Linguagens Formais e Autômatos (LFA)

Aula de 30/09/2013

Máquinas de Moore e Mealy
Conceito

Autômatos Finitos Determinísticos

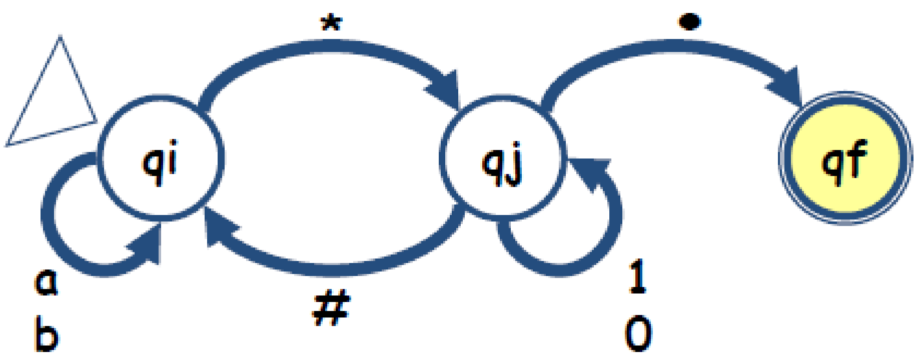
- Simples dispositivos de reconhecimento de sentenças
- Informação de saída é bastante limitada
  - Aceitação ou rejeição da cadeia de entrada
- Existem extensões de AFDs que procuram ampliar as possibilidades de uso
  - Transdutores finitos: máquinas de Moore e Mealy

AFD com saída

- Um autômato com saída é aquele que não apenas lê uma fita de entrada, mas também grava uma fita de saída.
  - São máquinas particularmente úteis para representar computacionalmente várias coisas que nos cercam.
- As condições de gravação obviamente imprimem certas características à representação. Por exemplo:
  - O vocabulário de saída e sua relação com a entrada
  - O que determina a gravação: se o estado, apenas, ou estado combinado ao símbolo da fita

A ideia

- Seja L uma linguagem regular que aceita sentenças definidas por este AFD:



Que coisas interessantes poderíamos representar a respeito das cadeias processadas pelo AFD em uma e outra condição?

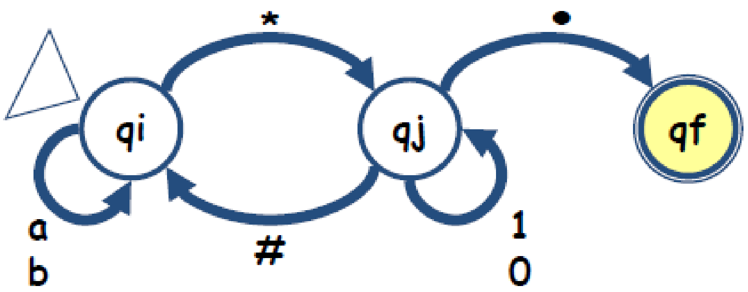
- Imaginem se pudéssemos **gravar símbolos** em 2 condições:
  1. sempre que estivéssemos em um estado genérico qk; ou
  2. ao realizarmos uma transição para qk, tendo lido na fita o símbolo α.

Representações que geram representações

- Autômatos capazes de "gravar" símbolos têm, necessariamente, um vocabulário de saída.
  – Como este vocabulário está associado a estados resultantes do processamento simbólico da fita de entrada, podemos de fato elaborar uma "linguagem de saída", para representar a cadeia de entrada (ou aspectos dela).

    - Esta possibilidade, além de ser importante para praticamente toda a Computação, é também o coração do que se conhece por "representação de conhecimento", que constitui a base para sistemas capazes de fazer "inferências" a respeito de dados de entrada (i.e. aplicações de Inteligência Artificial).

