Linguagens Formais e Autômatos

Aula 11 - Linguagens não regulares

Prof. Dr. Daniel Lucrédio
Departamento de Computação / UFSCar
Última revisão: ago/2015

Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
    - Capítulo 4 - Seção 4.1
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
    - Capítulo 1 - Seção 1.4

Linguagens não regulares

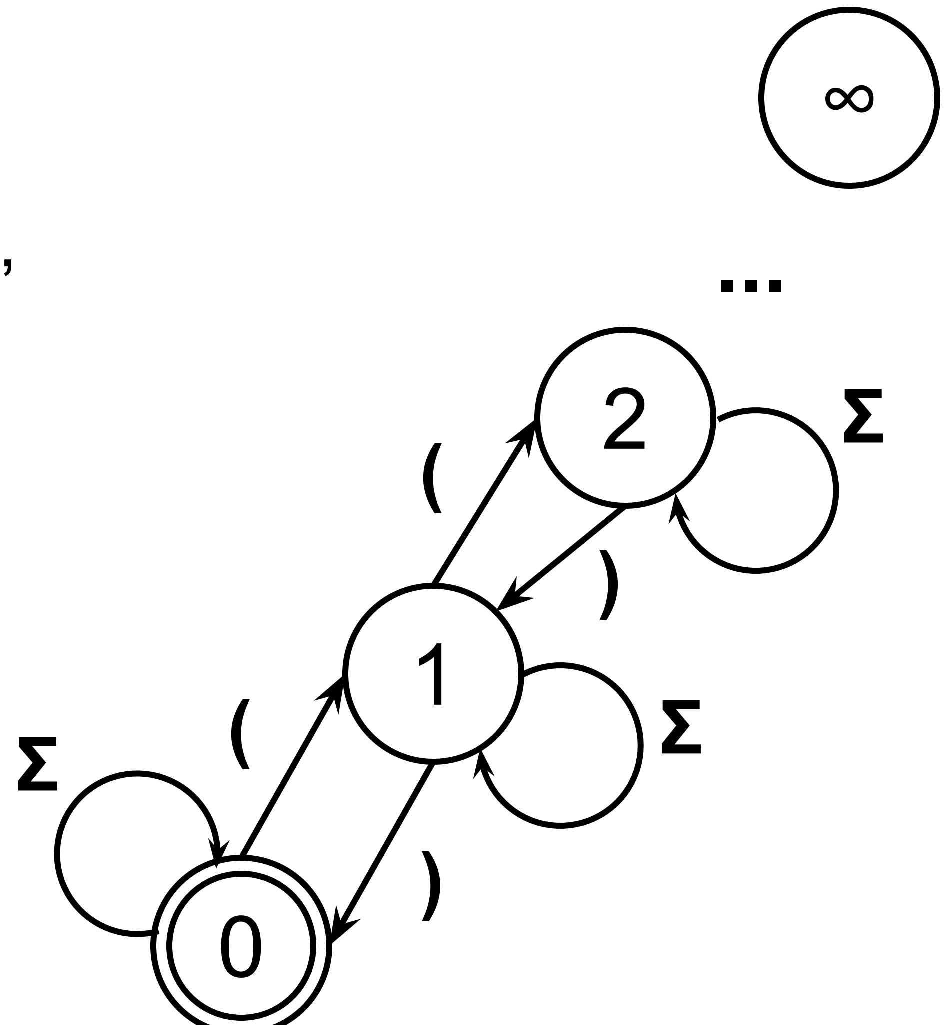
- Até agora vimos que: linguagens regulares são aquelas reconhecidas por autômatos finitos
  - Não foi feita nenhuma definição do que é uma linguagem regular
  - Um ser humano, ao olhar para uma linguagem, dificilmente consegue dizer se é ou não regular
  - Na verdade, não existe tal definição
- Mas existe uma distinção
  - Linguagens regulares vs não-regulares
  - A linha divisória é o fato de que
    - Autômatos finitos não conseguem contar

Linguagens não regulares

- Linguagens que exigem um contador
  - Ex: comentários dentro de comentários, escopos aninhados em uma linguagem, parêntesis aninhados, etc
  - Ex:
    - (1+2*(3-5+(7*7))-6)
    - É preciso contar quantos parêntesis são abertos e quantos são fechados
- Tente imaginar um autômato que reconheça tais cadeias
  - Estados são a “memória” do autômato

Linguagens não regulares

- Para reconhecer infinitos níveis de parêntesis aninhados, seriam necessários infinitos estados



