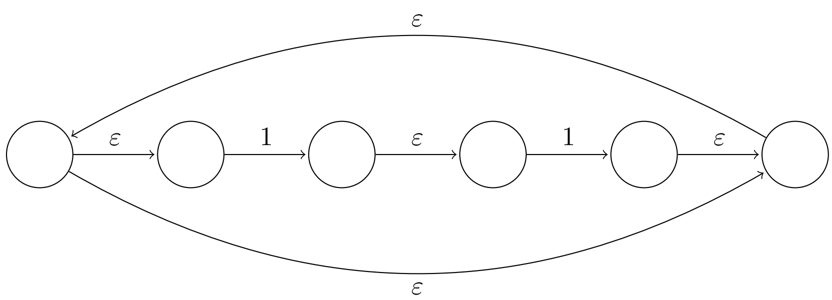
- $E1* = (00)*$



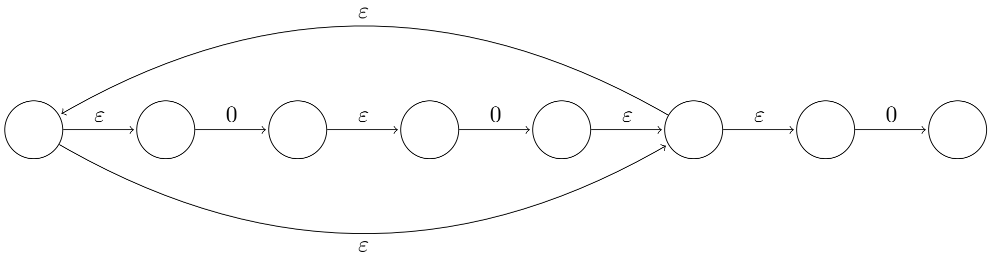
- $E2 = 11$



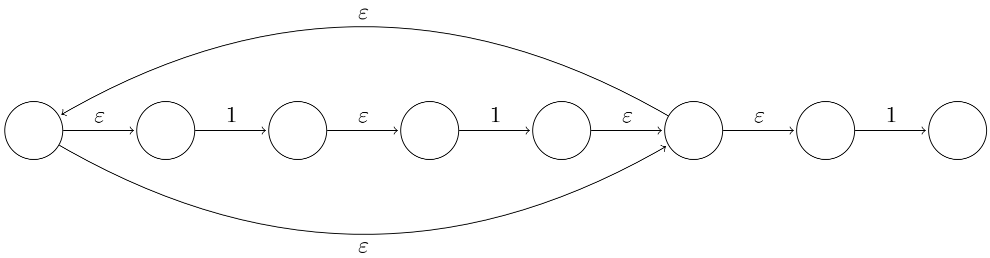
- $E2* = (11)*$



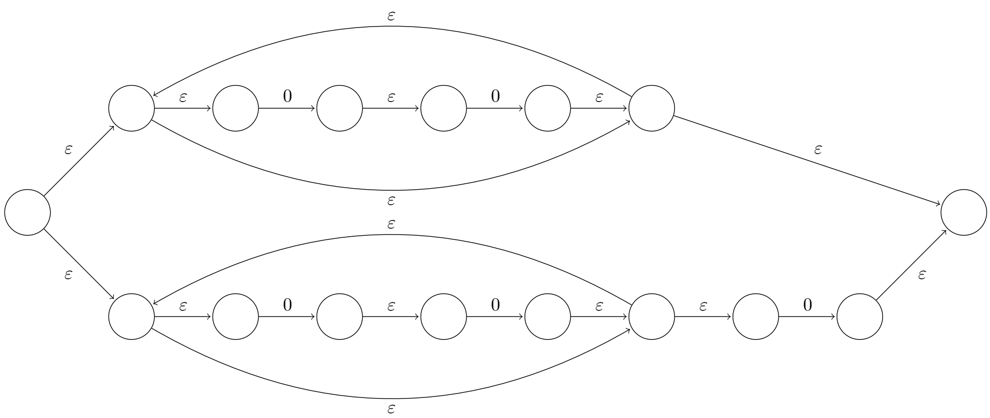
- $E3 = (00) * 0$



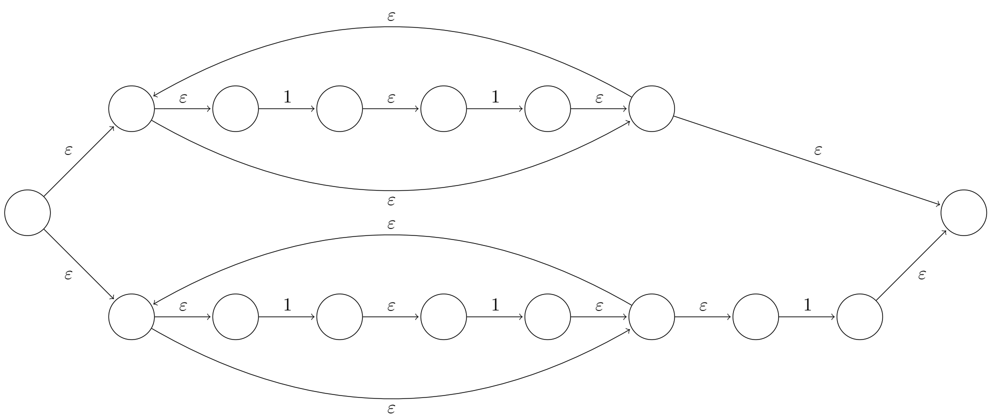
- $E4 = (11) * 1$



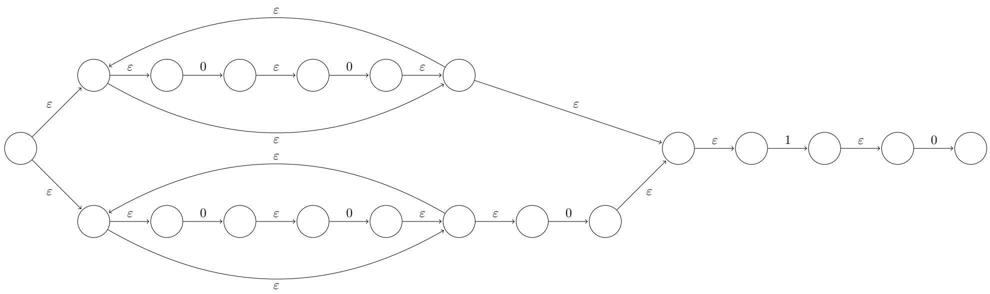
- $E5 = (00) * +(00) * 0$



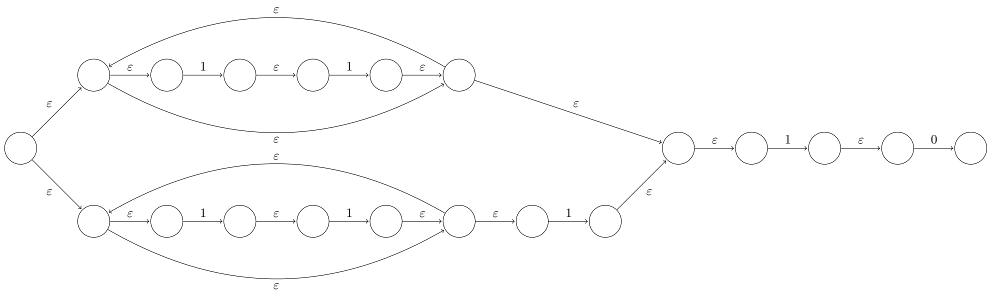
- $E6 = (11) * +(11) * 1$



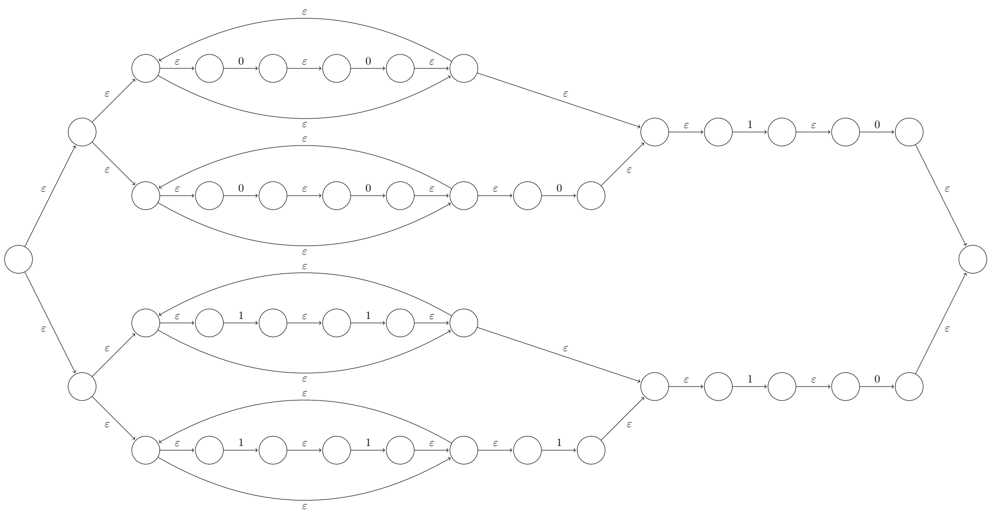
- $E7 = ((00)*+(00)*0)10$



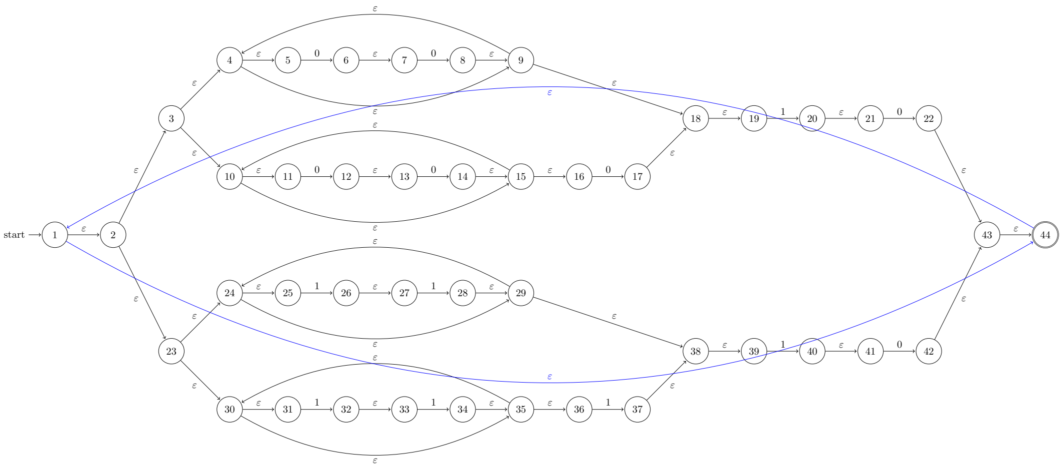
- $E8 = ((11)*+(11)*1)10$



- $E9 = (((00)*+(00)*0)10+((11)*+(11)*1)10)$



- $E9* = (((00)*+(00)*0)10 + ((11)*+(11)*1)10)* = E$



2. Generate the equivalent DFA

