UNIFAL-MG
Universidade Federal de Alfenas

Projeto e Análise de Algoritmos

# NP Completude – Parte 2

Prof. Humberto Brandão
humberto@bcc.unifal-mg.edu.br

*Universidade Federal de Alfenas*
*Departamento de Ciências Exatas*
*versão da aula: 0.2*

# Última aula

UNIFAL-MG
Universidade Federal de Alfenas

- Problemas fáceis vs. Problemas difíceis:
  - O quanto podem ser parecidos;

# Última aula

UNIFAL-MG
Universidade Federal de Alfenas

- Problemas fáceis vs. Problemas difíceis:
  - O quanto **podem ser parecidos**;

- O problema da Satisfabilidade Boleana (SAT):
  - Porque é um problema tão especial;

# Última aula

UNIFAL-MG
Universidade Federal de Alfenas

- Problemas fáceis vs. Problemas difíceis:
  - O quanto **podem ser parecidos**;
- O problema da Satisfabilidade Boleana (**SAT**):
  - **Porque é um problema tão especial**;
- Importância da Redução de problemas;

# Última aula

UNIFAL-MG
Universidade Federal de Alfenas

- **Problemas fáceis vs. Problemas difíceis:**
  - O quanto **podem ser parecidos**;
- O problema da Satisfabilidade Boleana (**SAT**):
  - **Porque é um problema tão especial**;
- Importância da **Redução de problemas**;
- A classe de problemas NP-Completo e como seus elementos estão relacionados;

UNIFAL-MG
Universidade Federal de Alfenas

# Transformação Polinomial

# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;

# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;

- Considere os problemas P1 e P2;

# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;
- Considere os problemas P1 e P2;
- Suponha que existe um algoritmo para transformar qualquer instância de P1 em uma instância de P2;

# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;
- Considere os problemas P1 e P2;
- Suponha que existe um algoritmo para transformar qualquer instância de P1 em uma instância de P2;
- Suponha que conhecemos um algoritmo A2 para resolver P2;

# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;
- Considere os problemas P1 e P2;
- Suponha que existe um algoritmo para transformar qualquer instância de P1 em uma instância de P2;
- Suponha que conhecemos um algoritmo A2 para resolver P2;
- Suponha que é possível transformar o resultado de P2 em um resultado de P1;

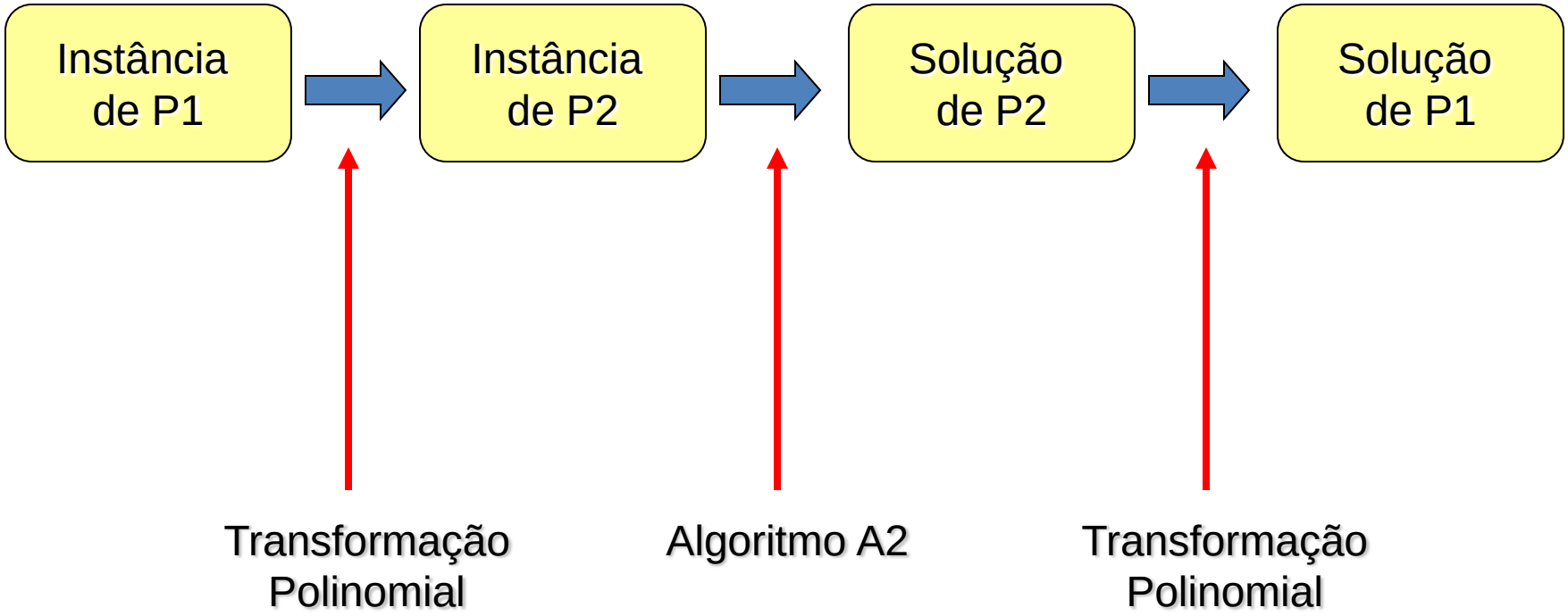
# Transformação Polinomial

UNIFAL-MG
Universidade Federal de Alfenas

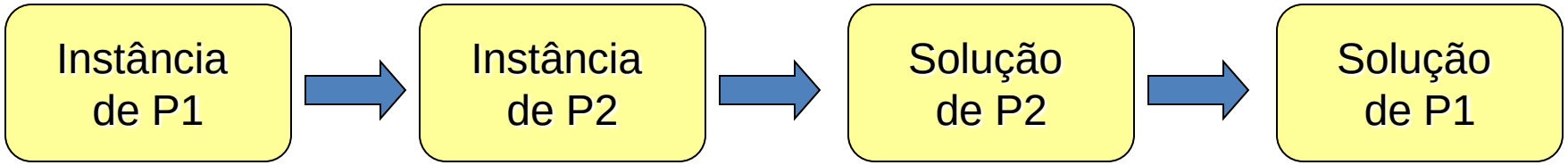
- O conceito de transformação polinomial é importante para definirmos a classe NP-Completo;
- Considere os problemas P1 e P2;
- Suponha que existe um algoritmo para transformar qualquer instância de P1 em uma instância de P2;
- Suponha que conhecemos um algoritmo A2 para resolver P2;
- Suponha que é possível transformar o resultado de P2 em um resultado de P1;
- Então, é possível resolver P1 através de A2.

