NP-Completude: Redução

Análise de Algoritmos – Ciência da Computação

Prof. Daniel Saad Nogueira Nunes

IFB – Instituto Federal de Brasília, Campus Taguatinga

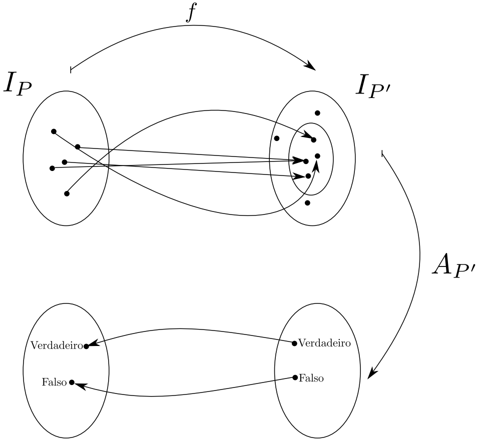
Sumário

Sumário

Redução Entre Problemas

- Sejam P e P' problemas de decisão.
- Uma redução de um problema P para um problema P' consiste de uma função computável que mapeia instâncias de I_P de P em instâncias de $I_{P'}$ de P'.
- Além disso, a redução deve manter a propriedade de que, I_P leva a **Verdadeiro** se, e somente se, $I_{P'}$ leva a **Verdadeiro** também.
- Podemos resolver P a partir de P'.

$\mathcal{P}$ vs $\mathcal{NP}$



Reduções Polinomiais

- Como estamos interessados na questão $\mathcal{P}$ vs $\mathcal{NP}$, estamos interessados em reduções polinomiais.
- Isto é, em mapeamentos de instâncias que podem ser computados em tempo $O(n^k)$.
- Transformamos uma instância de P em outra de P' em tempo polinomial.

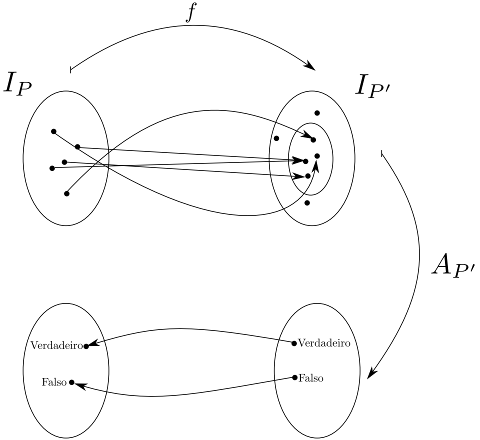
Reduções Polinomiais

Definição (Redução Polinomial)

Sejam P e P' problemas de decisão.

Uma redução polinomial de um problema de decisão P para um problema de decisão P' é uma função computável em tempo polinomial que mapeia instâncias I_P de P para instâncias $I_{P'}$ de P', de modo que a saída de I_P é **Verdadeira** se e somente se a saída de $I_{P'}$ é **Verdadeira**.

$\mathcal{P}$ vs $\mathcal{NP}$



Problemas $\mathcal{NPC}$

- Com posse da definição de redução, podemos estudar os problemas mais difíceis desta classe, os problemas NP-completos.
- Eles são a chave e ponto central da questão $\mathcal{P}$ vs $\mathcal{NP}$.

Sumário

2 $\mathcal{NPC}$

$\mathcal{NP}$-Completude

Definição (Problemas $\mathcal{NP}$-Difícil)

Um problema é dito $\mathcal{NP}$-difícil se todos os problemas em $\mathcal{NP}$ se reduzem a ele em tempo polinomial

$\mathcal{NP}$-Completude

Definição (Problemas $\mathcal{NP}$-Completo)

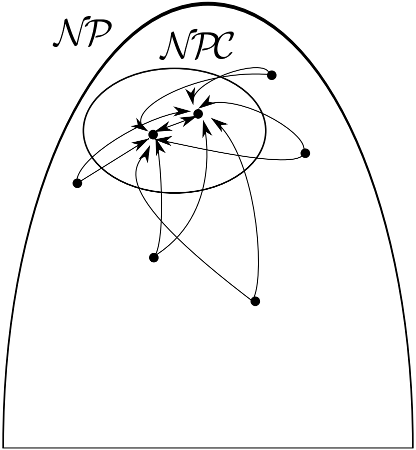
Um problema é dito $\mathcal{NP}$-completo se:

1. Ele está em $\mathcal{NP}$.
2. Ele é $\mathcal{NP}$-difícil.

$\mathcal{NP}$

- Os problemas da classe $\mathcal{NPC}$ são os mais difíceis de $\mathcal{NP}$.
- Se uma solução polinomial for encontrada para qualquer problema em $\mathcal{NPC}$, então demonstramos que $\mathcal{P} = \mathcal{NP}$, visto que todos os problemas em $\mathcal{NP}$ podem ser reduzidos para um problema em $\mathcal{NPC}$.
- Se for demonstrado que um problema em $\mathcal{NPC}$ não pode ser resolvido em tempo polinomial, então demonstramos que $\mathcal{P} \neq \mathcal{NP}$.
- Mostrar que um problema está em $\mathcal{NPC}$, é atestar a sua dificuldade!

