Garret Mook, Jorge Calderon, Katie Pell
Dr. Ravikumar
CS454
22 April 2022

Final Project Documentation

Table Of Contents

RegEx

Note: this assumes RegEx Validation is not needed.

Function that parses a string and puts every char in a list:

```
def Parse_Input(s):
    parseList = []

    for i in s:
        parseList.append(i.split())

    print(parseList)

s = "(a+a.b)*.(a+ε)"
Parse_Input(s)
```

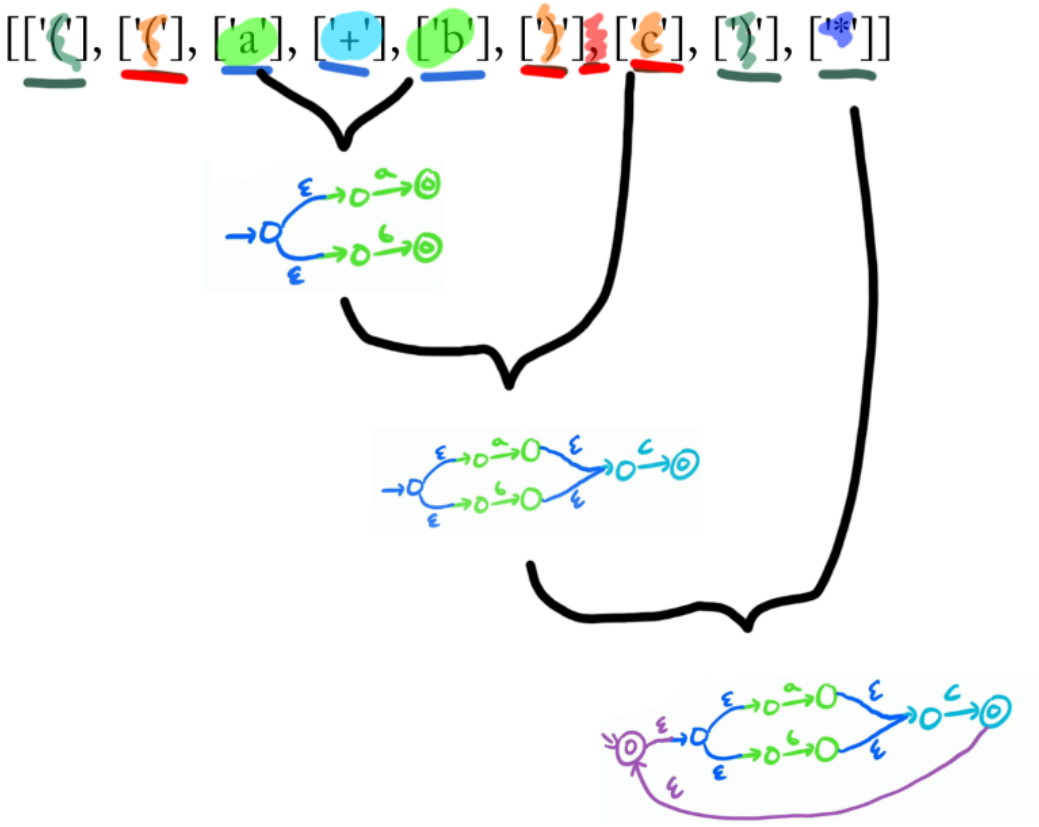
Output:

```
[['('], ['a'], ['+'], ['a'], ['.'], ['b'], [')'], ['*'], ['.'], ['('],
['a'], ['+'], ['ε'], [')']]
```

