Redução de A para B para prova de indecidibilidade

A: problema indecidível cuja indecibilidade já foi provada
B: problema que eu quero provar que é indecidível

Prova por contradição:
- Supor que B é decidível e que a mT R decide B
- Construir uma mT S que decide A usando a mt R
- Como A é indecidível, chegamos a uma contradição, então B é indecidível

A escrita da mT S (usando R) é a redução

Problemas indecidíveis

$$\mathcal{L}_{MT_{reg}} = \{\langle M \rangle \mid M \text{ é uma MT e } L(M) \text{ é uma linguagem regular}\}.$$

- A linguagem reconhecida por uma máquina de Turing também pode ser reconhecida por um dispositivo computacional mais simples?
- $\mathcal{L}_{MT_{reg}} \equiv$ Uma certa máquina de Turing é equivalente a algum autômato finito?

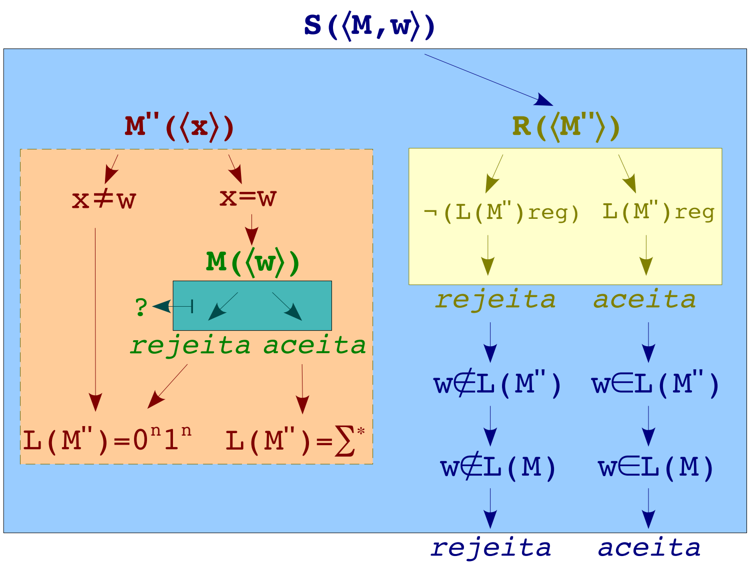
Redução de LMT para LMT (reg)

Problemas indecidíveis

$\mathcal{L}_{MT_{reg}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é uma linguagem regular$\}$.

Teorema 3.20

A linguagem $\mathcal{L}_{MT_{reg}}$ não é decidível.



Problemas indecidíveis

Esquema da Prova

- Supor que $\mathcal{L}_{MT_{reg}}$ é decidível e mostrar que a linguagem $\mathcal{L}_{MT}$ é decidível (contradição).
- Supor que máquina de Turing R decide $\mathcal{L}_{MT_{reg}}$ e construir máquina de Turing S que decide $\mathcal{L}_{MT}$.
  - Como S pode usar R como subrotina???

Problemas indecidíveis

Esquema da Prova

- Como S pode usar R como subrotina???
  - S recebe $\langle M, w\rangle$ e modifica M tal que a máquina M'' resultante reconheça uma linguagem regular se e somente se M aceita w.
  - Se M não aceita w, M'' reconhece a linguagem $\{0^n1^n \mid n \geq 0\}$.
  - Se M aceita w, M'' reconhece a linguagem regular Σ^*.

Problemas indecidíveis

$\mathcal{L}_{MT_{reg}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é uma linguagem regular$\}$.

Teorema 3.20

A linguagem $\mathcal{L}_{MT_{reg}}$ não é decidível.

Demonstração.

- Funcionamento de M'' com a cadeia x:
  1. Se x é da forma 0^n1^n, M'' *aceita*.
  2. Caso contrário, M'' chama M com entrada w.
  3. Se M aceita, M'' *aceita*.

M'' funciona aceitando automaticamente todas as cadeias de 0n1n.
Adicionalmente se M aceita w, M'' aceita todas as outras cadeias.

