Linguagens Formais e Autômatos

Expressões Regulares

Felipe Cunha

Material extraído do livro e slides do Prof. Newton Vieira (http://dcc.ufmg.br/~nvieira) e do Prof. Andrei Rimsa

Expressões Regulares

- Até agora foram vistas duas formas de se especificar uma linguagem regular
  - Usando notação de conjuntos
    - **Aplicação:** desenvolvimento da teoria
  - Desenhando um diagrama de estados em forma de grafo
    - **Aplicação:** processo de concepção de um reconhecedor

- Uma nova forma de especificar linguagem regular é através de **expressões regulares**, mediante uma expressão que a denota
  - **Aplicação:** na teoria para manipulações formais e na prática para referência compacta a conjunto de palavras (editores de texto e comandos de sistema operacional)

Expressões Regulares

- Dado um AF M, seria possível obter uma formulação *linear* para $L(M)$? Ou seja, seria possível obter uma expressão r que denotasse $L(M)$?

Sim, através de uma expressão regular!

- Dada uma expressão regular, é possível criar um AF que reconhece a linguagem denotada por ela
  - Logo, a família das linguagens denotadas por expressões regulares é exatamente a família das linguagens regulares

Expressões Regulares

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

  a) $\varnothing$, λ e a para qualquer $a \in \Sigma$ são expressões regulares; tais ER's denotam, respectivamente, os conjuntos $\varnothing$, $\{\lambda\}$ e $\{a\}$

  b) Se r e s são expressões regulares, então são expressões regulares: $(r+s)$, (rs) e r*; tais ER's denotam, respectivament $L(r) \cup L(s)$, $L(r)L(s)$ e $L(r)^*$

Um conjunto que pode ser denotado por uma ER é usualmente denominado **conjunto regular**.

Exemplo de Expressões Regulares

- Exemplos de ERs sobre o alfabeto $\Sigma = \{0, 1\}$ e os conjuntos regulares denotados por elas

ER	Linguagem denotada
$\varnothing$	$\varnothing$
λ	$\{\lambda\}$
(01)	$\{0\}\{1\} = \{01\}$
$(0+1)$	$\{0\} \cup \{1\} = \{0,1\}$
$((0+1)(01))$	$\{0,1\}\{01\} = \{001,101\}$
0^*	$\{0\}^* = \{0^n \mid n \geq 0\}$
$(0+1)^*$	$\{0,1\}^* = \Sigma^*$
$(((0+1)^*1)(0+1))$	$\{0,1\}^*\{1\}\{0,1\}$

Omissão de Parênteses

- Uma ER com muitos parênteses torna díficil sua escrita e compreensão. Algumas regras para omissão de parênteses:
  a) Como a união é associativa, pode-se escrever $(r_1 + r_2 + \ldots + r_n)$, omitindo-se os parênteses internos
     - Ex.: ((0 + (11)) + 1) ou (0 + ((11) + 1)) → (0 + (11) + 1)
  b) Idem para concatenação, já que também é associativa
     - Ex.: ((01)((00)1)) → (01001)
  c) Se a expressão tem parênteses externos, esses podem ser omitidos
     - Ex.: *a)* → 0 + (11) + 1
       b) → 01001
  d) Precedência: Fecho de Klene > Concatenação > União
     - Ex. $(0 + (10^*)) \rightarrow 0 + 10^*$

Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{\mathrm{a}, \mathrm{b}\}$ com pelo menos um b?

Como obter uma expressão regular que denote uma linguagem L?

Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{a, b\}$ com pelo menos um b?

Primeira abordagem: tentar visualizar a linguagem como um todo, utilizando **não-determinismo** quando possível

- Ex.: uma palavra de L tem um b que é precedido de zero ou mais símbolos (inclusive de b's: não-determinismo) e seguido por zero ou mais símbolos

$$(a + b)^* b (a + b)^*$$

Um mesmo conjunto pode ser denotado por várias ER's?

Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{\mathrm{a}, \mathrm{b}\}$ com pelo menos um b?

