

ITC: Introdução à Teoria da Computação

Marcos Castilho

DInf/UFPR

19 de julho de 2021

Motivação

Quais são os limites da computação?

O que é um Problema de decisão?

Um problema de decisão é um conjunto de perguntas, cada uma das quais tem um SIM ou um NÃO como resposta.

Exemplo

Seja o problema PSQ de determinar quando um número natural arbitrário é um quadrado perfeito.

Este problema consiste das seguintes questões:

- ▶ P_0 : 0 é um quadrado perfeito?
- ▶ P_1 : 1 é um quadrado perfeito?
- ▶ P_2 : 2 é um quadrado perfeito?
- ▶ ...

Solução

Uma solução para um problema de decisão P é um algoritmo que determina a resposta apropriada para cada questão p em P .

O que é um Algoritmo?

Não existe uma definição precisa para isto!

Podemos dizer que um algoritmo deve ser:

- ▶ Completo
- ▶ Mecanizável
- ▶ Determinístico

Um procedimento que atende estas propriedades é chamado de *efetivo*.

Máquinas de Turing

As computações de uma MT padrão são claramente mecanizáveis e determinísticas. Uma MT que para para toda entrada também é completa.

MT's como framework

- ▶ As MT's provêem um framework formal que podem ser usadas na construção de soluções para problemas de decisão.
- ▶ Um problema é respondido afirmativamente se a entrada é aceita pela MT e negativamente se for rejeitada.

Exemplo

- ▶ Seja o problema de colocar moedas para liberar o jornal na máquina de jornais: 30 centavos em moedas de 5, 10 ou 25 liberam o jornal.
- ▶ Se mais de 30 centavos for colocado, a máquina não devolve o troco.
- ▶ Agora considere um “pão-duro” que se recusa a perder dinheiro, ele quer colocar o valor exato.

Problema associado

É determinar quando um conjunto de moedas contém uma combinação cujo valor total seja exatamente 30 centavos.

Problema de decisão associado

É a transformação do problema de uma forma que seja formulado como um problema de aceitação de uma string.

Exemplo

- ▶ Seja o conjunto contendo c , d , v como conjunto de entrada, onde c : cinco, d : dez, v : vinte e cinco.
- ▶ Basta construir uma máquina de Turing que para quando uma dada string totaliza 30 centavos.
- ▶ Esta máquina não é difícil de construir.

Considerações

- ▶ A escolha da MT padrão (determinística) juntamente com a propriedade de completude, forçam que a computação termine para toda string.
- ▶ Assim, a linguagem aceita pela MT que resolve o problema de decisão é recursiva.

Considerações

- ▶ Por outro lado, toda MT determinística que aceita uma linguagem recursiva pode ser considerada como uma solução para um problema de decisão.
- ▶ A string w pertence a $L(M)$?

Decidibilidade

É o estudo dos problemas sobre os quais se pode decidir ou não.

Decidibilidade

- ▶ A relação entre problemas de decisão solúveis e as linguagens recursivas são exploradas para o estabelecimento da questão de decidibilidade de problemas de decisão.
- ▶ Um problema é decidível se ele tem uma representação na qual o conjunto de entradas aceitas formam uma linguagem recursiva.

Decidibilidade

- ▶ As máquinas de Turing padrão são suficientes!
- ▶ As outras MT's alternativas são equivalentes.
- ▶ É claro que para o estabelecimento da solução, as MT's alternativas facilitam o trabalho.

Soluções parciais

- ▶ Uma solução parcial para um problema de decisão P é um procedimento não necessariamente efetivo completo que retorna uma resposta SIM ou NÃO:
- ▶ Se a resposta for um SIM a MT para e aceita.
- ▶ Mas se a resposta for NÃO, a MT pode parar e não aceitar a entrada, mas também pode não terminar...

A tese de Church-Turing

- ▶ As máquinas de Turing foram introduzidas para processar entrada e gerar uma saída;
- ▶ Mas também podem servir para reconhecer strings;
- ▶ Vamos ver como podem ser usadas para problemas de decisão.

Decidibilidade

Preocupação com computação efetiva

- ▶ execução mecânica
- ▶ produção da mesma saída para as mesmas entradas
- ▶ execução em tempo finito.

Formalismos propostos

- ▶ máquinas de Turing
- ▶ sistemas de Post
- ▶ funções recursivas
- ▶ lambda calculus

Decidibilidade

- ▶ Computadores tem o mesmo poder computacional que uma MT?
- ▶ Linguagens como C e Java servem para facilitar a programação. mas não tem mais expressividade que uma máquina de Turing...
- ▶ Uma MT captura a parte essencial, aquela que em última instancia é usada para medir o poder computacional.

