INSTITUTO FEDERAL
CEARÁ

Ernani Andrade Leite

ernani@ifce.edu.br

Expressões Regulares

Expressões Regulares

As linguagens regulares podem ser representadas usando uma notação denominada expressão regular. Uma expressão regular utiliza apenas símbolos terminais e alguns caracteres especiais (metacaracteres), sendo que estes têm a função de especificar a quantidade de vezes que um terminal ou grupo de símbolos terminais se repetem dentro da formação de uma palavra pertencente à linguagem.

Definição. A definição formal de uma expressão regular pode ser feita recursivamente da seguinte maneira:

- Se **x** é um terminal, **x** é expressão regular.
- Se **r** e **s** são expressões regulares, então **r+s** é expressão regular. O sinal **+** indica que tanto **r** ou **s** formam palavras válidas
- Se **r** e **s** são expressões regulares, então **rs** é expressão regular, onde as palavras são formadas pela concatenação de **r** e **s**.
- Se **r** é expressão regular, $\mathbf{r}^*$ denota a expressão regular onde **r** é repetida zero ou mais vezes
- Se **r** é expressão regular, $\mathbf{r}^+$ denota a expressão regular onde **r** é repetida uma ou mais vezes. Portanto $\mathbf{r}^+ = \mathbf{rr}^*$
- Utilizam-se parêntesis para agrupar símbolos que se repetem conjuntamente.

Expressões Regulares

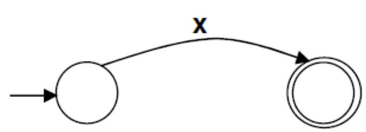
Exemplos:

- 001(0+1) denota L = {0010, 0011}
- 01(10+11) denota L = { 0110, 0111}
- (0 + 01)1 denota L = { 01, 011}
- ab^*a denota L = {aa, aba, abba, abbba, ...}, isto é, L = { w | w inicia com um **a** seguido de zero ou mais **b**'s e termina com um **a**}
- $a(bc)^*a$ denota L = {aa, abca, abcbca, abcbcbca, ...}, isto é, L = { w | w inicia-se com um **a** seguido de zero ou mais seqüências de **bc** e termina com **a**}
- ab^+a denota L = {aba, abba, abbba, ...}, isto é, L = { w | w inicia-se com um **a** seguido de um ou mais **b**'s e termina com **a**}
- (ab + bc)a denota L = {aba, bca}
- $(a+b)^*$ denota L = $\{a,b\}^*$, isto é todas as palavras possíveis de serem formadas a partir do símbolos **a** e **b**, inclusive a palavra vazia.
- $(a+b)^*c(a+b)^*$ denota L = { w | w possui exatamente um **c** sobre $\Sigma = \{a,b,c\}^*$ }

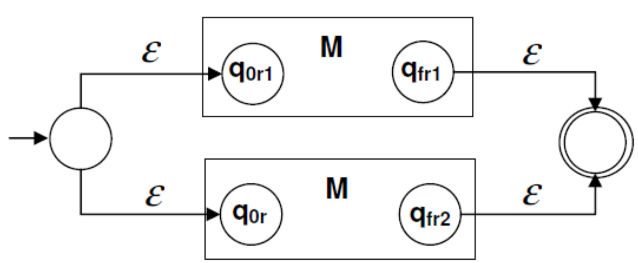
Transformação de Expressões Regulares em AFNε

Existe um algoritmo simples para conversão de expressões regulares em AFε. Segue sua descrição:

- Se uma expressão regular r é constituída por um único símbolo x, então r pode ser representada por um autômato de apenas 2 (dois) estados, tal que existe uma transição do primeiro para o segundo estado lendo o símbolo x.
- Se uma expressão regular r é da forma $r = r_1 + r_2$, isto é, as palavras válidas são palavras da linguagem descrita por r_1 ou da linguagem descrita por r_2, então r pode ser representada pela união dos autômatos M_1 e M_2, que reconhecem r_1 e r_2 respectivamente, de maneira que M_1 e M_2 formem caminhos exclusivos entre si utilizando transições ε.



