Universidade Federal de Alfenas

Linguagens Formais e Autômatos

Aula 10 – Autômatos Finitos Não Determinísticos (AFN)

humberto@bcc.unifal-mg.edu.br

Determinismo...

- Quando uma máquina está em um estado e lê o próximo símbolo de entrada, sabemos qual será o próximo estado: está determinado.

Determinismo...

- Quando uma máquina está em um estado e lê o próximo símbolo de entrada, sabemos qual será o próximo estado: está determinado.

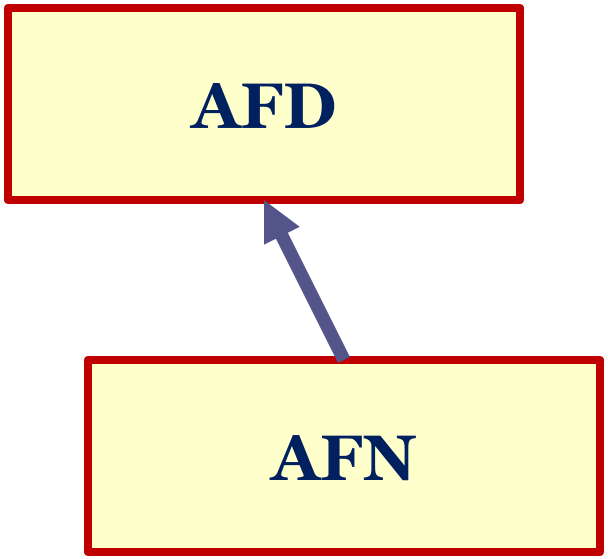
Chamamos isso de
COMPUTAÇÃO DETERMINÍSTICA

Autômatos Finitos Não Determinísticos

- Segundo algumas definições, todo AFD é automaticamente um AFN.

Autômatos Finitos Não Determinísticos

- Segundo algumas definições, todo AFD é automaticamente um AFN.
  - Podemos considerar que não determinismo é uma generalização de determinismo.



Autômatos Finitos Não Determinísticos

- Todo estado de um AFD sempre tem EXATAMENTE uma transição para cada símbolo no alfabeto.

Autômatos Finitos Não Determinísticos

- Todo estado de um AFD sempre tem EXATAMENTE uma transição para cada símbolo no alfabeto.

- Os AFNs podem violar esta regra.

Autômatos Finitos Não Determinísticos

- **Todo estado de um AFD sempre tem EXATAMENTE uma transição para cada símbolo no alfabeto.**
- Os **AFNs podem violar** esta regra.
- AFNs também podem conter transições vazias.

Como um AFN computa?

Como um AFN computa?

- Suponha que você esteja rodando um AFN sobre uma palavras de entrada e alcança para um estado com múltiplas maneiras de prosseguir:

Como um AFN computa?

- Suponha que você esteja rodando um AFN sobre uma palavras de entrada e alcança para um estado com múltiplas maneiras de prosseguir:
  - Após ler este símbolo, a máquina se divide em múltiplas cópias de si mesma e segue todas as possibilidades em paralelo.

Como um AFN computa?

- Suponha que você esteja rodando um AFN sobre uma palavras de entrada e alcança para um estado com múltiplas maneiras de prosseguir:
  - Após ler este símbolo, a máquina se divide em múltiplas cópias de si mesma e segue todas as possibilidades em paralelo.
  - **Se existem escolhas subseqüentes, a máquina divide-se novamente.**

Como um AFN computa?

- Se o próximo símbolo lido pelo AFN não aparece sobre qualquer uma das transições do estado corrente, então esta cópia "morre", juntamente com o ramo de computação associado a ela.

Como um AFN computa?

- Se o próximo símbolo lido pelo AFN não aparece sobre qualquer uma das transições do estado corrente, então esta cópia “morre”, juntamente com o ramo de computação associado a ela.

- Se qualquer uma das cópias da máquina está em um estado final, quando a palavra já foi totalmente lida, então a palavra é reconhecida pelo AFN.
  - Em outras palavras, o AFN aceita a palavra.

Como um AFN computa?

- Se um estado qualquer com pelo menos uma transição λ for alcançado, algo semelhante acontece:

Como um AFN computa?

- Se um estado qualquer com pelo menos uma transição λ for alcançado, algo semelhante acontece:

  - A máquina divide-se:
    - A primeira cópia continua no estado atual;
    - A s demais cópias se movem para o próximos estado que levam as transições λ.