  Segunda abordagem: visualizar como as palavras podem ser construídas deterministicamente, da esquerda para direita ou da direita para esquerda

  - Ex.: da esquerda para direita → pertence a L quando se encontra um b

$$a^*b(a+b)^*$$

  - Ex.: da direita para esquerda → pertence a L quando se encontra um b

$$(a+b)^*ba^*$$

Equivalência entre ER's

- Conforme visto, um conjunto pode ser denotado por várias expressões regulares

- A notação $r = s$ é comum para duas expressões regulares *r* e *s* que denotam uma mesma linguagem

$$L(r) = L(s)$$

- Às vezes pode ser interessante obter uma ER "mais simples", no sentido de visualizar com mais facilidade a linguagem por ela denotada
  - As propriedades de **união**, **concatenação** e **fecho de Kleene** permitem obter ER's equivalentes, eventualmente mais simples

Equivalência entre ER's

- Algumas equivalências entre ER's

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

Equivalência entre ER's

- Qualquer equivalência que não envolva fecho de Kleene pode ser derivada a partir das propriedades 1 a 7, mais as propriedades de associatividade de união e concatenação

- Quando envolve fecho de Kleene não há um conjunto finito de equivalência das quais se pode derivar qualquer outra

- Existem várias equivalências que são redundantes (podem ser obtidas através de outras)

$$
\begin{aligned}
\varnothing^* &= (r\varnothing)^* && \text{, por 5} \\
&= \lambda + r(\varnothing r)^*\varnothing && \text{, por 10} \\
&= \lambda + \varnothing && \text{, por 5} \\
&= \lambda && \text{, por 2}
\end{aligned}
$$

Simplificação de ER's

- Exemplo de uma série de simplificações de uma ER utilizando as equivalências da tabela, onde as sub-expressões simplificadas em cada passo estão sublinhadas

$$
\begin{aligned}
\underline{(00^* + 10^*)}0^*(1^* + 0)^* &= (0 + 1)\underline{0^*0^*}(1^* + 0)^* && \text{, por 6} \\
&= (0 + 1)0^*\underline{(1^* + 0)^*} && \text{, por 15} \\
&= (0 + 1)0^*\underline{(1 + 0)^*} && \text{, por 17} \\
&= (0 + 1)\underline{0^*(0 + 1)^*} && \text{, por 1} \\
&= (0 + 1)(0 + 1)^* && \text{, por 19}
\end{aligned}
$$

Simplificação de ER's

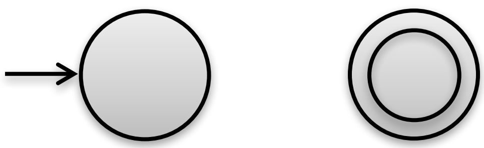
- Novas equivalências podem ser obtidas considerando as propriedades de:
  - **justaposição:** concatenação
  - **+:** união
  - ***:** fecho de Kleene

- Por exemplo:
  - $(r + rr + rrr + rrrr)^* = r^*$
  - $\underline{((0(0 + 1)1 + 11)0^*(00 + 11))^*}(0 + 1)^* = (0 + 1)^*$
    Como é da forma $\underline{r}(0+1)^*$, onde r denota um conjunto de palavras sobre o alfabeto que contém λ, então não precisa analisar r!
  - $r^*(r + s^*) = r^*s^*$

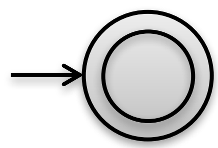
Linguagens Regulares

- Toda **expressão regular** denota uma **linguagem regular**

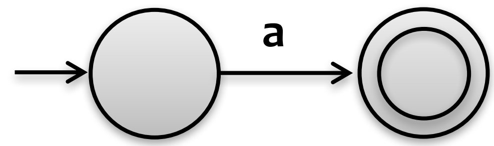
a) É possível construir AF's para os elementos primitivos dos conjuntos regulares: $\varnothing$, λ e para cada $a \in \Sigma$



