ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 20

Cap 5.1 – Problemas indecidíveis (parte 2)

Profa. Ariane Machado Lima
ariane.machado@usp.br

Na aula passada...

- Uma linguagem é Turing-decidível sse ela e seu complemento forem Turing-reconhecíveis
- Como provar que um problema B é indecidível usando a técnica de redutibilidade:
  - Assumo por contradição que B é decidível
  - Uso a MT decisora (R) de B para construir uma MT decisora (S) de um problema que sabemos que é indecidível (redução de A a B)
  - Contradição! Portanto R não pode existir!

Na aula de hoje

- Outros exemplos de provas de indecidibilidade por redutibilidade
- Reduções via histórias de computação

Exemplo da aula passada: Problema da Parada

- $PARA_{MT}$ = {<M,w> : M é uma MT e M pára sobre a entrada w}
- A_{MT} = {<M,w> : M é uma MT e M aceita w}

Ex: Problema da Parada

- Prova (tem que mostrar a redução!): assuma, por contradição, que uma MT R decida $PARA_{MT}$. Então construímos S que usa R para decidir A_{MT} :

S = “Sobre a entrada <M, w>, uma codificação de uma MT M e uma cadeia w:

1. Rode a MT R sobre a entrada <M, w>.
2. Se R rejeita, rejeite.
3. Se R aceita, simule M sobre w até ela pare.
2. Se M aceitou, aceite; se M rejeitou, rejeite.”

- Logo A_{MT} pode ser reduzido a $PARA_{MT}$
- Como A_{MT} é indecidível, $PARA_{MT}$ é indecidível

Vacuidade de uma linguagem de uma MT

- Como escrever isso em forma de linguagem?

Vacuidade de uma linguagem de uma MT

- V_{MT} = { <M>: M é uma MT e L(M) = Ø}
- Como podemos usar V_{MT} para resolver A_{MT}?

  A_{MT} = {<M,w> : M é uma MT e M aceita w}
- Se uma linguagem for vazia, ela não aceita w. Mas e se não for?
-

Vacuidade de uma linguagem de uma MT

- V_{MT} = { <M>: M é uma MT e L(M) = Ø}
- Como podemos usar V_{MT} para resolver A_{MT}?

  A_{MT} = {<M,w> : M é uma MT e M aceita w}
- Se uma linguagem for vazia, ela não aceita w. Mas e se não for?
- Ideia: construir uma versão de M que apenas teste w

M1 = “Sobre a entrada x:

1. Se x ≠ w *rejeite*
2. Se x = w, rode M sobre a entrada w e *aceite* se M aceita, e *rejeite* se M rejeita”

Vacuidade de uma linguagem de uma MT

- V_{MT} = { <M>: M é uma MT e L(M) = Ø}
- Como podemos usar V_{MT} para resolver A_{MT}?

  A_{MT} = {<M,w> : M é uma MT e M aceita w}

- Se uma linguagem for vazia, ela não aceita w. Mas e se não for?
- Ideia: construir uma versão de M que apenas teste w

M1 = “Sobre a entrada x:

1. Se x ≠ w *rejeite*
2. Se x = w, rode M sobre a entrada w e *aceite* se M aceita, e *rejeite* se M rejeita”

Ou seja, ou M1 aceita w ou não aceita nada

Vacuidade de uma linguagem de uma MT

- Suponha que R decide V_{MT}, vamos construir S que decide A_{MT}
- S = “Sobre a entrada <M,w>, uma codificação de uma MT M e uma cadeia w:

  1. Use a descrição de M e w para construir M1
  2. Rode R sobre M1
  3. Se R aceita, ? ; se R rejeita, ? ”

Vacuidade de uma linguagem de uma MT

- Suponha que R decide V_{MT}, vamos construir S que decide A_{MT}
- S = “Sobre a entrada <M,w>, uma codificação de uma MT M e uma cadeia w:
  1. Use a descrição de M e w para construir M1
  2. Rode R sobre M1
  3. Se R aceita, *rejeite*; se R rejeita, *aceite*.”

Mas como A_{MT} é indecidível, V_{MT} é indecidível

Classe da linguagem gerada por uma MT

- Dada um MT M, a linguagem gerada por ela poderia ser reconhecida por um modelo mais simples?
- Por ex: se a linguagem é regular
- Como escrever esse problema (identificar se a linguagem reconhecida por uma MT M é regular) em termos de linguagem?

