CC1004 - Modelos de Computação
Teóricas 7 e 8

Ana Paula Tomás

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Março 2021

Conversões: ER $\to$ AF e AF $\to$ ER

Vamos apresentar métodos para:

- Dada uma expressão regular r, construir um AFND-ε que reconhece $\mathcal{L}(r)$.
  - **Construção de Thompson** (i.e., de McNaughton-Yamada-Thompson)
- Dado um autómato finito A, determinar uma expressão regular r tal que $\mathcal{L}(r) = \mathcal{L}(A)$, ou seja, tal que r descreve a linguagem que A reconhece.
  - Método de Kleene (muito trabalhoso; não iremos aplicar)
  - **Método de eliminação de estados de Brzozowski e McCluskey**

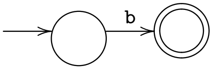
Construção de Thompson

Vamos ver como podemos construir um AFND-ε que reconhece a linguagem definida por uma expressão regular.

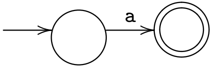
Por exemplo, para $(\mathtt{ba} + \mathtt{b})^\star$.

Exemplo: Construção de Thompson

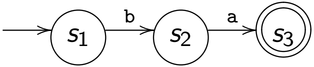
AFND-ε que reconhece a linguagem descrita por b



AFND-ε que reconhece a linguagem descrita por a



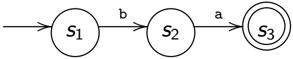
AFND-ε que reconhece a linguagem descrita por (ba)



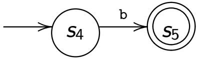
Para a **concatenação** (rs), identificamos o estado final do primeiro com o estado inicial do segundo.

Exemplo: Construção de Thompson

AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba})$



AFND-ε que reconhece a linguagem descrita por $\mathtt{b}$

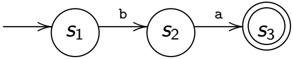


AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba} + \mathtt{b})$

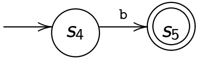
Para a **união** $(r + s)$, introduzimos um estado inicial s_6 e um estado final s_7, que será o único estado final da união, e acrescentamos transições por ε.

Exemplo: Construção de Thompson

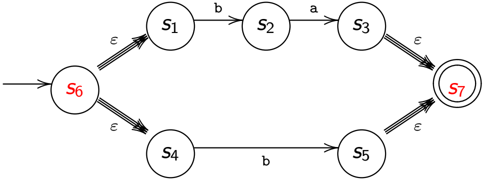
AFND-ε que reconhece a linguagem descrita por $(\texttt{ba})$



AFND-ε que reconhece a linguagem descrita por $\texttt{b}$



AFND-ε que reconhece a linguagem descrita por $(\texttt{ba} + \texttt{b})$



Para a **união** $(r + s)$, introduzimos um estado inicial s_6 e um estado final s_7, que será o único estado final da união, e acrescentamos transições por ε.

Exemplo: Construção de Thompson

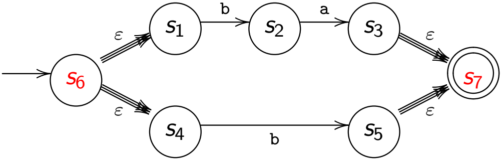
AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba} + \mathtt{b})$

AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba} + \mathtt{b})^\star$

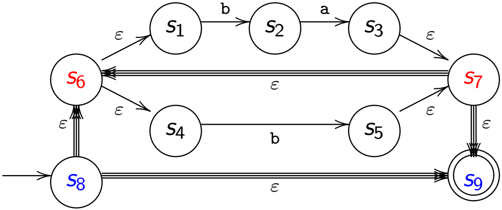
Para o **fecho de Kleene** $(r^\star)$, introduzimos um estado inicial s_8 e um estado final s_9, que será o único estado final do fecho, e acrescentamos transições por ε.

Exemplo: Construção de Thompson

AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba} + \mathtt{b})$



AFND-ε que reconhece a linguagem descrita por $(\mathtt{ba} + \mathtt{b})^\star$

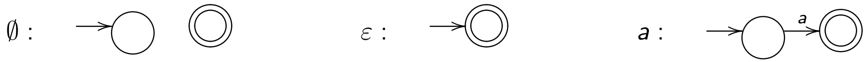


Para o **fecho de Kleene** $(r^\star)$, introduzimos um estado inicial s_8 e um estado final s_9, que será o único estado final do fecho, e acrescentamos transições por ε.