AF para $\varnothing$



AF para $\{\lambda\}$

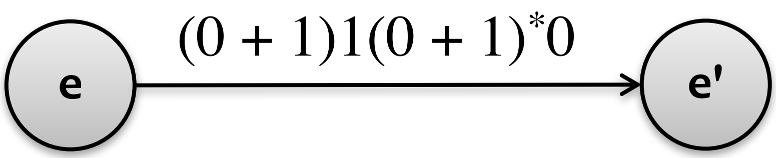


AF para $\{a\}$

b) Dados AF's para L_1 e L_2 é possível construir AF's para $L_1 \cup L_2$, L_1L_2, L_1^*

Diagrama ER

- Um **diagrama ER** sobre Σ é um diagrama de estados cujas arestas, ao invés de serem rotuladas com símbolos do alfabeto Σ, são rotuladas com ER's sobre Σ



  - Do estado e há transição para e' sob w, se e somente se, $w \in L((0 + 1)1(0 + 1)^{*}0)$, ou seja, o segundo símbolo de w é 1 e o último é 0
    - Ex.: As menores palavras são 010 e 110

Expressões Regulares

- Toda **linguagem regular** é denotada por alguma **expressão regular**

Seja um AFD $M = (E, \Sigma, \delta, i, \{f_1, f_2, \ ..., f_n\})$

- $L_1(M) \cup L_2(M) \cup \ldots \cup L_n(M)$, tal que
  $L_k(M) = \{w \in \Sigma^* \mid \hat{\delta}(i, w) = f_k\}$

- Seja p_k uma expressão regular para $L_k(M)$, então $L(M)$ seria denotada por $p_1 + p_2 + \ldots + p_k$

Obtenção de Expressão Regular

- Para obter uma expressão regular a partir de um AFN $M = (E, \Sigma, \delta, I, F)$

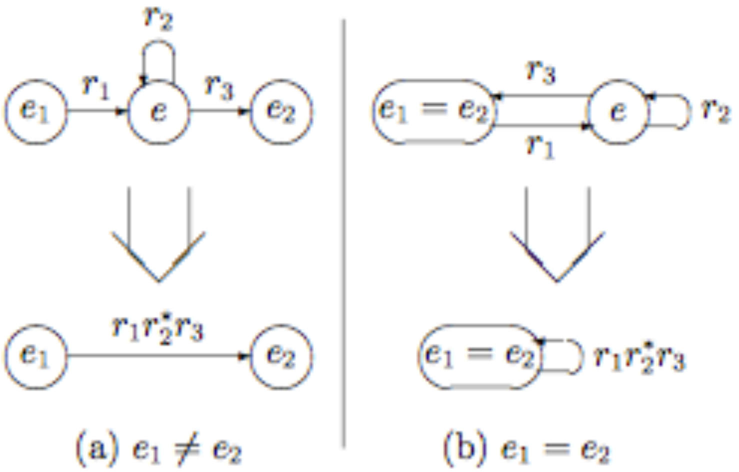
1. Obter AFNλ $M' = (E', \Sigma, \delta, i, \{f\})$ equivalente a *M* tal que:
   a) $i \notin \delta(e, a)$ para todo par $(e, a) \in E' \times \Sigma$
   b) $\delta(f, a) = \varnothing$ para todo $a \in \Sigma$

2. Obter diagrama ER a partir de M' substituindo transições de e para e' sob $s_1, s_2, \ldots, s_n$ por uma só transição de e para e' sob $s_1 + s_2 + \ldots + s_n$

Obtenção de Expressão Regular

- Para obter uma expressão regular a partir de um AFN $M = (E, \Sigma, \delta, I, F)$