A tese de Church-Turing

Se uma função é efetivamente computável, então ela é computável por meio de uma MT.

- ▶ Todo algoritmo pode se expresso mediante uma MT.
- ▶ “Computação efetiva” não é definida formalmente.
- ▶ Evidência: nenhum formalismo “moderno” superou ainda a MT.

A tese estendida de Church-Turing

Um problema de decisão P é parcialmente solúvel se, e somente se, existe uma MT que aceita precisamente os elementos de P cuja resposta é SIM.

Computação algorítmica

- ▶ A tese de Church-Turing pode ser considerada para prover uma definição de computação algorítmica.
- ▶ A robustez da MT dá indícios que a tese está correta.
- ▶ Outra evidência é que até hoje sempre se conseguiu implementar MT's que resolvem problemas de decisão.

Decidibilidade

- ▶ Caso dos problemas de decisão: se existe solução para um, então existe uma MT que o resolve.
- ▶ Para mostrar que não existe solução para um PD então basta mostrar que não existe MT para ele.

Indecidibilidade

Se, para um problema, não existe uma MT, o problema é dito indecidível.

Decidibilidade

- ▶ Interessante: Uma MT pode receber como entrada uma outra MT!
- ▶ Ou: um programa em C pode receber um outro programa em C como dado!
- ▶ Conceito: maquinas de Turing universais: uma MT que simula outra MT dada como entrada.

Há limites para a computação algorítmica?

- ▶ Esta é a questão de um milhão de dólares!
- ▶ Notem que a questão é relativa aos problemas, não aos limites de memória ou processador das máquinas atuais!!!
- ▶ Mas, se não pode ser resolvida por uma MT, evidentemente não pode ser resolvida por uma máquina real.

Máquina de Turing Universal

- ▶ Construir MTs capazes de simular qualquer outra MT!
- ▶ Como representar uma MT como entrada para a MTU?
- ▶ Existem várias maneiras, representando estados e strings, usando-se várias fitas, etc.

Entscheidungsproblem

- ▶ Entscheidungsproblem: termo alemão para “problema de decisão”.
- ▶ É um problema da lógica simbólica que consiste em achar um algoritmo genérico para determinar se um dado enunciado da lógica de primeira ordem pode ser provado;
- ▶ Conhecido como o terceiro problema de Hilbert (1928);
- ▶ Hilbert acreditava que não existiria um problema insolúvel.

Entscheidungsproblem

- ▶ Em 1936, de forma independente, tanto Alonzo Church quanto Alan Turing mostraram que o enunciado é falso.
- ▶ Church provou que, na sua teoria do λ -calculus, não existe algoritmo (função computável) que decide para duas expressões do cálculo λ se elas são equivalentes ou não.
- ▶ Turing provou o Teorema da Parada!

O problema da parada

- ▶ Dadas uma MT qualquer M , um alfabeto Σ e uma string $w \in \Sigma^*$ qualquer, determinar se a computação de M com a entrada w para.

Teorema

O problema da parada é indecidível.

Prova, por contradição

- ▶ A representação de uma palavra w suprida como entrada para uma MT tenha como representação ela própria, isto é, $R[w] = w$.
- ▶ Notacao $R[M, w]$ é a representação de entrada para a máquina M com entrada w .

Como obter esta representação?

Símbolo	Codificação
0	1
1	11
B	111
q_0	1
q_1	11
...	...
q_n	11...1 (n vezes)
L	1
R	11

Como obter esta representação?

Transições	Codificação
$\delta(q_0, B) = [q_1, B, R]$	101110110111011
$\delta(q_1, 0) = [q_0, 0, L]$	1101010101
$\delta(q_1, 1) = [q_2, 1, R]$	110110111011011
$\delta(q_2, 1) = [q_0, 1, L]$	1110110101101

Assim, esta é a representação de M , que aceita as palavras que iniciam com 11, além de λ :

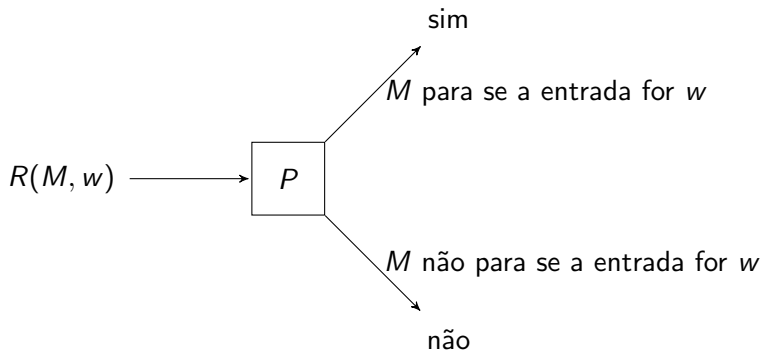
10111011011101111010101011101101110110111110110101101

Prova, por contradição

- ▶ Suponha que este problema (da parada) seja decidível.
- ▶ Neste caso seja uma MT P que solucione o problema.