   To generate the equivalent DFA, we'll follow the **subset construction method**, so we'll need the equivalent NFA. In the table 1(a) we have the transition table of $\mathcal{E}-NFA$ and in the table 1(b) the closures.

Table 1: Transition Table of $\mathcal{E}-NFA$

(a) Transition Table of $\mathcal{E}-NFA$

States	0	1	ε
1			2 44
2			3 23
3			4 10
4			5 9
5	6		
6			7
7	8		
8			9
9			4 18
10			11 15
11	12		
12			13
13	14		
14			15
15			10 16
16	17		
17			18
18			19
19		20	
20			21
21	22		
22			43
23			24 30
24			25 29
25		26	
26			27
27		28	
28			29
29			24 38
30			31 35
31		32	
32			33
33		34	
34			35
35			30 36
36		37	
37			38
38			39
39		40	
40			41
41	42		
42			43
43			44
44			1

(b) Closures of $\mathcal{E}-NFA$

	States	Closures
$\rightarrow *$	1	1 2 44 3 23 4 10 24 30 5 9 18 19 11 15 25 29 31 35 16 38 36 39
	2	2 3 23 4 10 24 30 5 9 11 15 25 29 31 35 18 16 38 36 19 39
	3	3 4 10 5 9 11 15 18 16 19
