

Análise léxica

Autômatos finitos

Prof. Edson Alves

Faculdade UnB Gama

Sumário

Reconhecedores

- ▶ Um reconhecedor é um programa que identifica, respondendo "Sim" ou "Nao", se uma cadeia é ou não uma sentença válida de uma determinada linguagem

Reconhecedores

- ▶ Um reconhecedor é um programa que identifica, respondendo "Sim" ou "Nao", se uma cadeia é ou não uma sentença válida de uma determinada linguagem
- ▶ Uma estratégia para a compilação de expressões regulares em reconhecedores é o uso de diagramas de transição generalizados, denominados autômatos finitos

Reconhecedores

- ▶ Um reconhecedor é um programa que identifica, respondendo "Sim" ou "Nao", se uma cadeia é ou não uma sentença válida de uma determinada linguagem
- ▶ Uma estratégia para a compilação de expressões regulares em reconhecedores é o uso de diagramas de transição generalizados, denominados autômatos finitos
- ▶ Um autômato finito pode ser determinístico ou não-determinístico: no segundo caso, podem existir duas ou mais transições com o mesmo rótulo partindo de um mesmo estado

Reconhecedores

- ▶ Um reconhecedor é um programa que identifica, respondendo "Sim" ou "Nao", se uma cadeia é ou não uma sentença válida de uma determinada linguagem
- ▶ Uma estratégia para a compilação de expressões regulares em reconhecedores é o uso de diagramas de transição generalizados, denominados autômatos finitos
- ▶ Um autômato finito pode ser determinístico ou não-determinístico: no segundo caso, podem existir duas ou mais transições com o mesmo rótulo partindo de um mesmo estado
- ▶ Autômatos determinísticos podem resultar em reconhecimentos mais rápidos do que os não-determinísticos, porém em geral são muito maiores, no que diz respeito ao número de estados e transições

Reconhecedores

- ▶ Um reconhecedor é um programa que identifica, respondendo "Sim" ou "Nao", se uma cadeia é ou não uma sentença válida de uma determinada linguagem
- ▶ Uma estratégia para a compilação de expressões regulares em reconhecedores é o uso de diagramas de transição generalizados, denominados autômatos finitos
- ▶ Um autômato finito pode ser determinístico ou não-determinístico: no segundo caso, podem existir duas ou mais transições com o mesmo rótulo partindo de um mesmo estado
- ▶ Autômatos determinísticos podem resultar em reconhecimentos mais rápidos do que os não-determinísticos, porém em geral são muito maiores, no que diz respeito ao número de estados e transições
- ▶ É possível representar expressões regulares em ambos tipos de autômatos finitos

Autômatos finitos não-determinísticos

Definição de AFN

Um autômato finito não-determinístico (AFN) é um modelo matemático que consiste em

1. um conjunto de estados S ,
2. um alfabeto Σ de símbolos de entrada,
3. uma função de transição que mapeia pares (estado, símbolo) em um conjunto de estados,
4. um estado s_0 , denominado estado inicial ou de partida, e
5. um conjunto F de estados de aceitação (ou estados finais).

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição
- ▶ Em um grafo de transição, os nós representam os estados e as arestas definem a função de transição

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição
- ▶ Em um grafo de transição, os nós representam os estados e as arestas definem a função de transição
- ▶ Os rótulos das arestas são os símbolos associados à transição, e a direção da aresta parte do estado atual para o próximo estado

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição
- ▶ Em um grafo de transição, os nós representam os estados e as arestas definem a função de transição
- ▶ Os rótulos das arestas são os símbolos associados à transição, e a direção da aresta parte do estado atual para o próximo estado
- ▶ Grafos de transição se assemelham aos diagramas de transição, com duas diferenças fundamentais

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição
- ▶ Em um grafo de transição, os nós representam os estados e as arestas definem a função de transição
- ▶ Os rótulos das arestas são os símbolos associados à transição, e a direção da aresta parte do estado atual para o próximo estado
- ▶ Grafos de transição se assemelham aos diagramas de transição, com duas diferenças fundamentais
- ▶ A primeira diferença é que um mesmo rótulo pode estar associado a duas ou mais arestas partindo de um mesmo estado

