Teoria da Computação

Aula Teórica 16:
Exemplos de conversão de AFDs em Expressões Regulares.

António Ravara

Departamento de Informática

8 de Maio de 2019

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ. A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA):
   considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

A ideia

1. Transforma um NFA num GNFA (Generalized NFA): considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

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 $\delta(i, ef) = k$

Método da remoção de estados e transições

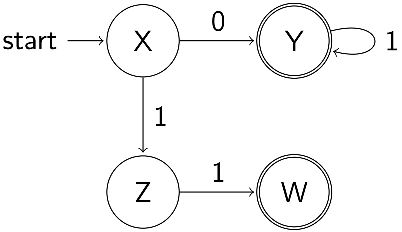
A ideia

1. Transforma um NFA num GNFA (Generalized NFA):
   considera-se Σ^* em vez de Σ.
   A etiqueta de uma transição passa a ser uma expressão.
2. O autómato obtido:
   2.1 tem no máximo 2 estados (um inicial e um final);
   2.2 a etiqueta da única transição é a expressao regular que denota a linguagem do autómato original.

Remoção iterativa de estados e transições

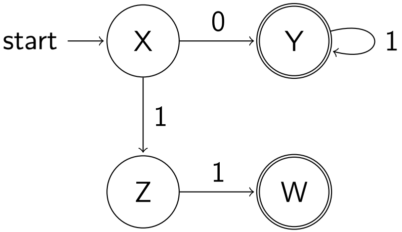
1. Ligam-se os estados finais por transições ϵ.
2. Se $\delta(i, e) = i$ então $\delta(i, e^*) = i$.
3. Se $\delta(i, e) = j$ e $\delta(i, f) = j$ então $\delta(i, e + f) = j$.
4. Se $\delta(i, e) = j$ e $\delta(j, f) = k$ então $\delta(i, ef) = k$

Aplicação do método: passo 1



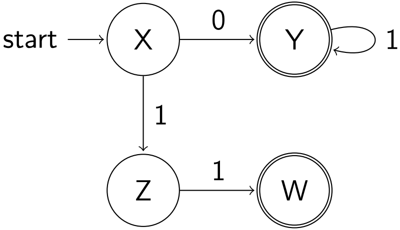
Apaga-se o estado Z, concatenando as etiquetas.

Aplicação do método: passo 1

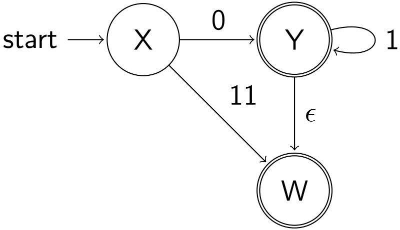


Apaga-se o estado Z, concatenando as etiquetas.

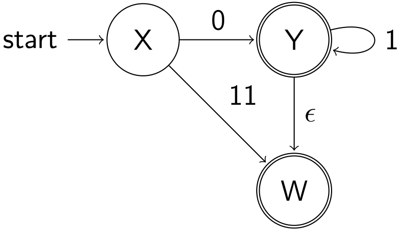
Aplicação do método: passo 1



Apaga-se o estado Z, concatenando as etiquetas.

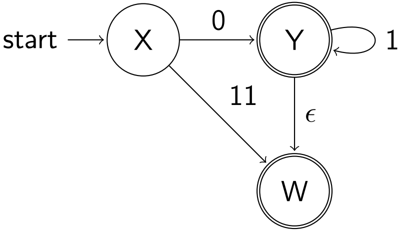


Aplicação do método: passo 2



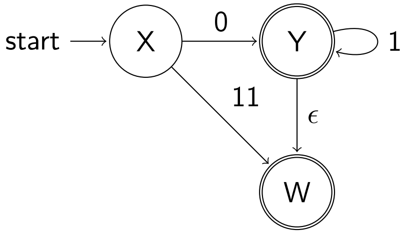
Fecho-* das transições re-entrantes.

Aplicação do método: passo 2

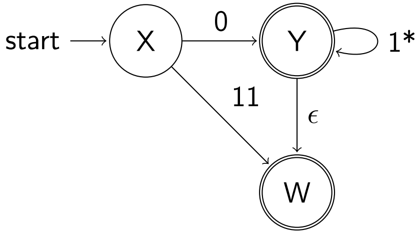


Fecho-* das transições re-entrantes.

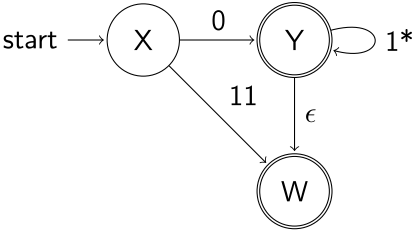
Aplicação do método: passo 2



Fecho-* das transições re-entrantes.

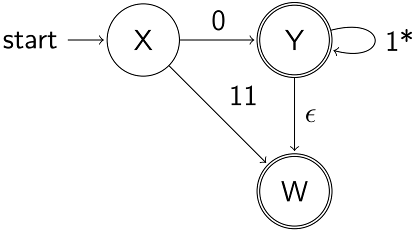


Aplicação do método: passo 3



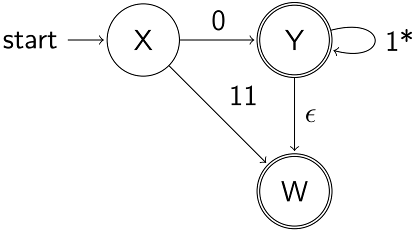
Concatenação das expressões de X para Y.