RegEx to ε-NFA

Input for ε-NFA
A list of characters

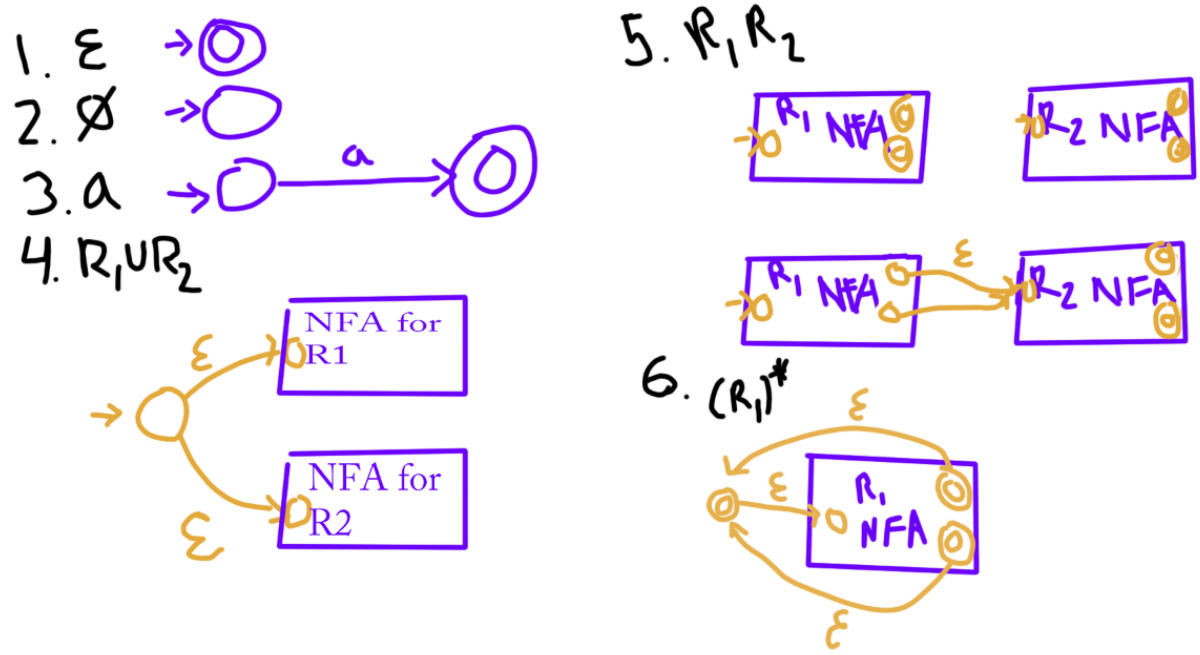
Example:
RegEx = "((a+b)c)*"
Parse_Input(RegEx) = [['('], ['('], ['a'], ['+'], ['b'], [')'], ['c'], [')'], ['*']]



Info about Regex of ε-NFA
We want to make a NFA out of a given RegEx(1).
Given: Regex
Want: NFA

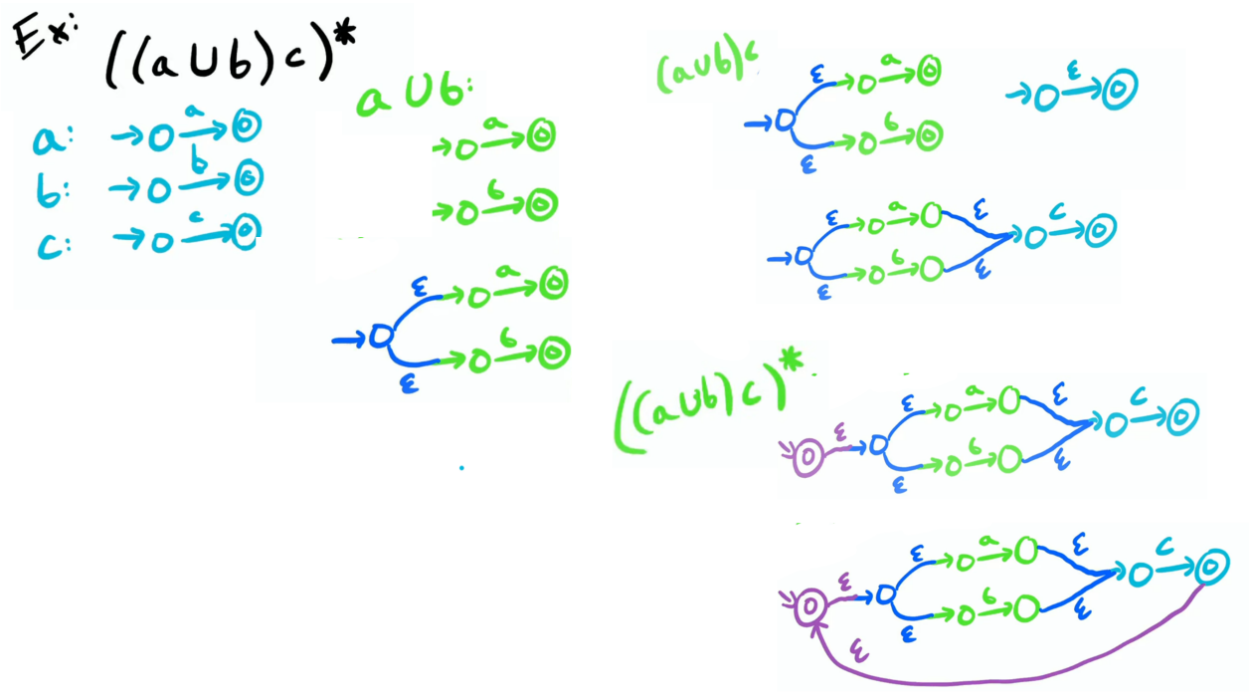
RegEx has six possibilities:

1. ε
2. $\varnothing$
3. a (a single character)
4. $R_1 \cup R_2$ (union of 2 RegEx's)
5. $R_1 R_2$ (concatenation of 2 RegEx's)
6. $(R_1)^*$

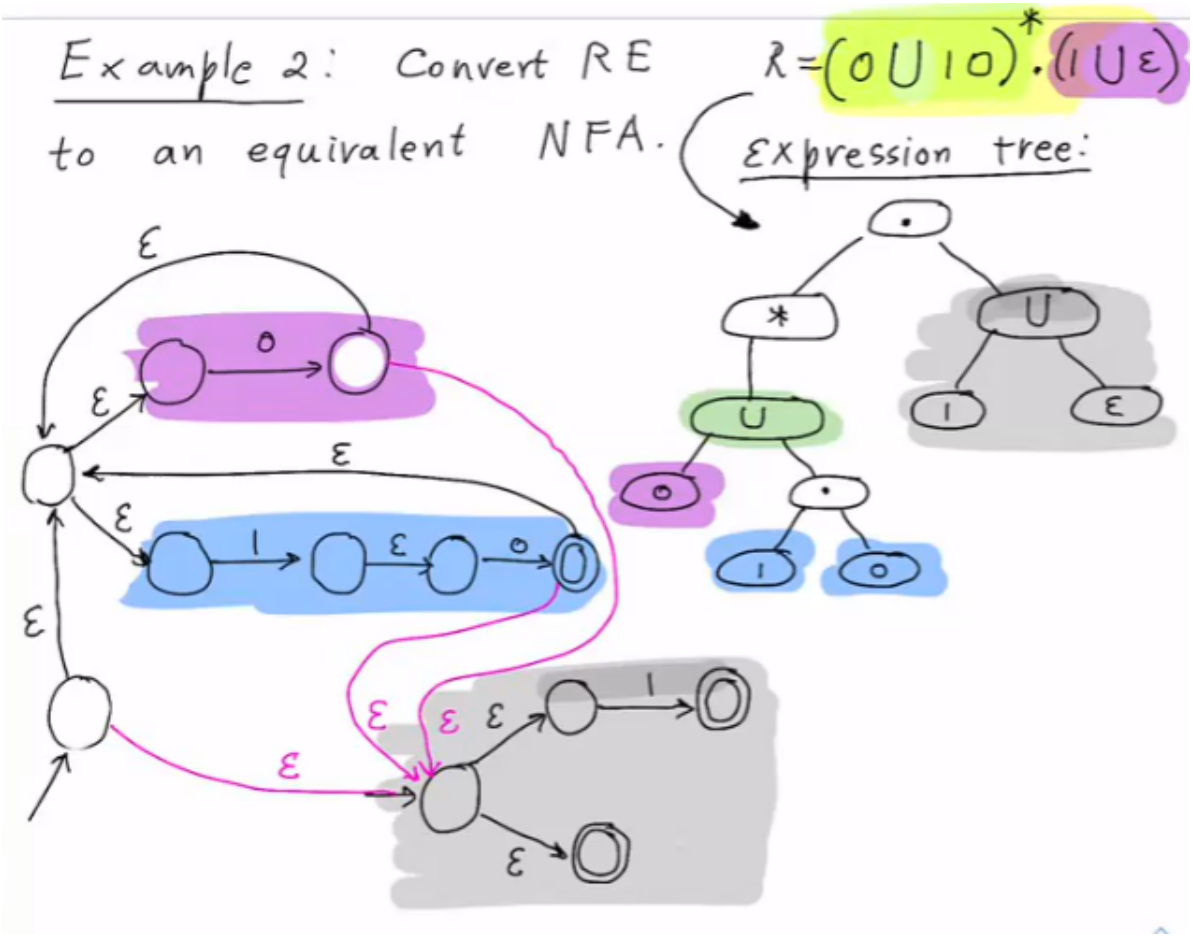


∴ Given any RegEx we can make a NFA for that particular RegEx. This shows that every language recognized by a regular expression is regular(1).

Let's look at an example of converting a RegEX to a NFA(2)



Infix to Postfix



size of a RE $\overbrace{R}$ = # of characters in R

Question: If a RE R of size n is converted to an NFA using the above algorithm, what is the size of the resulting NFA?