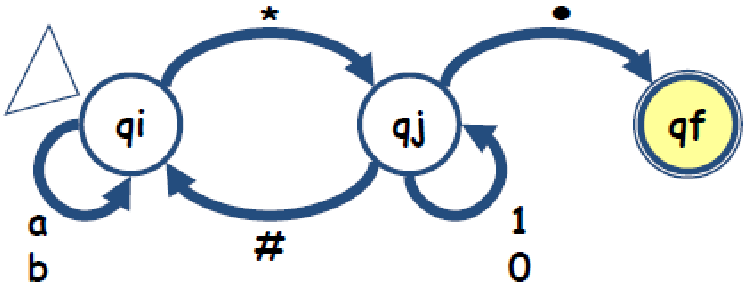
Exemplos de "conhecimento" que se pode expressar



- Coisas que poderíamos saber se pudéssemos gravar símbolos:

1. **sempre que estivéssemos em um estado genérico qk**
   - Se estou em qi, tanto posso não ter lido nada (é o estado inicial), como posso já ter lido uma cadeia de a's ou b's, como posso ter lido uma cadeia de a's ou b's seguida de *, de uma cadeia de 1's ou '0's, seguida de #. E como vou gravar apenas um símbolo de saída, este estado é bastante ambíguo.
   - Já se estou em qj sei que ao menos – garantidamente – já processei uma cadeia de a's ou b's seguida de *. Não posso dizer, porém, se já processei 1's ou 0's, #, etc.
   - Mas se estou em qf posso garantir, por exemplo, que a cadeia processada tem um e somente um ponto.
   - Vejam que, como a fita de saída contém um mesmo símbolo para cada visita de qk, ela contém informações interessantes, não? Quais?

Exemplos de "conhecimento" que se pode expressar



- Coisas que poderíamos saber se pudéssemos gravar símbolos:

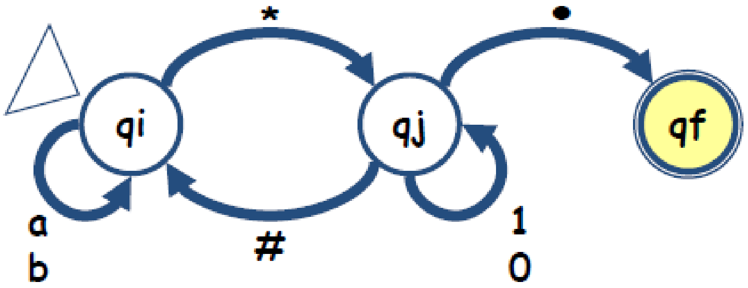
2. ao realizarmos uma transição para qk, tendo lido na fita o símbolo α.

Aqui as coisas ficam bem interessantes. Eis um exemplo:

- Se estou em qi tendo lido: "a", posso por exemplo gravar "%"; "b", posso por exemplo gravar "$"; e '#' posso por exemplo gravar "-".
- Se estou em qj tendo lido: "1", posso por exemplo gravar "&"; "0", posso por exemplo gravar "!"; e '*' posso por exemplo gravar "+";
- Finalmente, se estou em qf tendo lido "." posso gravar "~".

Vale notar que as gravações de saída deste exemplo "preservam toda a informação de entrada", criando um *homomorfismo*, que é a base para processos de conversão, compressão, criptografia, etc.

Representando abstrações: tipos e instâncias



Estas representações ABSTRAEM detalhes que podem não interessar para certos fins. Elas expressam TIPOS de informação, mas não INSTÂNCIAS.

Um caso interessante a examinar é se:

- estando em qi não gravamos nada se entram "a" ou "b", mas gravamos um símbolo α qualquer se entra "#" e estando em qj não gravamos nada se entram "0" ou "1" mas gravamos um símbolo ß qualquer se entra "*".
- A presença de "ß" nos permite inferir que foi processada uma cadeia de a's ou b's , da mesma forma que a presença de "α" nos permite inferir a presença de a's ou b's e de 0's ou 1's. Os símbolos na fita de saída nos permitiriam então contar quantas cadeias de "letras" e "algarismos" apareceram na fita.
- Se quiséssemos saber quantas letras apareceram em cada cadeia literal ou quantos algarismos apareceram em cada cadeia numérica poderíamos gravar um "l" ao ler "a" ou "b" e um "a" ao ler "1" ou "0", por exemplo.