# Transformação Polinomial

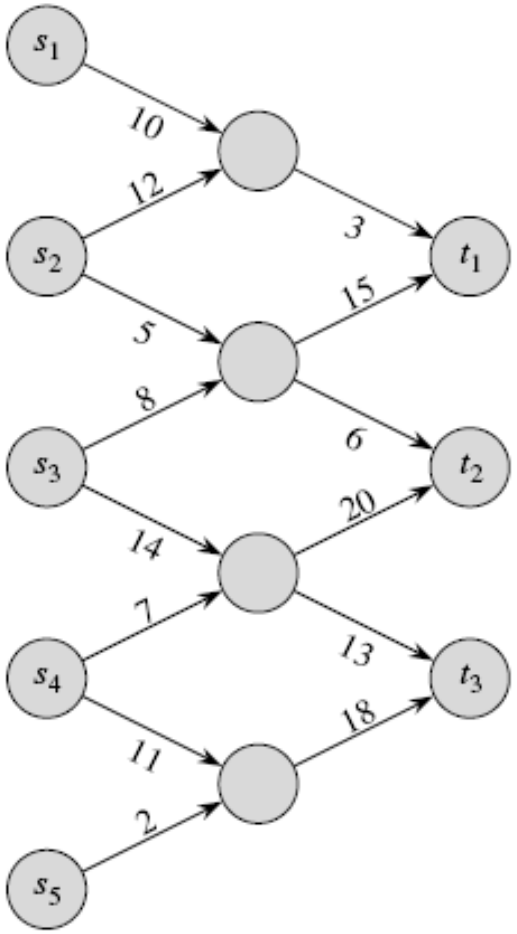
UNIFAL-MG
Universidade Federal de Alfenas



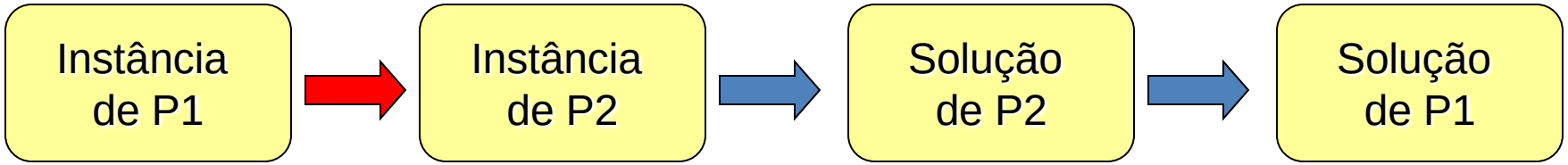
# Transformação Polinomial




- Exemplo:
  - P1: Desejamos calcular o fluxo máximo entre um conjunto de nós servidores e um conjunto de nós consumidores;
  - Não conhecemos algoritmo direto e simples para este problema...