Grafo de transições

- ▶ Um AFN pode ser representado por meio de um grafo direcionado e rotulado, denominado grafo de transição
- ▶ Em um grafo de transição, os nós representam os estados e as arestas definem a função de transição
- ▶ Os rótulos das arestas são os símbolos associados à transição, e a direção da aresta parte do estado atual para o próximo estado
- ▶ Grafos de transição se assemelham aos diagramas de transição, com duas diferenças fundamentais
- ▶ A primeira diferença é que um mesmo rótulo pode estar associado a duas ou mais arestas partindo de um mesmo estado
- ▶ A segunda é que o símbolo ϵ pode rotular uma aresta

Grafo de transição para a linguagem $(a \mid b)^*abb$

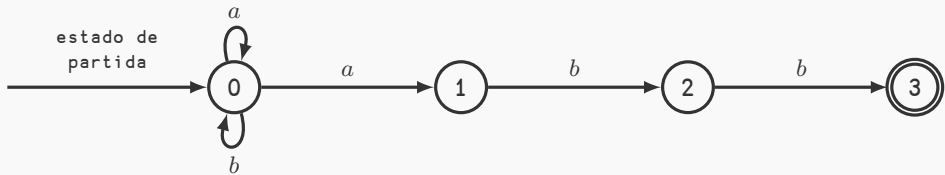


Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições

Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições
- ▶ Em uma tabela de transições cada linha representa um estado e cada coluna representa um rótulo

Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições
- ▶ Em uma tabela de transições cada linha representa um estado e cada coluna representa um rótulo
- ▶ Se necessário, é necessário adicionar uma coluna para o rótulo ϵ

Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições
- ▶ Em uma tabela de transições cada linha representa um estado e cada coluna representa um rótulo
- ▶ Se necessário, é necessário adicionar uma coluna para o rótulo ϵ
- ▶ A entrada da tabela posicionada na linha i , coluna c , contém o conjunto de estados que podem suceder o estado i quando o caractere c for lido na entrada

Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições
- ▶ Em uma tabela de transições cada linha representa um estado e cada coluna representa um rótulo
- ▶ Se necessário, é necessário adicionar uma coluna para o rótulo ϵ
- ▶ A entrada da tabela posicionada na linha i , coluna c , contém o conjunto de estados que podem suceder o estado i quando o caractere c for lido na entrada
- ▶ De fato, a tabela de transições corresponde à representação do grafo de transições como uma matriz de adjacências

Tabela de transições

- ▶ Uma alternativa para a implementação da função de transição é a tabela de transições
- ▶ Em uma tabela de transições cada linha representa um estado e cada coluna representa um rótulo
- ▶ Se necessário, é necessário adicionar uma coluna para o rótulo ϵ
- ▶ A entrada da tabela posicionada na linha i , coluna c , contém o conjunto de estados que podem suceder o estado i quando o caractere c for lido na entrada
- ▶ De fato, a tabela de transições corresponde à representação do grafo de transições como uma matriz de adjacências
- ▶ Outra alternativa é representar o grafo por meio de uma lista de adjacências

Tabela de transições do AFN da linguagem $(a \mid b)^*abb$

| Estado | Símbolo de entrada | |
|--------|--------------------|-------|
| | a | b |
| 0 | { 0, 1 } | { 0 } |
| 1 | - | { 2 } |
| 2 | - | { 3 } |

- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição

Caminhos

- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição
- ▶ Os rótulos das arestas, quando concatenados, formam uma cadeia s