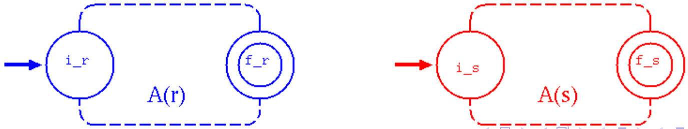
ER $\to$ AF: Construção de Thompson

Construção de Thompson (McNaughton-Yamada-Thompson)
Dada uma expressão regular r, obtém um AFND-ε, com um **único estado final**. Do estado final não saem transições. No estado inicial não entram transições.

Para as **expressões elementares** $\emptyset$, ε e a, com $a \in \Sigma$, o AFND-ε é:



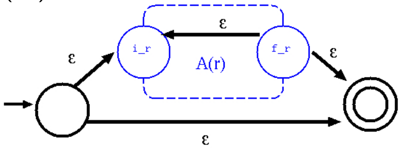
Para **as expressões com operadores** $(r^\star)$, (rs) e $(r+s)$, construimos os AFND-ε para as subexpressões r e s, por aplicação do método. Sejam esses autómatos $A(r)$ e $A(s)$ denotados esquematicamente por



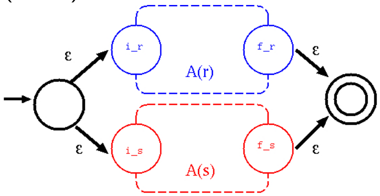
Conversão: Expressão Regular para AFND-ε

Construção de Thompson (McNaughton-Yamada-Thompson) para $(r^\star)$, (rs), e $(r+s)$

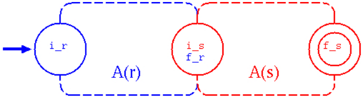
$(r^\star)$:



$(r+s)$:



(rs):

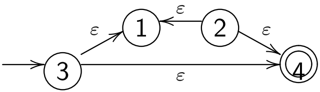


Para a concatenação, identifica o estado final de $A(r)$ com o inicial de $A(s)$

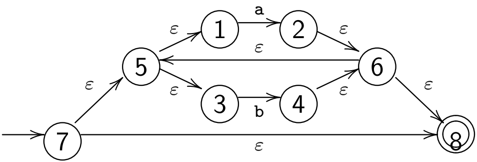
Um estado final. Não saem transições do estado final, nem entram no estado inicial.

Exemplos: Construção de Thompson

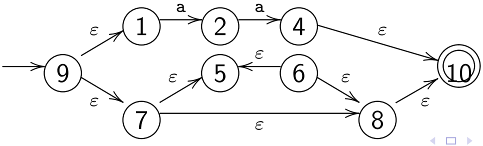
- $(\emptyset^\star)$



- $((\mathtt{a}+\mathtt{b})^\star)$



- $((\mathtt{aa})+(\emptyset^\star))$



Conversões: ER $\rightarrow$ AF e AF $\rightarrow$ ER

Conclusão (ER $\rightarrow$ AF)

As linguagens regulares (i.e., as linguagens que podem ser descritas por expressões regulares) podem ser reconhecidas por autómatos finitos.

AF $\rightarrow$ ER ? Vamos ver que:

As linguagens que podem ser reconhecidas por autómatos finitos são regulares.

- Dado um autómato finito A, os dois métodos seguintes determinam uma expressão regular r tal que $\mathcal{L}(r) = \mathcal{L}(A)$:
  - **Método de Kleene (1956)** (muito trabalhoso; não iremos usar)
  - Método de eliminação de estados de Brzozowski e McCluskey (1963)

Método de Kleene

Dado um autómato finito $A = (S, \Sigma, \delta, s_1, F)$, com estados numerados de 1 a n, seja $r_{ij}^{(k)}$ a expressão que descreve a linguagem determinada pelos percursos de i para j que passam quando muito por estados intermédios *etiquetados* com números não superiores a k.

$$r_{ii}^{(0)} = \begin{cases} \varepsilon & \textit{sse} \quad \text{não existe qualquer lacete em } i \\ \varepsilon + a_1 \ldots + a_p & \textit{sse} \quad \text{os lacetes em } i \text{ estão etiquetados com } a_1, \ldots, a_p \end{cases}$$

$$r_{ij}^{(0)} = \begin{cases} \emptyset & \textit{sse} \quad \text{não existe qualquer arco } (i,j) \\ a_1 + \ldots + a_p & \textit{sse} \quad a_1, \ldots, a_p \text{ etiquetam os arcos } (i,j) \end{cases}$$

Define-se agora $r_{ij}^{(k)}$, para $k \geq 1$, recursivamente assim:

$$r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}.$$

A linguagem $\mathcal{L}(A)$ é definida pela expressão $\sum_{s \in F} r_{1s}^{(n)}$.