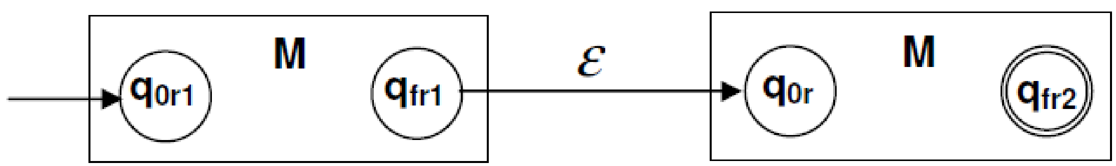
$r = x$



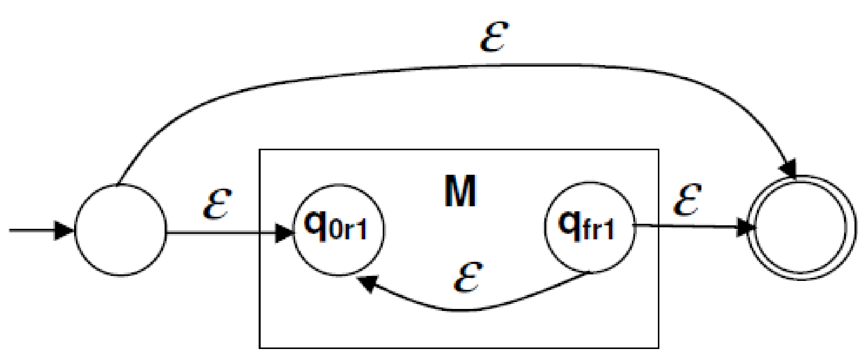
$r = r_1 + r_2$

Transformação de Expressões Regulares em AFNε

- Se uma expressão regular r é da forma $r = r_1 r_2$, isto é, as palavras válidas são obtidas da concatenação das palavras da linguagem descrita por r_1 com palavras da linguagem descrita por r_2, então r pode ser representada pelo seqüenciamento dos autômatos M_1 e M_2, que reconhecem r_1 e r_2 respectivamente, de maneira que M_2 segue M_1 utilizando uma transição **ε**.
- Apenas o estado que não possui transições com origem nele é um estado final.



$r = r_1 r_2$



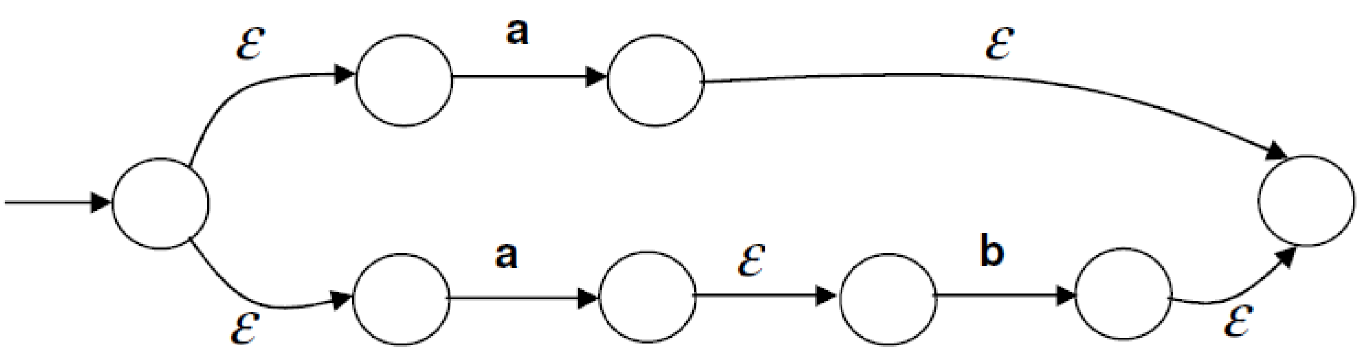
$r = r_1^*$

Transformação de Expressões Regulares em AFNε - Exemplo

Inicialmente procura-se desenhar os autômatos dos símbolos isoladamente, que são de fácil representação. Os autômatos obtidos são combinados segundo as regras anteriores até se obter o autômato que representa a expressão regular completa.