Caminhos

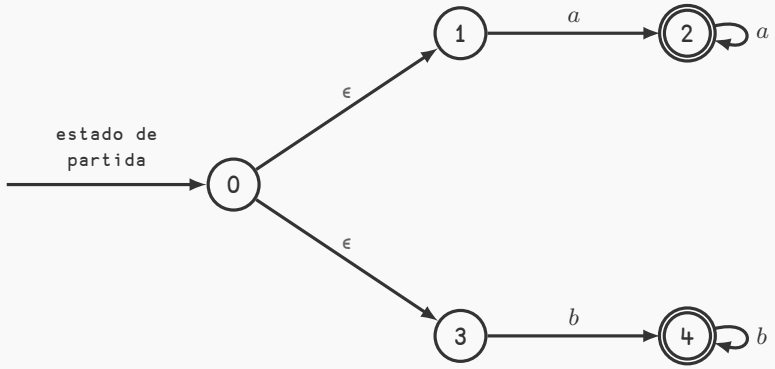
- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição
- ▶ Os rótulos das arestas, quando concatenados, formam uma cadeia s
- ▶ Caso o símbolo ϵ seja o rótulo de uma ou mais arestas de um caminho, na concatenação dos rótulos este símbolo é descartado

- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição
- ▶ Os rótulos das arestas, quando concatenados, formam uma cadeia s
- ▶ Caso o símbolo ϵ seja o rótulo de uma ou mais arestas de um caminho, na concatenação dos rótulos este símbolo é descartado
- ▶ Um AFN aceita uma cadeia de entrada s se, e somente se, existe um caminho no grafo de transições que parte do estado inicial e que termina em algum estado de aceitação

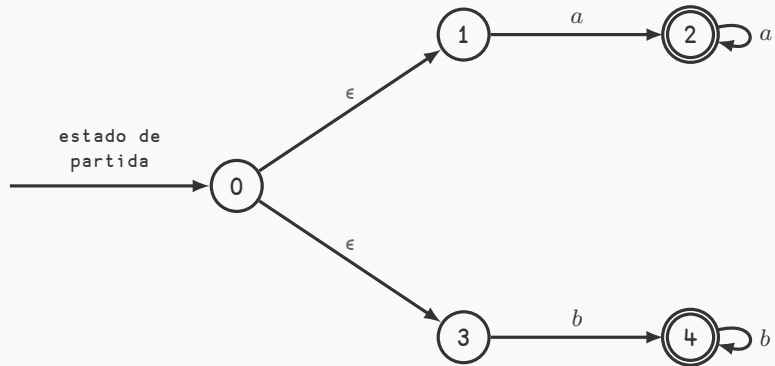
- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição
- ▶ Os rótulos das arestas, quando concatenados, formam uma cadeia s
- ▶ Caso o símbolo ϵ seja o rótulo de uma ou mais arestas de um caminho, na concatenação dos rótulos este símbolo é descartado
- ▶ Um AFN aceita uma cadeia de entrada s se, e somente se, existe um caminho no grafo de transições que parte do estado inicial e que termina em algum estado de aceitação
- ▶ Pode existir mais de um caminho que leva a um estado de aceitação

- ▶ Um caminho em um grafo de transições é uma sequência de arestas de transição
- ▶ Os rótulos das arestas, quando concatenados, formam uma cadeia s
- ▶ Caso o símbolo ϵ seja o rótulo de uma ou mais arestas de um caminho, na concatenação dos rótulos este símbolo é descartado
- ▶ Um AFN aceita uma cadeia de entrada s se, e somente se, existe um caminho no grafo de transições que parte do estado inicial e que termina em algum estado de aceitação
- ▶ Pode existir mais de um caminho que leva a um estado de aceitação
- ▶ A linguagem definida por um AFN é o conjunto de cadeias que são aceitas

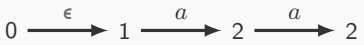
AFN da linguagem $aa^* \mid bb^*$



AFN da linguagem $aa^* \mid bb^*$



Caminho que aceita
a cadeia aa



Autômatos finitos determinísticos

Definição de AFD

Um autômato finito determinístico (AFD) é um caso especial de AFN no qual

1. nenhum estado possui uma transição rotulada pelo símbolo ϵ (denominada transição- ϵ); e
2. para cada estado s existe no máximo uma transição rotulada com o caractere c partindo de s .

