**INF 02779 Análise de Algoritmos - 2002/2** 01/03/03

# Aula 15: Maria Cecilia

Instrutor: Berilhes Borges Garcia Escriba:

*Example:* Conjunto Independente $\leq^P$ Clique
Dada uma instância $(G, k)$ do problema do Conjunto Independente, onde $G = (V, E)$, nós transformamos esta para $(G', k)$, onde $G' = (V, E')$, ou seja G' possui os mesmos vértices de $G$, mas $E'$ consiste exatamente das arestas que não estão presentes em $G$. $U$ é um conjunto independente em G se, e somente se, $U$ é um clique em $G'$. ⊠

**Exercício 1.** *Mostre que Conjunto Independente $\leq^P$ Cobertura por Vértice .*

**Definição 1.** *(NP_hard): Um problema $R$ é NP_hard se, e somente se, todo problema $L \in NP$ satisfaz $L \leq^P R$ .*

**Definição 2.** *(NP_completo): Um problema $R$ é NP_completo se, e somente se, $R$ é NP_hard e $R \in NP$.*

Note que definir um objeto não é o suficiente para que este objeto exista. Dessa forma não é obvio que problemas NP_completo existam !

**Teorema 1.** *(Teorema de Cook (Cook 71)) Satisfabilidade é NP_completo.*

A prova deste teorema é longa e obscura, dessa forma nós a omitiremos. No entanto, esta é uma das provas mais bonitas de toda a Ciência da Computação, de modo que os alunos são fortemente encorajados a estuda-lá.
[Implicações do Teorema de Cook] Existe pelo menos um problema interessante super-difícil na classe NP.

*Mas isto é realmente importante?*

**SIM!!!** Existem muitos (muitíssimos problemas) que podem ser resolvidos de forma eficiente se nós tivessemos um algoritmo de tempo polinomial para Satisfabilidade.
Muitos destes problemas são exatamente tão difíceis quanto o próprio problema da Satisfabilidade.

*Uma propriedade útil das reduções de Karp.*

**Teorema 2.** *Se $L \leq^P R$ e $R \leq^P S$ então $L \leq^P S$ .*

*Proof.* (Prova Esquemática) Componha a redução $T$ de $L$ para $R$ com a redução $T'$ de $R$ para $S$ de modo a obter uma nova redução $T''(x) = T'(T(x))$ de $L$ para $S$. □

**Corolário 1.** *$R$ é NP_hard se, e somente se, Satisfabilidade $\leq^P R$ .*

*Proof.* Se $R$ é NP_hard então todo problema em NP pode ser reduzido em tempo polinomial à R, em particular Satisfabilidade uma vez que este está em NP.
Para qualquer problema $L$ em NP, $L \leq^P$ Satisfabilidade e assim se Satisfabilidade $\leq^P$ R nós temos que $L \leq^P R$ . Portanto, $R$ é NP_hard se, e somente se, Satisfabilidade $\leq^P R$. □

**Teorema 3.** *(Outro problema NP_completo) Satisfabilidade $\leq^P$ Conjunto Independente.*

*Proof.* A chave da redução consiste em mapear uma fórmula CNF $\phi$ para um par $< G, k >$. $m$ é o número de clausúlas de $\phi$. Crie um vértice em G para cada literal em $\phi$. Une dois vértices u, v em G por um aresta se, e somente se:

- u e v correpondem a literais na mesma clausúla de $\phi$ (arestas verdes) ou
- u e v correspondem a literais x e $\rceil x$ (ou vice-versa) para alguma variável proposicional x.

A seguir ajuste K para m. Note que esta translação claramente pode ser feita em tempo polinomial.
$\phi$ : $(x_1 \vee \rceil x_3 \vee x_4) \wedge (x_2 \vee \rceil x_4 \vee x_3) \wedge (x_2 \vee \rceil x_1 \vee x_3)$
(FIGURA)
Correção da Translação
Se $\phi$ é satisfatível então existe alguma atribuição que satisfaz pelo menos um literal em cada clausúla.
$\Longrightarrow$ Considere o conjunto U em G correspondente ao primeiro literal satisfeito em cada clausúla.

- $\mid$ U $\mid$ = m
- Uma vez que U tem somente um vértice por clausúla, nenhum par de vértices em U é unido por arestas verdes.
- Uma vez que uma atribuição verdade nunca satisfaz ambos x e $\rceil x$, U não contém vértices rotulados ambos de x e $\rceil x$ e assim nenhum par de vértices em U é unido por um aresta vermelha.
- Portanto G possui um conjunto independente U, de tamanho pelo menos m.
- Logo, $< G, m >$ é uma instância Sim para o problema do conjunto independente.

$\Longleftarrow$ Se $< G, m >$ é uma instância Sim para o problema do conjunto independente, então existe um conjunto U de $m$ vértices em G que não compartilham nenhuma aresta.
Portanto, U tem precisamente um vértice por clausúla por causa das arestas verdes em G.
Devido as arestas vermelhas em G, U também não contém vértices de x e $\rceil x$. Nós podemos construir uma atribuição A que faz todos os literais rotulando vértices em U com valor *true* e para qualquer variável não rotulando um vértice em U nós atribuimos seu valor verdade de uma forma arbitrária. Por esta construção fica claro que A satisfaz $\phi$. Portanto, $\phi$ é uma instância SIM para satisfabilidade.
Nós acabamos de mostrar que o problema Conjunto Independente é NP_hard e nós já sabemos que o problema Conjunto Independente está em NP. Portanto, Conjunto Independente é NP_completo. $\square$

**Corolário 2.** *Clique é NP_completo.*

*Proof.* Nós já mostramos que Conjunto Independente $\leq^P$ Clique e Clique está em NP. $\square$

Problemas que nós já sabemos que são NP_completos:

- Satisfabilidade
- Conjunto Independente

- Clique
- Cobertura por Vértice
- Existem mais de 1000 problemas práticos que são NP_completos, por exemplo, scheduling, layout ótimo de circuitos VLSI, etc.