Como um AFN computa?

- Se um estado qualquer com pelo menos uma transição λ for alcançado, algo semelhante acontece:
  - A máquina divide-se:
    - A primeira cópia continua no estado atual;
    - A s demais cópias se movem para o próximos estado que levam as transições λ.
  - É importante dizer que **nenhum símbolo da palavra em teste de aceitação é consumido**.

Como um AFN computa?

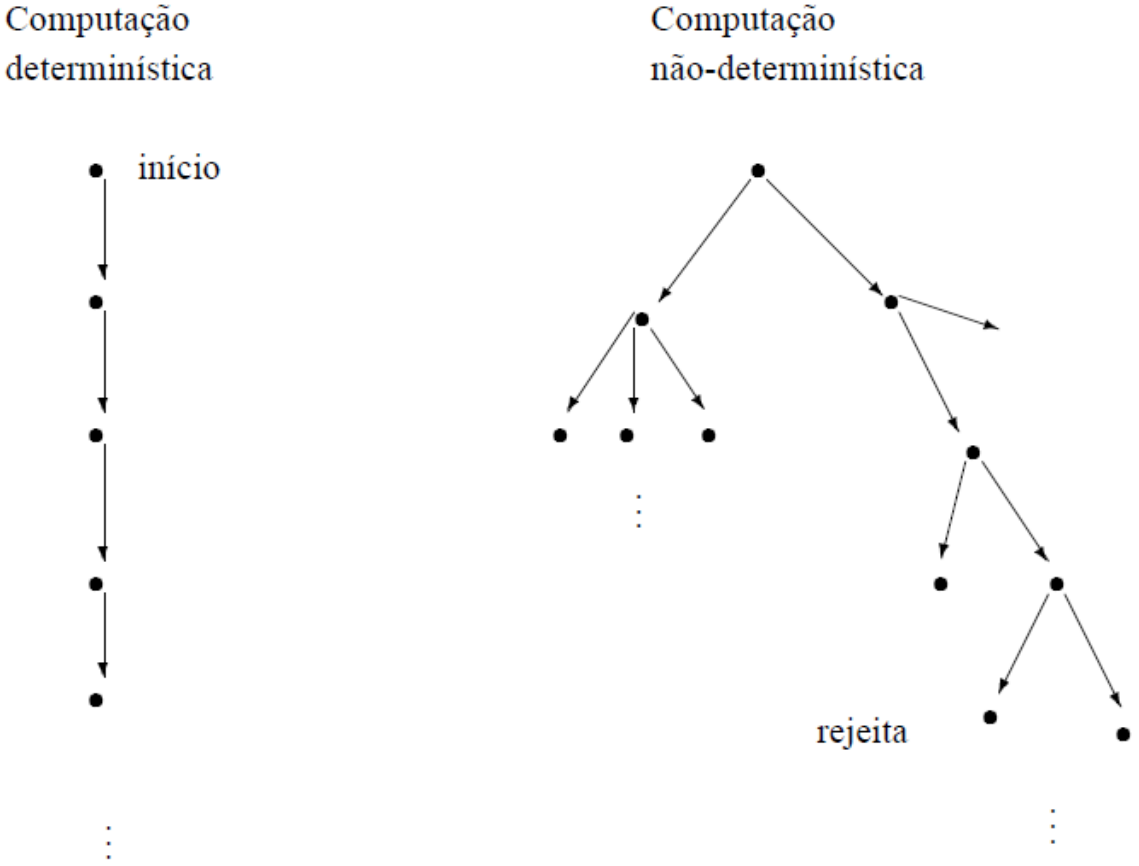
- O não determinismo pode ser visto como uma espécie de computação paralela, na qual múltiplos e independentes "processos" ou "*threads*" podem estar rodando concorrentemente.

Como um AFN computa?

- O não determinismo pode ser visto como uma espécie de computação paralela, na qual múltiplos e independentes "processos" ou "*threads*" podem estar rodando concorrentemente.
  - Quando o AFN se divide, isso corresponde ao processo bifurcar em vários filhos.

Como um AFN computa?

- Outra maneira de pensar em uma computação não determinística, é imaginar uma árvore de possibilidades.



Como um AFN computa?

- árvore de possibilidades.