Observação: em um AFD, cada entrada da tabela de transições contém um único estado, o que simplifica o processo de verificação de aceitação de uma cadeia.

Pseudocódigo para verificação de cadeias por meio de um AFD

Input: Uma cadeia de entrada x terminada por um EOF

Output: "Sim", caso a cadeia seja uma sentença válida da linguagem, ou "Nao", caso contrário

```
1:  $s \leftarrow s_0$ 
2:  $c \leftarrow \text{PRÓXIMO CARACTERE}()$ 

3: while  $c \neq \text{EOF}$  do
4:    $s \leftarrow \text{TRANSIÇÃO}(s, c)$ 
5:    $c \leftarrow \text{PRÓXIMO CARACTERE}()$ 

6: if  $s \in F$  then
7:   return "Sim"
8: else
9:   return "Nã o"
```


Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem

Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem
- ▶ A ideia central da conversão de um AFN para um AFD é fazer com que cada estado do AFD corresponda a um conjunto de estados do AFN

Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem
- ▶ A ideia central da conversão de um AFN para um AFD é fazer com que cada estado do AFD corresponda a um conjunto de estados do AFN
- ▶ Seja N um AFN e D um AFD

Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem
- ▶ A ideia central da conversão de um AFN para um AFD é fazer com que cada estado do AFD corresponda a um conjunto de estados do AFN
- ▶ Seja N um AFN e D um AFD
- ▶ O primeiro passo para a conversão é construir uma tabela de transições D_{trans} para D

Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem
- ▶ A ideia central da conversão de um AFN para um AFD é fazer com que cada estado do AFD corresponda a um conjunto de estados do AFN
- ▶ Seja N um AFN e D um AFD
- ▶ O primeiro passo para a conversão é construir uma tabela de transições D_{trans} para D
- ▶ Cada estado do AFD corresponderá a um conjunto de estados do AFN e D_{trans} será construída de forma a simular, “em paralelo”, todas as possíveis transições de N para uma dada entrada

Conversão de um AFN em um AFD

- ▶ Dado um AFN, é possível determinar um AFD que reconheça a mesma linguagem
- ▶ A ideia central da conversão de um AFN para um AFD é fazer com que cada estado do AFD corresponda a um conjunto de estados do AFN
- ▶ Seja N um AFN e D um AFD
- ▶ O primeiro passo para a conversão é construir uma tabela de transições D_{trans} para D
- ▶ Cada estado do AFD corresponderá a um conjunto de estados do AFN e D_{trans} será construída de forma a simular, “em paralelo”, todas as possíveis transições de N para uma dada entrada
- ▶ Para esta tarefa, são necessárias algumas operações que envolvem um estado s de N e um conjunto T de estados de N

Operações sobre os estados de um AFN

| Operação | Descrição |
|---------------------------|---|
| FECHAMENTO- $\epsilon(s)$ | Conjunto de estados do AFN atingíveis a partir do estado s somente por meio de transições- ϵ |
| FECHAMENTO- $\epsilon(T)$ | Conjunto de estados do AFN atingíveis a partir de algum estado $s \in T$ somente por meio de transições- ϵ |
| MOVIMENTO(T, a) | Conjunto de estados do AFN para o qual existe uma transição $s \in T$ cujo rótulo é o símbolo da entrada a |

Relação entre as operações sobre os estados de um AFN

- ▶ Antes mesmo de ver o primeiro símbolo da entrada, N pode estar em qualquer estado pertencente a $\text{FECHAMENTO-}\epsilon(s_0)$, onde s_0 é o estado inicial

Relação entre as operações sobre os estados de um AFN

- ▶ Antes mesmo de ver o primeiro símbolo da entrada, N pode estar em qualquer estado pertencente a $\text{FECHAMENTO-}\epsilon(s_0)$, onde s_0 é o estado inicial
- ▶ Seja T o conjunto de todos os estados atingíveis a partir de s_0 e uma dada sequência de símbolos da entrada, e seja a o próximo símbolo

