Fundamentos Teóricos da Computação

CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. João Paulo Aramuni

UNIVERSIDADE FUMEC

Expressões Regulares

* **Expressões Regulares**

Expressões Regulares

* Outra forma de especificar uma linguagem regular.
* Trata-se de uma expressão (ou formulação linear) para especificar uma linguagem $L(M)$
* A notação é semelhante à utilizada para especificar uma linguagem regular por meio de conjuntos

Expressões Regulares

* Até o momento, foram vistas duas formas de especificar uma linguagem regular: uma, mediante o uso de notação usual de teoria de conjuntos, e outra, por meio do desenho de um diagrama de estados.

Um olhar para o futuro

* Veremos duas outras formas de especificar linguagens regulares: expressões regulares e gramáticas regulares. A primeira especifica uma linguagem por meio de uma expressão que a *denota* e a segunda, por intermédio de um conjunto de regras que a *gera*.

* A partir de uma expressão regular ou de uma gramática regular, pode-se construir um AF de forma automática. Assim, os dois formalismos são também dois instrumentos alternativos que podem ser utilizados como etapa intermediária para a obtenção de reconhecedores.

Expressões Regulares

* A denotação de uma linguagem via expressão regular pode ser útil quando se deseja uma maneira concisa de se referir à linguagem como um todo.

* Essa utilidade manifesta-se tanto no plano da teoria, na qual as expressões regulares são adequadas para as manipulações formais, quanto no plano prático, em que as expressões regulares, ou notações derivadas delas, têm sido utilizadas para referência compacta a conjuntos de palavras.

* Por exemplo, em editores de texto e comandos de sistemas operacionais.

Definição

* Uma expressão regular (*ER*) sobre um alfabeto Σ é definida recursivamente como:

  * Ø, λ e a para qualquer $a \in \Sigma$ são expressões regulares. Tais Ers denotam, respectivamente, os conjuntos Ø, $\{\lambda\}$, $\{a\}$;
  * Se r e s são expressões regulares, também são expressões regulares $(r+s)$, (rs) e r^*. Tais ERs denotam, respectivamente, $L(r) \cup L(s)$, $L(r)L(s)$, $L(r)^*$

Exemplos

ER	Conjuntos Regulares
* (01)	denota {0}{1} = {01}
* (0+1)	denota {0} U {1} = {0,1}
* (0+1)(01)	denota {0,1}{01} = {001,101}
* 0*	denota {0}* = $\{0^n \mid n >= 0\}$
* (0+1)*	denota {0,1}* = Σ*
* (0+1)*(1)(0+1)	denota ({0,1}*{1}{0,1}) = $\{w \in \Sigma^* \mid$ o penúltimo símbolo de w é 1$\}$.

Eliminação de Parêntesis

* $(r + s) + q = r + (s + q) = r + s + q$
* $(rs)q = r(sq) = rsq$
* Prioridade dos operadores (em ordem)
  * Fecho de Kleene (r^*)
  * Concatenação (sr)
  * União ($s + r$)
  * Exemplo: $(0 + (1(0^*))) = 0 + 10^*$

Exemplo 1

* $\{w \in \{a,b\}^* \mid w$ possui, ao menos, um b$\}$

* Visualizando-se a linguagem como um todo
  * $(a + b)^*b(a + b)^*$
  * Visualização não determinística

* Visualizando-se da esquerda para a direita
  * $a^*b(a + b)^*$
* Visualizando-se da direita para a esquerda
  * $(a + b)^*ba^*$

Equivalências de ERs

1. $r + s = s + r$
2. $r + \emptyset = r$
3. $r + r = r$
4. $r\lambda = \lambda r = r$
5. $r\emptyset = \emptyset r = \emptyset$
6. $(r + s)t = rt + st$
7. $r(s + t) = rs + rt$
8. $(r + s)^* = (r^*s)^*r^*$
9. $(r + s)^* = r^*(sr^*)^*$
10. $(rs)^* = \lambda + r(sr)^*s$
11. $r^{**} = r^*$
12. $r^* = (rr)^*(\lambda + r)$
13. $\emptyset^* = \lambda$
14. $\lambda^* = \lambda$
15. $r^*r^* = r^*$
16. $rr^* = r^*r$
17. $(r^* + s)^* = (r + s)^*$
18. $(r^*s^*)^* = (r + s)^*$
19. $r^*(r + s)^* = (r + s)^*$
20. $(r + s)^*r^* = (r + s)^*$