$\mathcal{NP}$



Demonstrando a Dificuldade de Problemas

- Para tentar demonstrar a intratabilidade de um problema P, podemos tentar provar que ele está em $\mathcal{NPC}$.
- Como fazer isto?
  1. Precisamos mostrar que P admite um verificador que rode em tempo polinomial.
  2. Precisamos demonstrar que qualquer problema em $\mathcal{NP}$ se reduz em tempo polinomial a P.
     - Não precisamos fazer isto para cada problema em $\mathcal{NP}$, basta reduzir um problema de $\mathcal{NPC}$ a P.

Demonstrando a Dificuldade de Problemas

Framework para demonstrar que $P \in \mathcal{NPC}$

1. Mostrar que P está em $\mathcal{NP}$.
2. Mostrar que um problema $P' \in \mathcal{NPC}$ possui uma redução em tempo polinomial para P, tornando P pelo menos tão difícil quanto P'.

Demonstrando a Dificuldade de Problemas

- Para este *framework* funcionar, precisamos conhecer problemas em $\mathcal{NPC}$, no entanto não vimos nenhum até o momento.

Sumário

2. $\mathcal{NPC}$
   - SAT

SAT

Definição (SAT)

O problema SAT consiste em determinar se uma fórmula proposicional na CNF é satisfazível.

- **Entrada:** Uma fórmula φ na CNF. Exemplo:

$$\varphi := (p \lor q \lor r) \land w \land (\neg q \lor p)$$

- **Saída:**

$$\begin{cases} \textbf{Verdadeiro}, & \text{se } \varphi \text{ é satisfazível.} \\ \textbf{Falso}, & \text{caso contrário.} \end{cases}$$

SAT

- Stephen Cook mostrou que a satisfazibilidade é um problema difícil!

Teorema (Cook 1971)

$\texttt{SAT} \in \mathcal{NPC}$.

SAT

- Como SAT está em $\mathcal{NPC}$, podemos partir dele para mostrar que outros problemas também estão em $\mathcal{NPC}$

Sumário

Sumário

Satisfazibilidade

3SAT

- **Entrada:** uma fórmula $\varphi \in \mathcal{L}_p$ na CNF em que cada cláusula tem exatamente três variáveis.
- **Saída**:

$$\begin{array}{ll} \textbf{Verdadeiro}, & \text{se } \varphi \text{ é satisfazível.} \\ \textbf{Falso}, & \text{caso contrário.} \end{array}$$

Satisfazibilidade

Theorem (3SAT $\in \mathcal{NPC}$)

3SAT *está em* $\mathcal{NPC}$.

3SAT

Demonstração

- 3SAT está em $\mathcal{NPC}$.

Claramente 3SAT está em $\mathcal{NP}$. Dado uma valoração que torna φ verdadeira, é possível verificar em tempo polinomial que φ é satisfazível.

3SAT

Demonstração

A ideia da prova é mostrar que toda cláusula φ na CNF pode ser convertida em cláusulas na 3-CNF. Seja C uma cláusula da forma $c_1 \vee c_2 \vee \ldots \vee c_n$ na instância de SAT. Mostraremos que existe uma fórmula C' equivalente a C na 3-CNF. Temos 3 casos:

- $n = 3$.
- $1 \leq n < 3$.
- $4 \leq n$.

3SAT

Demonstração

- $n = 3$

Não precisamos fazer nada, a cláusula já está na 3-CNF.

3SAT

Demonstração

- $n < 3$

Se $n = 1$, $C = (c_1)$ e
$C' = (c_1 \vee x \vee y) \wedge (c_1 \vee \neg x \vee \neg y) \wedge (c_1 \vee x \vee \neg y) \wedge (c_1 \vee \neg x \vee y)$.