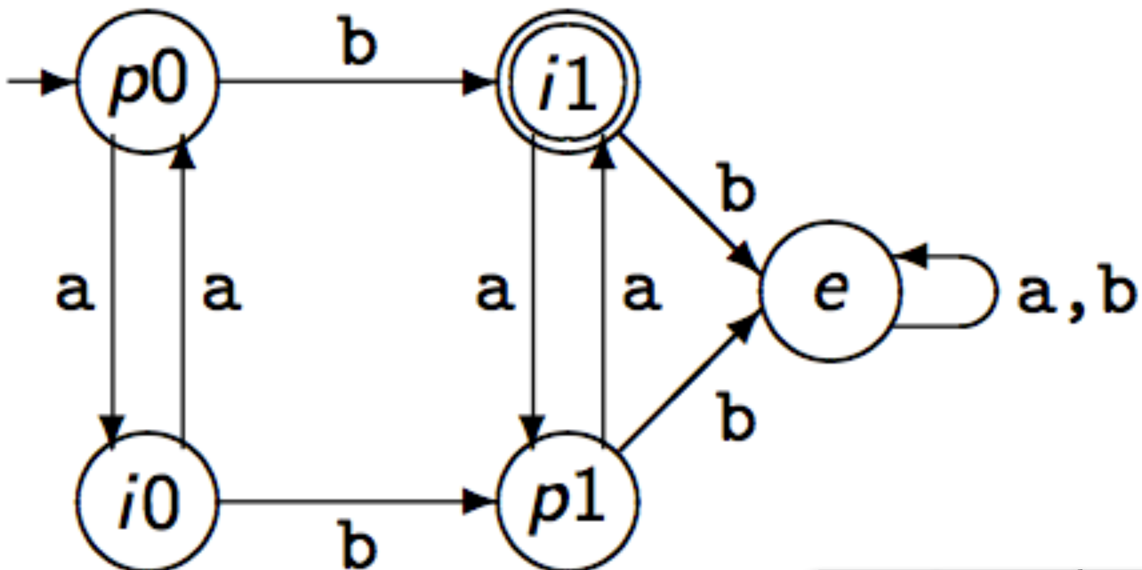
3. Eliminar um a um os estados do diagrama ER com exceção dos estados i e f_k
   - Simular todas passagens por e para cada par de estados $[e_1, e_2]$ tais que há transição de e_1 para e e de e para e_2 ($e_1 \neq e$ e $e_2 \neq e$)



   - Se havia transição de e_1 para e_2 sob s substituir por transição de e_1 para e_2 sob $\mathrm{s} + \mathrm{r}_1\mathrm{r}_2^*\mathrm{r}_3$

Exemplo

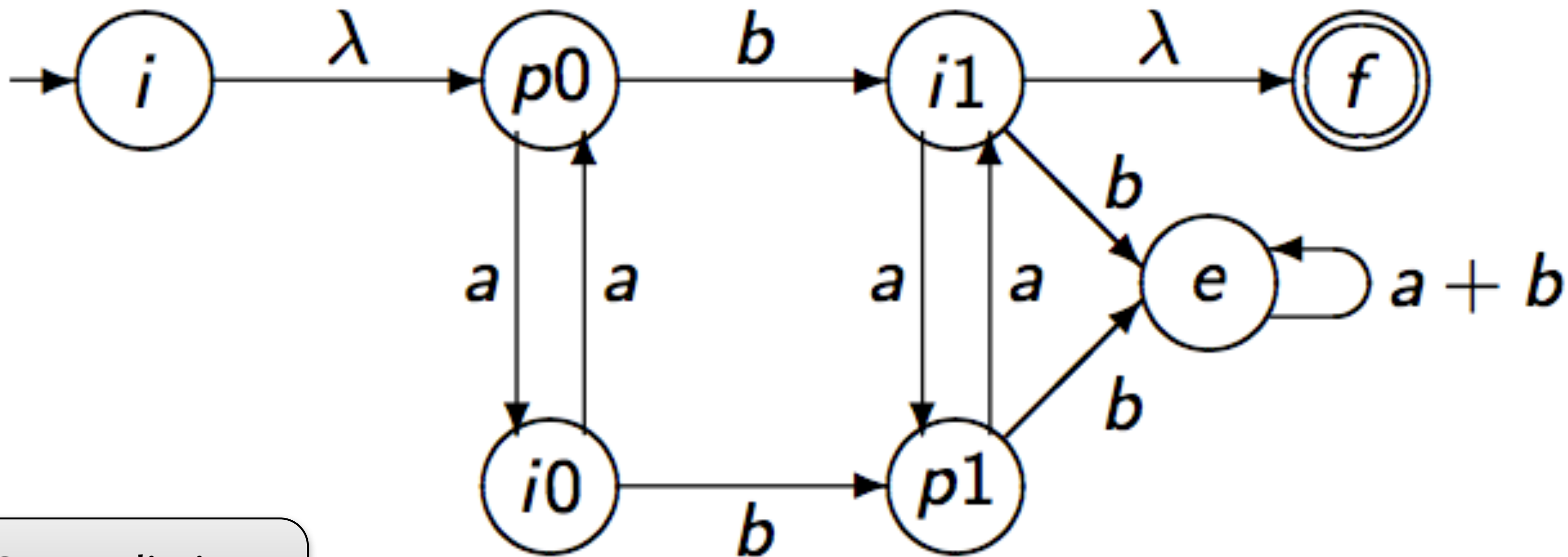
- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w|$ é ímpar e w contém exatamente um b$\}$



