LFA0001 – Linguagens Formais e Autômatos
Aula 08
Propriedades das Linguagens Regulares

Karina Girardi Roggia
karina.roggia@udesc.br

Departamento de Ciência da Computação
Centro de Ciências Tecnológicas
Universidade do Estado de Santa Catarina

2016

Sumário

Propriedades das Ling. Regulares

Ling. Regulares são representadas por formalismos

- de pouca complexidade
- com grande eficiência
- de fácil implementação

Classe relativamente simples de linguagens, logo

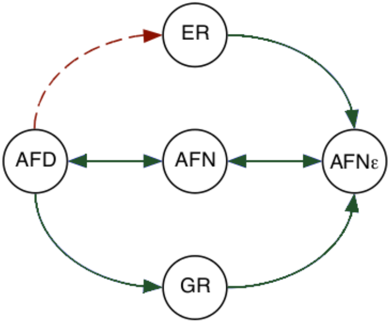
- é restrita e limitada
- facilmente se define linguagens não regulares

Questões sobre Linguagens Regulares

- Como determinar se uma linguagem é regular?
- Linguagens são conjuntos, portanto as linguagens regulares são..
  - fechadas para união?
  - fechadas para concatenação?
  - fechadas para intersecção?
  - ...
- Como verificar se uma Linguagem Regular é finita, infinita ou vazia?
- É possível analisar duas Linguagens Regulares e concluir se são iguais ou diferentes?

Análise das Propriedades

A análise será desenvolvida somente para **um** dos formalismos estudados



Outro formalismo? Tradução entre eles.

Obs: Complexidade de Tempo para Autômatos Finitos

Autômatos finitos estão na classe de algoritmos mais eficientes em termo de tempo de processamento

- supondo que toda a entrada deva ser lida
- se a hipótese for relaxada, pode-se obter algoritmos mais eficientes para algumas linguagens (ex: Σ^*)

Qualquer autômato finito é igualmente eficiente

- a menos de eventual redundância de estados
- o que não influi no tempo de processamento

Eliminação de redundância de estados

- Autômato Finitos (Determinístico) Mínimo

Lema do Bombeamento

Ideia básica

- Se uma linguagem é regular, então
  - é aceita por um AFD com n estados.
- Se o AFD reconhece alguma palavra w tal que $|w| \geq n$
  - o processamento passa por algum estado q mais de uma vez
  - portanto existe um ciclo na função programa que passa por q

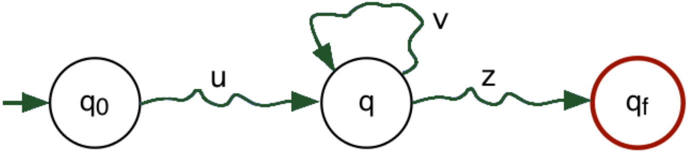
Bombeamento para Ling. Regulares

Dado AFD sendo

- n o número de estados
- w palavra aceita pelo AFD com $|w| \geq n$

Então w pode ser subdividida em três subpalavras $w = uvz$

- $|uv| \leq n$ e $|v| \geq 1$
- v é a parte de w reconhecida pelo ciclo
- o ciclo pode ser executado (*bombeado*) zero ou mais vezes



Para qualquer $i \geq 0$, uv^iz é palavra aceita pelo AFD

Lema do Bombeamento para Ling. Regulares

Teorema (Lema do Bombeamento para Ling. Regulares)

Se L é uma Linguagem Regular, então:

- *existe constante n tal que*
- *para qualquer palavra $w \in L$ onde $|w| \geq n$*
- *$w = uvz$ onde*
    - $v \neq \varepsilon$
    - $|uv| \leq n$
    - *para todo $i \in \mathbb{N}$, $uv^iz \in L$.*

UDESC

Demonstração

Se L é Ling. Regular, então existe um AFD $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ tal que $\text{ACEITA}(M) = L$

Suponha que:

- n é o cardinal de Q
- existe $w = a_1 a_2 \dots a_m \in L$ tal que $m \geq n$
- $\delta(q_0, a_1) = q_1$
- $\delta(q_1, a_2) = q_2$
- $\dots$
- $\delta(q_{m-1}, a_m) = q_m$