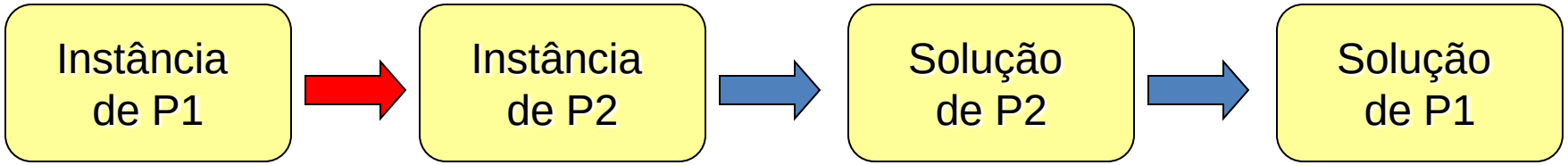


# Transformação Polinomial

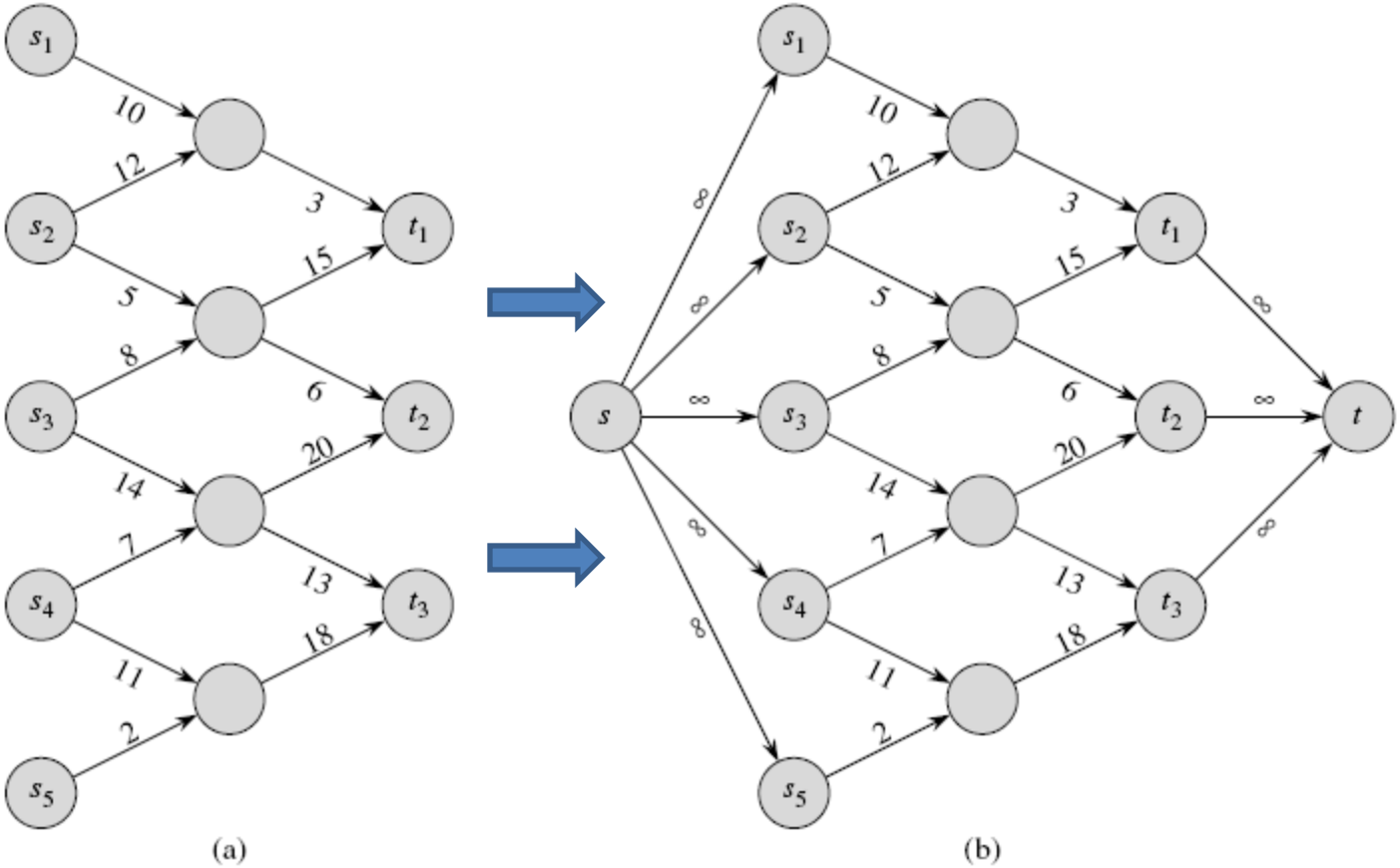



- Exemplo:
  - Podemos transformar a instância $G=(V,A)$ em uma instância modificada $G'=(V',A')$ para problema de fluxo máximo de única origem e único destino, onde conhecemos algoritmo;
  - $G'=(V',A')$, onde:
    - $V'=V \cup \{s,t\}$
    - $A''=\{(s,x, \infty) \mid x \text{ é origem em } G\}$
    - $A'''=\{(y,t, \infty) \mid y \text{ é destino em } G\}$
    - $A'=A \cup A'' \cup A'''$

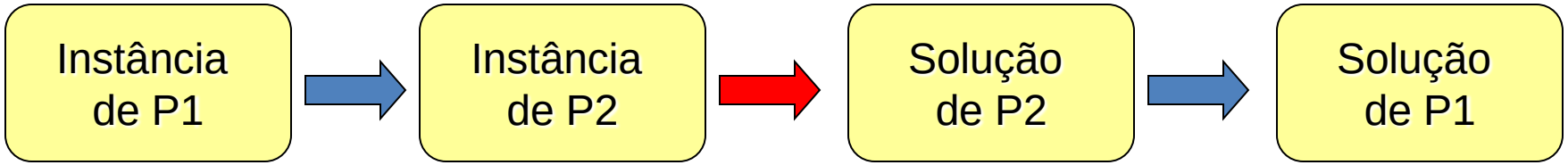
# Transformação Polinomial



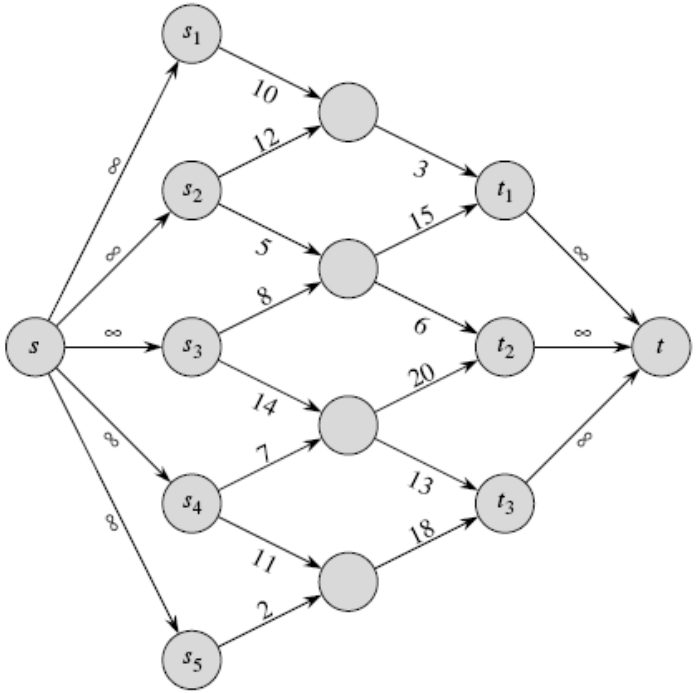

- Exemplo:



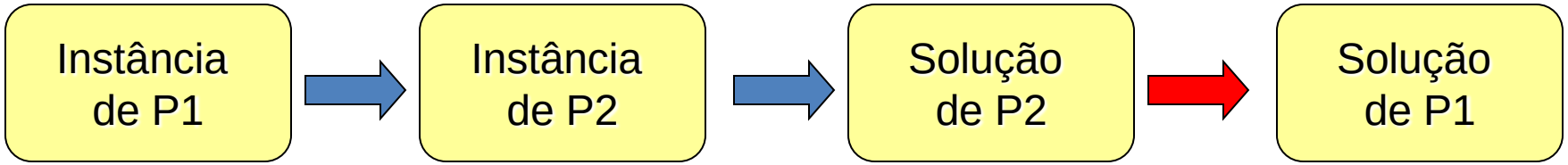
# Transformação Polinomial



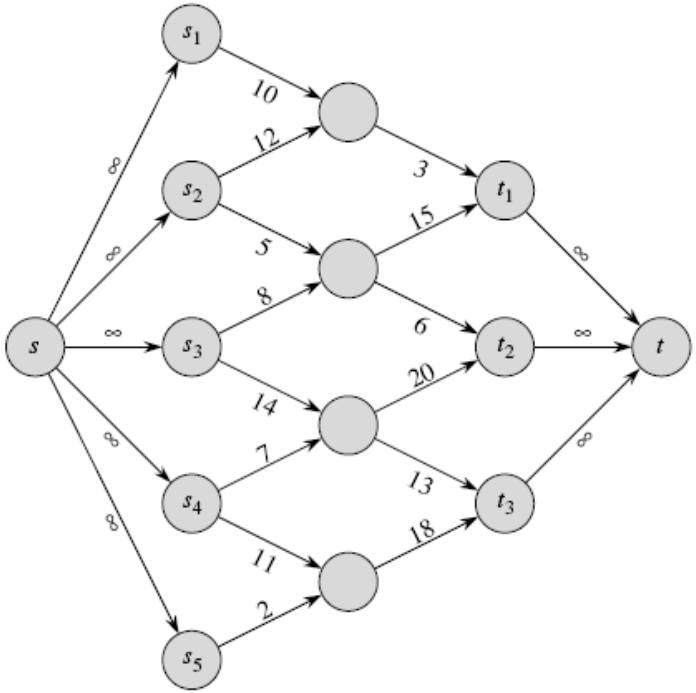

- Exemplo:
  - Com a instância de P2, podemos agora aplicar o algoritmo de fluxo máximo conhecido em P2: Por exemplo, Ford-Fulkerson;



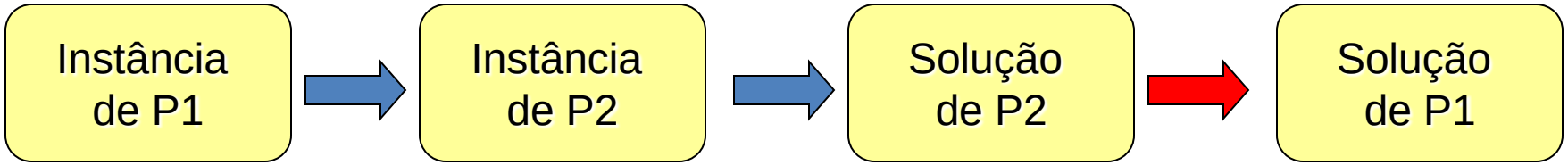
# Transformação Polinomial




- Exemplo:
  - Obtendo a solução de P2, podemos convertê-la em solução de P1, eliminando as arestas e nós criados artificialmente para P2.
  - Assim, conhecemos o fluxo máximo de P1.