Como obter o diagrama ER com transições λ?

Exemplo

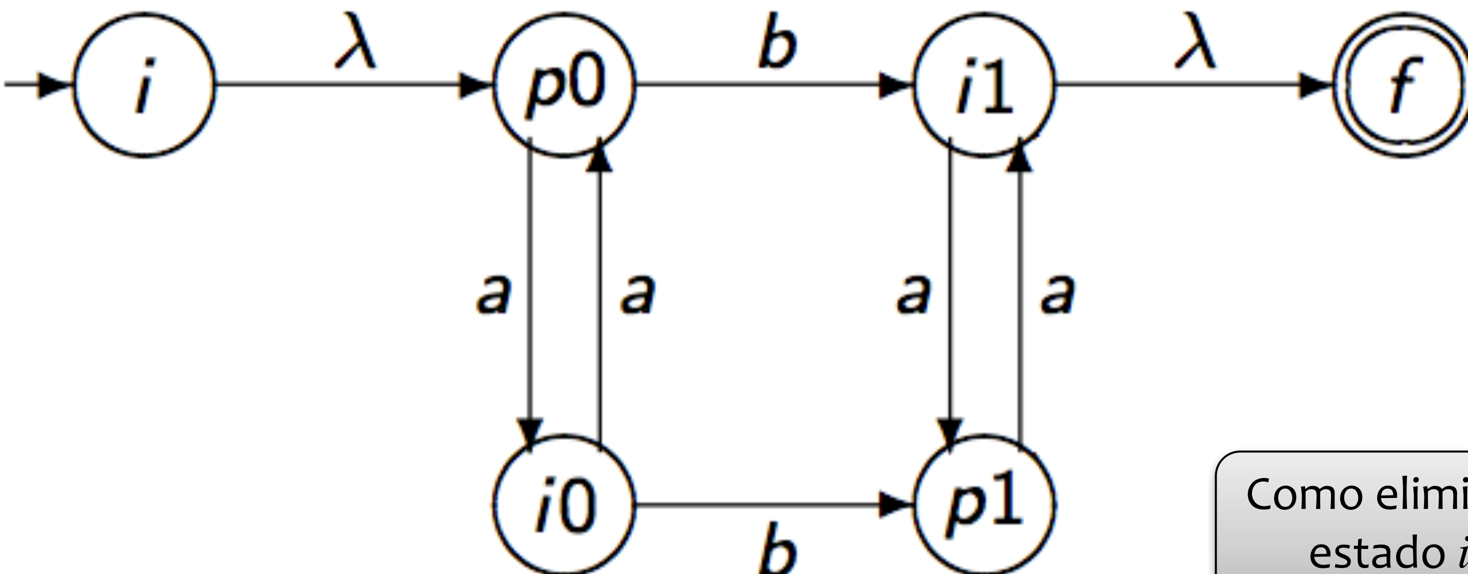
- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w|$ é ímpar e w contém exatamente um b$\}$



Como eliminar o estado *e*?