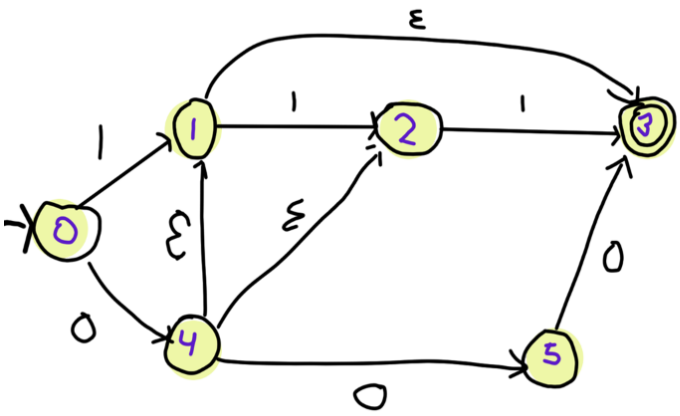
(# of states)

Answer: $\leq cn$ for some constant c.

Source: (3)

Output of ε-NFA
The output will be a transition table

Example of a transition table from ε-NFA



	0	1	ε
0	4	1	∅
1	∅	2	3
2	∅	3	∅
3	∅	∅	∅
4	5	∅	12
5	3	∅	∅

Example in Python:

```
import pandas as pd
df=pd.DataFrame({'0':[4,{},{},{},5,3],'1':[1,2,3,{},{},{}],'ε':[{},3,{},{},
12,{}]})
print(df)
```

Output:

```
    0   1   ε
0   4   1  {}
1  {}   2   3
2  {}   3  {}
3  {}  {}  {}
4   5  {}  12
5   3  {}  {}
```

ε-NFA to ε-free NFA

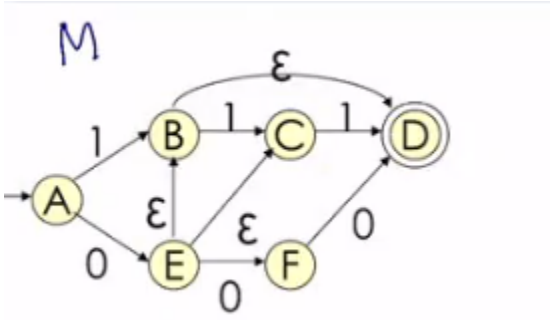
Input for ε-free NFA
Transition table

Info about ε-free NFA

ε-NFA → ε-free NFA (does not increase the # of states) (4)
Can convert any ε-NFA to an equivalent ε-free NFA (5)

Example:

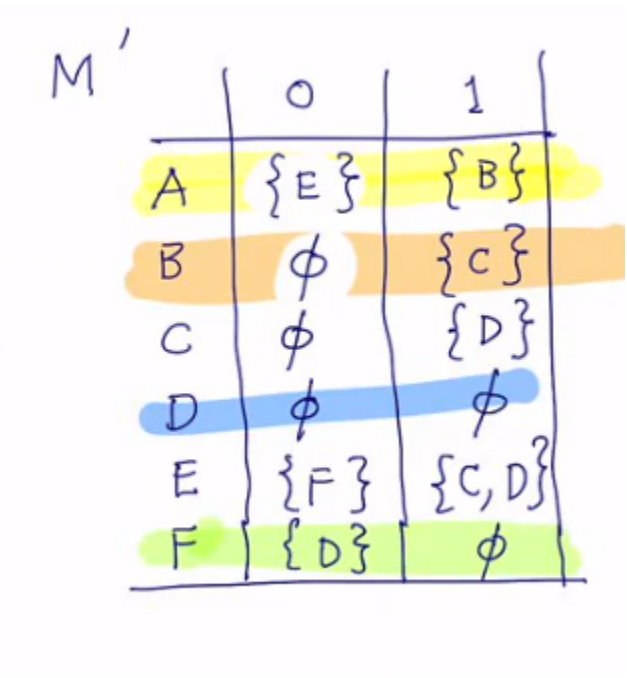
Step 1: Compute epsilon closure on ε-NFA M

