

Compilers Prog. Assign.

Semester [CMP403]

Authors

Evram Yousef
evramyousef@gmail.com
Sec. 1 Bn. 8

Omar Ahmed
omar.ahmed983@eng-st.cu.edu.eg
Sec. 1 Bn. 36

Supervised by:

Eng. Mahmoud Khaled

Main assumptions for validation

- 1- No spaces, any added space is discarded.
- 2- Digits allowed and treated like characters.
- 3- ? , . , - , _ = + ..etc. all treated like characters, the only ones that matter are | , (,) , * .
- 4- Any missing brackets are not added, and validated as false RE.
- 5- Any missing placement such as a| , |a etc. are detected.
- 6- “Epsilon” is called “eps”.

Main assumptions for parsing

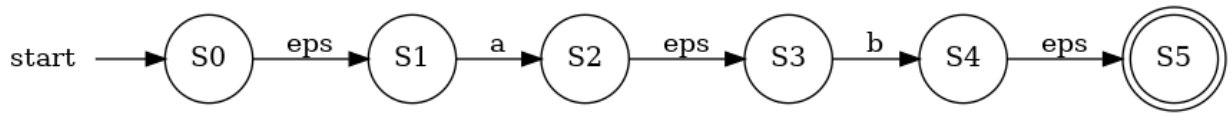
What we did at parsing was not ideal parsing as we understand it, we figured that we build a system that will make it easier for us when we implement the states (main) function, that we parse all substrings with rounded brackets, so that this RE: ab^* will become $(a(b^*))$.

Parsing with brackets made the algorithm for building the states and their connection more straight forward.

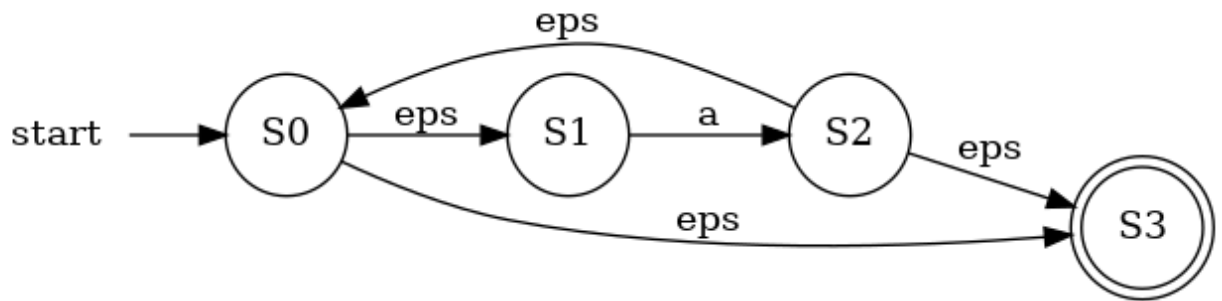
Main assumptions for states

We added more “epsilons” that’s all. Adding more “eps” won’t harm the algorithm much, since we already know that the following steps in creating DFA will remove them, but -as we learned- NFA are more understandable. Thus we added “eps” between every concatenation, so that if we come across a “*” we can deal with it easily. Also, it made it easier for us when dealing with nested ORs.

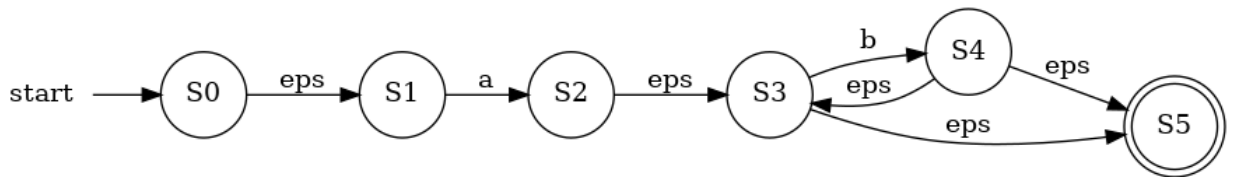
Figures for main parts



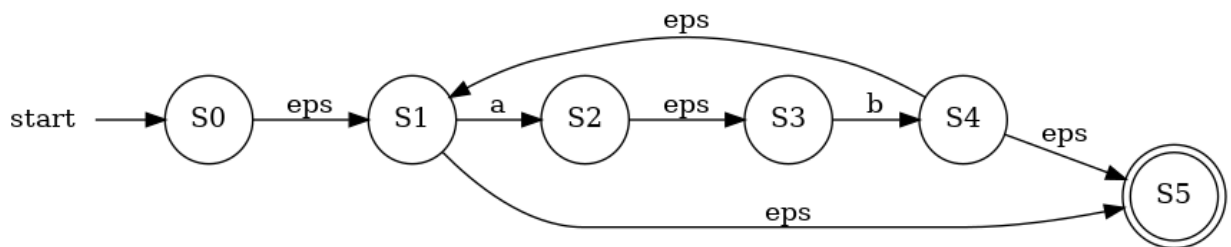
(ab)



