Insper

Lógica da Computação - 2020/1

Aula 20/T10 - 06/May/2020

Raul Ikeda - rauligs@insper.edu.br

Objetivos

1. P vs NP
2. NP-Completude
3. Complexidade de Espaço

Complexidade Parte 2

1. Dado um conjunto de possíveis caracteres Σ, crie um programa de computador para quebrar uma senha de 4 caracteres na força bruta. Admita que existe uma função *testa(senha)* que retorna *True* caso a senha esteja certa ou *False* caso contrário.

2. Agora crie um programa para quebrar uma senha com número de caracteres indefinidos.

Relembrando a aula da semana passada:

- **Problemas P**: conseguimos **decidir** sobre o problema em tempo **polinomial**.
- **Problemas NP**: conseguimos **decidir** sobre o problema em tempo **polinomial** com uma **MT não determinística**. Ou ainda, conseguimos **reconhecer** sobre o problema em tempo polinomial.

Ainda no problema de alocação dos dormitórios, **reconhecer** que uma dada solução do problema é satisfatório é fácil, mas uma solução que analisa **todas as possibilidades** é **intratável**.

Nova dica do milhão:

> **Definição**: "Problema da Satisfazibilidade: Uma fórmula booleana é dita satisfazível se algum conjunto de atribuições às variáveis booleanas resulta em valor verdadeiro."

Por Exemplo: $\phi = (\bar{x} \wedge y) \vee (x \wedge \bar{z})$.

Se atribuirmos os valores $x = F$, $y = V$ e $z = F$, ϕ será verdadeiro.

Defini-se então:

$$SAT = \{C(\phi) \mid \phi \; \acute{e} \; uma \; f\acute{o}rmula \; booleana \; satisfaz\acute{i}vel\}$$

E o atalho:

> **Teorema de Cook-Levin**: "$SAT \in P$ sse $P = NP$" Sipser, Pag. 288.

O desenvolvimento do teorema implica que para provar que $P = NP$ (e ficar milionário) basta achar um algoritmo de tempo polinomial para um determinado subconjunto de problemas NP. Esses problemas são chamados de **NP-Completos**.

> **Definição** (Sipser Pag. 292): Uma linguagem B é NP-Completa se satisfaz duas condições:
>
> 1. B está em NP
> 2. toda A em NP é **redutível em tempo polinomial** a B
>
> **Definição** (Sipser Pag. 288): A linguagem A é **redutível em tempo polinomial** à linguagem B, se existe uma **função computável em tempo polinomial** $f : \Sigma^* \rightarrow \Sigma^*$, onde para toda w,
>
> $$w \in A \Leftrightarrow f(w) \in B$$
>
> A função f é denominada redução de tempo polinomial de A para B, ou $A \leq_p B$.
>
> **Definição** (Sipser Pag. 288): Uma função $f : \Sigma^* \rightarrow \Sigma^*$ é uma **função computável em tempo polinomial** se existe alguma MT de tempo polinomial que para com exatamente $f(w)$ na sua fita, quando iniciada sobre qualquer entrada w.

Portanto, completando o Teorema de Cook-Levin:

> **Teorema**: "Se $A \leq_p B$ e $B \in P$, então $A \in P$" Sipser, Pag. 289.
>
> **Teorema**: "Se B for NP-Completo e $B \in P$ então $P = NP$" Sipser, Pag. 292.

O que as definições acima dizem é que se um problema é NP-Completo, logo ele pode ser reduzido em outro problema NP-Completo em tempo polinomial. Dentre os problemas mais conhecidos, poderíamos montar o seguinte diagrama:

Exercício: Mostre que toda LLC é um membro de P. Ver Teorema 7.16 do Sipser.
Exercício: Mostre que SAT é NP-Completo. Ver Teorema 7.37 do Sipser.

Complexidade no Espaço

Primeiramente precisamos definir como medir complexidade de espaço:

> **Definição** (Sipser Pag. 322): "Seja uma MT determinística que para sobre todas as entradas. A **complexidade de espaço** de M é uma função $f: N \to N$, onde $f(n)$ é o número máximo de células de fita que M visita sobre qualquer entrada de comprimento n. Se a complexidade de espaço de M é $f(n)$, também dizemos que M roda em espaço $f(n)$.
> Se M é uma MT não determinística na qual todos os ramos param sobre todas as entradas, definimos sua complexidade de espaço $f(n)$ como o número máximo de células de fita que M visita sobre qualquer ramo de sua computação para qualquer entrada de comprimento n."

Exemplo: SAT é NP-Completo e PSpace.

Exercício: Definir formalmente as classes $PSpace$ e $NPSpace$. Sipser Cap 8.2.

Problema de Z$ 1.000.000,00

> **Teorema de Savitch**: "Para qualquer função $f : N \to R^+$, onde $f(n) \geq n$, $NSPACE(f(n)) \subseteq SPACE(f^2(n))$" Sipser Pag. 324.

Qual é a conclusão?

`>> Ver Sipser Pag. 324.`

Outras classes: Co-NP, RP e ZPP

```
>> Ver Hopcroft et al. Cap 11.4.
```

To Go Further: L, NL, Co-NL, EXPSPACE. Ver Sipser e Hopcroft.

Próxima aula:

- Provas de Teoremas

Referências:

- Sipser Cap 0.4
- Hopcroft Cap 1