Teoria da Computação

Problemas Indecidíveis

Thiago Alves

Problema da Parada

- H = {“M”#”w” | M pára com w}
- Uma máquina de Turing M_H para essa linguagem deve receber uma outra máquina M e uma entrada w para M
- Será que é possível criar um programa que diz se outro programa vai parar?

Problema da Parada

- H = {“M”#“w” | M pára com w}
- Será que existe M_H que reconhece H?

Problema da Parada

- H = {“M”#“w” | M pára com w}
- Será que existe M_H que reconhece H?
- M_H pode ser a máquina de Turing universal
- Note que a máquina de Turing universal pode não parar
  - Basta M não parar com w

Problema da Parada

- H = {“M”#“w” | M pára com w}
- M_H pode ser a máquina de Turing universal
  - Pode não parar
- A linguagem H é r.e.
- Será que é recursiva?

Problema da Parada

- H não é recursiva
  - Prova: Suponha que existe uma máquina de Turing M_H que decide H
  - Vamos construir uma outra máquina de Turing M_D que tem como entrada máquinas de Turing "M"
    - M_D simula M_H com "M"#"M"
    - Se M_H aceita, fazemos M_D entrar em loop
    - Se M_H rejeita, M_D aceita

Problema da Parada

M_D(“M”)

se M_H(“M”#“M”) aceita

forçar entrar em loop

senão

rejeita

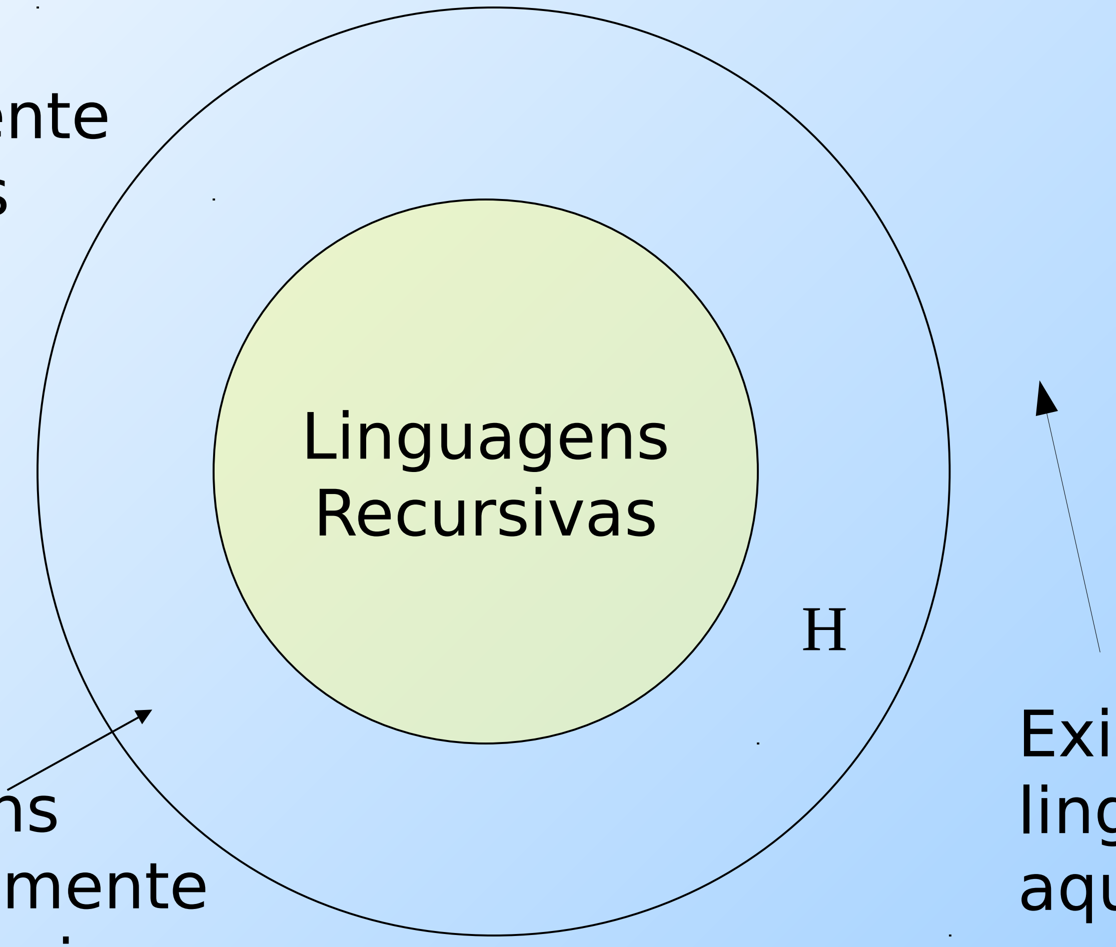
Problema da Parada

- Agora vamos executar M_D com "M_D"
- Suponha que M_D pára com "M_D"
  - M_H rejeita "M_D"#"M_D"
    - M_D não pára com "M_D"
- Suponha que M_D não pára com "M_D"
  - M_H aceita "M_D"#"M_D"
    - M_D pára com "M_D"
- Absurdo!!

Problema da Parada

- Logo, o problema da parada representado pela linguagem H = {“M”#“w” | M pára com w} não é recursivo
- H não é decidida por nenhuma máquina de Turing
- Isso significa que não existe algoritmo para este problema

Visão Geral


Linguagens
Não
Recursivamente
Enumeráveis
Linguagens
Recursivas
H
Linguagens
Recursivamente
Enumeráveis
Existem
linguagens
aqui fora?

Linguagem Não R.E.

- Agora queremos uma linguagem que não tenha máquina de Turing qualquer
- Mesmo que seja uma máquina que não pára para algumas entradas

Problema do Loop

- Loop = {“M”#“w” | M não pára com w}
- Saber se uma máquina de entrada M com entrada w não pára

Problema do Loop

- Loop não é r.e.
- Prova:
  - Suponha que Loop é r.e.
  - Seja M_1 uma máquina que reconhece Loop
  - Seja M_2 uma máquina que reconhece H
  - Podemos construir uma máquina M_H que decide H

Problema do Loop

- Loop não é r.e.
- Prova:
  - Podemos construir uma máquina M_H que decide H
  - M_H simula M_1 e M_2 com entrada w alternadamente
  - Se M_1 aceitar, M_H aceita
  - Se M_2 aceitar, M_H rejeita