Relação entre as operações sobre os estados de um AFN

- ▶ Antes mesmo de ver o primeiro símbolo da entrada, N pode estar em qualquer estado pertencente a $\text{FECHAMENTO-}\epsilon(s_0)$, onde s_0 é o estado inicial
- ▶ Seja T o conjunto de todos os estados atingíveis a partir de s_0 e uma dada sequência de símbolos da entrada, e seja a o próximo símbolo
- ▶ Ao ver a , N pode seguir para qualquer estado em $\text{MOVIMENTO}(T, a)$

Relação entre as operações sobre os estados de um AFN

- ▶ Antes mesmo de ver o primeiro símbolo da entrada, N pode estar em qualquer estado pertencente a $\text{FECHAMENTO-}\epsilon(s_0)$, onde s_0 é o estado inicial
- ▶ Seja T o conjunto de todos os estados atingíveis a partir de s_0 e uma dada sequência de símbolos da entrada, e seja a o próximo símbolo
- ▶ Ao ver a , N pode seguir para qualquer estado em $\text{MOVIMENTO}(T, a)$
- ▶ Se existem transições- ϵ , N pode estar em qualquer estado em $\text{FECHAMENTO-}\epsilon(M)$, onde $M = \text{MOVIMENTO}(T, a)$, após ver a

- ▶ O conjunto de todos os estados de D é denominado *Estados- D*

- ▶ O conjunto de todos os estados de D é denominado *Estados- D*
- ▶ Cada estado de D corresponde a um conjunto de estados de AFN que poderiam ser atingidos em N após uma sequência de símbolos da entrada, incluindo as possíveis transições- ϵ antes ou depois dos símbolos serem vistos

- ▶ O conjunto de todos os estados de D é denominado *Estados- D*
- ▶ Cada estado de D corresponde a um conjunto de estados de AFN que poderiam ser atingidos em N após uma sequência de símbolos da entrada, incluindo as possíveis transições- ϵ antes ou depois dos símbolos serem vistos
- ▶ O estado de partida de D é FECHAMENTO- $\epsilon(s_0)$

- ▶ O conjunto de todos os estados de D é denominado *Estados- D*
- ▶ Cada estado de D corresponde a um conjunto de estados de AFN que poderiam ser atingidos em N após uma sequência de símbolos da entrada, incluindo as possíveis transições- ϵ antes ou depois dos símbolos serem vistos
- ▶ O estado de partida de D é FECHAMENTO- $\epsilon(s_0)$
- ▶ Os demais estados são adicionados segundo o algoritmo descrito a seguir

- ▶ O conjunto de todos os estados de D é denominado *Estados- D*
- ▶ Cada estado de D corresponde a um conjunto de estados de AFN que poderiam ser atingidos em N após uma sequência de símbolos da entrada, incluindo as possíveis transições- ϵ antes ou depois dos símbolos serem vistos
- ▶ O estado de partida de D é FECHAMENTO- $\epsilon(s_0)$
- ▶ Os demais estados são adicionados segundo o algoritmo descrito a seguir
- ▶ Os estados de aceitação de D são aqueles cujo conjunto de estados de AFN que representa contém ao menos um estado de aceitação de N

