Algoritmos

Pedro Hokama

Fontes

- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest, Ronald L. Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden

Apresentação Baseada:

- Stanford Algorithms
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V`
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6`
- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende

Satisfazibilidade de Fórmulas

- Uma fórmula booleana ϕ é composta de:
  - n variáveis booleanas: $x_1, x_2, \ldots, x_n$;
  - m conectivos booleanos: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - parênteses
- Ex. $\phi = ((x_1 \rightarrow x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$

Problema da Satisfazibilidade de Fórmulas - SAT

Dada uma fórmula booleana, decidir se existe uma atribuição das variáveis booleanas que faz com que ela seja avaliada como 1 (verdadeira).

- No exemplo ϕ é satisfazível com a atribuição $< x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1 >$
- Um algoritmo ingênuo que testa todas as possibilidades é inviável pois existem 2^n atribuições diferentes.

Teorema

SAT é NP-completo

Lema

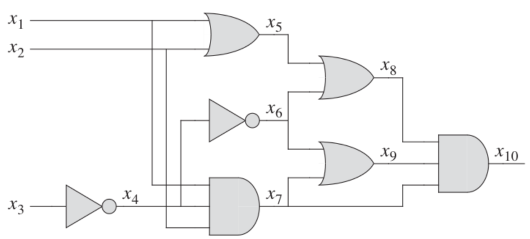
$SAT \in NP$

Lema

SAT $\in$ *NP-Difícil*

- Para mostrar que SAT $\in$ NP, mostramos que um certificado consiste em uma atribuição satisfatória das variáveis. Esse certificado pode ser verificado substituindo cada variável pelo valor dessa atribuição e avaliando a expressão, que pode ser feito em tempo polinomial. Se o resultado for 1 o algoritmo verificou que a formula é satisfazível.

- Para mostrar que SAT $\in$ NP-Difícil mostraremos que CIRCUIT-SAT $\leq_p$ SAT.
- Considere um circuito C qualquer. Para cada fio x_i no circuito, a fórmula ϕ vai ter uma variável x_i, expressamos cada porta lógica como uma cláusula na formula booleana que representa o seu comportamento. No fim fazemos a conjunção das cláusulas.



$$
\begin{aligned}
\phi = x_{10} & \wedge (x_4 \leftrightarrow \neg x_3) \\
& \wedge (x_5 \leftrightarrow (x_1 \vee x_2)) \\
& \wedge (x_6 \leftrightarrow \neg x_4) \\
& \wedge (x_7 \leftrightarrow (x_1 \wedge x_2 \wedge x_4)) \\
& \wedge (x_8 \leftrightarrow (x_5 \vee x_6)) \\
& \wedge (x_9 \leftrightarrow (x_6 \vee x_7)) \\
& \wedge (x_{10} \leftrightarrow (x_7 \wedge x_8 \wedge x_9)) .
\end{aligned}
$$

Lema

C é satisfazível se e somente se ϕ é satisfazível.

- Esse algoritmo de redução executa em tempo polinomial.
- ($\rightarrow$) Se C tem uma atribuição que satisfaz, cada fio tem um valor bem definido e a saída do circuito é 1.
- Portanto se atribuirmos os valores de cada fio para as respectivas variáveis, a formula também terá valor 1.
- ($\leftarrow$) Se uma atribuição faz ϕ se verdadeira. Podemos atribuir o valor de cada fio com o valor das suas respectivas variáveis. E C terá saída igual a 1.
- Portanto mostramos que CIRCUIT-SAT $\leq_p$ SAT. $\square$

Satisfazibilidade 3-CNF

- Uma formula está na forma normal conjuntiva (*conjunctive normal form* - CNF) se é expressa como uma conjunção (ANDs) de clausulas, e cada clausula como disjunções (ORs) de uma ou mais literais.
- Uma fórmula booleana está na forma normal 3-conjuntiva (3-CNF) se está na forma normal conjuntiva e cada cláusula tem exatamente três literais distintas.
- Por exemplo:

$$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

3-CNF-SAT

Dada uma fórmula booleana na forma normal 3-conjuntiva, decidir se existe uma atribuição das variáveis booleanas que faz com que ela seja avaliada como 1 (verdadeira).