□

Problemas indecidíveis

$\mathcal{L}_{MT_{reg}} = \{\langle M\rangle \mid M$ é uma MT e $L(M)$ é uma linguagem regular$\}$.

Teorema 3.20
A linguagem $\mathcal{L}_{MT_{reg}}$ não é decidível.

Demonstração.

- Supor que máquina de Turing R decide $\mathcal{L}_{MT_{reg}}$.
- Construir máquina de Turing S que decide $\mathcal{L}_{MT}$.
- Funcionamento de S com a entrada $\langle M, w\rangle$:
  1. Usar a descrição de M e w para construir a máquina M''.
  2. Chamar R com entrada $\langle M''\rangle$.
  3. Se R aceita, S *aceita*. Se R rejeita, S *rejeita*.
- Se R decidisse $\mathcal{L}_{MT_{reg}}$, S decidiria $\mathcal{L}_{MT}$!!!

$\square$

Redução a partir de Lhalt ={<M,w> I mT M pára com a cadeia w}.

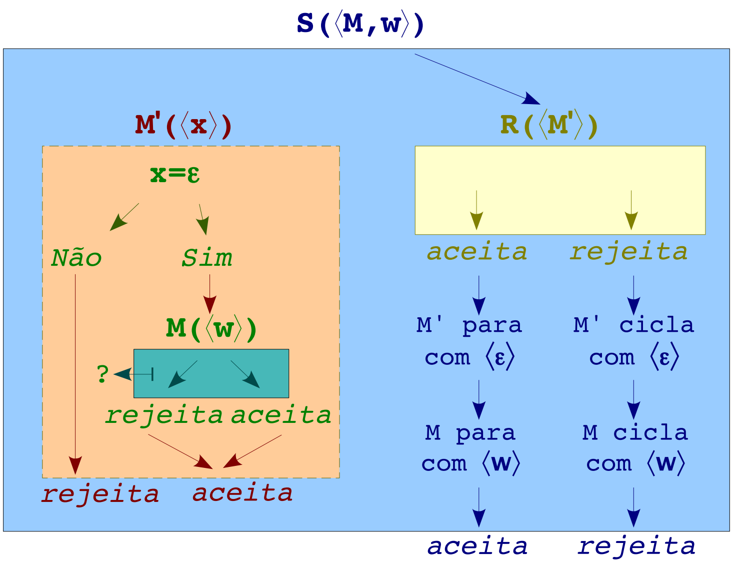
Problemas indecidíveis

Supor que LMTe é decidida por R e construir S para decidir Lhalt

$\mathcal{L}_{MT_\varepsilon}$ = {⟨M⟩ | M é uma MT e $M(\langle\varepsilon\rangle)$ pára}.

Teorema 3.21

A linguagem $\mathcal{L}_{MT_\varepsilon}$ não é decidível.



Construir M’ com base em M e w de forma que L(M’) = e sse M pára com w,

isto é,

M’ pára com e sse M pára com w.

\

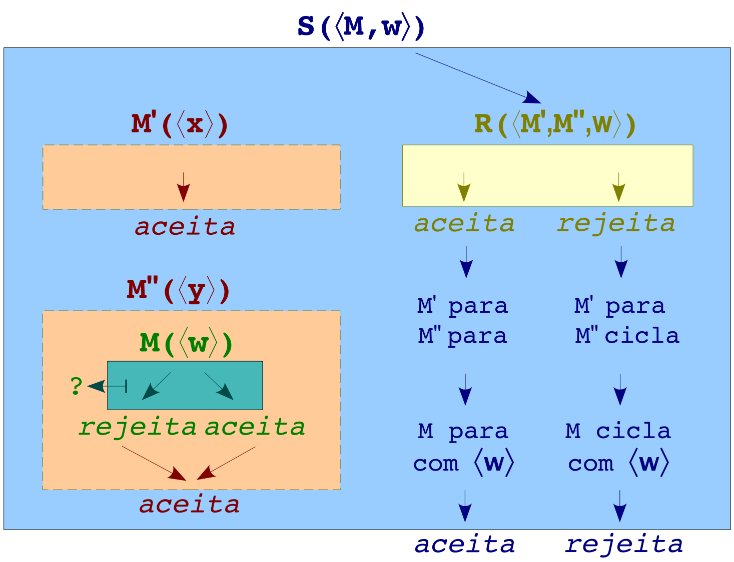
Redução a partir de Lhalt ={<M,w> I mT M pára com a cadeia w}.