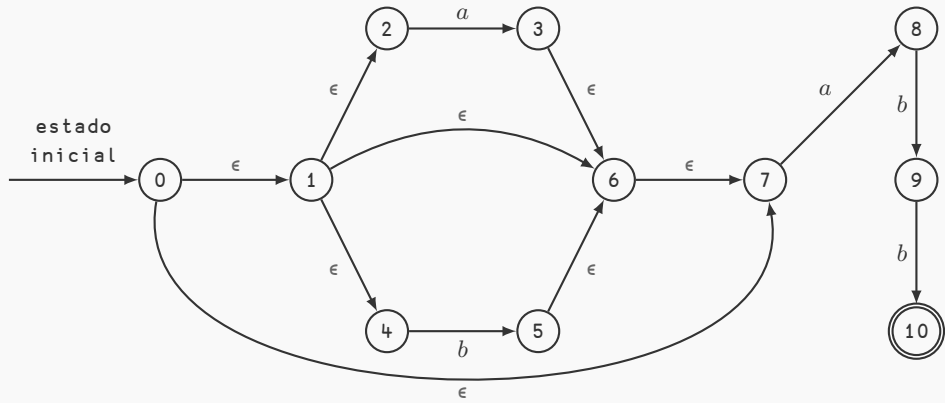
Construção de subconjuntos

- 1: $F \leftarrow \text{FECHAMENTO-}\epsilon(s_0)$
- 2: Desmarque F
- 3: Inclua em F em $\text{Estados-}D$
- 4: **while** existe um estado $T \in \text{Estados-}D$ não marcado **do**
- 5: Marque T
- 6: **for** cada símbolo de entrada a **do**
- 7: $M \leftarrow \text{MOVIMENTO}(T, a)$
- 8: $U \leftarrow \text{FECHAMENTO-}\epsilon(M)$
- 9: **if** $U \notin \text{Estados-}D$ **then**
- 10: Desmarque U
- 11: Inclua U em $\text{Estados-}D$
- 12: $Dtrans[T, a] \leftarrow U$

Cálculo de fechamento- ϵ (T)

- 1: Seja P uma pilha
- 2: Empilhe em P todos os estados em T
- 3: Inclua T em FECHAMENTO- ϵ (T)
- 4: **while** P não estiver vazia **do**
- 5: Desempilhe o topo t de P
- 6: **for** cada estado u que é chegada de uma aresta partindo de t com rótulo ϵ **do**
- 7: **if** $u \notin \text{FECHAMENTO-}\epsilon(T)$ **then**
- 8: Inclua u em FECHAMENTO- ϵ (T)
- 9: Empilhe u

AFN para a linguagem $(a \mid b)^*abb$



Convertendo o AFN em um AFD

- ▶ O estado inicial do AFD será o estado

$$A = \text{FECHAMENTO-}\epsilon(0) = \{0, 1, 2, 4, 7\}$$

Convertendo o AFN em um AFD

- ▶ O estado inicial do AFD será o estado

$$A = \text{FECHAMENTO-}\epsilon(0) = \{0, 1, 2, 4, 7\}$$

- ▶ Veja que no cálculo de $\text{FECHAMENTO-}\epsilon(0)$, o estado 0 é atingível por meio de um caminho vazio

Convertendo o AFN em um AFD

- ▶ O estado inicial do AFD será o estado

$$A = \text{FECHAMENTO-}\epsilon(0) = \{0, 1, 2, 4, 7\}$$

- ▶ Veja que no cálculo de $\text{FECHAMENTO-}\epsilon(0)$, o estado 0 é atingível por meio de um caminho vazio
- ▶ O alfabeto da linguagem é $\Sigma = \{ a, b \}$

Convertendo o AFN em um AFD

- ▶ O estado inicial do AFD será o estado

$$A = \text{FECHAMENTO-}\epsilon(0) = \{0, 1, 2, 4, 7\}$$

- ▶ Veja que no cálculo de $\text{FECHAMENTO-}\epsilon(0)$, o estado 0 é atingível por meio de um caminho vazio
- ▶ O alfabeto da linguagem é $\Sigma = \{ a, b \}$
- ▶ Seguindo o algoritmo, o próximo estado será $B = \text{FECHAMENTO-}\epsilon(M)$, onde $M = \text{MOVIMENTO}(A, a)$

Convertendo o AFN em um AFD

- ▶ O estado inicial do AFD será o estado

$$A = \text{FECHAMENTO-}\epsilon(0) = \{0, 1, 2, 4, 7\}$$

- ▶ Veja que no cálculo de $\text{FECHAMENTO-}\epsilon(0)$, o estado 0 é atingível por meio de um caminho vazio
- ▶ O alfabeto da linguagem é $\Sigma = \{a, b\}$
- ▶ Seguindo o algoritmo, o próximo estado será $B = \text{FECHAMENTO-}\epsilon(M)$, onde $M = \text{MOVIMENTO}(A, a)$
- ▶ Segue que

$$M = \text{MOVIMENTO}(A, a) = \text{MOVIMENTO}(\{0, 1, 2, 4, 7\}, a) = \{3, 8\}$$