Se $n = 2$, $C = (c_1 \vee c_2)$ e $C' = (c_1 \vee c_2 \vee x) \wedge (c_1 \vee c_2 \vee \neg x)$

3SAT

Demonstração

- $n > 3$

$C = (c_1 \vee c_2 \vee \ldots \vee c_n)$. Então:

$C' = (c_1 \vee c_2 \vee x_1) \wedge (c_3 \vee \neg x_1 \vee x_2) \wedge (c_4 \vee \neg x_2 \vee x_3) \wedge \ldots \wedge (c_{n-1} \vee c_n \vee \neg x_{n-3})$

3SAT

Demonstração

- O que queremos mostrar: C é satisfazível sse C' é. Mostraremos para $n > 3$, uma vez que é fácil mostrar para $n < 3$.
- $\Rightarrow$). Se C é satisfazível, então existe pelo menos um c_i que é verdadeiro.
  - Podemos valorar os x do jeito que queremos...
  - Exemplo: $c_3 = V$. Colocamos $x_1 = V, x_2 = F$ e o restante dos x para V.

3SAT

Demonstração

- $\Leftarrow$). Se C' é satisfazível, pelo menos um dos c_i é verdadeiro. Caso contrário, teríamos uma expressão da forma $(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \ldots \wedge (\neg x_{n-4} \vee x_{n-3})(\neg x_{n-3})$, que é insatisfazível.

$\square$

Sumário

3 Reduções
- 3SAT
- CLIQUE
- VC
- DS

Clique

Definição (Clique)

Uma clique é um grafo completo. O tamanho de uma clique é a sua quantidade de vértices.

`CLIQUE`

- **Entrada:** um grafo $G = (V, E)$ e um inteiro k.
- **Saída:** sim se G contém uma clique de tamanho $\geq k$, não caso contrário.

Clique

Teorema

`CLIQUE` está em $\mathcal{NPC}$.

- `CLIQUE` está em $\mathcal{NP}$. Basta verificar se cada nó do certificado se conecta aos demais nós do certificado.
- A ideia da prova é mostrar que `SAT` se reduz a `CLIQUE` em tempo polinomial.

Clique

Demonstração

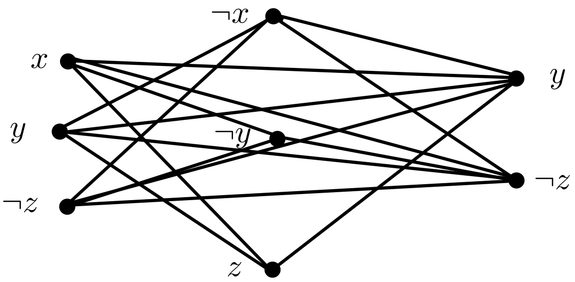
- A partir de uma instância de `SAT` vamos construir uma instância para `CLIQUE`.
- Cada cláusula será uma coluna de vértices. Cada vértice representa um literal.
- As colunas se conectam entre si, exceto se dois vértices representarem um literal e a sua negação.
- Teremos com esta redução que, uma fórmula φ em `SAT` é satisfazível se e somente se existe uma clique de tamanho m, em que m é o número de cláusulas.

Clique

Demonstração

$$\varphi = (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (y \vee \neg z)$$

$G =$



Clique

Demonstração

- Vamos mostrar que φ é satisfazível sse existe uma clique de tamanho m, onde m é a quantidade de cláusulas em φ.
- $\Rightarrow$). Suponha φ satisfazível. Existe uma valoração em que cada cláusula possui pelo menos um literal verdadeiro. Esse vértice tem que estar em uma clique de tamanho m que envolve os vértices desta valoração, uma vez que só não existe arestas entre dois literais opostos.

Clique

Demonstração

- Vamos mostrar que φ é satisfazível sse existe uma clique de tamanho m, onde m é a quantidade de cláusulas em φ.
- $\Leftarrow$). Suponha que G tem uma clique de tamanho m. Logo, φ é satisfazível, e a valoração é aquela apontada pela clique.
- Note que dois vértices da mesma coluna nunca estão conectados e que o literal e seu oposto também não.

Sumário

Cobertura de Vértices

Definição (Cobertura de Vértices)

Uma cobertura de vértices é um **conjunto** de vértices de modo que cada aresta de G incide em pelo menos um desses vértices

VC

VC

- **Entrada:** um Grafo $G = (V, E)$ e um inteiro k.
- **Saída:**

$$\begin{cases} \textbf{Verdadeiro}, & \text{se } G \text{ tem uma cobertura de vértices de tamanho} \leq k \\ \textbf{Falso}, & \text{caso contrário.} \end{cases}$$

Cobertura de Vértices

Theorem ($\texttt{VC} \in \mathcal{NPC}$)

$\texttt{VC}$ *está em* $\mathcal{NPC}$.