# Transformação Polinomial




- Considerações:
  - Este exemplo mostrado, é a transformação de um problema polinomial em outro polinomial;
  - Ou seja, a transformação polinomial não serve apenas para o estudo da classe de problemas NP-Completo;
  - Transformando o fluxo máximo de várias origens e destinos (P1) no fluxo máximo de origem e destino único (P2), estamos dizendo:
    - P2 é no mínimo tão difícil quanto P1.

UNIFAL-MG
Universidade Federal de Alfenas

# NP-Completo

*Complementando...*

# NP-Completo

UNIFAL-MG
Universidade Federal de Alfenas

- Para provar que um problema é NP-Completo, são necessários dois passos:

# NP-Completo

UNIFAL-MG
Universidade Federal de Alfenas

- Para provar que um problema é NP-Completo, são necessários dois passos:
  - 1) Mostre que o problema está em NP:
    - apresentando pelo menos um algoritmo não determinístico polinomial para o problema; ou
    - Verificando sua solução em tempo polinomial;

# NP-Completo

UNIFAL-MG
Universidade Federal de Alfenas

- Para provar que um problema é NP-Completo, são necessários dois passos:
  - 1) Mostre que o problema está em NP:
    - apresentando pelo menos um algoritmo não determinístico polinomial para o problema; ou
    - Verificando sua solução em tempo polinomial;
  - 2) Mostre que pelo menos um problema NP-Completo pode ser reduzido diretamente ao problema estudado.
    - SAT ou qualquer outro NP-Completo.

# NP-Completo

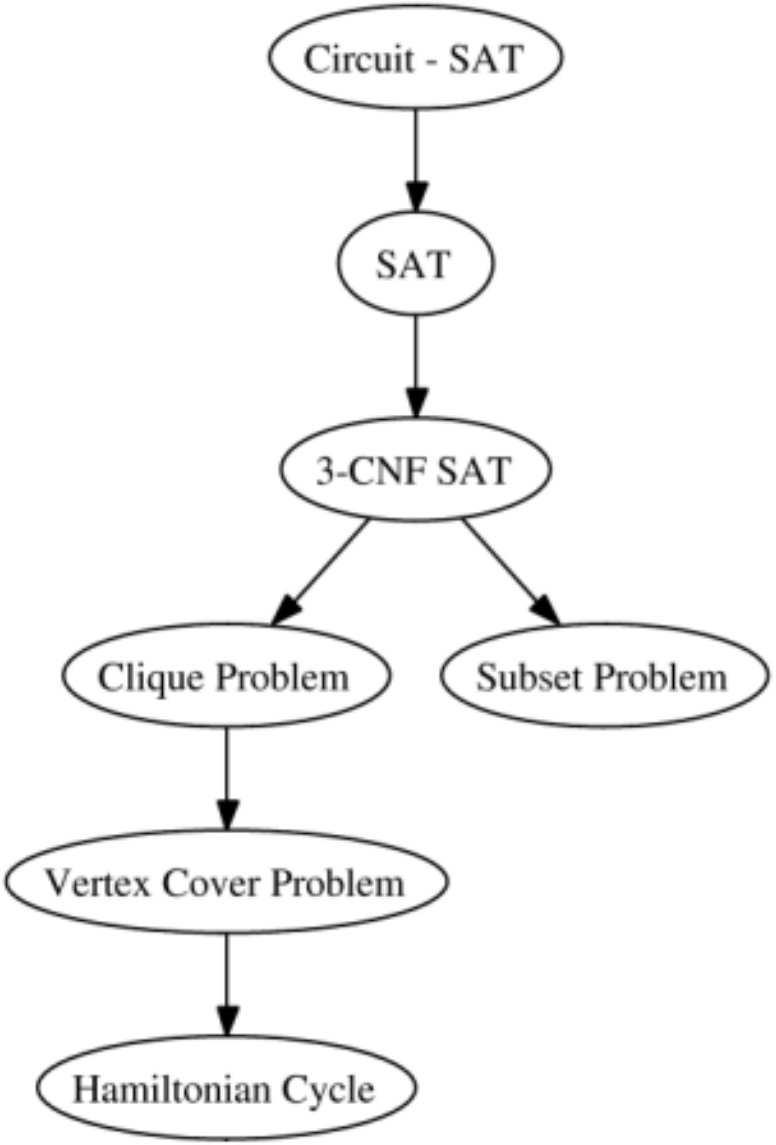
UNIFAL-MG
Universidade Federal de Alfenas

- Isso só é possível porque Cook em 1971 apresentou uma prova direta da SAT ser um problema NP-Completo, além do fato de que a redução polinomial é transitiva:
  - Se (SAT $\propto$ P1) e (P1 $\propto$ P2) então
    - SAT $\propto$ P2
  - Como temos a prova de que qualquer instância de qualquer problema com algoritmo pode ser transformado em uma instância da SAT, então, P2 $\propto$ SAT.

# NP-Completo

- Vamos supor que conhecemos os seguintes problemas na classe NP-Completo:

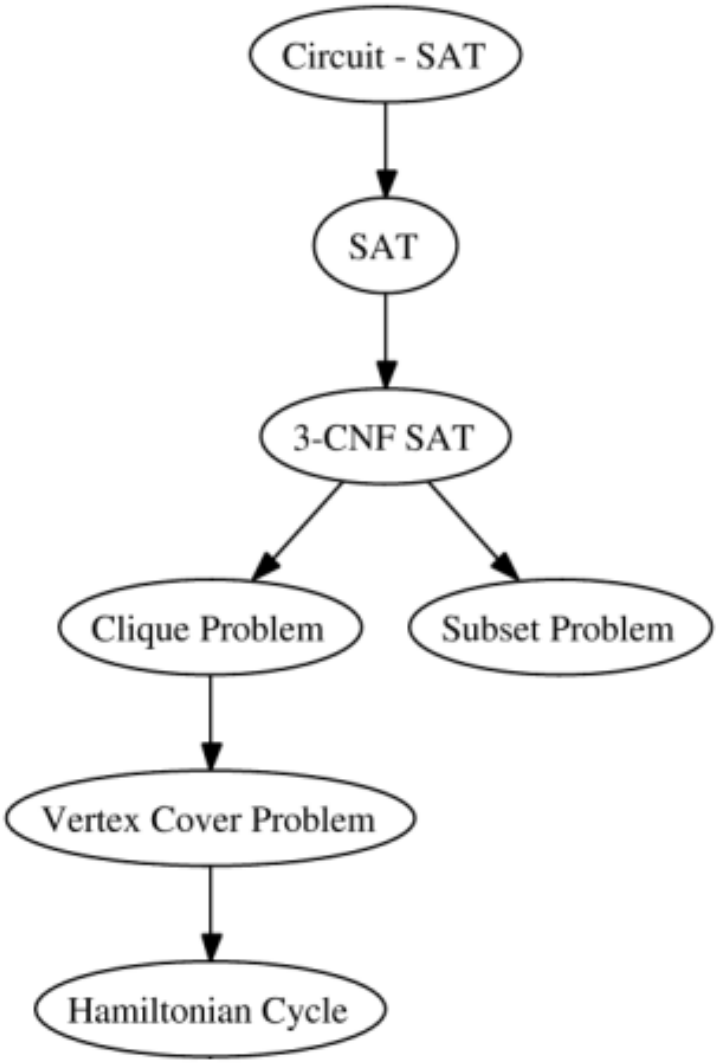
- A seta do problema P para o problema Q, indica que P pode ser reduzido em tempo polinomial ao problema Q.



