Decidibilidade III

Juliana Kaizer Vizzotto

Universidade Federal de Santa Maria

Disciplina de Teoria da Computação/Slides baseados no livro: **Introdução a Teoria da Computação. Michael Sipser**.

Roteiro

- Exemplos de problemas decidíveis

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- Podemos determinar se uma expressão regurlar gera uma dada cadeia. Seja

  $A_{EXR} = \{\langle R, w\rangle | R \text{ eh uma expressão regular que gera uma cadeia} w\}$

- **Teorema**: A_{EXR} é uma linguagem decidível.

- **Exercício**: Prove esse teorema.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- A prova do teorema anterior usa o seguinte teorema:
- **Teorema**: Toda expressão regular tem um autômato finito não-determinístico (AFN) equivalente.
- **Ideia da prova**: converta a expressão regular R para um AFN equivalente.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- Os Teoremas anteriores ilustram que, para os propósitos de decidibilidade, entregar à MT um AFD, AFN ou uma expressão regular, é tudo equivalente, pois a MT é capaz de converter uma forma de codificação na outra.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- Agora vamos considerar um tipo diferente de problema concernete a autômatos finitos: *testar vacuidade* para a linguame de autômato finito.

- Precisamos determinar se um autômato finito aceita *alguma* cadeia:

$$V_{AFD} = \{\langle A \rangle | A \text{ eh um AFD e } L(A) = \emptyset\}$$

- **Teorema**: V_{AFD} é uma linguagem decidível.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- **Prova**: Um AFD aceita alguma cadeia sse é possível atingir um estado de aceitação a partir do estado inicial passando pelas setas do AFD. Para testar essa condição, podemos projetar uma MT T, que usa um algoritmo de marcação similar aquele usado no exemplo do grafo direcionado conexo.
- T="Sobre a entrada $\langle A \rangle$, onde A é um AFD:
  1. Marque o estado inicial de A.
  2. Repita até que nenhum estado novo venha a ser marcado:
  3. Marque qualquer estado que tenha uma transição chegando nele a partir de qualquer estado que já está marcado.
  4. Se nenhum estado de aceitação estiver marcado, aceite; caso contrário, rejeite."

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- O próximo teorema afirma que determinar se dois AFDs reconhecem a mesma linguagem é decidível. Seja:

$$EQ_{AFD} = \{\langle A, B\rangle | A \text{ e } B \text{são AFDs e } L(A) = L(B)\}$$

- **Teorema**: EQ_{AFD} é uma linguagem decidível.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- **Prova**: Para provar esse teorema usamos o teorema anterior da vacuidade. Construimos um novo AFD C a partir de A e B, tal que C aceita aquelas cadeias que são aceitas ou por A ou por B, mas não por ambos. Consequentemente, Se A e B reconhecem a mesma linguagem, C não aceitará nada.

- A linguagem C é:

$$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- Essa expressão é chamada de diferença simétrica de $L(A)$ e $L(B)$.
- A diferença simétrica é interessante, pois $L(C) = \emptyset$ sse $L(A) = L(B)$.
- Podemos obter C a partir de A e B com as construções utilizadas para provar que a classe das linguagens regulares é fechada sob complemento, união e interseção. Essas construções são algoritmos que podem ser realizados por MTs.
- Uma vez tendo construído C, podemos usar o teorema da vacuidade para testar se $L(C)$ é vazia.
- Se ela for vazia, $L(A)$ e $L(B)$ têm de ser iguais.

Decidibilidade: problemas decidíveis concernentes a Linguagens Regulares

- $F =$ "Sobre a entrada $\langle A, B \rangle$, onde A e B são AFDs:
  1. Construa o AFD C.
  2. Rode a MT do Teorema anterior sobre a entrada $\langle C \rangle$.
  3. Se T aceita, aceite. Se T rejeita, rejeite.'

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- Considere o problema de se determinar se GLC gera uma cadeia específica. Seja:

$$A_{GLC} = \{\langle G, w\rangle | G \text{ é uma GLC que gera a cadeia } w\}.$$

- **Teorema**: A_{GLC} é uma linguagem decidível.

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- **Prova**: Para a GLC G e a cadeia w, queremos determinar se G gera w. Uma ideia é usar G para passar por todas as derivações para determinar se alguma deles é uma derivação de w.
- Essa ideia não funciona, pois uma quantidade infinita de derivações pode ter que ser testada.
- Se G não gera w, esse algoritmo nunca pararia.
- Essa ideia leva a uma máquina que é um reconhecedor, mas não um decisor para A_{GLC}.

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- Para tornar essa MT um decisor, precisamos garantir que o algoritmo tenta somente uma quantidade finita de derivações.
- Convertendo G para a forma normal de Chomsky, qualquer derivação de w teria $2n-1$ passos, onde n é o comprimento de w.
- Nesse caso, verificar apenas as derivações com $2n-1$ passos para determinar se G gera w seria suficiente.
- Existe somente uma quantidade finita de tais derivações.
- Exercício: estudo como converter G para a forma normal de Chomsky.

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- **Prova:** A MT S para A_{GLC}, segue:
- S="Sobre a entrada $\langle G, w\rangle$, onde G é uma GLC e w uma cadeia:
  1. Converta G para uma gramática equivalente na forma normal de Chomsky.
  2. Liste todas as derivações com $2n-1$ passos, onde n é o comprimento de w.
  3. Se alguma das derivações gera w, aceite; se não, rejeite."

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- O problema de se determinar se uma GLC gera uma cadeia específica está relacionado ao problema de compilar linguagens de programação.
- O algoritmo na MT S é muito ineficiente e nunca seria utilizado na prática, mas é fácil de descrever e no momento não estamos preocupados com eficiência.
- *Pesquise um algoritmo mais eficiente para reconhecer linguagens livres de contexto.*
- Equivalência entre GLCs e APs!

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- O problema de se testar vacuidade para uma linguagem de uma GLC:

$$V_{GLC} = \{\langle G\rangle | G \text{ é uma } GLC \text{ e } L(G) = \emptyset\}.$$

- **Teorema:** V_{GLC} é decidível.
- **Prova:** poderíamos usar a MT S do Teorema anterior. Ele afirma que podemos testar se uma GLC gera alguma cadeia específica w. Para testar se $L(G) = \emptyset$, o algoritmos poderia tentar passar por todos os ws possíveis. Mas existe uma quantidade infinita de ws para se testar.

Decidibilidade: problemas decidíveis concernentes a Linguagens Livres-do-Contexto

- **Prova:** para determinar se a linguagem de uma gramática é vazia, precisamos testar se a variável inicial pode gerar uma cadeia de terminais. O algoritmo faz isso resolvendo um problema mais geral. Ele determina, para cada variável, se ela é capaz de gerar uma cadeia de terminais.

- Quando o algoritmo tiver determinado que uma variável pode gerar uma cadeia de terminais, ele mantém registro dessa informação, colocando uma marca sob essa variável.