Propriedades da Solucionabilidade

Teoria da Computação

INF05501

Estudo da Solucionabilidade

- O Universo de Todos os Problemas é dividido em:
  - Solucionáveis
  - Não-Solucionáveis

- Problemas Não-Solucionáveis podem ser:
  - Parcialmente Solucionáveis
  - Totalmente Insolúveis

Estudo da Solucionabilidade (cont.)

- Avalia-se a solucionabilidade de um problema através da **análise da linguagem que o traduz** através de uma codificação bijetora

- Desta forma, transforma-se a questão em um **problema de decisão**

- Com isto, classificam-se os problemas em decidíveis e não-decidíveis, os quais podem ser semidecidíveis

Propriedades da Solucionabilidade

- Como a questão da solucionabilidade está relacionada à linguagem correspondente ao problema, sabe-se que:
  - Um problema é solucionável se a sua linguagem correspondente é recursiva
  - Um problema é não-solucionável se a linguagem correspondente é não-recursiva
  - Um problema é parcialmente solucionável se a linguagem correspondente é enumerável recursivamente
  - Um problema é totalmente insolúvel se a linguagem correspondente é não enumerável recursivamente

Propriedades da Solucionabilidade (cont.)

- Em relação à linguagens, vimos seguintes **teoremas**:
  - **Teorema 1:** O complemento de uma linguagem recursiva é uma linguagem recursiva
  - **Teorema 2:** Uma linguagem é recursiva sss ela e seu complemento são enumeráveis recursivamente

Propriedades da Solucionabilidade (cont.)

- Aplicando-se os teoremas a problemas:
  - O complemento de um problema solucionável é um problema solucionável
  - Um problema é solucionável sss ele e seu complemento são parcialmente solucionáveis

Propriedades da Solucionabilidade (cont.)

- Com base nisto, podemos saber, por exemplo, que
  - O Problema da Parada é parcialmente solucionável
  - O Problema da Parada é não-solucionável
  - Logo, o Problema da Não-Parada (Vacuidade) é não-solucionável

Propriedades da Solucionabilidade (cont.)

- Com base nisto, podemos saber, por exemplo, que
  - O Problema da Parada é parcialmente solucionável
  - O Problema da Parada é não-solucionável
  - Logo, o Problema da Não-Parada (Vacuidade) é não solucionável
- Consequência direta: **não existe um algoritmo genérico para detecção de loops infinitos em programas**

Teorema 1

O complemento de um problema solucionável é um problema solucionável

Portanto, se uma linguagem L sobre um alfabeto Σ é recursiva, então o seu complemento $\Sigma^* - L$ também é uma linguagem recursiva

Prova do Teorema 1

Assumindo-se que L seja recursiva, então existe um Máquina Universal M que aceita a linguagem e sempre para para qualquer entrada, tal que:

$ACEITA(M) = L$

$REJEITA(M) = \Sigma^* - L$

$LOOP(M) = \emptyset$

Prova do Teorema 1 (cont.)

Seja M' uma Máquina Universal construída a partir de M, invertendo-se as condições de $ACEITA$ e $REJEITA$, tal que:

$ACEITA(M') = \Sigma^* - L$

$REJEITA(M') = L$

$LOOP(M') = \emptyset$

Prova do Teorema 1 (cont.)

Seja M' uma Máquina Universal construída a partir de M, invertendo-se as condições de $ACEITA$ e $REJEITA$, tal que

$ACEITA(M') = \Sigma^* - L$

$REJEITA(M') = L$

$LOOP(M') = \emptyset$

Claramente, M' também sempre para para qualquer entrada e, por consequência, $\Sigma^* - L$ é recursiva

Teorema 2

Um problema é solucionável sss ele e seu complemento são parcialmente solucionáveis

Portanto, uma linguagem L sobre um alfabeto Σ é recursiva sss L e $\Sigma^* - L$ são enumeráveis recursivamente

Prova do Teorema 2

Provamos que, **se L é recursiva, então L e $\Sigma^* - L$ são enumeráveis recursivamente**

Prova do Teorema 2

Provamos que, **se L é recursiva, então L e $\Sigma^* - L$ são enumeráveis recursivamente**

Suponha uma linguagem recursiva L sobre um alfabeto Σ

Prova do Teorema 2

Provamos que, **se L é recursiva, então L e $\Sigma^* - L$ são enumeráveis recursivamente**

Suponha uma linguagem recursiva L sobre um alfabeto Σ

Pelo Teorema 1, necessariamente, $\Sigma^* - L$ é recursiva

Prova do Teorema 2

Provamos que, **se L é recursiva, então L e $\Sigma^* - L$ são enumeráveis recursivamente**

Suponha uma linguagem recursiva L sobre um alfabeto Σ

Pelo Teorema 1, necessariamente, $\Sigma^* - L$ é recursiva

Como, por definição, toda linguagem recursiva é também enumerável recursivamente, então L e $\Sigma^* - L$ são enumeráveis recursivamente

Prova do Teorema 2 (cont.)