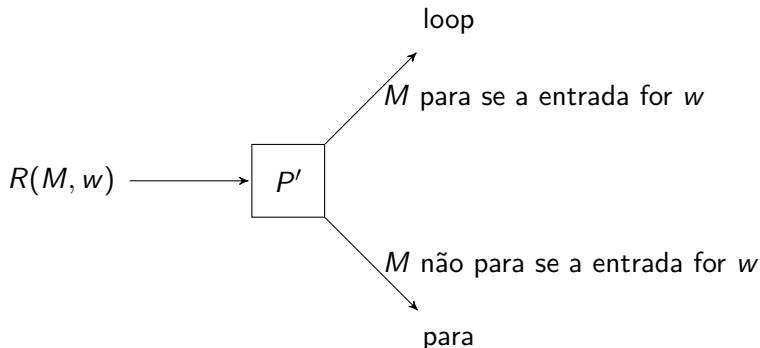
Prova

Considere esta MT P que recebe como entrada a representação $R(M, w)$ da máquina M com sua entrada w :



Prova

A partir desta MT P , seria possível contruir uma MT P' cujo comportamento se dá assim:



Prova

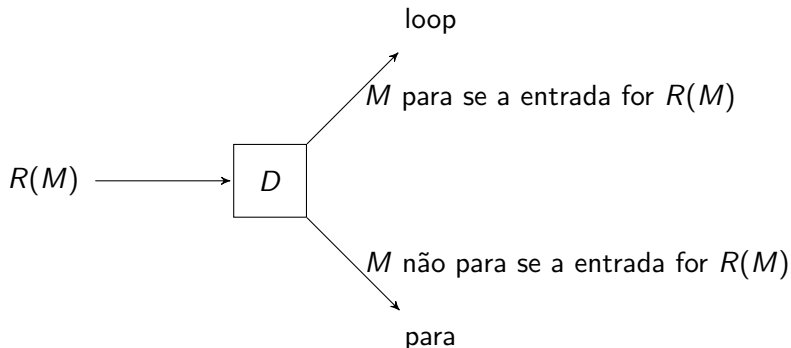
- ▶ Esta máquina P' foi construída para entrar em loop toda vez que a máquina M com entrada w para e vice-versa:
- ▶ P' entra em loop se e somente se M para com entrada w .

Prova

- ▶ Basta fazer, para cada par (f, a) tal que $\delta(f, a)$ seja indefinido, onde f é um estado final de P e a é um símbolo da fita de P $\delta(f, a) = (\text{Novoestado}, a, D)$, onde *novoestado* é um novo estado.
- ▶ Neste último estado, faz-se a máquina entrar em loop fazendo $\delta(l, a) = [l, a, D]$ para todo a de P .
- ▶ Isto é, acrescentamos apenas duas transições em P para obter P' .

Prova

A partir de P' , pode-se obter uma outra máquina D conforme a figura:



Prova

- ▶ O que acontece se $R[D]$ for dada como entrada para a MT D .
- ▶ Se D entra em loop para a entrada $R[D]$ é porque D para se a entrada por $R[D]$
- ▶ e se D para para a entrada $R[D]$ é porque D não para se a entrada for $R[D]$
- ▶ Absurdo!
- ▶ Como D pode ser construída a partir de P , então P não pode existir e portanto o problema da parada é indecidível.

Corolário

A linguagem de todas as entradas da forma $R(M, w)$ Onde $R(M, w)$ é uma representação de uma máquina de Turing M tal que M para com entrada w sobre o alfabeto $0,1$ não é recursiva.

Outros problemas indecidíveis

- ▶ Saber se as linguagens de duas GLC's são disjuntas
- ▶ Saber se a linguagem de uma GLC é Σ^*
- ▶ Saber se uma GLC é ambígua
- ▶ Muitos outros...

Licença

- ▶ Slides feitos em $\text{\LaTeX}$ usando beamer e tikz, editados com vim.
- ▶ Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>