Teoria da Computação

Aula Teórica 16:
Conversão de AFDs em Expressões Regulares.

António Ravara

Departamento de Informática

8 de Maio de 2019

O Teorema de Kleene

Linguagens Regulares = Linguagens reconhecidas por AFDs

1. Mostrou-se a inclusão $\subseteq$ através da definição de dois algoritmos:
   1.1 Conversão de Expressões Regulares em AFNs.
   1.2 Conversão de AFNs em AFDs.
2. Mostra-se agora a inclusão $\supseteq$.

Como "transformar" AFDs em Expressões regulares?

Foram propostos vários algoritmos:

1. Método do fecho transitivo
   construção iterativa da expressão, pelas palavras que permitem transitar entre estados

2. Método da remoção de estados
   expressões regulares como etiquetas das transições

3. Método algébrico de Brzozowski.
   resolução de um sistema de equações lineares

Método do fecho transitivo

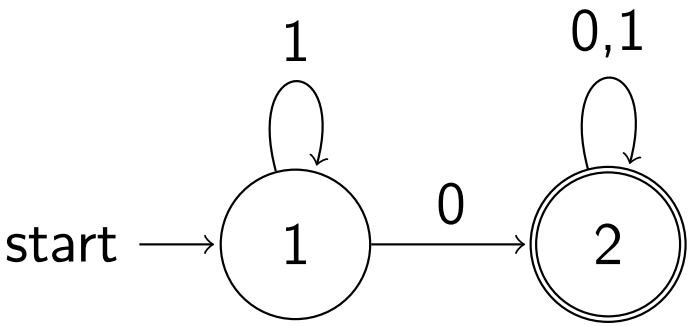
A ideia

1. Para cada par de estados (i, j) define-se a expressão R_{ij} que denota o conjunto das palavras que levam de i a j.
2. A expressão regular que denota a linguagem do autómato é a soma de todas as expressões R_{sf_k}, para cada $f_k \in F$, sendo s o estado inicial.

Definição (recursiva) de $R_{ij} \stackrel{\text{def}}{=} R_{ij}^{|S|}$

1. $$R_{ij}^0 = \begin{cases} a & \text{se } i \neq j \wedge \delta(i,a) = j \\ a + \epsilon & \text{se } i = j \wedge \delta(i,a) = j \\ \emptyset & \text{caso contrário} \end{cases}$$
2. $$R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}.$$

Método do fecho transitivo - exemplo



1. $R_{11}^0 = \epsilon + 1$
2. $R_{12}^0 = 0$
3. $R_{21}^0 = \emptyset$
4. $R_{22}^0 = \epsilon + 0 + 1$

1. $R_{11}^1 = R_{11}^0 + R_{11}^0 (R_{11}^0)^* R_{11}^0 = (\epsilon+1)+(\epsilon+1)(\epsilon+1)^*(\epsilon+1) = 1^*$
2. $R_{12}^1 = R_{12}^0 + R_{11}^0 (R_{11}^0)^* R_{12}^0 = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 = 1^* 0$
3. $R_{21}^1 = R_{21}^0 + R_{21}^0 (R_{11}^0)^* R_{11}^0 = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) = \emptyset$
4. $R_{22}^1 = R_{22}^0 + R_{21}^0 (R_{11}^0)^* R_{12}^0 = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0 = \epsilon + 0 + 1$
1. $R_{11}^2 = R_{11}^1 + R_{12}^1 (R_{22}^1)^* R_{21}^1 = 1^* + 1^* 0(\epsilon + 0 + 1)^* \emptyset = 1^*$
2. $R_{12}^2 = R_{12}^1 + R_{12}^1 (R_{22}^1)^* R_{22}^1 =$
   $1^* 0 + 1^* 0(\epsilon + 0 + 1)^*(\epsilon + 0 + 1) = 1^* 0(0 + 1)^*$
3. $R_{21}^2 = R_{21}^1 + R_{22}^1 (R_{22}^1)^* R_{21}^1 = \emptyset + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^* \emptyset = \emptyset$
4. $R_{22}^2 = R_{22}^1 + R_{22}^1 (R_{22}^1)^* R_{22}^1 =$
   $\epsilon + 0 + 1 + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*(\epsilon + 0 + 1) = (0 + 1)^*$

Método da remoção de estados

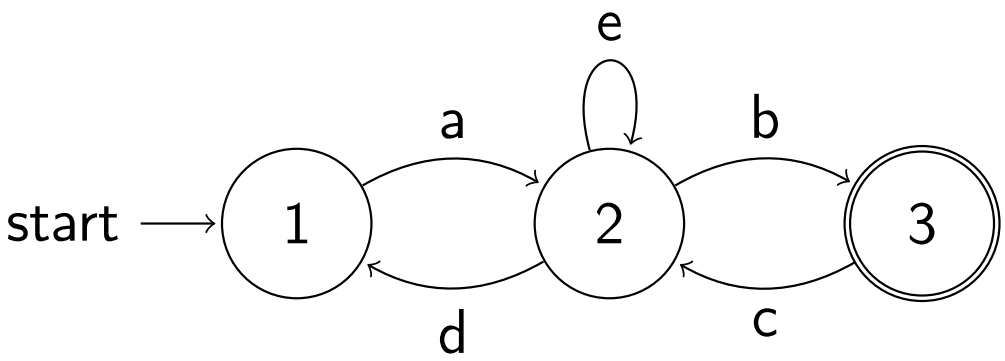
A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
2. A etiqueta de uma transição passa a ser uma expressão.