Teorema

3-CNF-SAT é NP-completo

Lema

3-CNF-SAT $\in$ NP

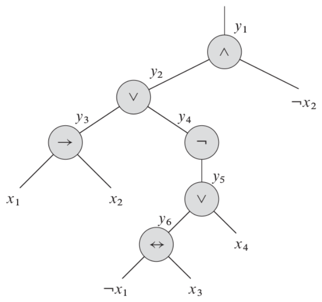
Prova igual à prova que SAT $\in$ NP.

Lema

3-CNF-SAT $\in$ NP-Difícil

- Para mostrar que 3-CNF-SAT $\in$ NP-Difícil mostraremos que SAT $\leq_p$ 3-CNF-SAT.
- Considere ϕ uma formula booleana qualquer.
- Primeiramente construímos uma árvore de análise para ϕ em que cada literal é uma folha e os conectivos são nós internos. Essa construção sempre é possível (usando a associatividade podemos colocar parenteses para explicitar uma ordenação)

- Por exemplo $\phi = ((x_1 \to x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$.



- Depois dessa construção, introduzimos uma variável para cada aresta que sobe dos nós internos.
- A seguir escrevemos cláusulas para cada nó interno. E fazemos a conjunção dessas cláusulas e criamos uma formula ϕ'.
- No exemplo ao lado a formula obtida seria:

$$\begin{aligned}\phi' \;=\; & y_1 \wedge (y_1 \leftrightarrow (y_2 \wedge \neg x_2)) \\ & \wedge (y_2 \leftrightarrow (y_3 \vee y_4)) \\ & \wedge (y_3 \leftrightarrow (x_1 \to x_2)) \\ & \wedge (y_4 \leftrightarrow \neg y_5) \\ & \wedge (y_5 \leftrightarrow (y_6 \vee x_4)) \\ & \wedge (y_6 \leftrightarrow (\neg x_1 \leftrightarrow x_3))\,.\end{aligned}$$

- Dessa forma obtemos clausulas que tem no máximo 3 literais. Mas ainda não está na forma normal 3-conjuntiva. Então podemos escrever a tabela verdade de cada clausula, e encontrar os valores que tornam ela FALSA. Por exemplo a cláusula $\phi'_1 = y_1 \leftrightarrow (y_2 \wedge \neg x_2))$

y_1	y_2	x_2	$(y_1 \leftrightarrow (y_2 \wedge \neg x_2))$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

- Analisando as linhas da tabela que tornam a cláusula FALSA obtemos:

$$(y_1 \wedge y_2 \wedge x_2) \vee (y_1 \wedge \neg y_2 \wedge x_2) \vee (y_1 \wedge \neg y_2 \wedge \neg x_2) \vee (\neg y_1 \wedge y_2 \wedge \neg x_2)$$

- Agora aplicamos a lei de DeMorgan

$$\phi''_1 = (\neg y_1 \vee \neg y_2 \vee \neg x_2) \wedge (\neg y_1 \vee y_2 \vee \neg x_2) \wedge (\neg y_1 \vee y_2 \vee x_2) \wedge (y_1 \vee \neg y_2 \vee x_2)$$

- Caso a cláusula resultante só tenha 2 literais $(l_1 \vee l_2)$, então incluímos uma literal auxiliar e a seguinte formula $(l_1 \vee l_2 \vee p) \wedge (l_1 \vee l_2 \vee \neg p)$
- No caso de uma cláusula com uma única literal l, incluímos 2 literais auxiliares $(l \vee p \vee q) \wedge (l \vee p \vee \neg q) \wedge (l \vee \neg p \vee q) \wedge (l \vee \neg p \vee \neg q)$.
- Seja ϕ''' a fórmula em 3-CNF-SAT resultante.

Lema

ϕ''' é satisfazível se e somente se ϕ é satisfazível.

- Como todas as transformações preservam o valor algébrico da formula, tanto ϕ quanto ϕ''' são equivalentes.
- A redução pode ser calculada em tempo polinomial. Construir ϕ' a partir de ϕ introduz no máximo uma variável e uma cláusula por conectivo.
- Construir ϕ'' a partir de ϕ' introduz no máximo oito cláusulas para cada cláusula. E a construção de ϕ''' no máximo multiplica por 4 o número de cláusulas.
- Portanto mostramos uma redução de tempo polinomial de SAT para 3-CNF-SAT e concluímos a prova. $\square$

CLICK

Problema do CLICK

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe um subgrafo G' induzido de G que tenha k vértices e seja completo.