Determinação de se a linguagem gerada por uma MT é regular

- $REGULAR_{MT}$ = { <M> | M é uma MT e L(M) é regular}

Determinação de se a linguagem gerada por uma MT é regular

- $REGULAR_{MT}$ = { <M> | M é uma MT e L(M) é regular}
- $REGULAR_{MT}$ é indecidível
- Ideia da Prova:

Determinação de se a linguagem gerada por uma MT é regular

- $REGULAR_{MT}$ = { <M> | M é uma MT e L(M) é regular}
- $REGULAR_{MT}$ é indecidível
- Ideia da Prova:
  - Supomos que existe uma MT R que decide $REGULAR_{MT}$ e usamos R em uma MT S para decidir A_{MT}
  - R deve analisar se uma MT M2 é regular, sendo que M2 reconhece uma linguagem regular (Σ*) sse M aceita w

Determinação de se a linguagem gerada por uma MT é regular

- S que decide A_{MT} usando R
- S = “Sobre a entrada <M,w>, onde M é uma MT e w é uma cadeia:

1. Construa a MT M2:

   M2 = “Sobre a entrada x:

   1. Se x tem a forma 0^n1^n, *aceite*
   2. senão, rode M sobre a entrada w e *aceite* se M aceita w, *rejeite* se M rejeita”

-

Determinação de se a linguagem gerada por uma MT é regular

- S que decide A_{MT} usando R
- S = “Sobre a entrada <M,w>, onde M é uma MT e w é uma cadeia:

1. Construa a MT M2:

Ou seja, M2 aceita qualquer x (Σ*, que é regular) se M aceitar w. Se M não aceitar w, então M2 só aceita 0^n1^n, que NÃO é regular

M2 = “Sobre a entrada x:

1. Se x tem a forma 0^n1^n, *aceite*
2. senão, rode M sobre a entrada w e *aceite* se M aceita w, *rejeite* se M rejeita”

-

Determinação de se a linguagem gerada por uma MT é regular

- S que decide A_{MT} usando R
- S = “Sobre a entrada <M,w>, onde M é uma MT e w é uma cadeia:

1. Construa a MT M2:

   M2 = “Sobre a entrada x:

   1. Se x tem a forma 0^n1^n, *aceite*
   2. senão, rode M sobre a entrada w e *aceite* se M aceita w, *rejeite* se M rejeita”

Ou seja, M2 aceita qualquer x (Σ*, que é regular) se M aceitar w. Se M não aceitar w, então M2 só aceita 0^n1^n, que NÃO é regular

2. Rode R sobre a entrada <M2>
3. Se R aceita, ; se R rejeita, ”

-

Determinação de se a linguagem gerada por uma MT é regular

- S que decide A_{MT} usando R
- S = “Sobre a entrada <M,w>, onde M é uma MT e w é uma cadeia:

1. Construa a MT M2:

Ou seja, M2 aceita qualquer x (Σ*, que é regular) se M aceitar w. Se M não aceitar w, então M2 só aceita 0^n1^n, que NÃO é regular

M2 = “Sobre a entrada x:

1. Se x tem a forma 0^n1^n, *aceite*
2. senão, rode M sobre a entrada w e *aceite* se M aceita w, *rejeite* se M rejeita”

M2 é regular → M aceita w

2. Rode R sobre a entrada <M2>

M2 não é regular → M não aceita w

3. Se R aceita, *aceite*; se R rejeita, *rejeite*”

-

Determinação de se a linguagem gerada por uma MT é regular

- S que decide A_{MT} usando R
- S = “Sobre a entrada <M,w>, onde M é uma MT e w é uma cadeia:

1. Construa a MT M2:

    M2 = “Sobre a entrada x:

    1. Se x tem a forma 0^n1^n, *aceite*
    2. senão, rode M sobre a entrada w e *aceite* se M aceita w, *rejeite* se M rejeita”

Ou seja, M2 aceita qualquer x (Σ*, que é regular) se M aceitar w. Se M não aceitar w, então M2 só aceita 0^n1^n, que NÃO é regular

2. Rode R sobre a entrada <M2>

M2 é regular → M aceita w

M2 não é regular → M não aceita w

3. Se R aceita, *aceite*; se R rejeita, *rejeite*”

- Ops, com R eu decidiria A_{MT}!!! Mas A_{MT} é indecidível, então $REGULAR_{MT}$ também é

Determinação de propriedades da linguagem gerada por uma MT

- Da mesma forma, os seguintes problemas são indecidíveis (para uma dada MT M)
  - Determinar se L(M) é livre-de-contexto
  - Determinar se L(M) é sensível ao contexto
  - Determinar se L(M) é decidível (recursiva)
  - ...
  - Na verdade, determinar qualquer propriedade de L(M) (Teorema de Rice)