Linguagens não regulares

- Outros exemplos:
  - $\{0^n1^n | n \geq 0\}$
  - {w|w tem número igual de 0s e 1s}
  - {ww| w seja uma cadeia sobre qualquer alfabeto}
- Mas veja esse exemplo:
  - {w|w tem um número igual de ocorrências de 01 e 10 como subcadeias}
  - Aparentemente, precisa contar
    - Mas essa linguagem é regular! (faça depois como exercício a prova, se duvidar)
- Formalmente:
  - Lema do bombeamento para linguagens regulares
  - Permite definir exatamente quais linguagens não são regulares

Lema do bombeamento para linguagens regulares

- Formalmente:
  - Se A é uma linguagem regular, então existe um número p (o comprimento de bombeamento) tal que, se s é qualquer cadeia de A de comprimento no mínimo p, então s pode ser dividida em três partes, s=xyz, satisfazendo as seguintes condições:
    - Para cada $i \geq 0$, $xy^iz \in A$
    - $|y| > 0$
    - $|xy| \leq p$
- Informalmente:
  - Toda cadeia da linguagem contém uma parte que pode ser repetida um número qualquer de vezes (bombeada), com a cadeia resultante permanecendo na linguagem

Lema do bombeamento para linguagens regulares

- Essa repetição ou bombeamento é a característica que faz com que seja sempre possível definir um número finito de estados para um autômato que reconheça a linguagem
- Uso do lema do bombeamento:
  - Provar que B não é regular
  - Contradição: suponha que B seja regular
  - 1. Encontre um p de forma que todas as cadeias de comprimento p ou maiores possam ser bombeadas
  - 2. Encontre uma cadeia s em B que tenha comprimento p ou mais, mas que não possa ser bombeada
  - 3. Demonstre que s não pode ser bombeada considerando todas as maneiras de dividir s em x,y e z, conforme o lema

Lema do bombeamento para linguagens regulares

- Ex: $\{0^n 1^n | n \geq 0\}$
  - 1. Seja p o comprimento de bombeamento
  - 2. Escolha $s = 0^p 1^p$
    - s é maior que p (conforme o lema)
    - Portanto, o lema diz que s pode ser dividida em 3 partes, $s=xyz$, onde para qualquer $i \geq 0$, xy^iz está em B
      - Ou seja, deve ser possível “bombear” y
  - 3. Mas é impossível!!
    - Primeira possibilidade: Suponha que y contém apenas 0s
      - Ex: s = 000111, x=0, y=00, z=111
      - Sempre que bombearmos y, teremos como resposta uma cadeia que não pertence à linguagem
      - Pois teremos como resultado mais 0s do que 1s

Lema do bombeamento para linguagens regulares

- Ex: $\{0^n1^n | n \geq 0\}$
  - 3. Mas é impossível!! (continuação)
    - Segunda possibilidade: Suponha que y contém apenas 1s
      - Ex: s = 000111, x=000, y=11, z=1
      - Sempre que bombearmos y, teremos como resposta uma cadeia que não pertence à linguagem
      - Pois teremos como resultado mais 1s do que 0s
    - Terceira possibilidade: y contém 0s e 1s
      - Ex: s = 000111, x=00, y=01, z=11
      - Sempre que bombearmos y, teremos como resposta uma cadeia que não pertence à linguagem
      - Pois teremos como resultado a presença de 0s e 1s alternados

Lema do bombeamento para linguagens regulares

- Ex: $\{0^n1^n | n \geq 0\}$
  - Ou seja, é impossível existir uma divisão de s de acordo com o lema do bombeamento
  - Isso é uma contradição!
- Ou seja, se não fizemos nada de errado, a suposição de que B é regular é falsa
  - Portanto, B não é regular

Lema do bombeamento para linguagens regulares

- O “truque” é encontrar o s
- Requer um pouco de pensamento criativo
- Tentativa e erro
- Busca pela “essência” da não-regularidade de B
- Conhecimento das restrições do lema
    - ($|y| > 0$, $|xy| \leq p$, etc)

Linguagens não regulares

- Ok, descobri que uma linguagem não é regular
  - Como resolver o problema?
  - Como obter uma implementação?
- Bom, se o problema é que um autômato finito não consegue contar...
  - ... basta adicionar um contador!
- Essa é exatamente a solução
  - Mais poder aos autômatos
  - Classe maior de linguagens
  - Mais detalhes nas próximas aulas!

Fim

Aula 11 - Linguagens não regulares