

Processamento de Linguagens e Compiladores

Regras para Conversão de ER para AND

Pedro Rangel Henriques

7 de Novembro de 2021

Expressões Regulares e Autómatos

Para reconhecer com muita eficiência *ocorrências de padrões* definidos por Expressões Regulares (ER) em textos (sequências de caracteres) é necessário recorrer a um algoritmo simples, iterativo, guiado pela função de transição de um Autômato Determinista (AD).

De modo a que tal solução seja viável é necessário transformar sistematicamente a ER que define o padrão a pesquisar. Porém o processo formal apenas assegura que uma dada ER seja transformada num Autômato Não-Determinista (AND).

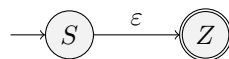
Posteriormente esse AND será transformado num AD, também rigorosamente e de forma passível de ser automatizada. Essa transformação é baseada numa função designada por ε -fecho() que dado um conjunto de estados aglutina nesse conjunto todos os estados que são alcançáveis de cada elemento do conjunto por caminhos de custo ε .

Porém esse processo não será aqui explicado. Em vez disso mostra-se na subsecção seguinte através de um exemplo como se pode informalmente passar de uma ER para um AD.

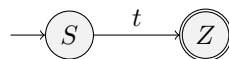
Conversão formal de ER em AND

Apresentam-se a seguir as **Regras para transformar uma dada ER e num AND**.

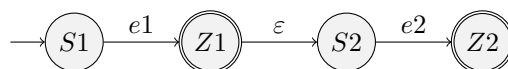
Regra 1 (frase nula) $e = \varepsilon$



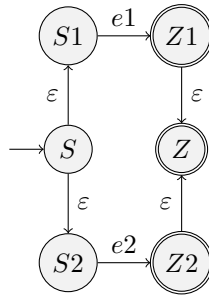
Regra 2 (símbolo terminal) $e = t, t \in T$



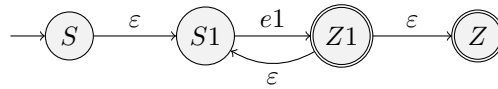
Regra 3 (concatenação) $e = e1 \cdot e2$



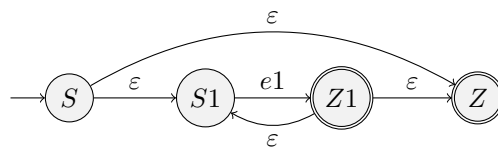
Regra 4 (união) $e = e1 \mid e2$



Regra 5 (fecho transitivo) $e = e1^+$



Regra 6 (fecho de Kleene) $e = e1^*$



Exemplos da Conversão formal de ER a AND

Para ilustrar a aplicação prática destas 6 regras de conversão a uma ER concreta mais complexa mostram-se a seguir dois exemplos.

Exemplo 1:

Desenhe um autômato determinista correspondente à expressão regular:

$$e = a (c \mid db) a$$

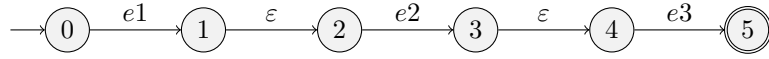
Para fazer a conversão é importante analisar as operações de maior ou menor prioridade para identificar quais os grupos de subexpressões que compõem a expressão inicial e . Neste caso é fácil perceber que e é a concatenação de 3 componentes

$$e = e1 \cdot e2 \cdot e3$$

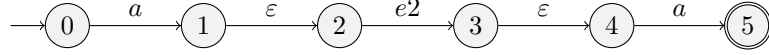
com

$$\begin{aligned} e1 &= a \\ e2 &= (c \mid db) \\ e3 &= a \end{aligned}$$

e daí ser fácil perceber que o AND final terá a seguinte estrutura:



No próximo passo substituímos as subexpressões pelos respectivos subautômatos. No caso de $e1$ e $e3$ é muito simples porque se trata de ER atômicas (símbolos terminais); assim por aplicação da Regra 2, obtém-se:

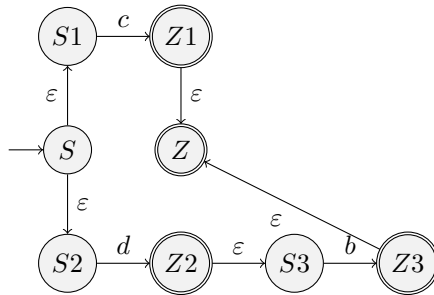


Quanto a $e2$ é bom de ver que não é uma ER atômica e que requer que, recursivamente, se aplique o mesmo método de conversão, identificando as prioridades dos operadores e analisando os seus operandos. Neste caso trata-se de uma união entre outras duas subexpressões que podemos denotar como $e21$ e $e22$ sendo:

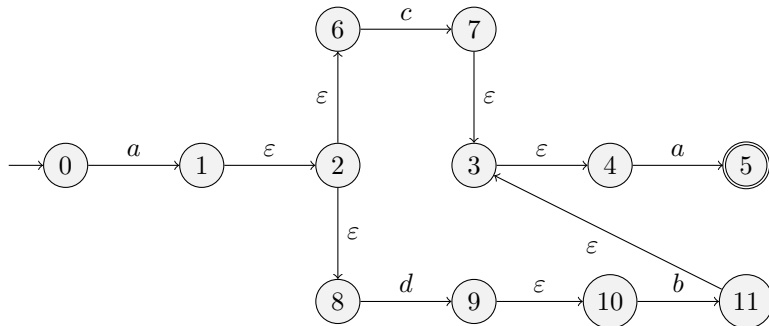
$$\begin{aligned} e21 &= c \\ e22 &= d \cdot b \end{aligned}$$

Relativamente a $e21$ atingimos a atomicidade e podemos usar a Regra 2. No caso de $e22$ identificamos uma concatenação de duas ER que são simples símbolos terminais, sendo possível aplicar a Regra 3.

Analizadas as duas subexpressões de $e2$, pode aplicar-se a Regra 4 para converter a união identificada e então obtém-se o AD seguinte:



Por fim o subautômato correspondente a $e2$ deve ser integrado no inicial para se obter o AND equivalente a e .



Exemplo 2:

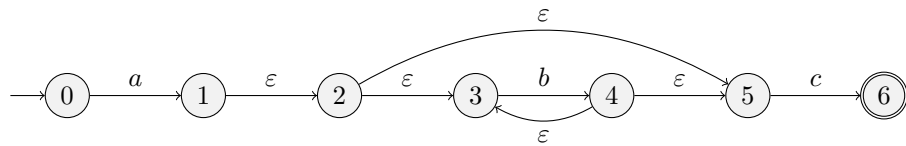
Desenhe um autômato determinista correspondente à expressão regular:

$$e = a b^* c$$

al como no exemplo anterior, para fazer a conversão é importante analisar as operações de maior ou menor prioridade para identificar quais os grupos de subexpressões que compõem a expressão inicial e .

Neste exemplo consta-se que há de novo duas concatenações reconhecendo-se como acima três subexpressões em que a primeira e a última são atômicas (símbolos terminais a que se aplica a Regra 2) e em que a segunda é mais complexa mas de novo atômica (um fecho de Kleene). Para esta $e2$ pode aplicar-se a Regra 6, sendo apenas necessário analisar a estrutura da subexpressão $e21$ que está na base desse expoente $*$. No caso presente $e21$ é apenas um terminal e portanto o seu subautômato obtém-se por aplicação da Regra 2.

Assim sendo pode concluir-se que o AND equivalente à ER e dada se representa conforme o esquema seguinte:



Passagem informal de ER a AD

Exemplo:

Desenhe um autômato determinista correspondente à expressão regular:

$$y (aa \mid cb) (cb)^* a$$

Por análise do significado de cada operador da ER dada e de cada um dos seus operandos e percebendo que o padrão define uma sequência de caracteres escritos da esquerda (estado inicial) para a direita (estado final) é fácil (para ER simples) representar o processo determinista de reconhecimento usando uma *máquina de transição de estado* (que corresponde ao AD pretendido).