Simplificação de ERs

* Qualquer equivalência sobre ERs que não contenha o fecho de Kleene pode ser derivada utilizando-se as regras 1 a 7 da tabela anterior, mais as propriedades de associatividade da união e concatenação

* Não existe um conjunto de equivalências finito que permita derivar qualquer equivalência que envolva o fecho de Kleene
  * No entanto, muitas são redundantes e podem ser derivadas a partir de outras

Exemplo 2

$(00^*+10^*)0^*(1^*+0)^* = (0+1)0^*0^*(1^*+0)^*$ por 6

$= (0+1)0^*(1^*+0)^*$ por 15

$= (0+1)0^*(1+0)^*$ por 17

$= (0+1)0^*(0+1)^*$ por 1

$= (0+1)(0+1)^*$ por 19

* Obs: Note que esta linguagem não contém a palavra vazia (λ)

Exemplo 3

* ((0(0+1)1 + 11)0*(00+11))*(0+1)* = r(0+1)*
* r = ((0(0+1)1 + 11)0*(00+11))*
  * r contém λ e é constituída por {0,1}

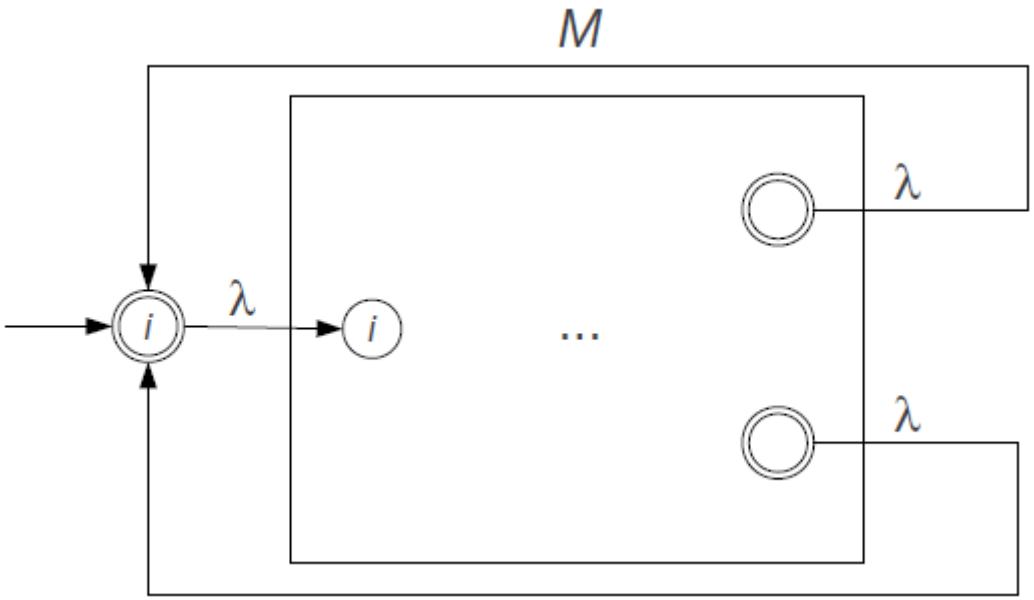
$$L(r) \subseteq L((0+1)^*)$$

* Então r(0+1)* = (0+1)*
  * Neste caso, não precisa-se simplificar r para chegar à fórmula simplificada

AFNλ a partir de ERs

* Pode-se construir facilmente AFNλs a partir de Ers
  * O primeiro passo é identificar a maior sub-expressão r que se saiba escrever um AFD (ou AFN) dentro da ER
    * Construa um AFD (ou AFN) para r
  * Repita o primeiro passo até que se tenha criado AFDs (ou AFNs) para cada sub-expressão da ER
  * Conecte os AFDs (ou AFNs) construídos utilizando transições sob λ de acordo com o tipo da operação
    * Se concatenação, conecte os estados finais do primeiro autômato com os estados iniciais do segundo
    * Se união, crie um estado inicial que leve aos estados iniciais dos autômatos
    * Se fecho de Kleene?