# NP-Completo

- Como mostrar que o Problema do Caixeiro Viajante (PCV) é NP-Completo?

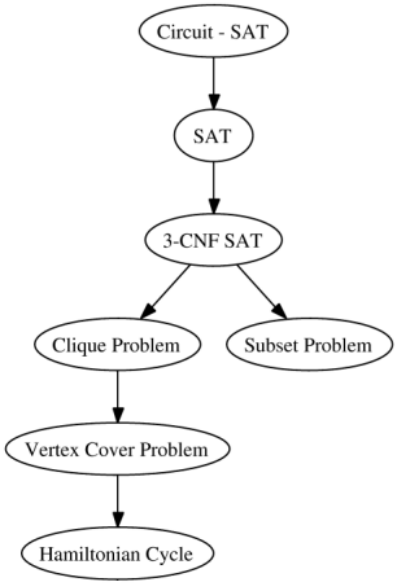
- Quais são mesmo os passos para a prova?



# NP-Completo

UNIFAL-MG
Universidade Federal de Alfenas

- Como mostrar que o PCV é NP-Completo?




- Quais são mesmo os passos para a prova?
  - 1: Mostrar que existe algoritmo ND para o PCV;
  - 2: Reduzir qualquer problema NP-Completo para o PCV;

# NP-Completo

UNIFAL-MG
Universidade Federal de Alfenas

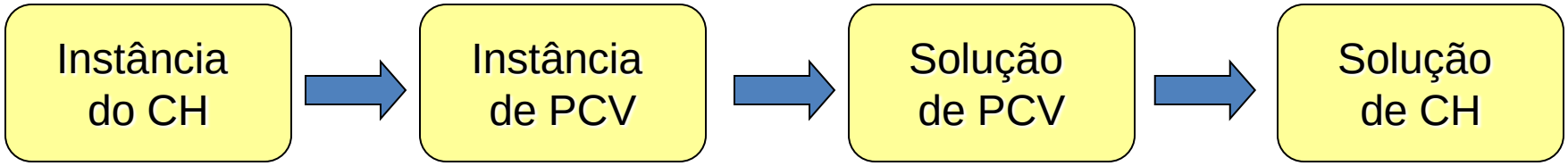
- 1: Mostrar que existe algoritmo ND para o PCV;

- Algoritmo ND para o PCV:

  - Início
    - Para todo vértice *v* do grafo G=(V,A) faça
      - Antecessor[v] ← escolhe( adjacentes(v) );
    - Fim para.
  - Fim

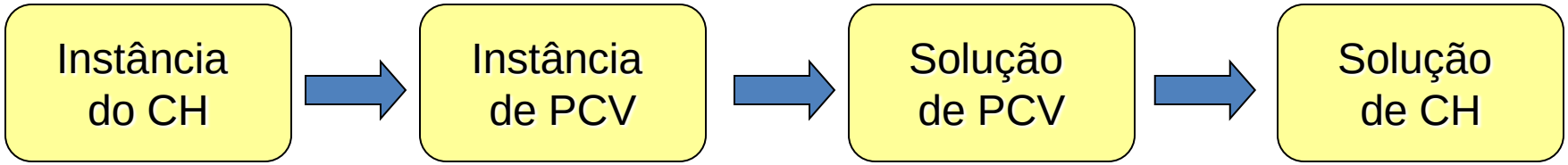
# NP-Completo

- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;



# NP-Completo

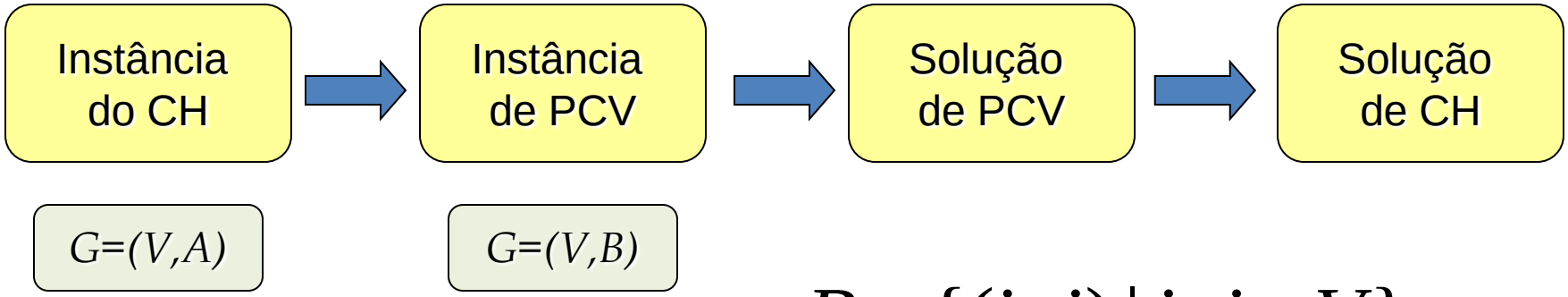
- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;




$G=(V,A)$

# NP-Completo

- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;