Cobertura de Vértices

Demonstração

- `VC` está em $\mathcal{NP}$. Basta verificar se cada aresta incide em um dos vértices da cobertura.
- Vamos mostrar que `CLIQUE` se reduz polinomialmente a `VC`.
- Seja G, k uma instância do problema `CLIQUE`.
- Tome o complemento do grafo, $\bar{G}$ e tome $k' = n - k$.
- Agora, basta mostrar que G tem uma clique de tamanho k sse $\bar{G}$ tem uma cobertura de vértices de tamanho $n - k$.

Cobertura de Vértices

Demonstração

- G tem uma clique de tamanho k sse $\bar{G}$ tem uma cobertura de vértices de tamanho $n-k$.
- $\Rightarrow$). Se G tem uma clique $G=(U,F)$ de tamanho k, então em $\bar{G}$, todos os vértices de $V-U$ possuem todas as arestas.
  - O miolo da clique está vazio.
- Logo, existe uma cobertura de tamanho $n-k$: os vértices $V-U$.

Cobertura de Vértices

Demonstração

- G tem uma clique de tamanho k sse $\bar{G}$ tem uma cobertura de vértices de tamanho $n - k$.
- $\Leftarrow$). Suponha que $\bar{G}$ tenha uma cobertura D de tamanho $n - k$.
- Não pode existir arestas entre os vértices de $V - D$.
  - Caso contrário não teríamos a cobertura D.
- No grafo G, existirão arestas entre todos os vértices $V - D$ e portanto, existe uma clique de tamanho $n - (n - k) = k$.
- $\square$

Sumário

Conjunto Dominante

Definição (Conjunto Dominante)

Um conjunto dominante é um conjunto de vértices D em $G = (V, E)$ de modo que todo vértice de G está em D ou se liga a um vértice de D.

DS

DS

- **Entrada:** Um grafo $G = (V, E)$ e um inteiro k.
- **Saída:**

$$\begin{cases} \textbf{Verdadeiro}, & \text{se } G \text{ tem um conjunto dominante de tamanho } \leq k. \\ \textbf{Falso}, & \text{caso contrário.} \end{cases}$$

Conjunto Dominante

Theorem ($\texttt{DS} \in \mathcal{NPC}$)

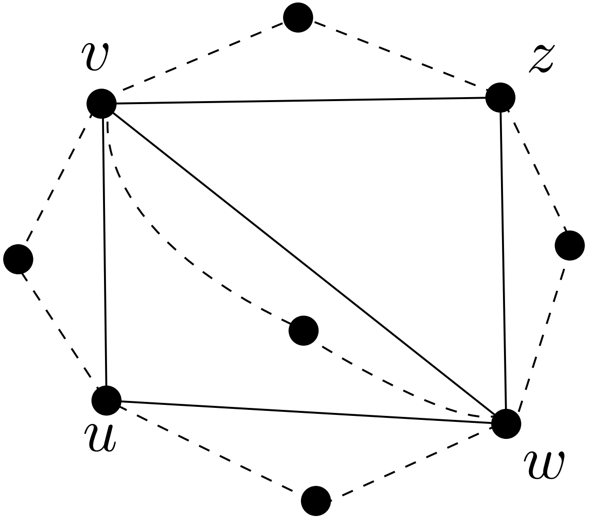
$\texttt{DS}$ *está em* $\mathcal{NPC}$.

Conjunto Dominante

Demonstração

- `DS` está em $\mathcal{NP}$.
- Vamos mostrar que `VC` se reduz polinomialmente à `DS`.
- Adicionaremos $2|E|$ arestas e $|E|$ vértices ao grafo G, formando assim o grafo G'.
  - Se temos uma aresta (v, u), teremos agora arestas $(v, u), (v, w), (w, u)$.
  - Transformamos toda aresta em um triângulo.

Conjunto Dominante



Conjunto Dominante

Demonstração

- G tem uma cobertura de tamanho k sse G' tem um conjunto dominante de tamanho k.
- $\Rightarrow)$. Se G tem uma cobertura de tamanho k, então G', por construção, tem um conjunto dominante de tamanho k.

Conjunto Dominante

Demonstração

- G tem uma cobertura de tamanho k sse G' tem um conjunto dominante de tamanho k.
- $\Leftarrow$. Suponha que G' tem um conjunto dominante de tamanho k.
  - Observação : se o conjunto dominante tem vértices que não existem em G, podemos trocar por um vértice que existe em G. Basta pegar um adjacente.
- Como existe um conjunto dominante de tamanho k contendo apenas vértices de G, então existe uma cobertura de vértices em G de tamanho k.