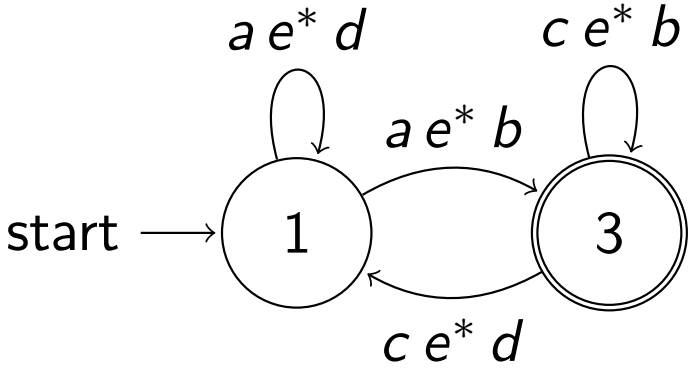
Remoção de estados

1. Ligam-se os estados finais por transições ϵ.
2. Se $\delta(i, e) = i$ então $\delta(i, e^*) = i$.
3. Se $\delta(i, e) = j$ e $\delta(i, f) = j$ então $\delta(i, e + f) = j$.
4. Se $\delta(i, e) = j$ e $\delta(j, f) = k$ então $\delta(i, ef) = k$ (remove-se o estado j).
5. Itera-se o processo até ficarem só o estado inicial e um final.

Método da remoção de estados - exemplo



passa a



Expressão: $(a\,e^*\,d)^*\,a\,e^*\,b\,(c\,e^*\,b + c\,e^*\,d\,(a\,e^*\,d)^*\,a\,e^*\,b)^*$

Método Algébrico

A ideia: estados como variáveis; acções como concatenações

1. Cria-se um sistema de equações sobre expressões regulares.
2. Resolve-se o sistema por substituição das variáveis que ocorrem no lado direito das equações (pelas suas expressões).
3. A expressão do estado inicial denota a linguagem do autómato.

Obtenção do sistema

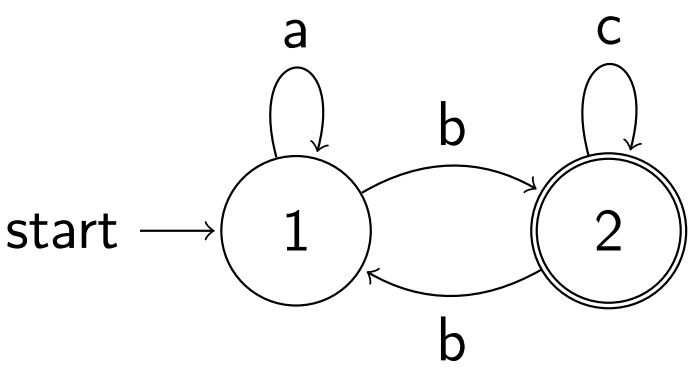
Se $\delta(i, a) = j$ então $i = aj$ e se $i \in F$ então $i = \epsilon$

Resolução do sistema

1. Se se tem $i = aj$ e $i = bk$ então substituem-se as duas equações por $i = aj + bk$.
2. Se se tem $i = aj$ e $i = \epsilon$ então substituem-se as duas equações por $i = aj + \epsilon$.

Método Algébrico

Um exemplo de aplicação.



1. $1 = a\,1$
2. $1 = b\,2$
3. $2 = b\,1$
4. $2 = c\,2$
5. $2 = \epsilon$

1. $1 = a\,1$
2. $1 = b\,2$
3. $2 = b\,1 + c\,2 + \epsilon$

e fica-se com

1. $1 = a\,1 + b\,2$
2. $2 = b\,1 + c\,2 + \epsilon$

Método Algébrico

Para resolver as equações recursivas usa-se o seguinte resultado.

Lema de Arden

Se $i = e\,i + f$ então $i = e^* f$.

1. De $2 = b\,1 + c\,2 + \epsilon$ obtém-se $2 = c^*(b\,1 + \epsilon)$
2. De $1 = a\,1 + b\,2$ obtém-se $1 = a\,1 + b\,c^*(b\,1 + \epsilon)$

$$\begin{aligned} 1 &= a\,1 + b\,c^*(b\,1 + \epsilon) \\ &= a\,1 + b\,c^*b\,1 + b\,c^*\epsilon \\ &= (a + b\,c^*b)\,1 + b\,c^* \\ &= (a + b\,c^*b)^*\,b\,c^* \end{aligned}$$

Comparação dos métodos

Método do Fecho Transitivo

- Fácil de implementar
- Difícil de usar manualmente
- Tende a gerar expressões longas e complexas

Método da Remoção de Estados

- Difícil de implementar
- Fácil de usar manualmente

Método Algébrico

- Difícil de implementar
- Fácil de usar manualmente
- Tende a gerar expressões compactas