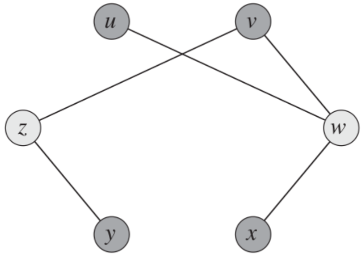
Teorema

CLICK é NP-completo

- Provado com 3-CNF-SAT $\leq_p$ CLICK

Cobertura por Vértices

- Dado um grafo não direcionado $G = (V, E)$ uma cobertura por vértices de G é um subconjunto $V' \subseteq V$ tal que para toda aresta $(u, v) \in E$, pelo menos um entre u e v deve estar em V'.
- Dizemos que um vértice em V' cobre todas as arestas adjacentes a ele. E em uma cobertura por vértices todas as arestas devem ser cobertas.
- O **tamanho** de uma cobertura por vértices é o número de vértices que contêm.



Problema da Cobertura por Vértices - VERTEX-COVER

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe uma cobertura por vértices $V' \subseteq V$ de tamanho k.

Teorema

VERTEX-COVER é NP-completo

Lema

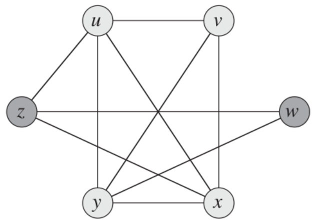
VERTEX-COVER $\in$ NP

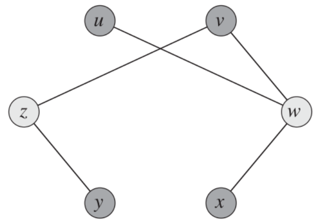
- Utilizando a própria cobertura V' como certificado, podemos verificar facilmente, em tempo polinomial, se $|V'| = k$ e se toda aresta (u, v), u ou v está em V'.

Lema

VERTEX-COVER $\in$ NP-Difícil

- Vamos mostrar que CLICK $\leq_p$ VERTEX-COVER.
- Definição: O **Complemento** de um grafo $G = (V, E)$ denotado por $\overline{G} = (V, \overline{E})$, em que, $\overline{E} = \{(u, v) : u, v \in V, u \neq v \text{ e } (u, v) \notin E\}$. Ou seja, o complemento de G é um grafo com os mesmos vértices e exatamente as arestas que não estão em G.
- A redução consiste em dada uma entrada $< G, k >$ do problema da CLICK. Calcular o complemento $\overline{G}$, o que pode ser feito em tempo polinomial.
- E então usar $\overline{G}$ como entrada para o problema do VERTEX-COVER, procurando uma cobertura de tamanho $|V| - k$.





Lema

G tem uma CLICK de tamanho k, se e somente se, $\overline{G}$ tem uma cobertura por vértices de tamanho $|V| - k$

- ($\rightarrow$) Se G tem uma CLICK C com k vértices, então $V \setminus C$ é uma cobertura em $\overline{G}$.
- Seja (u, v) qualquer aresta em $\overline{E}$, então (u, v) não está em G e portanto u e v não podem estar em C simultaneamente.
- Logo, pelo menos um deles está em $V \setminus C$ e portanto (u, v) está coberta.
- ($\leftarrow$) Se $\overline{G}$ tem uma cobertura por vértices $V' \subseteq V$ em que $|V'| = |V| - k$, então para todo $u, v \in V$ se $(u, v) \in \overline{E}$ então ou $u \in V'$ ou $v \in V'$ ou ambos. Pela contrapositiva se nenhum dos dois está em V' significa que (u, v) está em E e portanto $V - V'$ é uma CLICK, e tem tamanho $|V| - |V'| = k$
- Portanto mostramos um redução CLICK $\leq_p$ VERTEX-COVER. □

O Problema do Ciclo Hamiltoniano

- Definição: Um **ciclo hamiltoniano** de um grafo é um circuito que passa exatamente uma vez por todos os vértices.

Problema do Ciclo Hamiltoniano - HAM-CYCLE

Dado um grafo não orientado $G = (V, E)$, decidir se G tem um ciclo hamiltoniano.