Provamos agora que, **se L e $\Sigma^* - L$ são enumeráveis recursivamente, então L é recursiva**

Prova do Teorema 2 (cont.)

Provamos agora que, **se L e $\Sigma^* - L$ são enumeráveis recursivamente, então L é recursiva**

Suponha-se uma linguagem L sobre um alfabeto Σ tal que L e $\Sigma^* - L$ são enumeráveis recursivamente

Prova do Teorema 2 (cont.)

Provamos agora que, **se L e $\Sigma^* - L$ são enumeráveis recursivamente, então L é recursiva**

Suponha-se uma linguagem L sobre um alfabeto Σ tal que L e $\Sigma^* - L$ são enumeráveis recursivamente

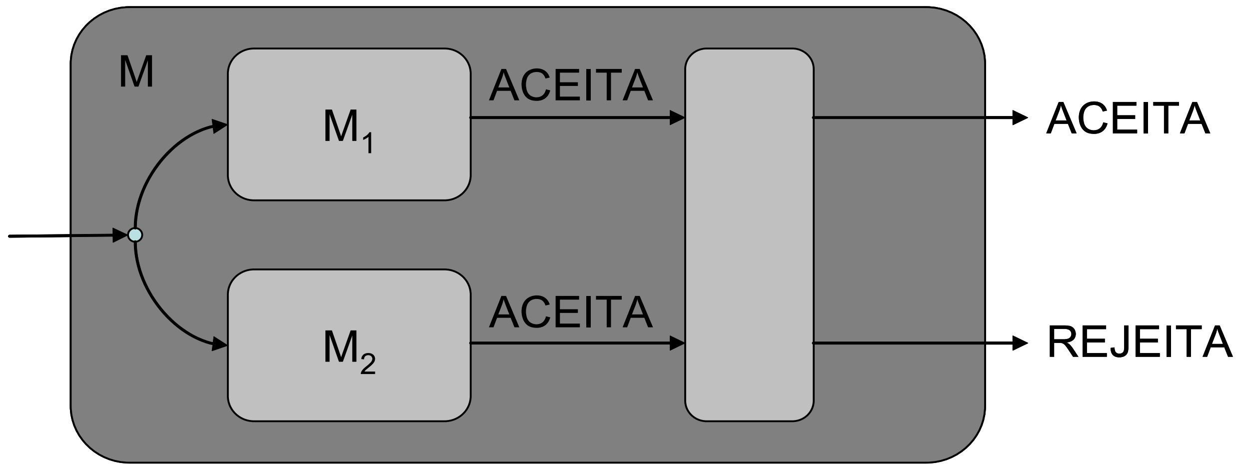
Neste caso, devem existir duas Máquinas Universais M_1 e M_2 tais que:

$ACEITA(M_1) = L$

$ACEITA(M_2) = \Sigma^* - L$

Prova do Teorema 2 (cont.)

Utilizando M_1 e M_2, podemos construir uma Máquina Universal Não-Determinística M da seguinte maneira:



Prova do Teorema 2 (cont.)

Logo, para qualquer palavra de entrada sobre Σ, M **aceita a entrada se** M_1 **aceitá-la e** M **rejeita a entrada se** M_2 **aceitá-la**

Prova do Teorema 2 (cont.)

Logo, para qualquer palavra de entrada sobre Σ, M **aceita a entrada se** M_1 **aceitá-la e** M **rejeita a entrada se** M_2 **aceitá-la**

Portanto, M sempre para e L é recursiva

Resumo

- Alfabetos e Linguagens
- Programas
- Máquinas
- Computações e Funções Computadas
- Equivalência de Programas e Máquinas
- Máquinas Universais e Hipótese de Church
- Solucionabilidade de Problemas