AFNλ para operação fecho de Kleene

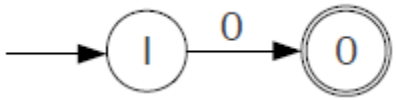


Exemplo 4

* Construir um AFNλ para a linguagem regular denotada pela ER (0+1)(0+1)*

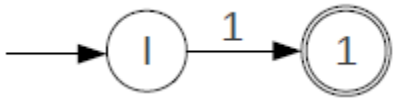
Exemplo 4

* AFN que reconhece 0



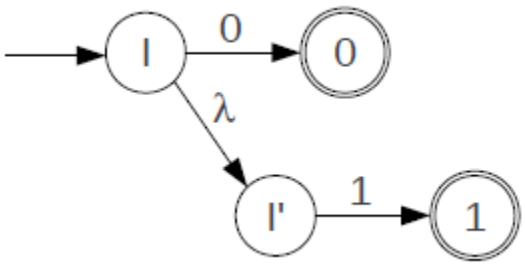
Exemplo 4

* AFN que reconhece 1



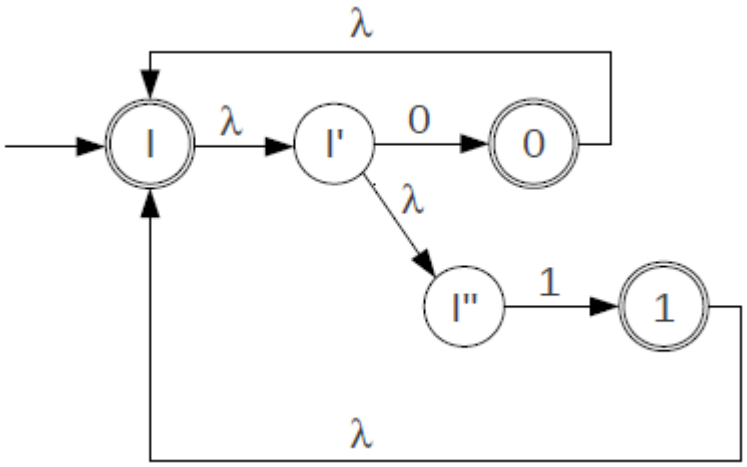
Exemplo 4

* AFNλ que reconhece (0+1)



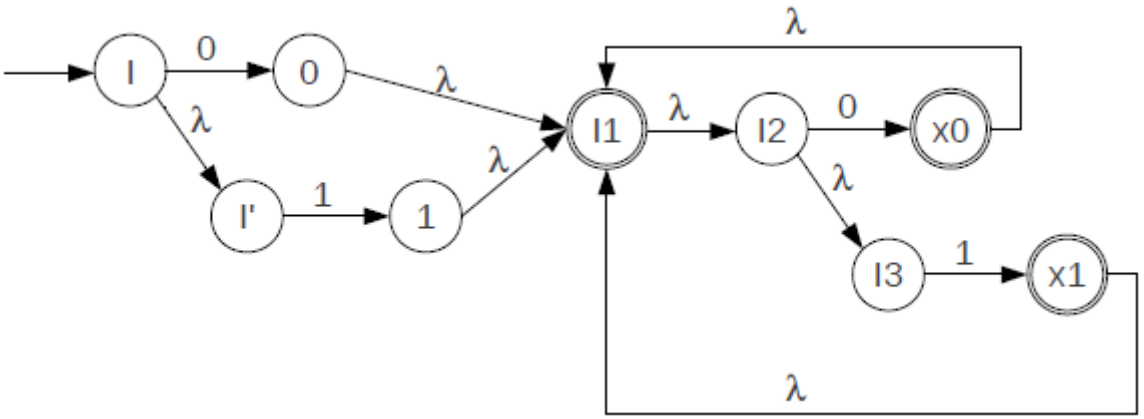
Exemplo 4

* AFNλ que reconhece (0+1)*



Exemplo 4

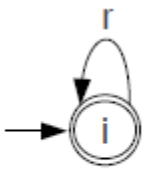
* AFNλ que reconhece (0+1)(0+1)*



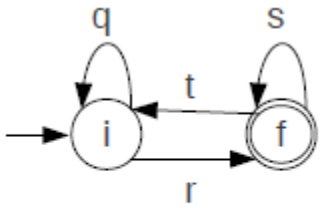
ERs a partir de AFDs

* É possível extrair uma ER a partir de um AFD. Para tal, utiliza-se uma notação similar a um AF, chamada diagrama ER.
  * Não veremos a definição formal, somente como utilizar o diagrama
* Inicialmente, o diagrama ER é igual ao AFD
* Elimina-se um estado (exceto o inicial e um dos finais) de cada vez até que se chegue a um dos diagramas básicos
  * Repete-se este procedimento para cada um dos estados finais. A ER será a união dos resultados de cada aplicação do procedimento