Teorema

HAM-CYCLE é NP-completo

Lema

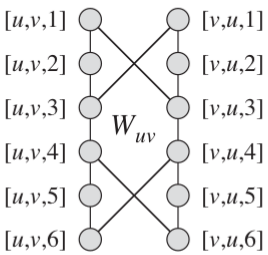
HAM-CYCLE $\in$ NP

Prova: Exercício

Lema

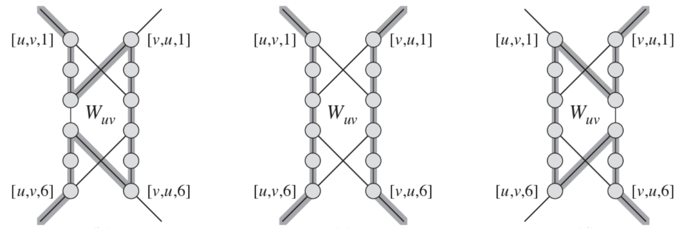
HAM-CYCLE $\in$ NP-Difícil

- Mostraremos que VERTEX-COVER $\leq_p$ HAM-CYCLE.
- Dado um grafo não dirigido $G = (V, E)$ e um inteiro k, construiremos um grafo não dirigido $G' = (V', E')$ que tem um ciclo hamiltoniano, se e somente se, G tem uma cobertura por vértices de tamanho k.
- A construção de G' se baseia em uma **engenhoca** (*widget*) que é um pedaço de um grafo que impõe certas propriedades.



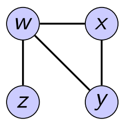
- Para cada aresta $(u, v) \in E$ o nosso grafo G' conterá uma cópia da engenhoca. Que denotaremos por W_{uv}.
- Cada vértice W_{uv} tem um nome e ao total ele tem 14 arestas.
- Para a engenhoca funcionar como queremos, ela vai se conectar ao resto do grafo apenas pelos vértices $[u, v, 1], [u, v, 6], [v, u, 1]$ e $[v, u, 6]$

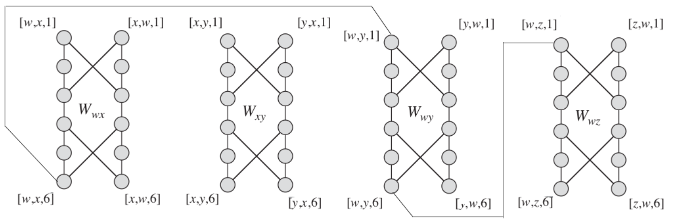
- Só existem três formas de um caminho entrar na engenhoca, passar por todos os vértices e sair.



- Em particular é impossível construir dois caminhos disjuntos nos vértices, um que ligue $[u, v, 1]$ a $[v, u, 6]$ e outro que ligue $[v, u, 1]$ a $[u, v, 6]$.
- Além das engenhocas serão adicionados k vértices seletores $s_1, s_2, \dots, s_k$. Usaremos as arestas que incidem nesses vértices para selecionar os k vértices que formarão a cobertura por vértices em G.

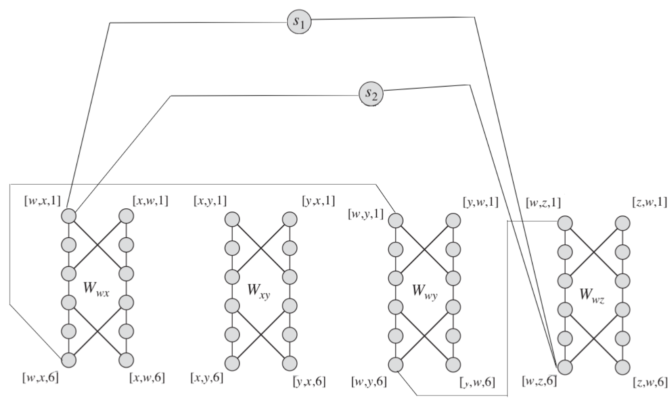
- Posteriormente para cada vértice $u \in V$ adicionamos arestas que criam um caminho em G' que passam por todas as engenhocas que correspondem a arestas incidentes a u.
- Ordenamos arbitrariamente os vértices adjacentes a u. E dada a ordenação $(v_1, v_2, v_{grau(u)})$, conectamos $[u, v_i, 6]$ com $[u, v_{i+1}, 1]$.
- No exemplo a seguir, w é vizinho de x, y e z (considerando essa ordem).