Convertendo o AFN em um AFD

► Daí

$$B = \text{FECHAMENTO-}\epsilon(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$$

Convertendo o AFN em um AFD

- ▶ Daí

$$\mathbf{B} = \text{FECHAMENTO-}\epsilon(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$$

- ▶ A transição entre estes dois estados deve ser registrada na tabela *Dtrans*:

$$Dtrans[\mathbf{A}, a] = \mathbf{B}$$

Convertendo o AFN em um AFD

- ▶ Daí

$$B = \text{FECHAMENTO-}\epsilon(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$$

- ▶ A transição entre estes dois estados deve ser registrada na tabela *Dtrans*:

$$Dtrans[A, a] = B$$

- ▶ O próximo estado *C* é computado a partir de $\text{MOVIMENTO}(A, b) = \{5\}$:

$$C = \text{FECHAMENTO-}\epsilon(\{5\}) = \{1, 2, 4, 5, 6, 7\}$$

Convertendo o AFN em um AFD

► Assim, $Dtrans[A, b] = C$

Convertendo o AFN em um AFD

- ▶ Assim, $Dtrans[A, b] = C$
- ▶ Seguindo o algoritmo, apenas mais dois estados distintos serão encontrado:

$$D = \{1, 2, 4, 5, 6, 7, 9\}$$

$$E = \{1, 2, 4, 5, 6, 7, 10\}$$

Convertendo o AFN em um AFD

- ▶ Assim, $Dtrans[A, b] = C$
- ▶ Seguindo o algoritmo, apenas mais dois estados distintos serão encontrado:

$$D = \{1, 2, 4, 5, 6, 7, 9\}$$

$$E = \{1, 2, 4, 5, 6, 7, 10\}$$

- ▶ Dentre os estados identificados, apenas E é estado de aceitação, pois contém o estado 10 de N

Convertendo o AFN em um AFD

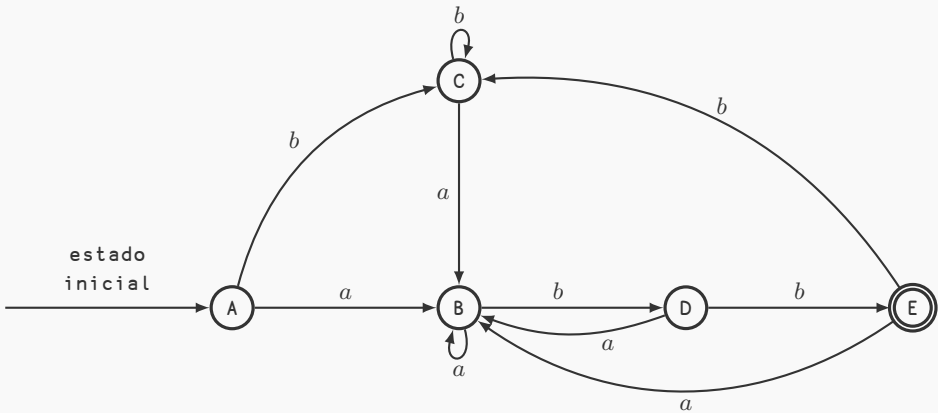
- ▶ Assim, $Dtrans[A, b] = C$
- ▶ Seguindo o algoritmo, apenas mais dois estados distintos serão encontrado:

$$D = \{1, 2, 4, 5, 6, 7, 9\}$$

$$E = \{1, 2, 4, 5, 6, 7, 10\}$$

- ▶ Dentre os estados identificados, apenas E é estado de aceitação, pois contém o estado 10 de N
- ▶ No pior caso, seriam identificados 2^n novos estados, onde n é o número de estados do AFN

Resultado da conversão do AFN para um AFD



Referências

1. **AHO**, Alfred V, **SETHI**, Ravi, **ULLMAN**, Jeffrey D. *Compiladores: Princípios, Técnicas e Ferramentas*, LTC Editora, 1995.