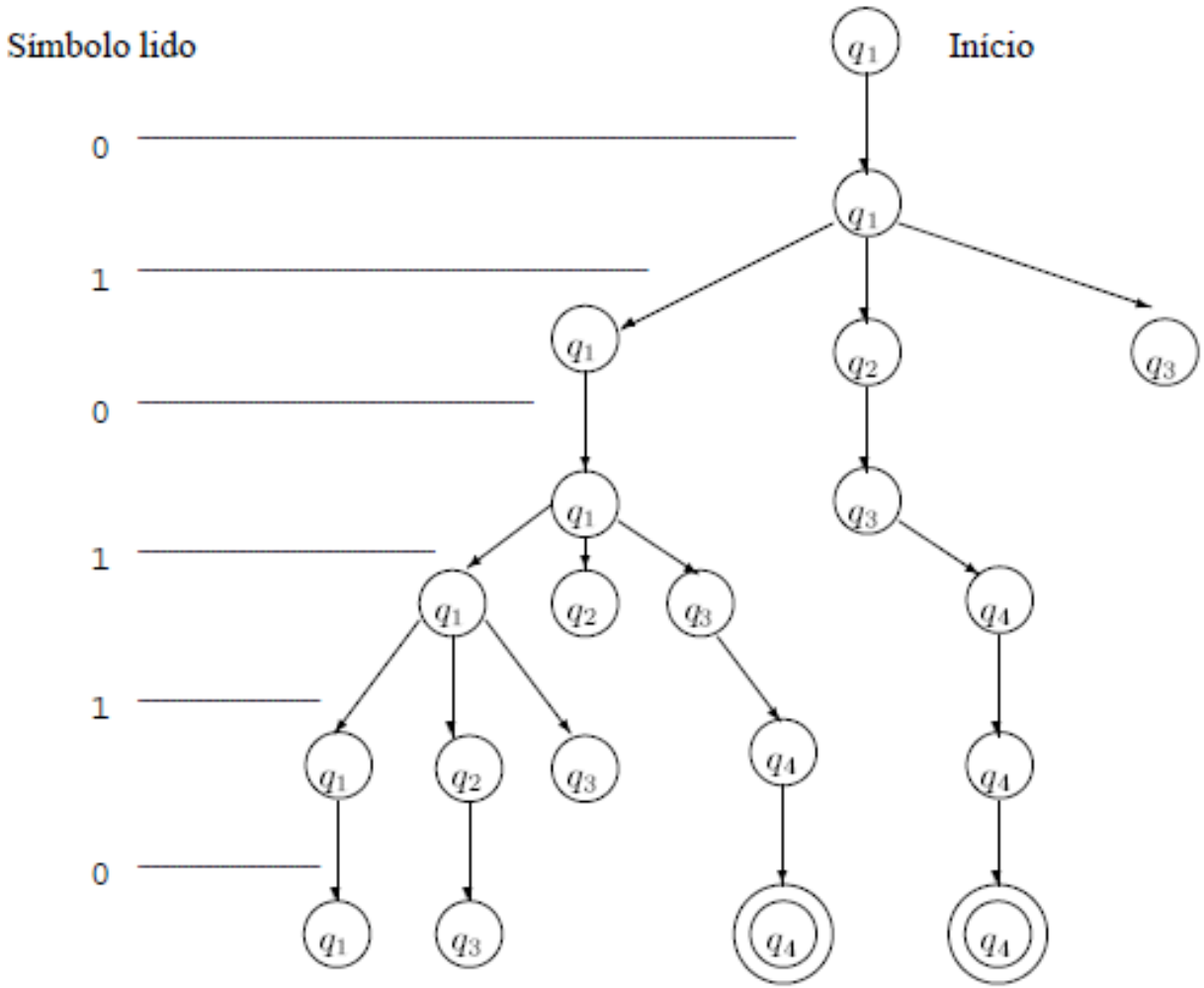


Figura 1.16: A computação de N_1 sobre a entrada 010110

Vantagens na utilização de AFNs

Vantagens

- Os AFNs são úteis em vários sentidos. Com mostraremos, todo AFN pode ser convertido num AFD equivalente.

Vantagens

- Os AFNs são úteis em vários sentidos. Com mostraremos, todo AFN pode ser convertido num AFD equivalente.
  - Construir AFNs é as vezes muito mais fácil que construir AFDs diretamente.