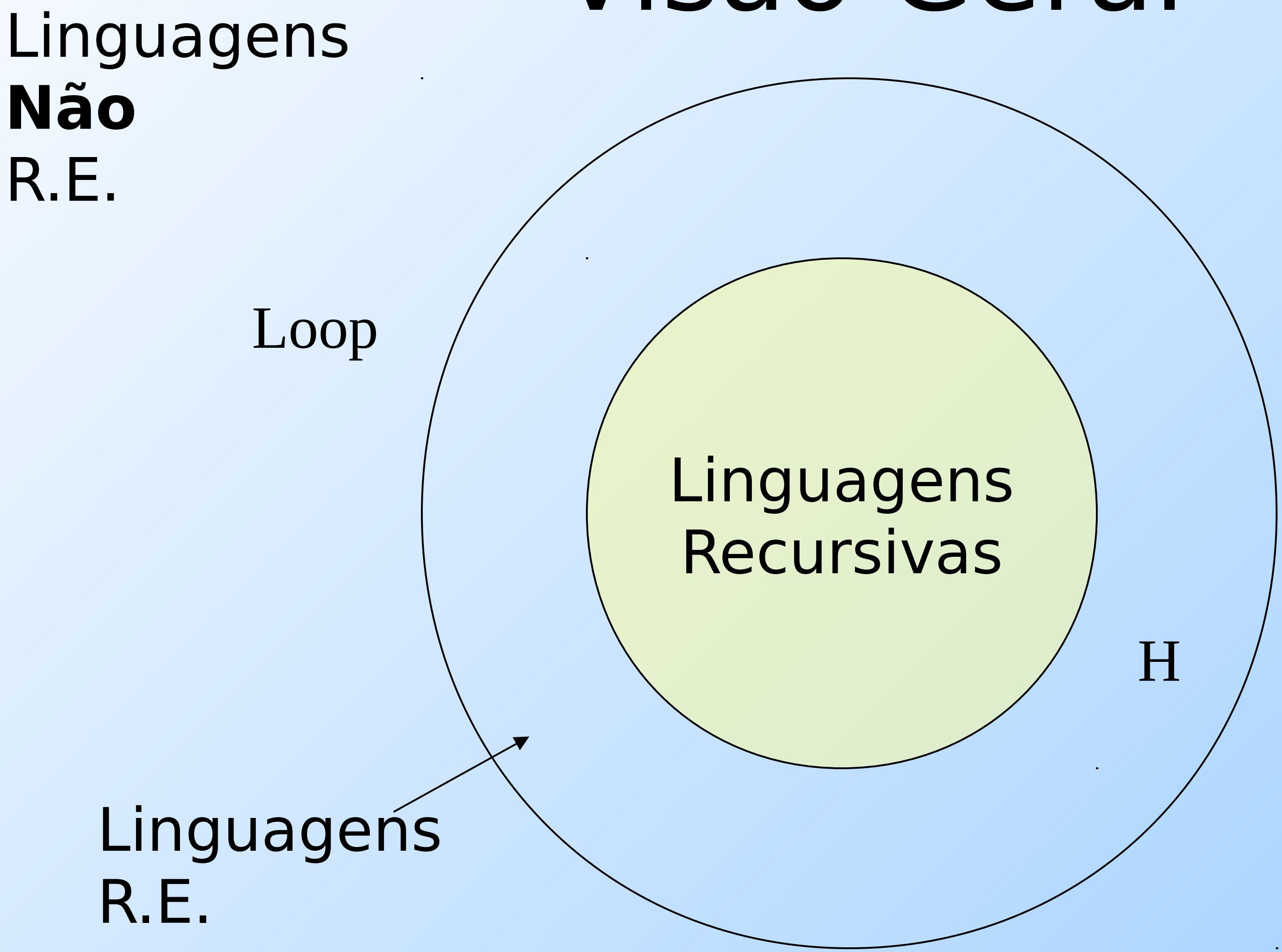
Problema do Loop

- Loop não é r.e.
- Prova:
  - Para toda string w, $w \in H$ ou $w \in Loop$
  - Para toda string w, M_1 aceita w ou M_2 aceita w
  - M_H pára pois pára quando M_1 aceita ou quando M_2 aceita
  - M_H decide H

Problema do Loop

- Loop não é r.e.
- Prova:
  - M_H decide H
  - Absurdo! Pois mostramos que H não é decidível!
  - Logo, Loop não é reconhecida por nenhuma máquina de Turing

Visão Geral


Linguagens
Não
R.E.
Loop
Linguagens
Recursivas
H
Linguagens
R.E.

Linguagem e Complemento

- A prova de que Loop não é r.e. indica que
  - se uma linguagem L é r.e e não é recursiva, o seu complemento não é r.e.
- A intuição é que existiria uma máquina para decidir a linguagem L

Linguagem e Complemento

- Teorema: Uma linguagem L é recursiva se e somente se L é r.e. e $\overline{L}$ é r.e.

Linguagem e Complemento

- Teorema: Uma linguagem L é recursiva se e somente se L é r.e. e $\overline{L}$ é r.e.
- Prova (ida):
  - Suponha L recursiva
  - $\overline{L}$ é recursiva pois basta trocar os estados q_a e q_r nas transições da máquina de L
  - Se são recursivas são r.e.

Linguagem e Complemento

- Prova (volta):
  - Suponha L r.e. e $\overline{L}$ r.e.