Método de Kleene

Dado um autómato finito $A = (S, \Sigma, \delta, s_1, F)$, com estados numerados de 1 a n, seja $r_{ij}^{(k)}$ a expressão que descreve a linguagem determinada pelos percursos de i para j que passam quando muito por estados intermédios *etiquetados* com números não superiores a k.

$$r_{ii}^{(0)} = \begin{cases} \varepsilon & \textit{sse} \quad \text{não existe qualquer lacete em } i \\ \varepsilon + a_1 \ldots + a_p & \textit{sse} \quad \text{os lacetes em } i \text{ estão etiquetados com } a_1, \ldots, a_p \end{cases}$$

$$r_{ij}^{(0)} = \begin{cases} \emptyset & \textit{sse} \quad \text{não existe qualquer arco } (i,j) \\ a_1 + \ldots + a_p & \textit{sse} \quad a_1, \ldots, a_p \text{ etiquetam os arcos } (i,j) \end{cases}$$

Define-se agora $r_{ij}^{(k)}$, para $k \geq 1$, recursivamente assim:

$$r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^{\star} r_{kj}^{(k-1)}.$$

A linguagem $\mathcal{L}(A)$ é definida pela expressão $\sum_{s \in F} r_{1s}^{(n)}$.

Método de Kleene

Dado um autómato finito $A = (S, \Sigma, \delta, s_1, F)$, com estados numerados de 1 a n, seja $r_{ij}^{(k)}$ a expressão que descreve a linguagem determinada pelos percursos de i para j que passam quando muito por estados intermédios *etiquetados* com números não superiores a k.

$$r_{ii}^{(0)} = \begin{cases} \varepsilon & \textit{sse} \quad \text{não existe qualquer lacete em } i \\ \varepsilon + a_1 \ldots + a_p & \textit{sse} \quad \text{os lacetes em } i \text{ estão etiquetados com } a_1, \ldots, a_p \end{cases}$$

$$r_{ij}^{(0)} = \begin{cases} \emptyset & \textit{sse} \quad \text{não existe qualquer arco } (i, j) \\ a_1 + \ldots + a_p & \textit{sse} \quad a_1, \ldots, a_p \text{ etiquetam os arcos } (i, j) \end{cases}$$

Define-se agora $r_{ij}^{(k)}$, para $k \geq 1$, recursivamente assim:

$$r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)} (r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}.$$

A linguagem $\mathcal{L}(A)$ é definida pela expressão $\sum_{s \in F} r_{1s}^{(n)}$.

Método de Kleene

Dado um autómato finito $A = (S, \Sigma, \delta, s_1, F)$, com estados numerados de 1 a n, seja $r_{ij}^{(k)}$ a expressão que descreve a linguagem determinada pelos percursos de i para j que passam quando muito por estados intermédios *etiquetados* com números não superiores a k.

$$r_{ii}^{(0)} = \begin{cases} \varepsilon & \textit{sse} \quad \text{não existe qualquer lacete em } i \\ \varepsilon + a_1 \ldots + a_p & \textit{sse} \quad \text{os lacetes em } i \text{ estão etiquetados com } a_1, \ldots, a_p \end{cases}$$

$$r_{ij}^{(0)} = \begin{cases} \emptyset & \textit{sse} \quad \text{não existe qualquer arco } (i, j) \\ a_1 + \ldots + a_p & \textit{sse} \quad a_1, \ldots, a_p \text{ etiquetam os arcos } (i, j) \end{cases}$$

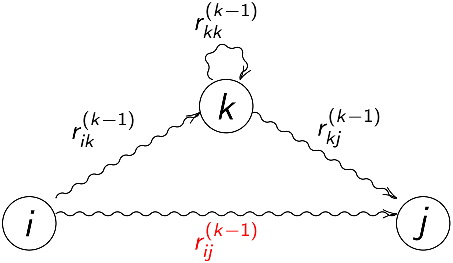
Define-se agora $r_{ij}^{(k)}$, para $k \geq 1$, recursivamente assim:

$$r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)} (r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}.$$

A linguagem $\mathcal{L}(A)$ é definida pela expressão $\sum_{s \in F} r_{1s}^{(n)}$.

Método de Kleene

Porquê $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$?