Problemas indecidíveis

$\mathcal{L}_{2MTs} = \{\langle M_1, M_2, w\rangle \mid M_1$ e M_2 são MT's e $M_1(\langle w\rangle) = M_2(\langle w\rangle)\}$.

Teorema 3.23

A linguagem $\mathcal{L}_{2MTs}$ não é decidível.



Problemas indecidíveis

Exercícios:
Provar que as seguintes linguagens são indecidíveis
através de redução

- $\mathcal{L}_{MT_{llc}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é uma $LLC\}$.
- $\mathcal{L}_{MT_{dec}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é decidível$\}$.
- $\mathcal{L}_{MT_{fin}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é uma linguagem finita$\}$.
- $\mathcal{L}_{MT_{\square}} = \{\langle M \rangle \mid M$ é uma MT e $L(M)$ é $\square\ \}$.

Problemas indecidíveis

$\mathcal{L}_{MT_=} = \{\langle M_1, M_2 \rangle \mid M_1$ e M_2 são MT's e $L(M_1) = L(M_2)\}$.

Teorema 3.27

A linguagem $\mathcal{L}_{MT_=}$ não é decidível.

- É possível provar este teorema fazendo a redução de $\mathcal{L}_{MT}$ para $\mathcal{L}_{MT_=}$.
- Pode-se reduzir qualquer outra linguagem não decidível para $\mathcal{L}_{MT_=}$. Por exemplo, $\mathcal{L}_{MT_\emptyset}$.

Problemas indecidíveis

$\mathcal{L}_{MT_=} = \{\langle M_1, M_2\rangle \mid M_1$ e M_2 são MT's e $L(M_1) = L(M_2)\}$.

Teorema 3.28

A linguagem $\mathcal{L}_{MT_=}$ não é decidível.

Esquema da Prova

- Mostrar que se $\mathcal{L}_{MT_=}$ é decidível, então $\mathcal{L}_{MT_\emptyset}$ também é decidível.
- Reduzir $\mathcal{L}_{MT_\emptyset}$ para $\mathcal{L}_{MT_=}$:
  - $\mathcal{L}_{MT_\emptyset}$: a linguagem de uma MT é vazia?
  - $\mathcal{L}_{MT_=}$: as linguagens de duas MT's são iguais?
  - Se uma das linguagens for vazia, testar se a segunda é vazia.
  - $\mathcal{L}_{MT_\emptyset}$ é um caso especial de $\mathcal{L}_{MT_=}$, onde uma das máquinas fica fixa para reconhecer a linguagem vazia.

Problemas indecidíveis

$$\mathcal{L}_{MT_=} = \{\langle M_1, M_2\rangle \mid M_1 \text{ e } M_2 \text{ são MT's e } L(M_1) = L(M_2)\}.$$

Teorema 3.28

A linguagem $\mathcal{L}_{MT_=}$ não é decidível.

Demonstração.

- Supor que máquina de Turing R decide $\mathcal{L}_{MT_=}$.
- Construir máquina de Turing S que decide $\mathcal{L}_{MT_\emptyset}$.
- Funcionamento de S com a entrada $\langle M\rangle$:
  1. Chamar R com entrada $\langle M, M'\rangle$, onde M' é uma máquina de Turing que rejeita todas as entradas.
  2. Se R aceita, S *aceita*. Se R rejeita, S *rejeita*.
- Se R decide $\mathcal{L}_{MT_=}$, S decide $\mathcal{L}_{MT_\emptyset}$.
- Pelo Teorema 3.19, $\mathcal{L}_{MT_\emptyset}$ não é decidível. Logo, $\mathcal{L}_{MT_=}$ também não pode ser decidível. □