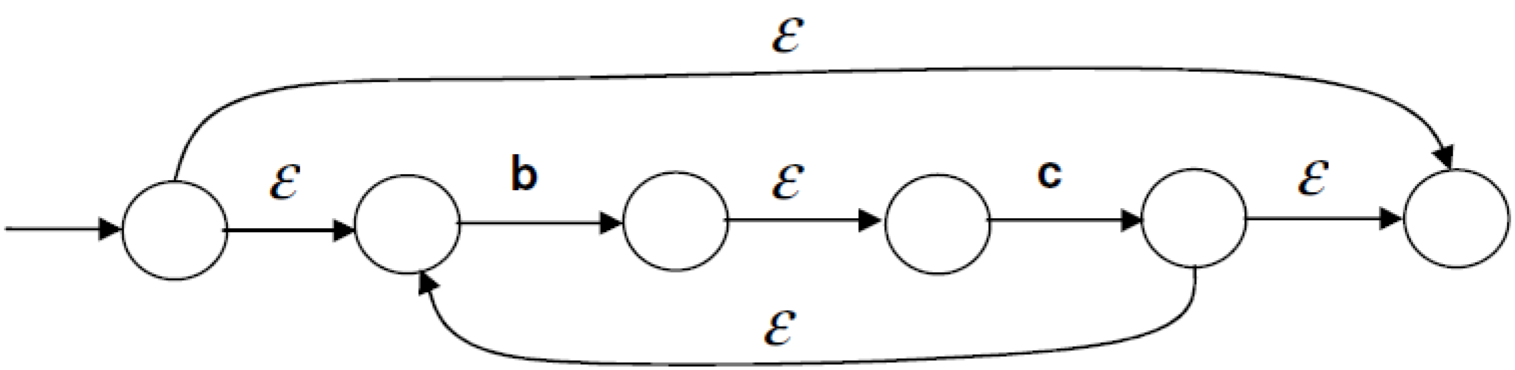
Exemplo: seja a expressão regular **(a + ab)(bc)***.
Observe que ela é a concatenação das subexpressões **(a +ab)** e **(bc)***. Por sua vez, **(a + ab)** indica que apenas as palavras **a** e **ab** são válidas dentro da subepressão. Já **(bc)*** é a concatenação dos símbolos **b** e **c**, repetidos em grupo, zero ou mais vezes.

Assim, primeiro montamos o autômato para a primeira subexpressão **(a + ab)** que será:



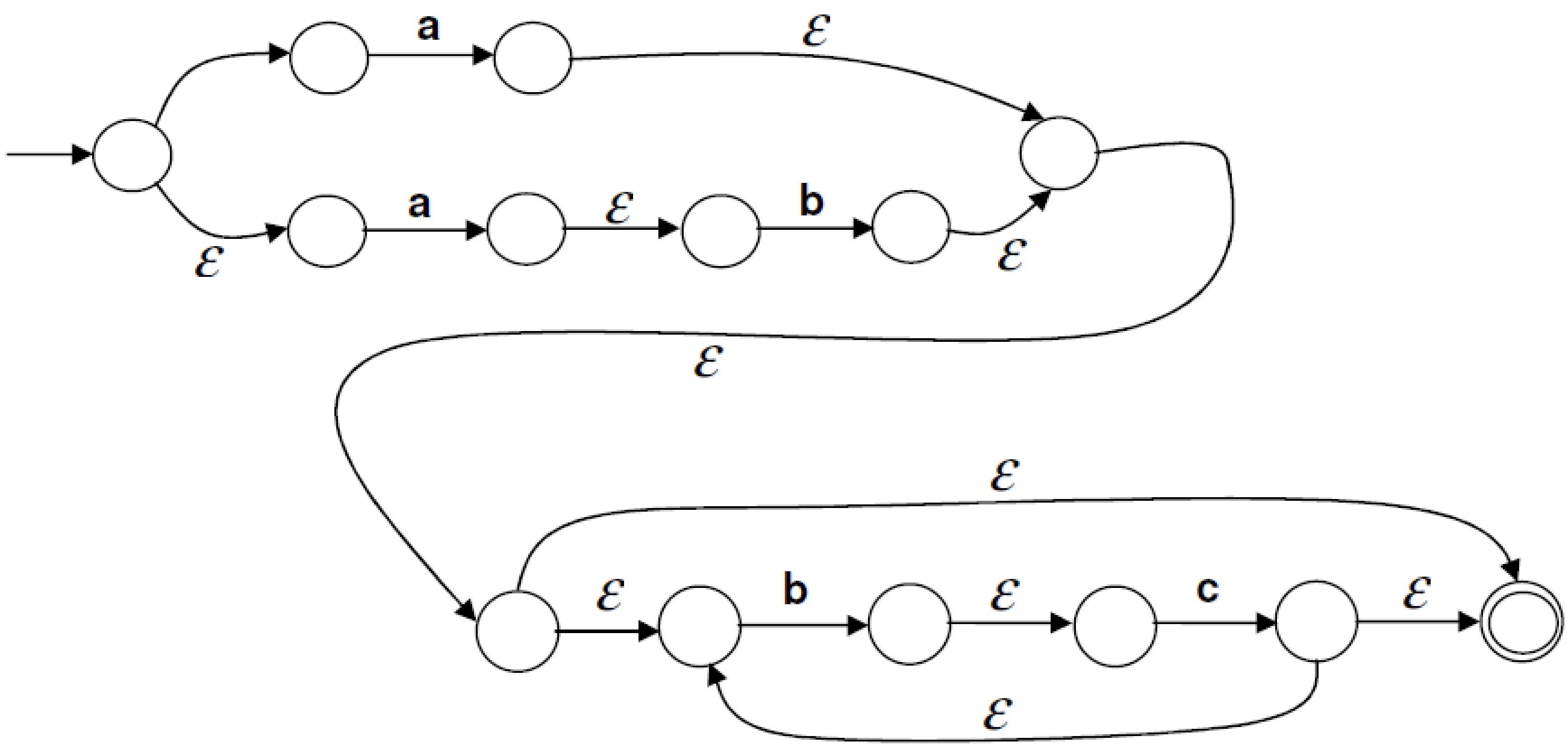
Transformação de Expressões Regulares em AFNε - Exemplo

Para montar o autômato para a segunda subexpressão **(bc)***, primeiro montamos o autômato para **bc** somente, e sobre o seu resultado aplicamos a regra que permite repeti-lo zero ou mais vezes, obtendo assim:



Agora, uma vez obtidos os autômatos para **(a +ab)** e **(bc)*** basta colocá-los em seqüência usando uma transição **ε** para que representem a concatenação das duas subexpressões, e definir o estado que não possui transições a partir dele como estado final, representando assim o autômato para **(a + ab)(bc)***.

Transformação de Expressões Regulares em AFNε - Exemplo



Referências Bibliográficas

- MENEZES, Paulo Blauth. **Linguagens formais e autômatos**. 5.ed. Porto Alegre (RS): Bookman, 2008. 215 p. (Livros Didáticos; v. 3)
- SIPSER, Michael. **Introdução à teoria da computação**. São Paulo (SP): Cengage Learning, 2011. 459 p. Tradução da 2ª edição americana.
- **Fonte:** **(material adaptado)** http://www.icmc.sc.usp.br/~mdgvnune/download/sce5832/Teoria.html