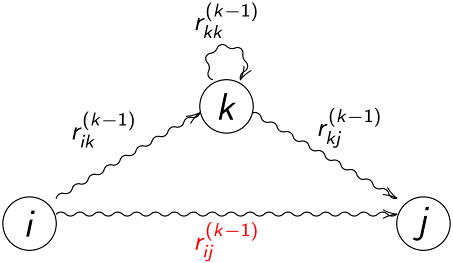


$r_{ij}^{(k)}$ define a linguagem determinada pelos percursos de i para j que só podem ter como **estados intermédios** os que têm números não superiores a k.

- Se não passar por k, a expressão é $r_{ij}^{(k-1)}$.
- Se passar por k, é uma concatenação de $r_{ik}^{(k-1)}$ com $(r_{kk}^{(k-1)})^\star$ e $r_{kj}^{(k-1)}$.

Método de Kleene

Porquê $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^{\star} r_{kj}^{(k-1)}$?



$r_{ij}^{(k)}$ define a linguagem determinada pelos percursos de i para j que só podem ter como **estados intermédios** os que têm números não superiores a k.

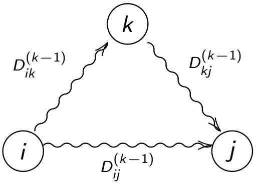
- Se não passar por k, a expressão é $r_{ij}^{(k-1)}$.
- Se passar por k, é uma concatenação de $r_{ik}^{(k-1)}$ com $(r_{kk}^{(k-1)})^{\star}$ e $r_{kj}^{(k-1)}$.

Método de Floyd-Warshall para Caminhos Mínimos

Um parentesis...

- O método de Kleene baseia-se numa técnica de concepção de algoritmos designada por **programação dinâmica**. A solução do problema é obtida à custa de soluções para subproblemas e pode ser calculada por fases, como no método de Kleene (em que cada fase corresponde a um valor de k). Cada subproblema é resolvido apenas uma vez e a sua solução é reutilizada sempre que for necessário.

- O **algoritmo de Floyd-Warshall**, que calcula **a distância mínima** $D_{ij}^{(n)}$ para todos os pares (i, j) em grafos com valores (distâncias) nos ramos, baseia-se numa ideia análoga. A distância é dada pela soma dos valores nos ramos do percurso. Como os **percursos mínimos não têm ciclos**, a recorrência fica:



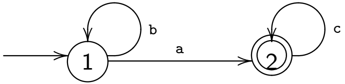
$$D_{ij}^{(k)} = \min(\ D_{ij}^{(k-1)},\ D_{ik}^{(k-1)} + D_{kj}^{(k-1)}\ )$$

$$D_{ii}^{(0)} = 0. \text{ Para } i \neq j,\ \ D_{ij}^{(0)} = \begin{cases} d_{ij} & \text{valor no ramo } (i,j) \\ \infty & \text{não tem ramo} \end{cases}$$

O algoritmo de Floyd-Warshall é estudado noutras UCs, por exemplo, CC2001 “Desenho e Análise de Algoritmos”.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso...



$r_{11}^{(0)} = \varepsilon + \mathtt{b}$ $\quad r_{22}^{(0)} = \varepsilon + \mathtt{c}$ $\quad r_{12}^{(0)} = \mathtt{a}$ $\quad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$, **para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

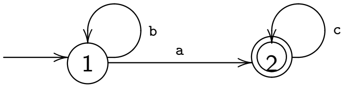
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star\mathtt{ac}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso...



$r_{11}^{(0)} = \varepsilon + \mathtt{b}$ $\quad r_{22}^{(0)} = \varepsilon + \mathtt{c}$ $\quad r_{12}^{(0)} = \mathtt{a}$ $\quad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$, **para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

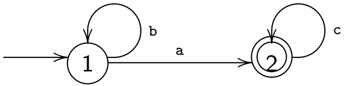
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star \mathtt{a}\mathtt{c}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso...



$r_{11}^{(0)} = \varepsilon + \mathtt{b} \qquad r_{22}^{(0)} = \varepsilon + \mathtt{c} \qquad r_{12}^{(0)} = \mathtt{a} \qquad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$, **para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

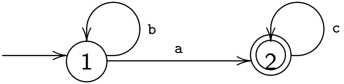
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star a c^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso...



$r_{11}^{(0)} = \varepsilon + \mathtt{b} \qquad r_{22}^{(0)} = \varepsilon + \mathtt{c} \qquad r_{12}^{(0)} = \mathtt{a} \qquad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$**, para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

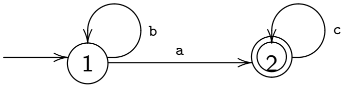
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star\mathtt{a} + \mathtt{b}^\star\mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star\mathtt{a}\mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star\mathtt{a}\mathtt{c}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso. . .