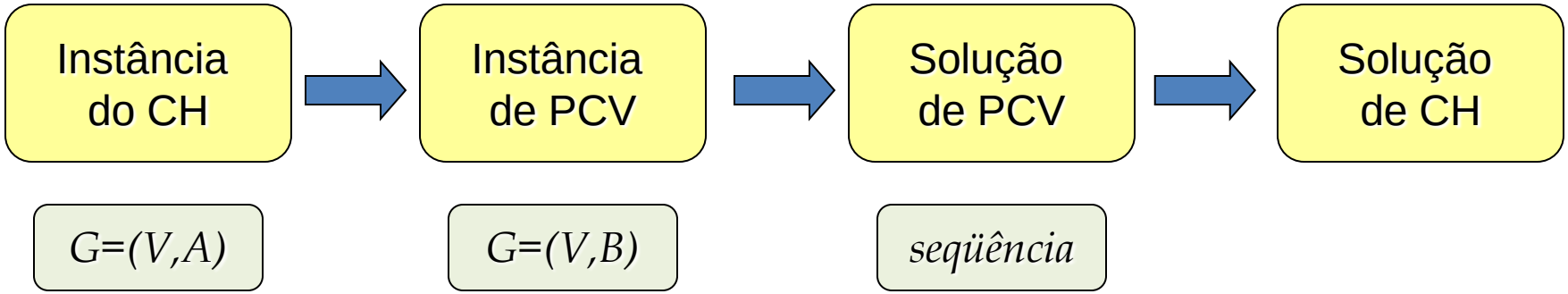

$$B = \{(i, j) \mid i, j \in V\}$$

$$c : V \times V \rightarrow \Re$$

$$c(i, j) = \begin{cases} 1, se\ (i, j) \in A \\ 2, se\ (i, j) \notin A \end{cases}$$

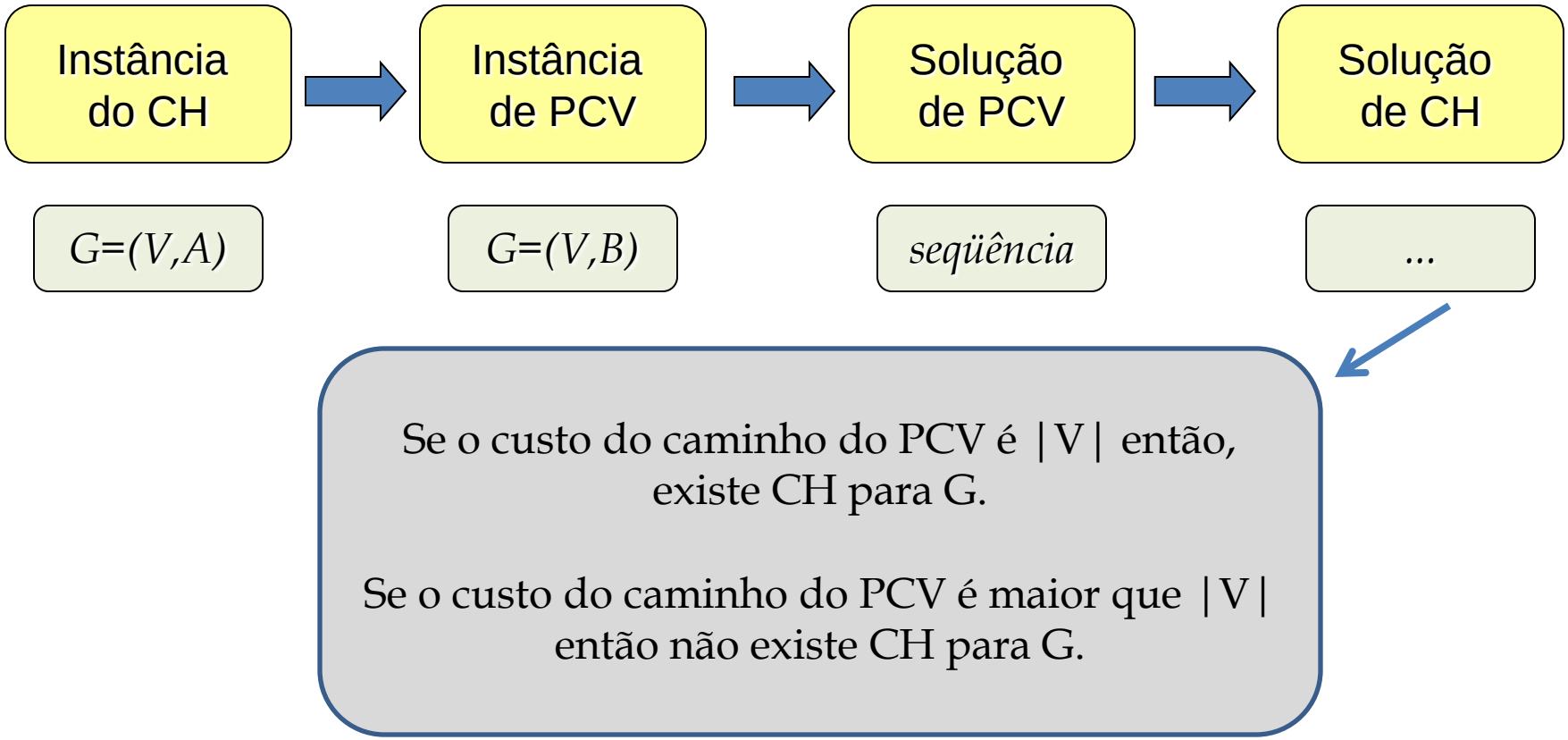
# NP-Completo

- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;



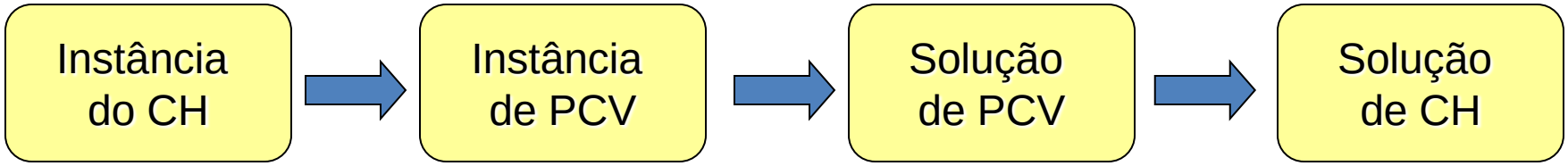
# NP-Completo

- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;



# NP-Completo

- 2: Reduzir qualquer problema NP-Completo para o PCV;
  - Vamos reduzir o Ciclo Hamiltoniano no PCV;




  - Ao resolver um problema NP-Completo através de outro problema Y, então Y também é NP-Completo, pois ele é no mínimo tão difícil quanto o problema NP-Completo.

UNIFAL-MG
Universidade Federal de Alfenas

# A classe NP-Difícil

# NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Também conhecida como NP-Árduo (*NP-Hard*);
- Para mostrar que um problema é NP-Difícil, basta satisfazer a seguinte condição:
  - Mostre que pelo menos um problema NP-Completo pode ser reduzido diretamente ao problema estudado.
- Lembrando que esta é uma condição necessária para o problema ser NP-Completo, ou seja:
  - TODO PROBLEMA NP-COMPLETO É TAMBÉM NP-DIFÍCIL.

# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

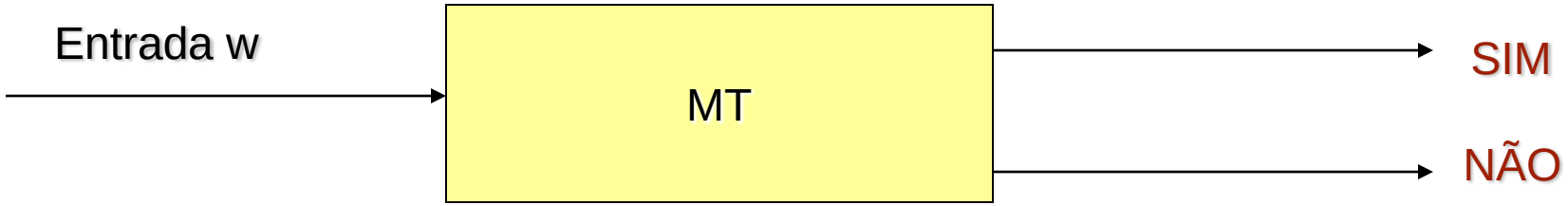
UNIFAL-MG
Universidade Federal de Alfenas

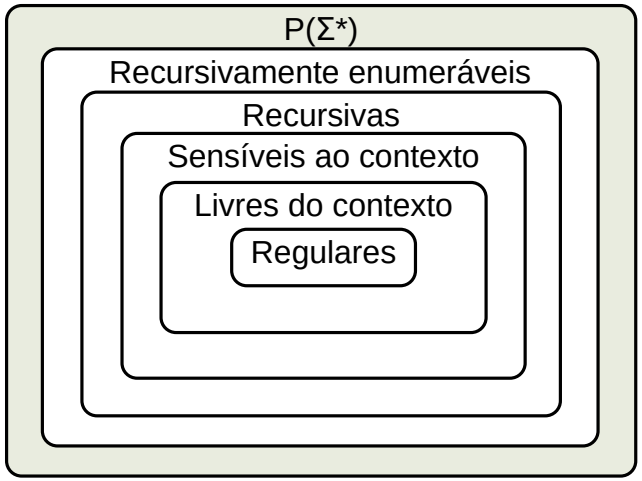
- Problema da Parada
  - Para mostrar que um problema é exclusivamente NP-Difícil, quais são os passos?
    - 1) Mostre que o problema *não* está em NP:
      - Mostrando que não existe nenhum algoritmo para ele;
    - 2) Mostre que pelo menos um problema NP-Completo pode ser reduzido diretamente ao problema estudado.

# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Problema da Parada: mostrando que o PP não está em NP.
  - Lembrando uma máquina de Turing...

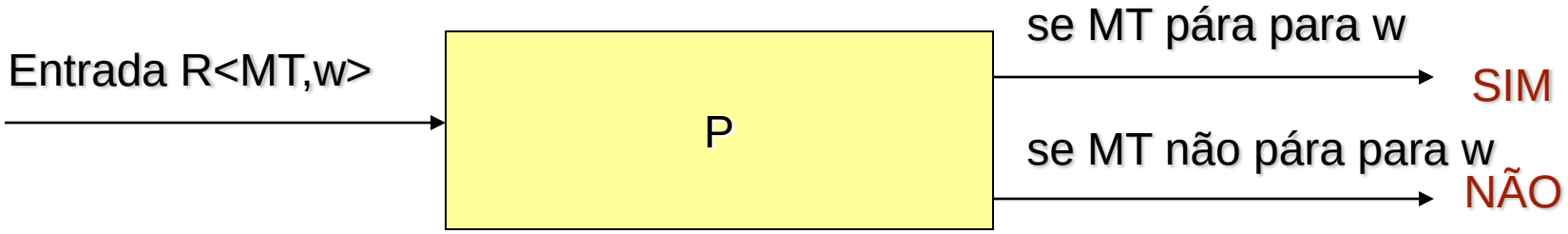


# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

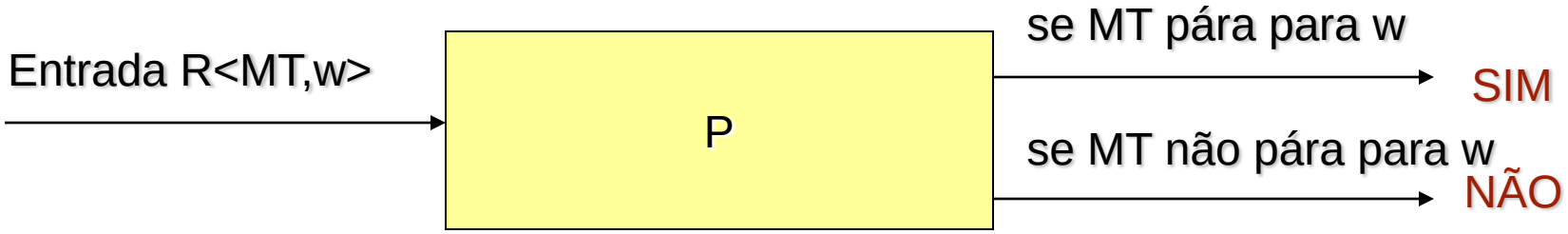
- Problema da Parada: mostrando que o Problema da Parada não está em NP.
  - Vamos supor que Problema da Parada seja decidível (existe algoritmo para ele);



# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

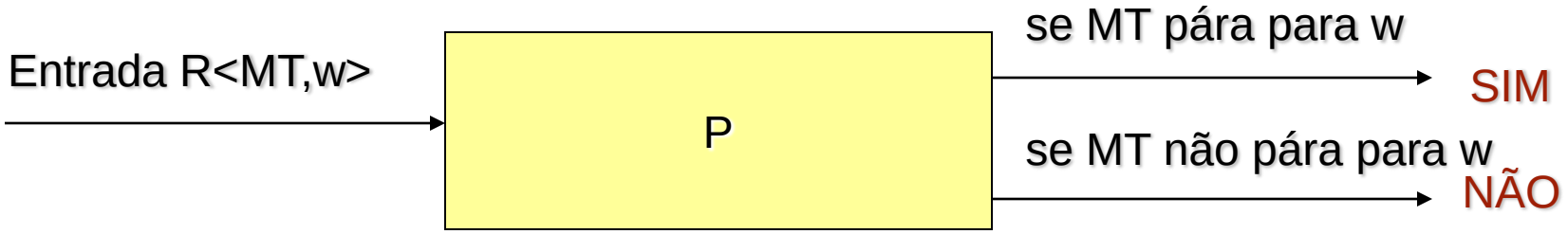
- Problema da Parada: mostrando que o Problema da Parada não está em NP.
  - Vamos supor que Problema da Parada seja decidível (existe algoritmo para ele);
  - Tal algoritmo, será aqui representado pela MT P;
    - Entrada 1: algoritmo;
    - Entrada 2: argumentos.