Exemplo

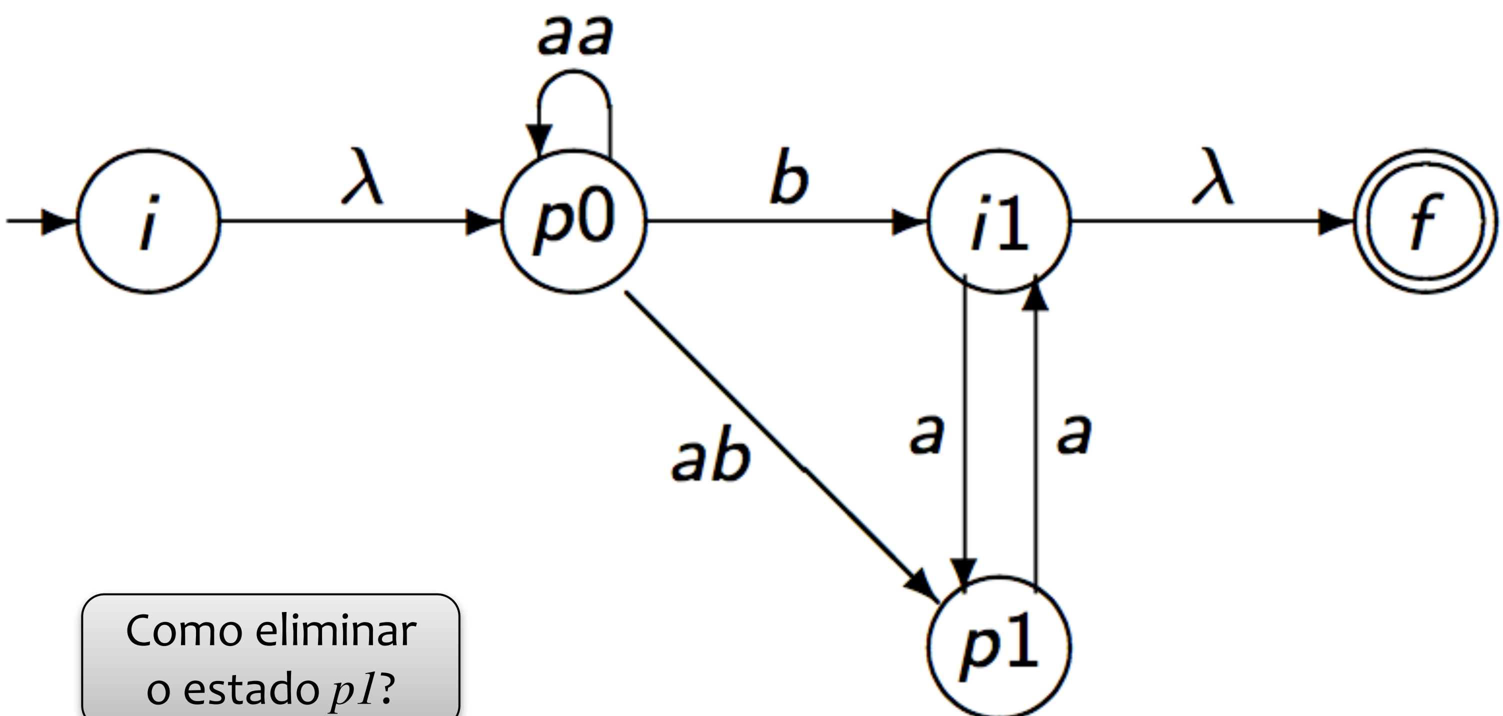
- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w|$ é ímpar e w contém exatamente um b$\}$



Como eliminar o estado *i0*?