$r_{11}^{(0)} = \varepsilon + \mathtt{b} \qquad r_{22}^{(0)} = \varepsilon + \mathtt{c} \qquad r_{12}^{(0)} = \mathtt{a} \qquad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$, **para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

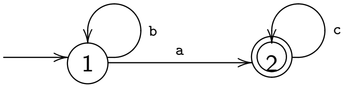
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star \mathtt{ac}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso...



$r_{11}^{(0)} = \varepsilon + \mathtt{b} \qquad r_{22}^{(0)} = \varepsilon + \mathtt{c} \qquad r_{12}^{(0)} = \mathtt{a} \qquad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$, **para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

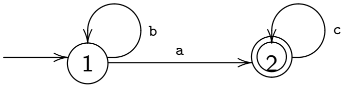
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star$.

Exemplo de Aplicação do Método de Kleene

Muito trabalhoso. . .



$r_{11}^{(0)} = \varepsilon + \mathtt{b} \qquad r_{22}^{(0)} = \varepsilon + \mathtt{c} \qquad r_{12}^{(0)} = \mathtt{a} \qquad r_{21}^{(0)} = \emptyset$

Recordar: $r_{ij}^{(k)} = r_{ij}^{(k-1)} + r_{ik}^{(k-1)}(r_{kk}^{(k-1)})^\star r_{kj}^{(k-1)}$**, para** $k \geq 1$

$$
\begin{aligned}
r_{11}^{(1)} &= r_{11}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \varepsilon + \mathtt{b} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star(\varepsilon + \mathtt{b}) = \mathtt{b}^\star \\
r_{22}^{(1)} &= r_{22}^{(0)} + r_{21}^{0}(r_{11}^{(0)})^\star r_{12}^{(0)} = \varepsilon + \mathtt{c} + \emptyset(\varepsilon + \mathtt{b})^\star \mathtt{a} = \varepsilon + \mathtt{c} \\
r_{12}^{(1)} &= r_{12}^{(0)} + r_{11}^{(0)}(r_{11}^{(0)})^\star r_{12}^{(0)} = \mathtt{a} + (\varepsilon + \mathtt{b})(\varepsilon + \mathtt{b})^\star \mathtt{a} = \mathtt{b}^\star \mathtt{a} \\
r_{21}^{(1)} &= r_{21}^{(0)} + r_{21}^{(0)}(r_{11}^{(0)})^\star r_{11}^{(0)} = \emptyset
\end{aligned}
$$

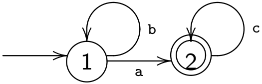
$$
\begin{aligned}
r_{11}^{(2)} &= r_{11}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{21}^{(1)} = r_{11}^{(1)} = \mathtt{b}^\star \\
r_{12}^{(2)} &= r_{12}^{(1)} + r_{12}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{b}^\star \mathtt{a} + \mathtt{b}^\star \mathtt{a}(\varepsilon + \mathtt{c})^\star(\varepsilon + \mathtt{c}) = \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star \\
r_{22}^{(2)} &= r_{22}^{(1)} + r_{22}^{(1)}(r_{22}^{(1)})^\star r_{22}^{(1)} = \mathtt{c}^\star \\
r_{21}^{(2)} &= \emptyset
\end{aligned}
$$

Conclusão: a expressão que descreve a linguagem aceite pelo AF é $r_{12}^{(2)}$ ou seja, $b^\star ac^\star$.

Se 1 e 2 fossem estados finais seria $r_{11}^{(2)} + r_{12}^{(2)} = \mathtt{b}^\star + \mathtt{b}^\star \mathtt{a} \mathtt{c}^\star$.

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

Obtém ER por eliminação de estados. Menos trabalhoso ...



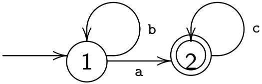
- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.

- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i, 2)$ com expressão $\varepsilon b^\star a$, ou seja $b^\star a$,

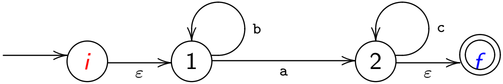
  - Estado 2: substituir percursos `i22*f` pelo ramo (i, f) com expressão $b^\star a c^\star \varepsilon \equiv b^\star a c^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

Obtém ER por eliminação de estados. Menos trabalhoso ...