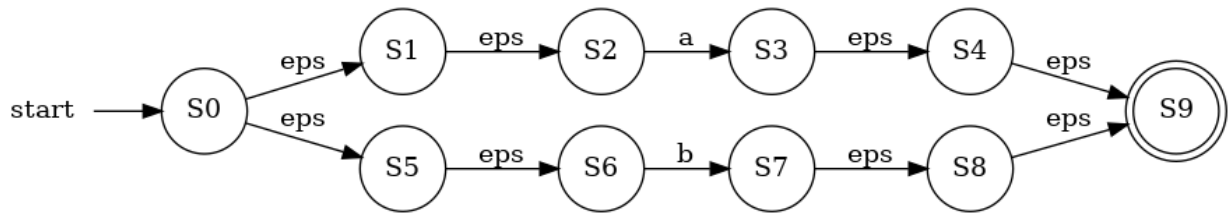
(a^*)



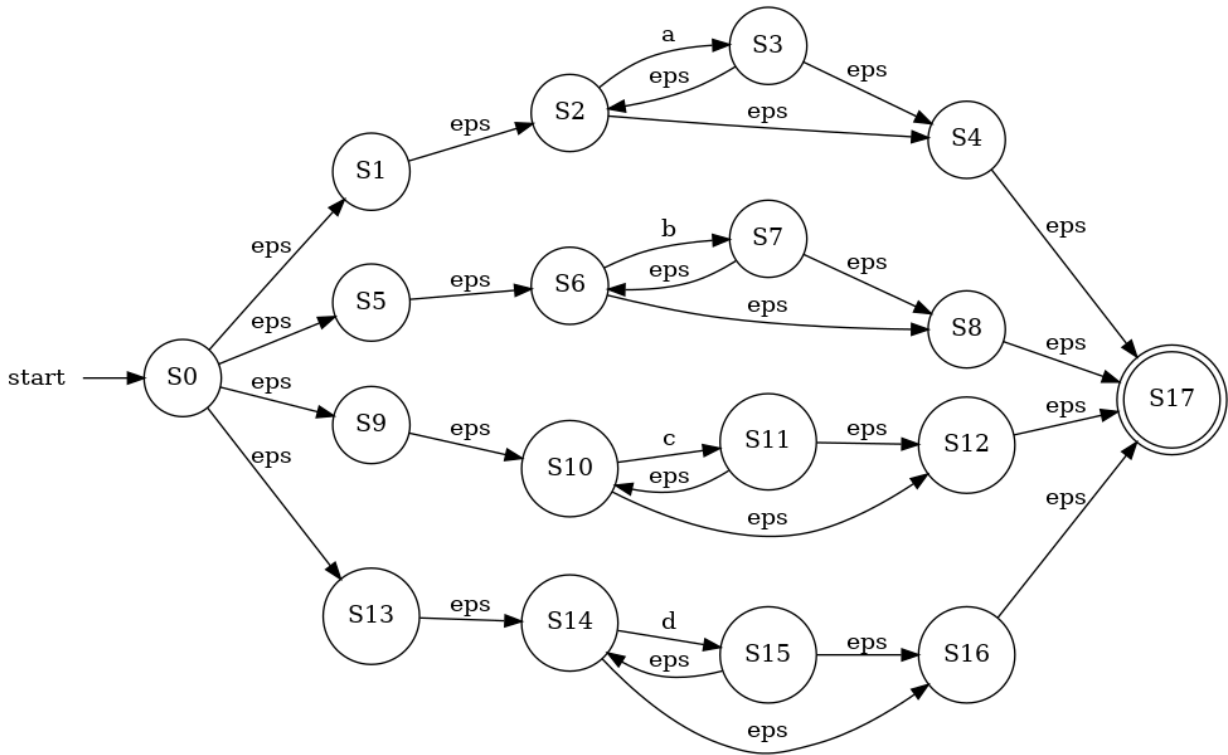
$ab^* \rightarrow (a(b^*))$



$(ab)^*$



$(a|b)$



$a^*|b^*|c^*|d^*$