Como $m \geq n$, então existe r e s com $0 \leq r < s \leq n$ tais que

- $q_r = q_s$
- $\delta^*(q_0, a_1 \dots a_r) = q_r$
- $\delta^*(q_r, a_{r+1} \dots a_s) = q_s$
- $\delta^*(q_s, a_{s+1} \dots a_m) = q_m$

ou seja, o autômato passa mais de uma vez no estado $q_r = q_s$

Demonstração

Sejam

- $u = a_1 \dots a_r$
- $v = a_{r+1} \dots a_s$
- $z = a_{s+1} \dots a_m$

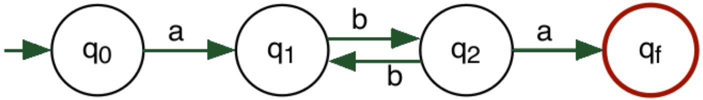
Como $r < s \leq n$, então

$$|v| \geq 1 \text{ e } |uv| \leq n$$

Como $q_r = q_s$, então v é reconhecida em um ciclo.

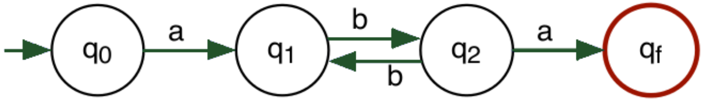
Portanto, para todo $i \in \mathbb{N}$, $uv^i z \in L$.

Exemplo



- $n = 4$
- para $w = abbba$
  - $q_r = q_s =$
  - $u =$
  - $v =$
  - $z =$

Exemplo



- $n = 4$
- para $w = abbba$
    - $q_r = q_s = q_1$
    - $u = a$
    - $v = bb$
    - $z = ba$

Investigação se uma Linguagem é Regular

Mostrar que L é Regular

- representar L a partir de algum dos formalismos regulares

Mostrar que L **não** é Regular

- desenvolvida caso a caso
- pode-se usar o Lema do Bombeamento

Exemplo

$$L = \{w \mid w \text{ possui o mesmo nro de símbolos } a \text{ e } b\}$$

Suponha L regular. Então há AFD com n estados que aceita L. Seja $w = a^n b^n$. $w \in L$ com $|w| = 2n \geq n$. Portanto, pelo lema do bombeamento, $w = uvz$ tal que

- $|uv| \leq n$
- $|v| \geq 1$
- $\forall i \in \mathbb{N}, uv^i z \in L$

Absurdo! Como $|uv| \leq n$

- uv é composta exclusivamente por símbolos a
- $uv^k z$ para $k \geq 2$ não pertence a L (possui mais as do que bs)

Operações Fechadas sobre Ling. Regulares

Álgebra de Linguagens Regulares

- Construção de novas linguagens a partir de conhecidas

Classe das Linguagens Regulares é fechada para

- união
- concatenação
- complemento
- intersecção

Operações Fechadas sobre Ling. Regulares

Teorema (Operações Fechadas sobre Ling. Regulares)

A Classe das Linguagens Regulares é fechada para as operações de:

- *união*
- *concatenação*
- *complemento*
- *intersecção*

Demonstração:

União e Concatenação

Trivial a partir da definição de Expressão Regular

Demonstração

Complemento
Suponha L uma Linguagem Regular sobre Σ^*. Então existe AFD

$$M = \langle \Sigma, Q, \delta, q_0, F \rangle$$

tal que ACEITA$(M) = L$.

Construiremos um AFD M_C que aceitará $\overline{L}$.

Ideia: inverter as condições de ACEITA e REJEITA de M.

- Relembrar que M pode rejeitar por indefinição.

Demonstração

$$M_C = \langle \Sigma, Q_C, \delta_C, q_0, F_C \rangle$$

- $Q_C = Q \uplus \{d\}$
- $F_C = Q_C \setminus F$
- δ_C é como δ, adicionando para todo $a \in \Sigma$ e $q \in Q$, as transições:
  - $\delta_C(q, a) = d$ se $\delta(q, a)$ não é definida
  - $\delta_C(d, a) = d$

Claramente, M_C é tal que

$$\text{ACEITA}(M_C) = \overline{L}, \text{ ou seja, } \text{ACEITA}(M_C) = \text{REJEITA}(M)$$

Demonstração

Intersecção

Suponha L_1 e L_2 linguagens regulares. Pela lei de DeMorgan

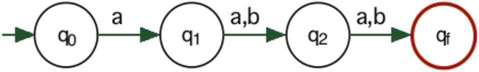
$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Como a Classe das Linguagens Regulares é fechada para complemento e união, também é fechada para intersecção.