- **Início:** Introduzir um estado inicial *i* e um estado final *f* para garantir que não saem transições de *f* nem chegam transições a *i*. Transformar os *labels* em ERs.

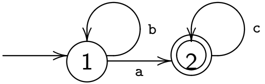


- **Eliminar os estados um a um, com excepção de *i* e *f*:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i, 2)$ com expressão $\varepsilon b^\star a$, ou seja $b^\star a$,

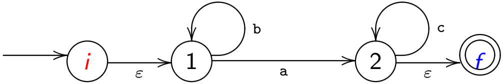
  - Estado 2: substituir percursos `i22*f` pelo ramo (i, f) com expressão $b^\star a c^\star \varepsilon \equiv b^\star a c^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

Obtém ER por eliminação de estados. Menos trabalhoso ...



- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.



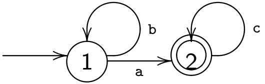
- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i, 2)$ com expressão $\varepsilon b^\star a$, ou seja

    $b^\star a$,

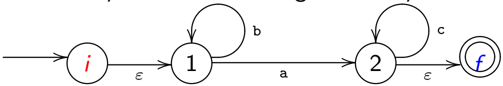
  - Estado 2: substituir percursos `i22*f` pelo ramo (i, f) com expressão $b^\star a c^\star \varepsilon \equiv b^\star a c^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

Obtém ER por eliminação de estados. Menos trabalhoso ...



- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.



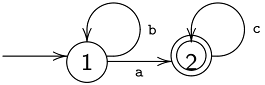
- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i,2)$ com expressão $\varepsilon \mathtt{b}^\star \mathtt{a}$, ou seja

    $\mathtt{b}^\star\mathtt{a}$,

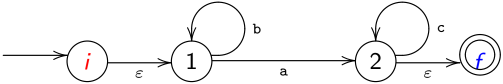
  - Estado 2: substituir percursos `i22*f` pelo ramo (i,f) com expressão $\mathtt{b}^\star\mathtt{ac}^\star\varepsilon \equiv \mathtt{b}^\star\mathtt{ac}^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

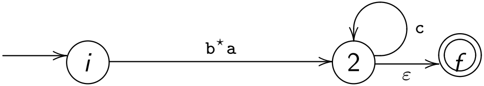
Obtém ER por eliminação de estados. Menos trabalhoso ...



- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.



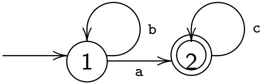
- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i, 2)$ com expressão $\varepsilon\mathtt{b}^\star\mathtt{a}$, ou seja $\mathtt{b}^\star\mathtt{a}$,



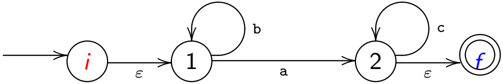
  - Estado 2: substituir percursos `i22*f` pelo ramo (i, f) com expressão $\mathtt{b}^\star\mathtt{a}\mathtt{c}^\star\varepsilon \equiv \mathtt{b}^\star\mathtt{a}\mathtt{c}^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

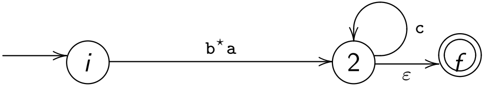
Obtém ER por eliminação de estados. Menos trabalhoso ...



- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.



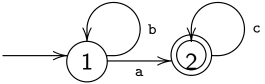
- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos $\mathtt{i11^{\star}2}$ pelo ramo $(i,2)$ com expressão $\varepsilon\mathtt{b}^\star\mathtt{a}$, ou seja $\mathtt{b}^\star\mathtt{a}$,



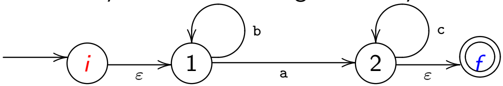
  - Estado 2: substituir percursos $\mathtt{i22^{\star}f}$ pelo ramo (i,f) com expressão $\mathtt{b}^\star\mathtt{ac}^\star\varepsilon \equiv \mathtt{b}^\star\mathtt{ac}^\star$. Fica

Ideia do Método de Eliminação de Estados de Brzozowski e McCluskey

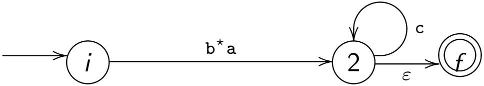
Obtém ER por eliminação de estados. Menos trabalhoso ...



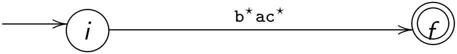
- **Início:** Introduzir um estado inicial i e um estado final f para garantir que não saem transições de f nem chegam transições a i. Transformar os *labels* em ERs.