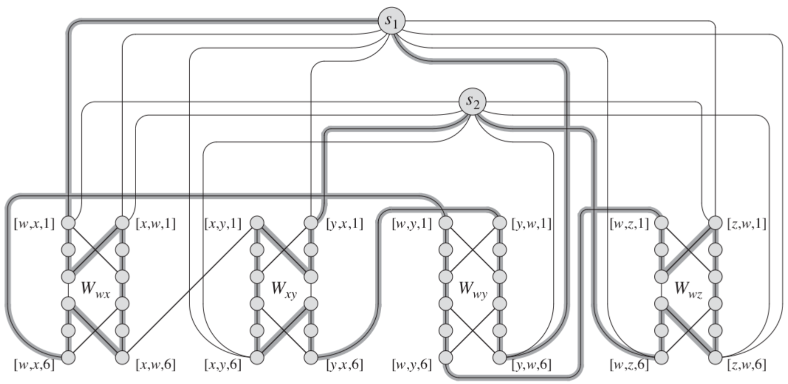


- A intuição é que se escolhermos um vértice u para a nossa cobertura, podemos fazer um caminho que passa por todas as engenhocas que correspondem as arestas adjacentes a u.

- O último tipo de aresta em E' une os vértices $[u, v_1, 1]$ e $[u, v_{grau(u)}, 6]$ a cada um dos seletores.



- Fazendo esse mesmo procedimento para todos os vértices obtemos o seguinte grafo, que é grande mas ainda é polinomial.



- Com 12 vértices por engenhoca, mais $k \leq |V|$ vértices seletores, em um total de

$$\begin{aligned} |V'| &= 12|E| + k \\ &\leq 12|E| + |V| \end{aligned}$$

- Para cada vértice $u \in V$ temos $grau(u) - 1$ arestas entre as engenhocas, em um total de

$$\sum_{u \in V} (grau(u) - 1) = 2|E| - |V|$$

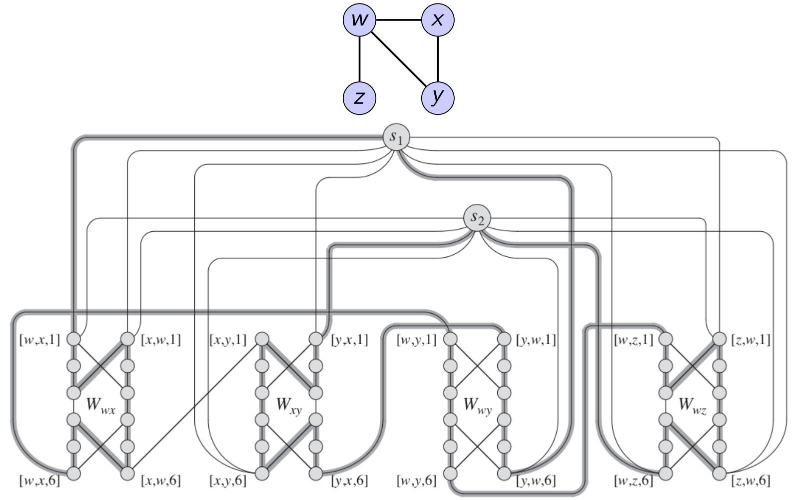
- Cada engenhoca tem 14 arestas, além das arestas entre os vértices seletores, no total

$$\begin{aligned} |E'| &= (14|E|) + (2|E| - |V|) + (2k|V|) \\ &= 16|E| + (2k - 1)|V| \\ &\leq 16|E| + (2|V| - 1)|V| \end{aligned}$$

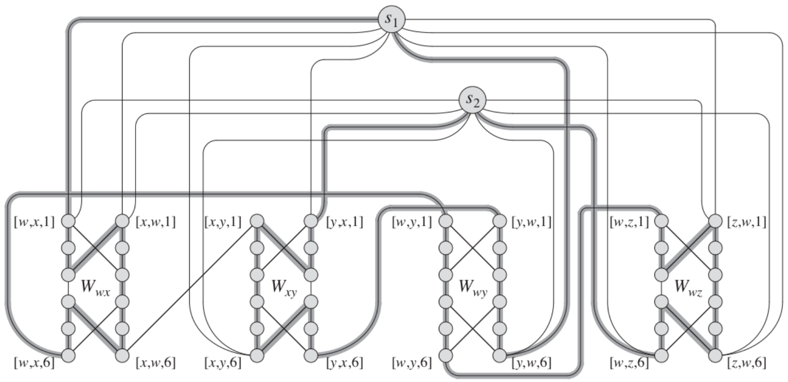
Lema

G tem uma cobertura por vértices de tamanho k, se e somente se, G' tem um caminho hamiltoniano.

