

Tarefa Prática 1 – CTC-34 – Expressões Regulares e Autômatos Finitos

Prazo: 3/setembro/2018 – Projeto para ser realizado em grupo de até 3 pessoas.

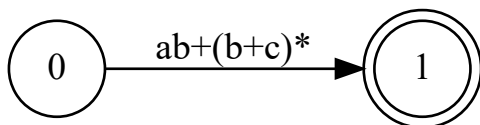
Entrega eletrônica, compartilhar o fonte em Python/JS/Java ou C e o relatório em PDF.

Serão fornecidos exemplos para teste que devem constar do relatório.

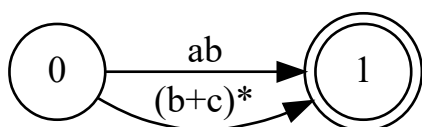
Os grafos produzidos devem ser gerados no formato Graphviz e apresentados como figura.
(pode usar o webgraphviz)

Prof. Carlos Forster

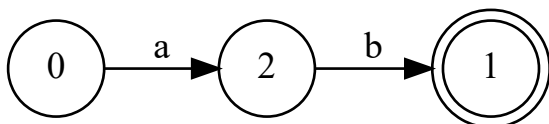
1 – Implementar o seguinte algoritmo, obtendo como entrada uma expressão regular e produzindo um autômato ε -AFN.



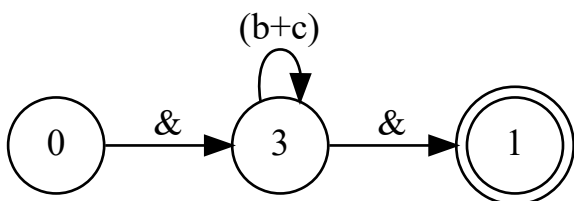
I – Inicia o autômato como um nó inicial e um nó final com transição especificando a expressão regular.
0, 1, $ab+(b+c)^*$
próximo nó: 2



II – verificar se é uma união de linguagens (expressa como $A+B$), separar a união em arcos
0, 1, ab
0, 1, $(b+c)^*$



III- verificar se é uma concatenação de linguagens, separar utilizando um novo nó
0, 2, a
2, 1, b
próximo nó: 3



IV – se a expressão for um fecho de Kleene, construir um loop em novo nó separado por ε -transições
0, 3, ε
3, 3, $(b+c)$
3, 1, ε



V – se a expressão estiver entre parênteses, remover os parênteses.

VI – repete II, III, IV, V e VI para cada arco até que todo arco tenha apenas um símbolo ou ε (repr &).

2 – Implemente a remoção das ε -transições de um autômato (isto pode resultar em múltiplos estados finais). Pode utilizar o seguinte procedimento:

I – computar o ε -fecho de cada estado

II – todo arco de A em X gera um arco de A em Y para cada Y no ε -fecho(X)

III – todo arco de Y em A para qualquer Y no ε -fecho(X) gera um arco de X para A.

IV – X é estado final se algum Y no ε -fecho(X) for final.

3 – Dados: (a) uma expressão regular, um ε -AFN ou um AFN conforme os itens acima e (b) uma cadeia de entrada, implemente uma função que encontre todas as sub-cadeias da cadeia do item (b) aceitas pela estrutura do item (a).

Dica: converter (a) para AFN usando (1) e (2), simular o AFN a partir de cada posição da cadeia (b).