	4	4 5 9 18 19
	5	5
	6	6 7
	7	7
	8	8 9 4 18 5 19
	9	9 4 18 5 19
	10	10 11 15 16
	11	11
	12	12 13
	13	13
	14	14 15 10 16 11
	15	15 10 16 11
	16	16
	17	17 18 19
	18	18 19
	19	19
	20	20 21
	21	21
$*$	22	22 43 44 1 2 3 23 4 10 24 30 5 9 18 19 11 15 25 29 31 35 16 38 36 39
	23	23 24 25 29 38 39 30 31 35 36
	24	24 25 29 38 39
	25	25
	26	26 27
	27	27
	28	28 29 24 38 25 39
	29	29 24 38 25 39
	30	30 31 35 36
	31	31
	32	32 33
	33	33
	34	34 35 30 36 31
	35	35 30 36 31
	36	36
	37	37 38 39
	38	38 39
	39	39
	40	40 41
	41	41
$*$	42	42 43 44 1 2 3 23 4 10 24 30 5 9 18 19 11 15 25 29 31 35 16 38 36 39
$*$	43	43 44 1 2 3 23 4 10 24 30 5 9 18 19 11 15 25 29 31 35 16 38 36 39
$*$	44	44 1 2 3 23 4 10 24 30 5 9 18 19 11 15 25 29 31 35 16 38 36 39

Removing the $\mathcal{E}$-transition, we obtain the transition table 2 of NFA.