Vantagens

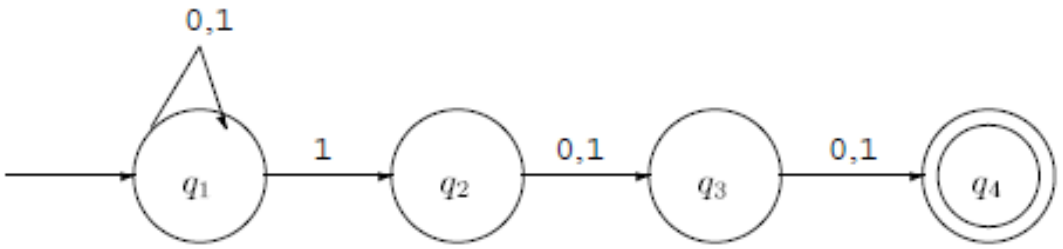
- Os AFNs são úteis em vários sentidos. Com mostraremos, todo AFN pode ser convertido num AFD equivalente.
  - Construir AFNs é as vezes muito mais fácil que construir AFDs diretamente.
  - **Um AFN pode ser muito menor que seu correspondente determinístico**.

Vantagens

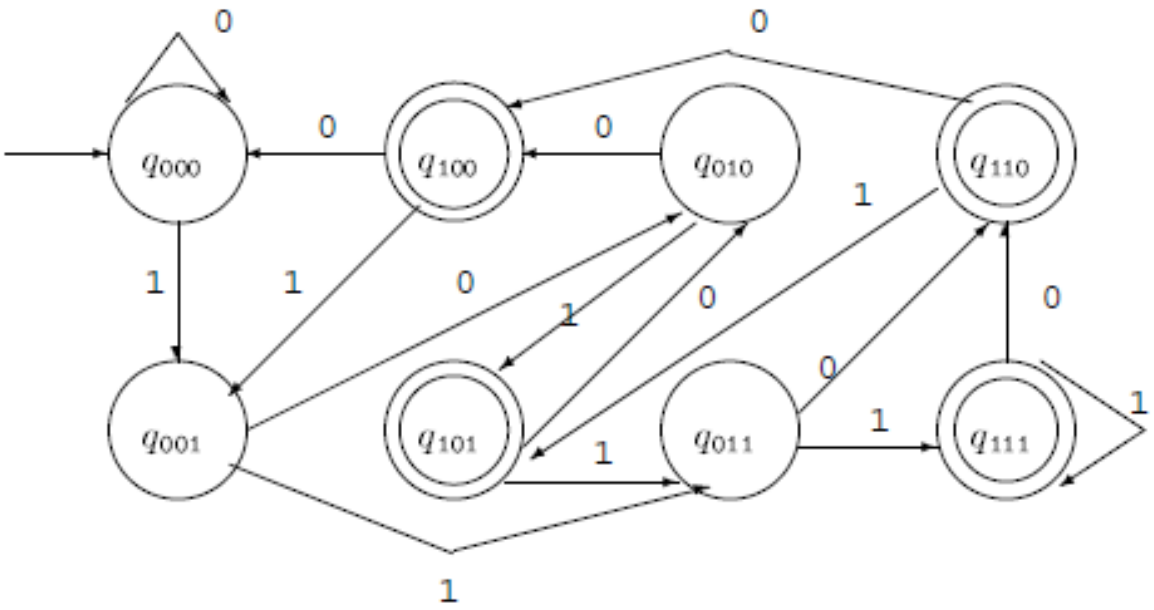
- Os AFNs são úteis em vários sentidos. Com mostraremos, todo AFN pode ser convertido num AFD equivalente.
  - Construir AFNs é as vezes muito mais fácil que construir AFDs diretamente.
  - Um AFN pode ser muito menor que seu correspondente determinístico.
  - Seu funcionamento pode ser muito mais fácil de ser entendido.