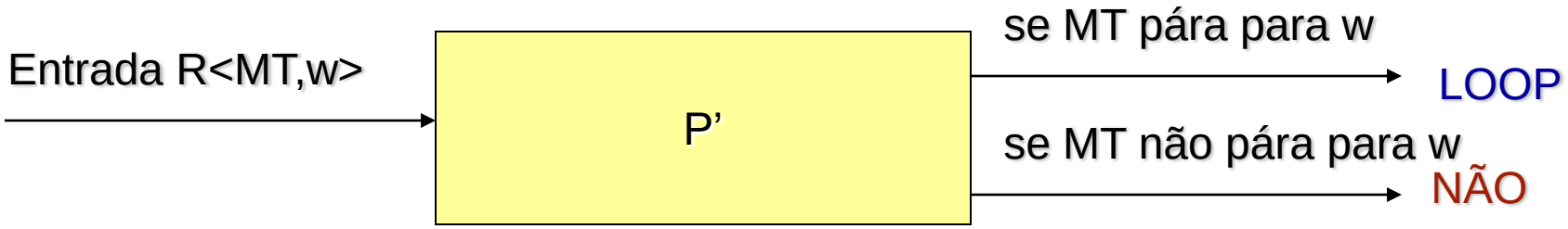


# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Problema da Parada: mostrando que o PP não está em NP.
  - Se conhecemos P, podemos criar P', com uma pequena modificação;

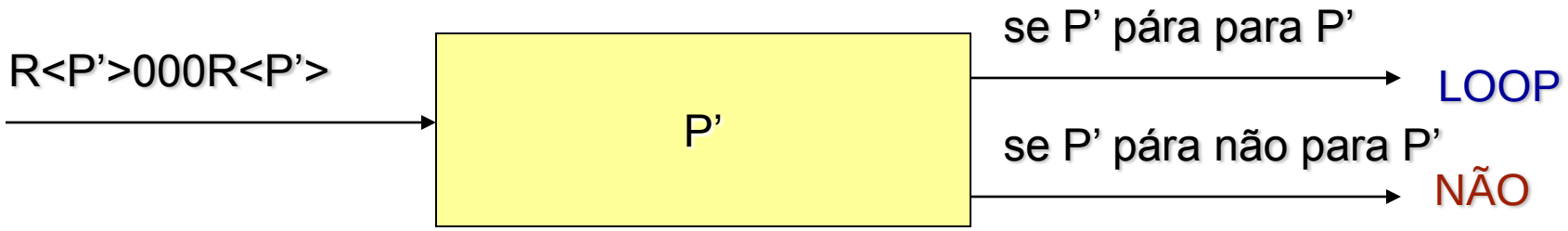


# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

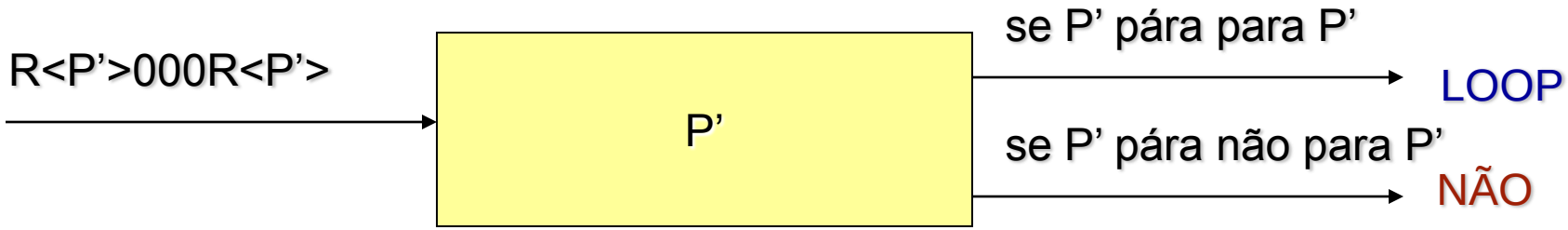
- Problema da Parada: mostrando que o PP não está em NP.
  - P', como é uma máquina de *turing*, pode ser codificada e servir como entrada para a própria P'; (emulação da emulação)



# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Problema da Parada: mostrando que o PP não está em NP.
  - P', como é uma máquina de *turing*, pode ser codificada e servir como entrada para a própria P'; (emulação da emulação)

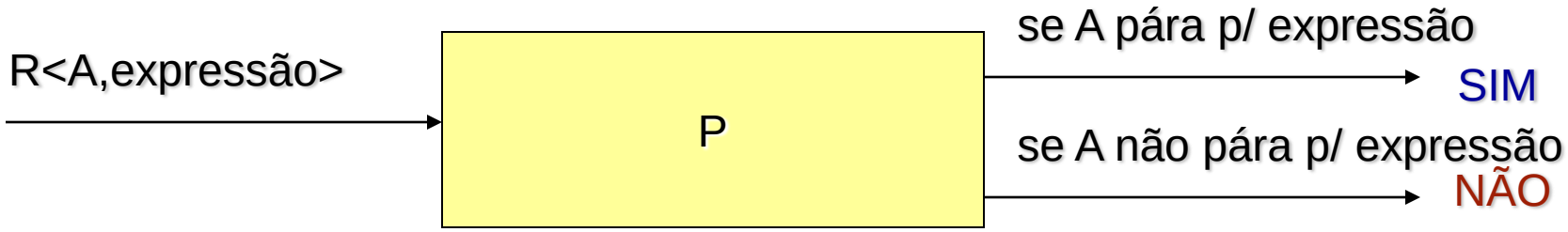



  - Contradição!!!
  - Portanto o problema da parada é indecidível. Não existe MT para ele. Portanto não existe algoritmo para ele;

# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Problema da Parada: mostrando SAT pode ser reduzida ao PP
  - Considere o algoritmo *A* cuja entrada é uma expressão booleana na forma normal conjuntiva com *n* variáveis;
  - Basta tentar $2^n$ possibilidades e verificar se é satisfatível.
    - Se a expressão for SAT, A pára;
    - Senão, entra em *loop* (não pára).




  - Logo, o problema da parada é NP-difícil, mas não é NP-completo.

# NP-Difícil
# Exemplo de Problema exclusivamente NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Problema da Parada:
  - O problema da parada é no mínimo tão difícil quanto o problema da SAT.
  - O problema da parada é no mínimo tão difícil quanto qualquer problema.

# NP-Difícil

UNIFAL-MG
Universidade Federal de Alfenas

- Podemos mostrar que o problema da parada pode ser reduzido a outros problemas;
  - O que isso quer dizer?
  - Em que classe de problemas este outro problema está?

# Bibliografia


- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos – Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus.

- TAMASSIA, ROBERTO; GOODRICH, MICHAEL T. (2004). Projeto de Algoritmos - Fundamentos, Análise e Exemplos da Internet.

- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;

- VIERA, N. J. "Introdução aos Fundamentos da Computação: Linguagens e Máquinas". Editora: Thomson Learning

- Material de aulas do Professor Loureiro (DCC-UFMG)
  - http://www.dcc.ufmg.br/~loureiro/paa/