Exemplo

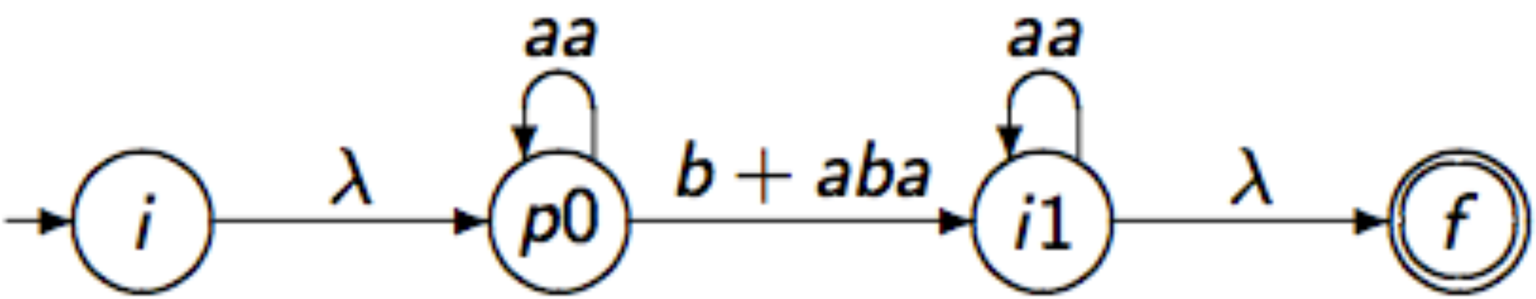
- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um b}\}$



Como eliminar
o estado *p1*?

Exemplo

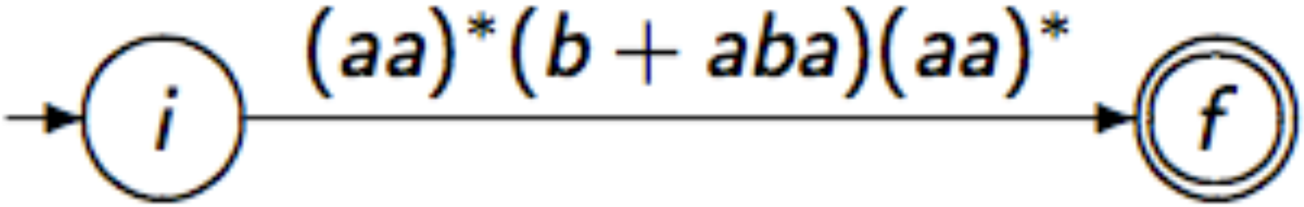
- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um b}\}$



Como eliminar os estados *p0* e *i1*?

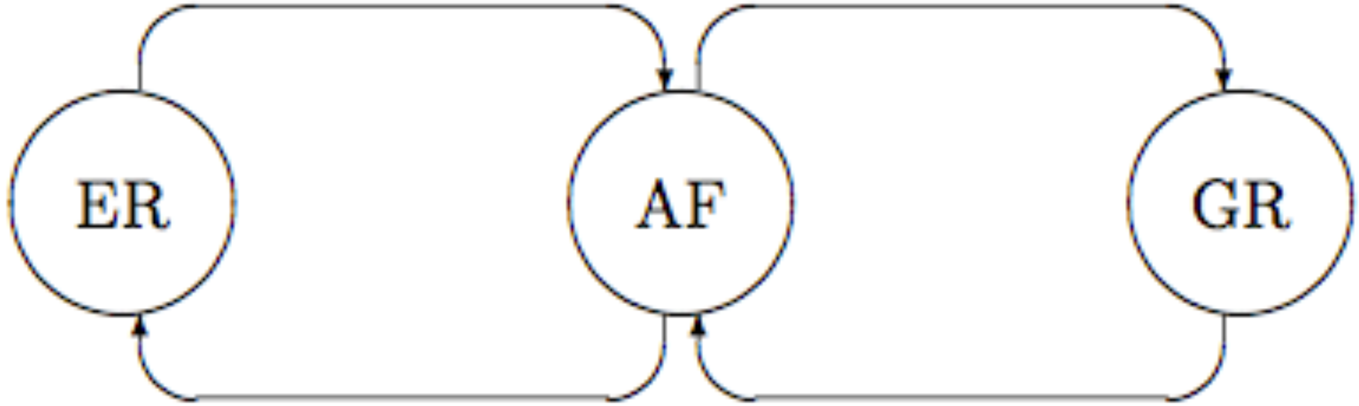
Exemplo

- Considere o AFD que reconhece a linguagem
  $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um b}\}$



Síntese

- AF's, ER's e GR's são formalismos alternativos para **linguagens regulares**



Transformações entre formalismos