- **Eliminar os estados um a um, com excepção de i e f:**
  - Estado 1: substituir percursos que passam no nó 1 por ramos. Neste caso, percursos `i11*2` pelo ramo $(i, 2)$ com expressão $\varepsilon\mathtt{b}^\star\mathtt{a}$, ou seja $\mathtt{b}^\star\mathtt{a}$,



  - Estado 2: substituir percursos `i22*f` pelo ramo (i, f) com expressão $\mathtt{b}^\star\mathtt{ac}^\star\varepsilon \equiv \mathtt{b}^\star\mathtt{ac}^\star$. Fica



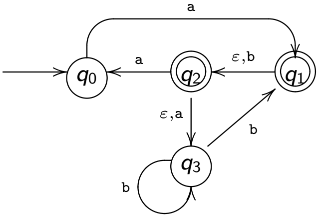
Eliminação de Estados de Brzozowski e McCluskey

Para $A = (S, \Sigma, \delta, s_0, F)$, o método de eliminação de estados obtém uma expressão regular para $\mathcal{L}(A)$. Partindo do diagrama de transição de A efetuamos:

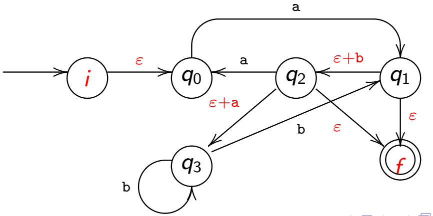
- **Primeiro passo:** Inserimos um estado inicial i e um novo estado final f, para garantir que não saem transições do estado final nem entram transições no estado inicial.
  - Definimos transições por ε: do novo estado inicial i para s_0; dos estados finais $s \in F$ para o estado f, que passa a ser o único final.
  - Substituimos os símbolos nos ramos por expressões regulares.
- **Procedemos à eliminação dos estados**, um a um, pela ordem que quisermos (se não for indicada uma ordem), mas mantemos i e f.
  - Para eliminar um estado s, devemos analisar **os ramos que entram** em s, os **ramos que saem** de s e, **o lacete** em s, se existir.
  - Para retirar s, **substituimos os percursos** (X, s, Y) que se formam com esses ramos por ramos (X, Y), com expressão $r_{Xs} r_{ss}^{\star} r_{sY}$, se tiver lacete (s, s). Se não, fica $r_{Xs} r_{sY}$. Se o ramo (X, Y) já existia, acrescenta-se a expressão nova usando $+$.

Exemplo: Método de eliminação de estados

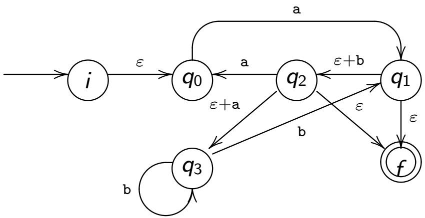
Vamos determinar uma expressão regular para a linguagem reconhecida pelo autómato seguinte:



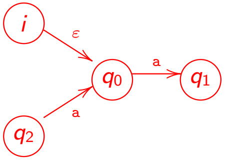
Primeiro passo: introduzir um estado inicial i e um estado final f.Não sairão transições de f nem chegam transições a i. Transformar os *labels* para expressões regulares.



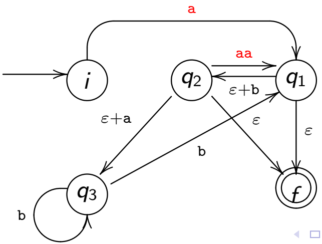
Exemplo: Método de eliminação de estados (cont.)



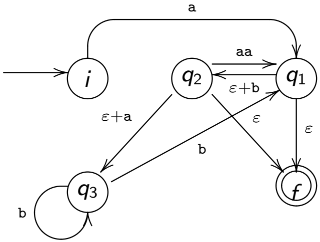
Para **eliminar** q_0, analisamos os ramos que entram em q_0 e os que saem de q_0



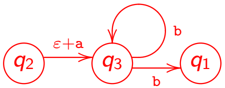
Para **eliminar** q_0, substituimos o percurso $q_2 \to^{a} q_0 \to^{a} q_1$ por um ramo (q_2, q_1) com expressão aa e substituimos $i \to^{\varepsilon} q_0 \to^{a} q_1$ por um ramo (i, q_1) com expressão εa $\equiv$ a.