	States	0	1
$\rightarrow *$	1	6 12 17	20 26 32 37 40
	2	6 12 17	20 26 32 37 40
	3	6 12 17	20
	4	6	20
	5	6	
	6	8	
	7	8	
	8	6	20
	9	6	20
	10	12 17	
	11	12	
	12	14	
	13	14	
	14	12 17	
	15	12 17	
	16	17	
	17		20
	18		20
	19		20
	20	22	
	21	22	
$*$	22	6 12 17	20 26 32 37 40
	23		26 32 37 40
	24		26 40
	25		26
	26		28
	27		28
	28		26 40
	29		26 40
	30		32 37
	31		32
	32		34
	33		34
	34		32 37
	35		32 37
	36		37
	37		40
	38		40
	39		40
	40	42	
	41	42	
$*$	42	6 12 17	20 26 32 37 40
$*$	43	6 12 17	20 26 32 37 40
$*$	44	6 12 17	20 26 32 37 40

Table 2: Transition Table of NFA

And the NFA:

Following the **subset construction method**, we obtain the table 3(a) and we can rename the states like in the table 3(b).

Table 3: Transition Table of DFA

(a) Transition Table of DFA

	States	0	1
→ * A	1	6_12_17	20_26_32_37_40
B	6_12_17	8_14	20
C	20_26_32_37_40	22_42	28_34_40
D	8_14	6_12_17	20
E	20	22	Dead
* F	22_42	6_12_17	20_26_32_37_40
G	28_34_40	42	26_40_32_37
* H	22	6_12_17	20_26_32_37_40
* I	42	6_12_17	20_26_32_37_40
J	26_40_32_37	42	28_34_40
	Dead	Dead	Dead

(b) Transition Table of DFA (Renamed)

	States	0	1
→ *	A	B	C
	B	D	E
	C	F	G
	D	B	E
	E	H	Dead
*	F	B	C
	G	I	J
*	H	B	C
*	I	B	C
	J	I	G
	Dead	Dead	Dead

Following the **state minimization algorithm** for DFAs, we obtain the table 4 where X are all initial pairs of distinguishable states, X are all pairs of distinguishable states in the first round and O are all pairs of equivalent states.

		J	* I	* H	G	* F	E	D	C	B	→ * A
	Dead	X	X	X	X	X	X	X	X	X	X
→ *	A	X	O	O	X	O	X	X	X	X	
	B	X	X	X	X	X	X	O	X		
	C	O	X	X	O	X	X	X			
	D	X	X	X	X	X	X				
	E	X	X	X	X	X					
*	F	X	O	O	X						
	G	O	X	X							
*	H	X	O								
*	I	X									

Table 4: Minimization Table of DFA

The Minimum-state DFA is:

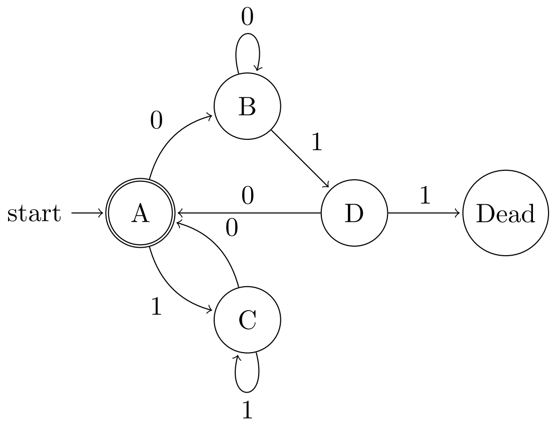
Table 5: Minimum-state Table of DFA

(a) Minimum-state Table of DFA

	States	0	1
→ *	A_I_H_F	B_D	C_J_G
	B_D	B_D	E
	E	A_I_H_F	Dead
	C_J_G	A_I_H_F	C_J_G
	Dead	Dead	Dead

(b) Minimum-state Table of DFA (Renamed)

	States	0	1
→ *	A	B	C
	B	B	D
	D	A	Dead
	C	A	C
	Dead	Dead	Dead



3. Implement it in a programming language (`Python`, `C/C++`, `Java`) following the table method.

   Solution 3.