Importância cognitiva dos TIPOS de informação

- Poder reconhecer e expressar "tipos" de informação é importantíssimo para fazermos inferências.
  - Sem eles, não conseguiríamos formular nosso conhecimento como REGRAS mais gerais do que os FATOS que elas descrevem.
    - Por exemplo, sabemos que "fogo" queima. Qual fogo?
      - Tanto o que já experimentamos (instâncias) quanto os que podemos vir a experimentar ou não (tipo).
        » Esta generalização de regras a partir de instâncias é para alguns autores a característica principal da inteligência humana.

E as máquinas de Moore e de Mealy?

- As máquinas de **Moore** são autômatos finitos que gravam símbolos de saída, dependendo somente dos ESTADOS do autômato.
  - São definidas por uma sétupla $(Q, \Sigma, \Delta, \delta, \lambda, q0, F)$:

    Q é um conjunto finito de estados

    Σ é o alfabeto terminal

    Δ é o alfabeto de símbolos de saída

    $\delta: Q \times \Sigma \rightarrow Q$, a função de transição de estados (leitura)

    $\lambda: Q \rightarrow \Delta^*$, a função de transdução (escrita)

    $q0 \in Q$ é o estado inicial

    $F \subset Q$ é o conjunto de estados finais de aceitação

E as máquinas de Moore e de Mealy?

- As máquinas de **Mealy** são autômatos finitos que gravam símbolos de saída, dependendo dos ESTADOS e das TRANSIÇÕES do autômato.
  - São definidas por uma sétupla $(Q, \Sigma, \Delta, \delta, \lambda, q0, F)$:

    Q é um conjunto finito de estados

    Σ é o alfabeto terminal

    Δ é o alfabeto de símbolos de saída

    $\delta: Q \times \Sigma \rightarrow Q$, a função de transição de estados (leitura)

    $\lambda: Q \times \Sigma \rightarrow \Delta^*$, a função de transdução (escrita)

    $q0 \in Q$ é o estado inicial

    $F \subset Q$ é o conjunto de estados finais de aceitação

Máquina de Moore: exemplo (Ramos,2009 p.230)

- $T = (Q, \Sigma, \Delta, \delta, \lambda, q0, F)$
  - $Q = \{q0,q1\}$
  - $\Sigma = \{a,b,c\}$
  - $\Delta = \{1\}$
  - $\delta = \{ (q0,a) \rightarrow q1, (q1,b) \rightarrow q1, (q1,c) \rightarrow q0 \}$
  - $\lambda = \{ q0 \rightarrow 1, q1 \rightarrow \varepsilon \}$
  - $F = \{q1\}$

- Simular no JFLAP

Máquina de Mealy: exemplo (Ramos,2009 p.231)

- $T = (Q, \Sigma, \Delta, \delta, \lambda, q0, F)$
  - $Q = \{q0,q1\}$
  - $\Sigma = \{a,b,c\}$
  - $\Delta = \{a,b,c\}$
  - $\delta = \{ (q0,a) \rightarrow q1, (q1,b) \rightarrow q1, (q1,c) \rightarrow q0 \}$
  - $\lambda = \{ (q0,a) \rightarrow ab, (q1,b) \rightarrow \varepsilon, (q1,c) \rightarrow c \}$
  - $F = \{q1\}$