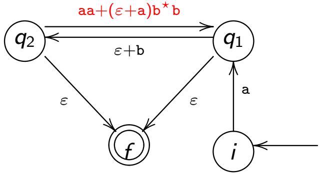
Exemplo: Método de eliminação de estados (cont.)



Para **eliminar** q_3, analisamos os ramos que entram em q_3 e os que saem de q_3

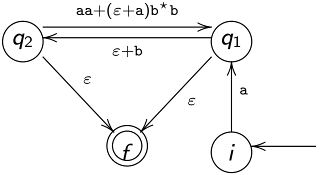


Substituimos percurso $q_2 q_3 q_3^\star q_1$ por um ramo (q_2, q_1) com expressão $(\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b}$. Como **já existia um ramo de q_2 para q_1**, substituimos a sua expressão aa por $\mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b}$.

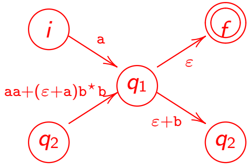


Deslocámos o estado *i* apenas para facilitar a compreensão do diagrama.

Exemplo: Método de eliminação de estados (cont.)



Para eliminar q_1, analisamos os ramos que entram e os que saem em q_1

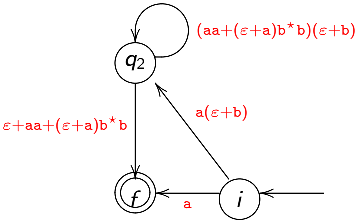


Para **eliminar** q_1, temos de substituir quatro "percursos":

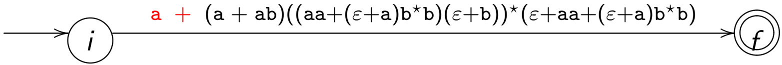
- iq_1f pelo ramo (i, f) com expressão $\mathtt{a}\varepsilon \equiv \mathtt{a}$
- iq_1q_2 pelo ramo (i, q_2) om expressão $\mathtt{a}(\varepsilon + \mathtt{b})$
- q_2q_1f pelo ramo (q_2, f) com expressão $(\mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b})\varepsilon \equiv \mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b}$
- $q_2q_1q_2$ pelo lacete (q_2, q_2) com expressão $(\mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b})(\varepsilon + \mathtt{b})$

Exemplo: Método de eliminação de estados (cont.)

Após eliminar q_1 fica:



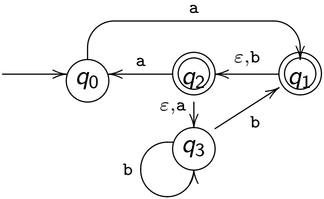
Para obter a expressão para $\mathcal{L}(A)$, resta eliminar q_2, substituindo percursos $iq_2q_2^\star f$ por um ramo (i,f). Como já existia um ramo (i,f), acrescenta-se a nova expressão.



Exemplo: Método de eliminação de estados (cont.)

Conclusão:

A expressão $\mathtt{a} + (\mathtt{a} + \mathtt{ab})((\mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b})(\varepsilon + \mathtt{b}))^\star(\varepsilon + \mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b})$ descreve a linguagem reconhecida pelo autómato



A expressão é equivalente a $\mathtt{a} + (\mathtt{a} + \mathtt{ab})(\mathtt{aa} + \mathtt{b} + \mathtt{ab})^\star$ porque

$$
\begin{array}{rcl}
(\ (\mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b})(\varepsilon + \mathtt{b})\ )^\star & \equiv & (\mathtt{aa} + \mathtt{bb}^\star + \mathtt{ab}^\star\mathtt{b} + \mathtt{aab} + \mathtt{bb}^\star\mathtt{b} + \mathtt{ab}^\star\mathtt{bb})^\star \\
& \equiv & (\mathtt{aa} + \mathtt{b} + \mathtt{ab}^\star\mathtt{b} + \mathtt{aab} + \mathtt{ab}^\star\mathtt{bb})^\star \\
& \equiv & (\mathtt{aa} + \mathtt{b} + \mathtt{ab})^\star \\
(\mathtt{aa} + \mathtt{b} + \mathtt{ab})^\star(\varepsilon + \mathtt{aa} + (\varepsilon + \mathtt{a})\mathtt{b}^\star\mathtt{b}) & \equiv & (\mathtt{aa} + \mathtt{b} + \mathtt{ab})^\star(\varepsilon + \mathtt{aa} + \mathtt{b}^\star\mathtt{b} + \mathtt{ab}^\star\mathtt{b}) \\
& \equiv & (\mathtt{aa} + \mathtt{b} + \mathtt{ab})^\star
\end{array}
$$