Simplicidade do AFN

- AFN:



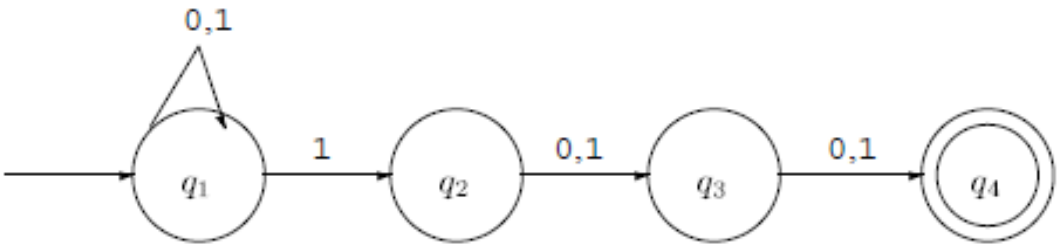
- AFD equivalente:



Trabalho prático

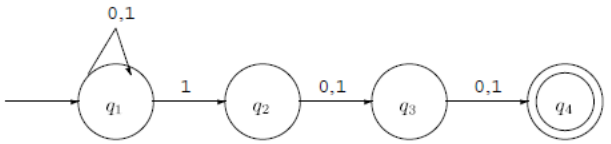
Arquivo de entrada

- q1: (q1,0); (q1,1); (q2,1);
- q2: (q3,0); (q3,1);
- q3: (q4,0); (q4,1);
- _q4: ;



- Podem haver transições vazias:
  - q5: (q7,);

Arquivo de entrada



- Após lido o AFN do arquivo de entrada, o seu programa deve ler outro arquivo com palavras:
  - 0000000111
  - 00
  - 010101
  - ...
- Depois da leitura das palavras, você deve simular a computação do AFN, informando quantas palavras foram aceitas, e quantas forma rejeitadas.

Trabalho Prático

- Devem ser criadas Threads quando bifurcações são encontradas:

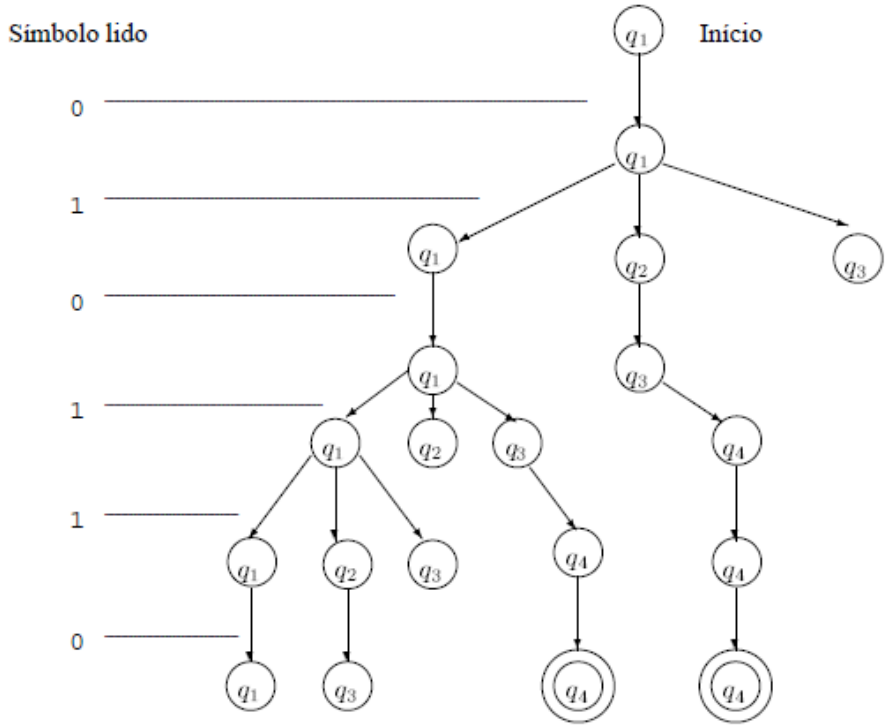


Figura 1.16: A computação de N_1 sobre a entrada 010110

Trabalho Prático

- Data de entrega: 25/05/2011
- O aluno também deve entregar um relatório descrevendo o TP.
- A entrega deve ser feita no email
  humberto (arroba) bcc.unifal-mg.edu.br
  até as 23:59 do dia 25/05/2011.
- Haverá apresentação do TP no dia 26/05/2011.

Bibliografia

- SIPSER, Michael. Introdução à Teoria da Computação. 2a ed.:São Paulo, Thomson, 2007.

- VIEIRA, Newton José. Introdução aos Fundamentos da Computação: Linguagens e Máquinas. 1a ed.: Rio de Janeiro: Thomson, 2006.