Exemplo

Complemento de Linguagem Regular

$$M = \langle \{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta, q_0, \{q_f\} \rangle$$

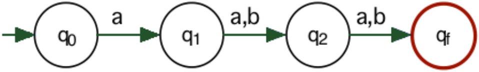


UDESC

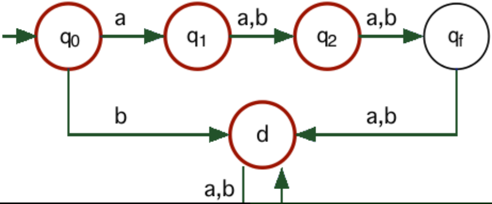
Exemplo

Complemento de Linguagem Regular

$$M = \langle \{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta, q_0, \{q_f\} \rangle$$



$$M_C = \langle \{a, b\}, \{q_0, q_1, q_2, q_f, d\}, \delta_C, q_0, \{q_0, q_1, q_2, d\} \rangle$$



Linguagem Regular Vazia, Finita ou Infinita

Teorema (Linguagem Regular Vazia, Finita ou Infinita)

Se L é uma Linguagem Regular aceita por um AFD $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ com n estados, então L é:

- Vazia *sse M **não** aceita qualquer palavra w tal que $|w| < n$*
- Finita *sse M **não** aceita qualquer palavra w tal que $n \leq |w| < 2n$*
- Infinita *sse M aceita palavra w tal que $n \leq |w| < 2n$*

Demonstração

Infinita sse M aceita palavra w tal que $n \leq |w| < 2n$

$(\Leftarrow)$
Verificar se M aceita alguma palavra w tal que $n \leq |w| < 2n$

- processar o autômato para todas as entradas w neste intervalo
- se existe $w \in L$, ela pode ser definida como $w = uvz$
  - $|uv| \leq n$
  - $|v| \geq 1$
- então para todo $i \in \mathbb{N}$, $uv^i z \in L$

Logo L é infinita.

Demonstração

($\Rightarrow$)
Se L é infinita, então existe w tal que $|w| \geq n$.
Duas possibilidades:
Caso 1: $|w| < 2n$

- prova está completa

Demonstração

Caso 2: $|w| \geq 2n$

Suponha que

- não existe palavra de comprimento menor de $2n$ aceita
- w é a menor palavra tal que $|w| \geq 2n$

w pode ser definida como $w = uvz$

- $|uv| \leq n|$ e $|v| \geq 1$
- em particular, $1 \leq |v| \leq n$

Logo, $uz \in L$, o que é absurdo, pois...

- $|uz| \geq 2n$
  Contradiz suposição de que w é menor palavra com $|w| \geq 2n$
- $|uz| < 2n$
  $n \leq |uz| < 2n$ (pois $|uvz| \geq 2n, 1 \leq |v| \leq n$)
  Contradiz suposição de que não existe w onde $n \leq |w| < 2n$

Demonstração

Vazia sse M não aceita qualquer palavra w tal que $|w| < n$

Processa M para todas as palavras de comprimento menor que n

Se rejeita todas as palavras: linguagem vazia.

- Exercício: detalhamento da prova.

Demonstração

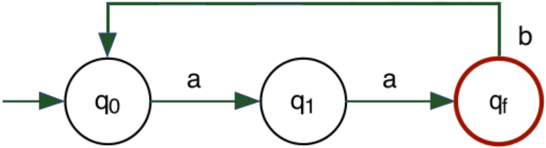
Finita sse M não aceita qualquer palavra w tal que $n \leq |w| < 2n$

Prova por contraposição, portanto a mesma prova para o caso de linguagem infinita.

- Exercício: explicar a afirmação acima.

Exemplo

Linguagem Regular Infinita



A linguagem é infinita sse aceita palavra w tal que $n \leq |w| \leq 2n$

- $aabaa$ é aceita
- $3 \leq |aabaa| < 6$

Logo, a linguagem aceita pelo autômato é infinita.

Igualdade de Linguagens Regulares

Existe algoritmo para **verificar** se dois autômatos finitos são equivalentes

Teorema (Igualdade de Ling. Regulares)

Se M_1 e M_2 são autômatos finitos, então existe algoritmo para determinar se

$$ACEITA(M_1) = ACEITA(M_2)$$

Demonstração

Suponha M_1 e M_2 autômatos finitos tal que ACEITA$(M_1) = L_1$ e ACEITA$(M_2) = L_2$.
Seja a linguagem L_3 tal que

$$L_3 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$$

L_3 é linguagem regular e, portanto, é possível construir M_3 tal que ACEITA$(M_3) = L_3$.
Porém, $L_1 = L_2$ sse L_3 é vazia.
Como há algoritmo para verificar se uma linguagem é vazia, o teorema está demonstrado.