- Simular no JFLAP

Observações sobre os transdutores

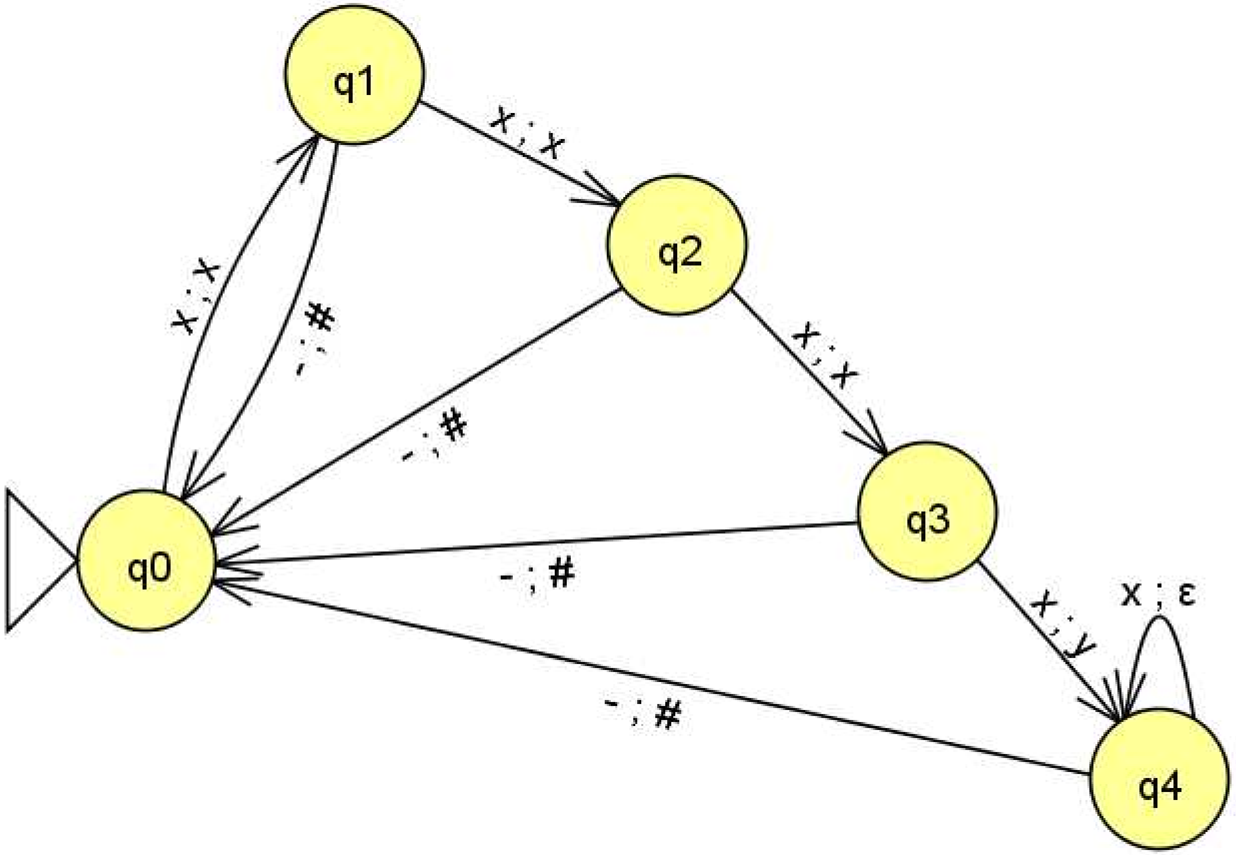
- Equivalência
  - toda máquina de Mealy pode ser simulada por uma máquina de Moore, e vice-versa.
  - A seleção do tipo de máquina leva em conta a conveniência de manipulação e representação, conforme o caso.

- Linguagem de saída (sentenças sobre Δ)
  - O tipo de linguagem gerada por um transdutor finito corresponde ao mesmo tipo da linguagem reconhecida pelo autômato subjacente (linguagens regulares)

Exemplo de equivalência entre transdutores finitos (Ramos 2009, p.232)

- Seja L1 = $xx^*(-xx^*)^*$, definida sobre Σ ={x,-}
- Seja L2, definida sobre Δ = {x,y,#}, modificando as cadeias de L1 da seguinte forma:
  - As subcadeias de entrada xx^* que contiverem três ou menos símbolos x devem ser reproduzidas de forma idêntica na saída (com um, dois ou três símbolos 'x');
  - As subcadeias de entrada xx^* que contiverem quatro ou mais símbolos x devem ser reproduzidas na saída como xxxy;
  - Todos os símbolos '-' da cadeia entrada devem ser substituídos pelo símbolo '#' na cadeia de saída

Máquina de Mealy para gerar L2



Máquina de Moore para gerar L2