Aplicação do método: passo 3

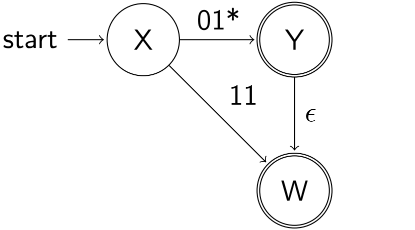


Concatenação das expressões de X para Y.

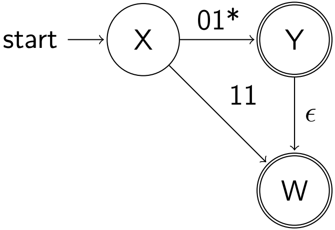
Aplicação do método: passo 3



Concatenação das expressões de X para Y.



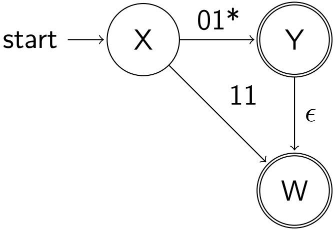
Aplicação do método: passo 4



Eliminação do estado Y.

Expressão regular obtida: 11+01*

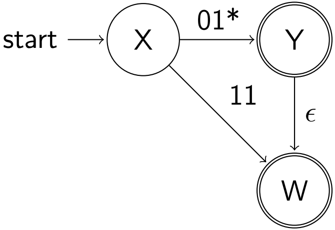
Aplicação do método: passo 4



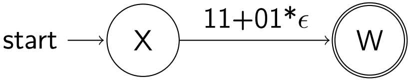
Eliminação do estado Y.

Expressão regular obtida: 11+01*

Aplicação do método: passo 4



Eliminação do estado Y.



Expressão regular obtida: 11+01*

Método das equações lineares

A ideia

1. Obtém-se uma equação de cada transição:
   Se $\delta(X, a) = Y$ então $X = aY$.
2. Obtém-se uma equação de cada estado final:
   Se $X \in F$ então $X = \epsilon$.
3. O sistema obtido é resolvido em ordem ao estado inicial.
   A expressão denota todas as palavras aceites pelo autómato (levam do estado inicial a um final).
4. Resolvem-se as equações recursivas pelo Lema de Arden
   $X = eX + f \Leftrightarrow X = e^*f$

Método das equações lineares

A ideia

1. Obtém-se uma equação de cada transição:
   Se $\delta(X, a) = Y$ então $X = aY$.
2. Obtém-se uma equação de cada estado final:
   Se $X \in F$ então $X = \epsilon$.
3. O sistema obtido é resolvido em ordem ao estado inicial.
   A expressão denota todas as palavras aceites pelo autómato (levam do estado inicial a um final).
4. Resolvem-se as equações recursivas pelo Lema de Arden
   $X = eX + f \Leftrightarrow X = e^*f$

Método das equações lineares

A ideia

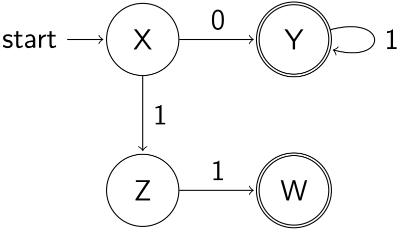
1. Obtém-se uma equação de cada transição:
   Se $\delta(X, a) = Y$ então $X = aY$.
2. Obtém-se uma equação de cada estado final:
   Se $X \in F$ então $X = \epsilon$.
3. O sistema obtido é resolvido em ordem ao estado inicial.
   A expressão denota todas as palavras aceites pelo autómato (levam do estado inicial a um final).
4. Resolvem-se as equações recursivas pelo Lema de Arden
   $X = eX + f \Leftrightarrow X = e^*f$

Método das equações lineares

A ideia

1. Obtém-se uma equação de cada transição:
   Se $\delta(X, a) = Y$ então $X = aY$.
2. Obtém-se uma equação de cada estado final:
   Se $X \in F$ então $X = \epsilon$.
3. O sistema obtido é resolvido em ordem ao estado inicial.
   A expressão denota todas as palavras aceites pelo autómato (levam do estado inicial a um final).
4. Resolvem-se as equações recursivas pelo Lema de Arden
   $X = eX + f \Leftrightarrow X = e^*f$

Método da remoção de estados e transições



Gera as equações:

1. $X = 0\,Y$
2. $X = 1\,Z$
3. $Y = 1\,Y$
4. $Y = \epsilon$
5. $Z = 1\,W$
6. $W = \epsilon$

Método da remoção de estados e transições

Resolução do sistema em ordem ao estado inicial

(ler de baixo para cima)

1. $X = 0\,Y$
2. $X = 1\,Z \;\Leftrightarrow\; X = 1\,1 \;\Leftrightarrow\; X = 0\,Y + 1\,1 \;\Leftrightarrow\; X = 0\,1^* + 1\,1$
3. $Y = 1\,Y \;\Leftrightarrow\; Y = 1\,Y + \epsilon \;\Leftrightarrow\; Y = 1^*\,\epsilon \;\Leftrightarrow\; Y = 1^*$
4. $Y = \epsilon$
5. $Z = 1\,W \;\Leftrightarrow\; Z = 1\epsilon \;\Leftrightarrow\; Z = 1$
6. $W = \epsilon$