  - Temos M_1 que reconhece L e M_2 que reconhece $\overline{L}$
  - Podemos construir uma M_L para decidir L simulando M_1 e M_2 alternadamente
  - Logo, L é recursiva.

Exercícios

- Mostre que a linguagem

  A ={“M”#“w” | M aceita w} é r.e. mas não é recursiva.

Exercícios

- Mostre que a linguagem

  A ={“M”#“w” | M aceita w} é r.e. mas não é recursiva.
- Para mostrar que A é r.e. basta usar a máquina universal M_H

Exercícios

- Mostre que a linguagem

  A ={“M”#“w” | M aceita w} não é recursiva
- Prova:
  - Seja M_A para decidir A
  - Vamos construir M_D:
    - Com entrada “M” simula M_A com “M”#“M”
    - M_D aceita se e somente se M_A rejeita

Exercícios

- Prova:
  - Seja M_A para decidir A
  - Vamos construir M_D:
    - Com entrada “M” simula M_A com “M”#“M”
    - M_D aceita se e somente se M_A rejeita
  - M_D com “M” aceita, se M não aceita “M”
  - M_D com “M” rejeita, se M aceita “M”

Exercícios

M_D(“M”)

se M_A(“M”#“M”) aceita

rejeita

senão

aceita

Exercícios

- Prova:
  - Seja M_A para decidir A
  - Vamos executar M_D com "M_D":
    - M_D aceita "M_D", se M_D rejeita "M_D"
    - M_D rejeita "M_D", se M_D aceita "M_D"
  - Absurdo!
  - A não é recursiva!