- ($\rightarrow$) Suponha que $G = (V, E)$ tem uma cobertura por vértices $V' \subseteq V$ de tamanho k.
- Para cada vértice $u \in V'$ com os vizinhos $(v_1, v_2, \ldots, v_{grau(u)})$, adicionamos no caminho as arestas que ligam as engenhocas.
- Além disso ligamos os vértices internos da engenhoca dependendo se a aresta é coberta por um ou por dois vértices.
- No exemplo se $k = 2$, temos uma cobertura formada por w e y.



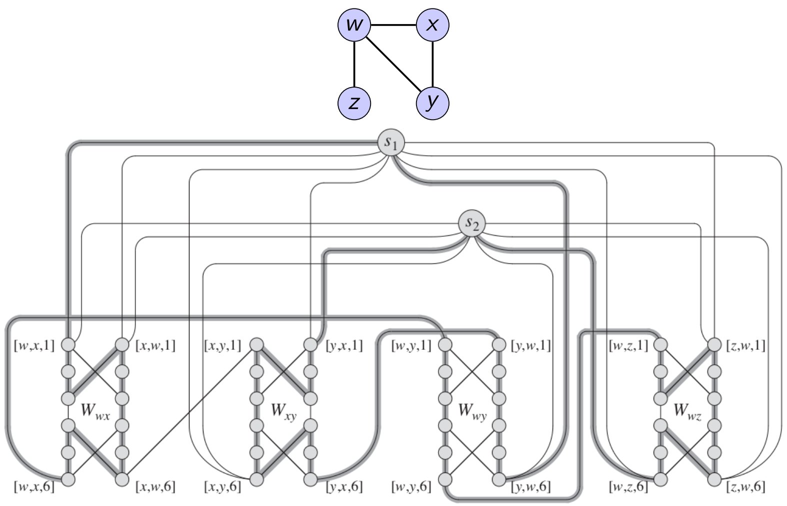
- Também incluímos as arestas com os vértices seletores da seguinte forma. Considere uma ordenação $(u_1, \ldots, u_k)$ qualquer dos vértices da cobertura. Conectamos o seletor s_i com o inicio do caminho formado por u_i, além disso ligamos o final do caminho ao próximo seletor s_{i+1}, por fim ligamos o final do caminho de u_k com o primeiro seletor, fechando o ciclo.



- Como a cobertura por vértices incide em todas as aresta, todas as engenhocas serão visitadas (uma ou duas vezes), assim como os vértices seletores. E portanto formamos um ciclo hamiltoniano.
- ($\leftarrow$) Suponha que $G' = (V', E')$ tem um ciclo hamiltoniano $C \subseteq E'$. Afirmamos que o conjunto

$$V' = \{u \in V : (s_j, [u, v, 1]) \in C \text{ para algum } 1 \leq j \leq k\}$$

- é uma cobertura por vértices para G.
- Como o caminho que sai de um seletor passa pelas engenhocas até chegar em algum seletor (já que é um ciclo hamiltoniano).
- Pela forma como G' foi construído, os vértices internos de cada engenhoca W_{uv} só podem ser visitados se o caminho teve inicio em u ou v, e portanto este estará na cobertura por vértices.



Resumindo:

- Mostramos que HAM-CYCLE $\in$ NP
- Mostramos que HAM-CYCLE $\in$ NP-Difícil
  - Mostrando uma redução de qualquer instância do VERTEX-COVER para HAM-CYCLE
  - Essa redução é de tempo polinomial
  - Essa redução é correta, ou seja a instância do VERTEX-COVER decide sim se e somente se a instância do HAM-CYCLE decide sim.
- Portanto demonstramos que HAM-CYCLE $\in$ NP-Completo. $\square$