Diagramas Básicos



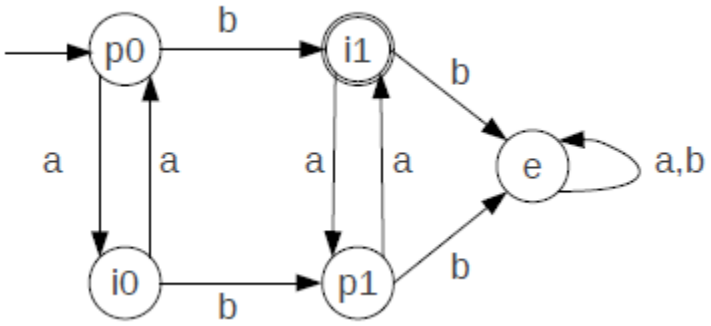
r*



q*r(s+tq*r)*

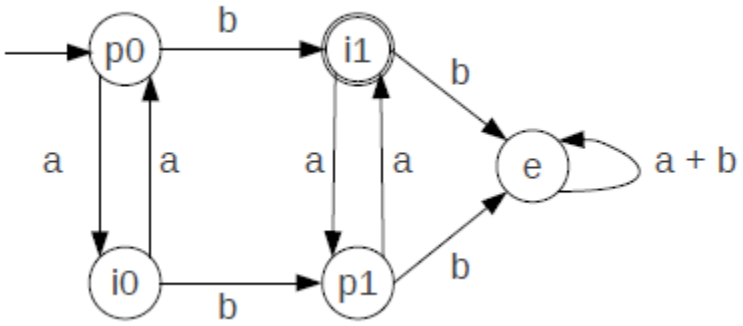
Exemplo 5

* Encontrar a ER para o seguinte AFD



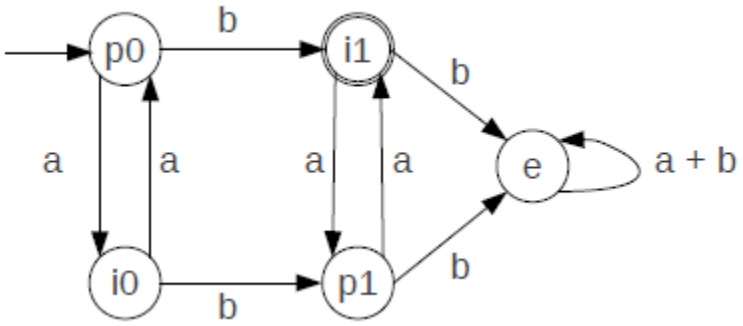
Exemplo 5

* Primeiro, transforma-se o AFD em diagrama ER



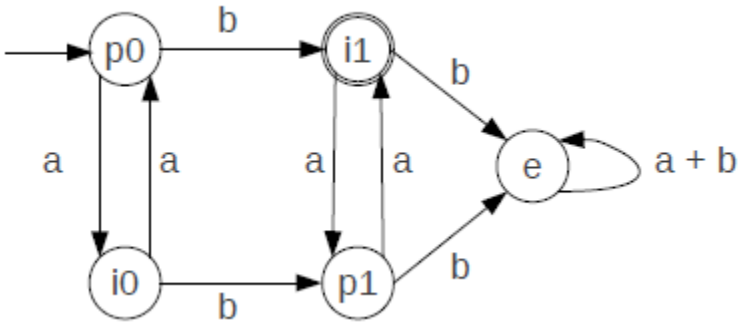
Exemplo 5

* Seleciona-se o primeiro estado final
  * Neste caso só existe i1



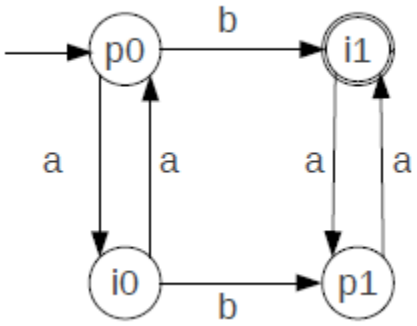
Exemplo 5

* Elimina-se os estados que não têm caminho que leva ao estado final



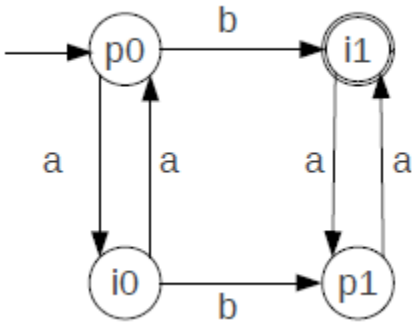
Exemplo 5

* Elimina-se os estados que não têm caminho que leva ao estado final



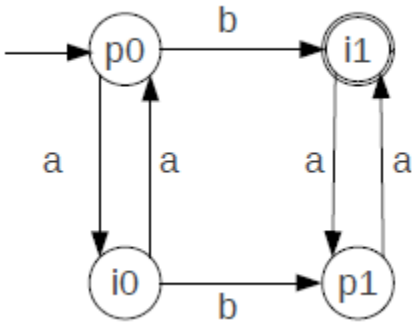
Exemplo 5