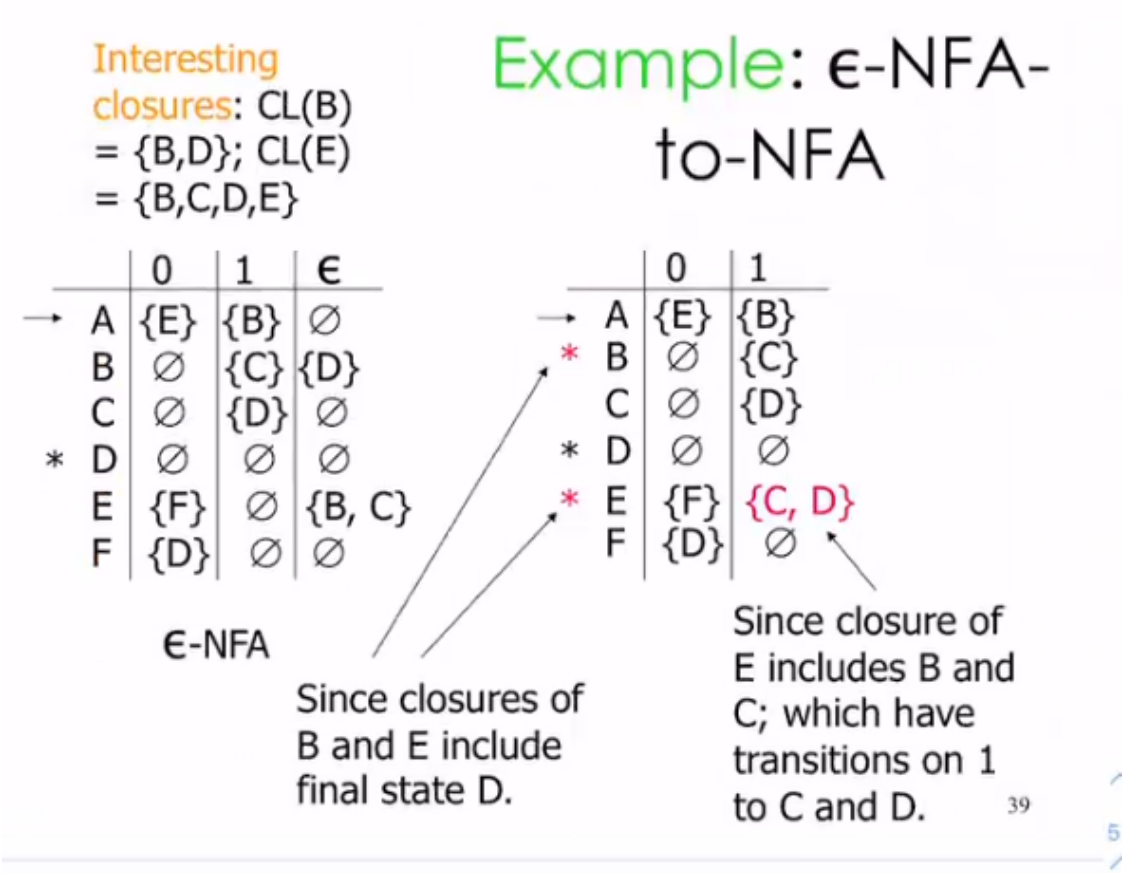


		0	1	ϵ
→	A	{E}	{B}	∅
	B	∅	{C}	{D}
	C	∅	{D}	∅
*	D	∅	∅	∅
	E	{F}	∅	{B, C}
	F	{D}	∅	∅

CLO(A) = {A}
CLO(B) = {B, D}
CLO(C) = {C}
CLO(D) = {D}
CLO(E) = {B, C, D, E}
CLO(F) = {F}

Step 2: Create the transition table for the ε-free NFA M'





Source: (6)

Other Thoughts on implementation

- Have an NFA class that contains a transition table
    - Transition table initializes to be the correct size but empty
    - Make sure that you are updating the fields NOT appending new ones
- Started with ε-NFA transition table, found epsilon closures
- See code for detailed comments

Output of ε-free NFA

Transition table

Work Cited

(1) “Regular Expression (Regex) to NFA Conversion - Easy Theory.” YouTube, uploaded by Easy Theory, 4 Aug. 2020, www.youtube.com/watch?v=HLOAwCCYVxE.

(2) “Conversion of Regular Expression (Regex) to NFA Example.” YouTube, uploaded by Easy Theory, 5 Aug. 2020, www.youtube.com/watch?v=c-IoxIZFeRQ.

(3) Dr.Ravikumar, Lecture Part 2, Feb 21st 00:25:18

(4) Dr.Ravikumar, Lecture Par 2, 23 Feb 2022. 00:08:21

(5) Dr.Ravikumar, Lecture Par 2, 9 Feb 2022. 00:31:23

(6) (5) Dr.Ravikumar, Lecture Par 1, 14 Feb 2022. 00:43:23