* Elimina-se os estados um a um, trocando-se as transições pela ER equivalente



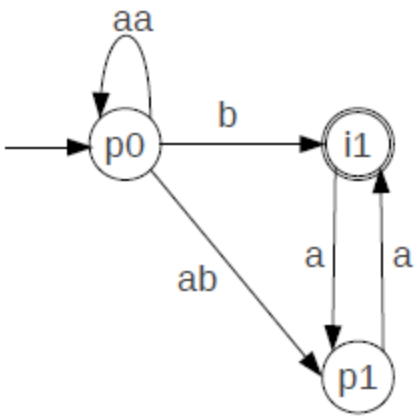
Exemplo 5

* Eliminando i0



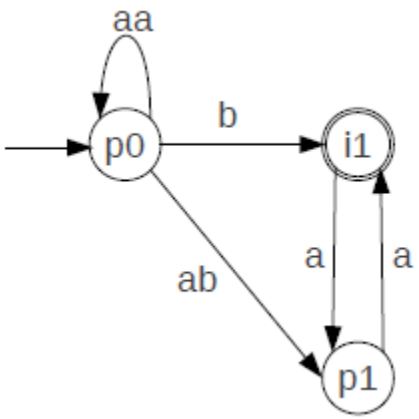
Exemplo 5

* Eliminando i0



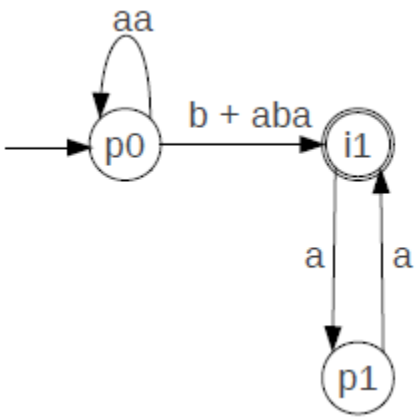
Exemplo 5

* Eliminando p1



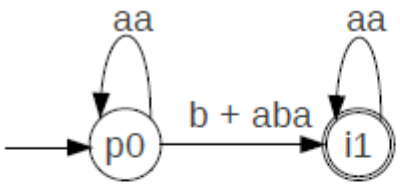
Exemplo 5

* Eliminando p1



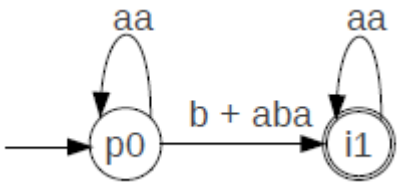
Exemplo 5

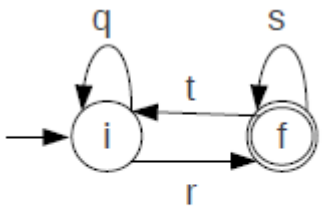
* Eliminando p1



Exemplo 5

* Chegou-se a um estado básico





q*r(s+tq*r)*

(aa)*(b+aba)(aa+Ø(aa)*(b+aba))* = (aa)*(b+aba)(aa)*

Colinha: Ø* = λ

Øa = Ø

Obrigado.

joaopauloaramuni@gmail.com
joaopauloaramuni@fumec.br

UNIVERSIDADE FUMEC
50 Anos