Homomorphisms in graphs

Jefferson Elbert Simões

Artigos:

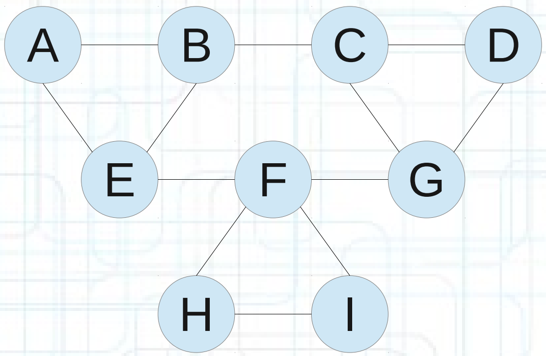
Algorithmic Aspects of Graph Homomorphisms (Hell, 2003)

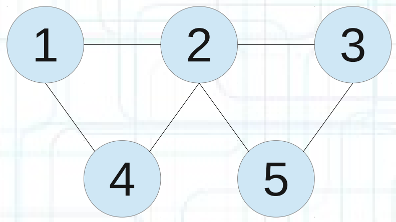
Graph Homomorphism Revisited for Graph Matching (Fan et al., 2010)

Homomorfismos

- Dados dois grafos G e H, um homomorfismo é uma função $f : V(G) \rightarrow V(H)$ que mapeia todas as arestas de G em arestas de H
  - Nenhuma restrição sobre "não-arestas"
  - Grafos podem ser loops
- Definição equivalente em grafos direcionados ou não-direcionados
  - Caso não-direcionado é mais geral
- H impõe restrições sobre o "papel" de vértices de G

Homomorfismos



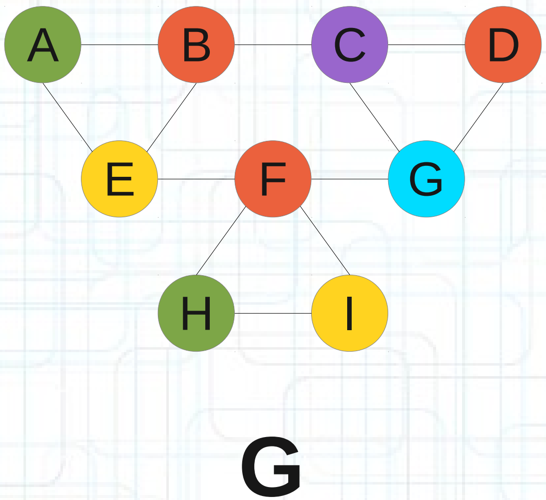


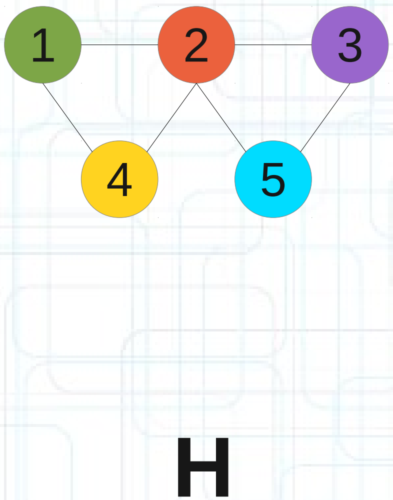
G

H

f(A) = f(H) = 1
f(B) = f(D) = f(F) = 2
f(C) = 3
f(E) = f(I) = 4
f(G) = 5

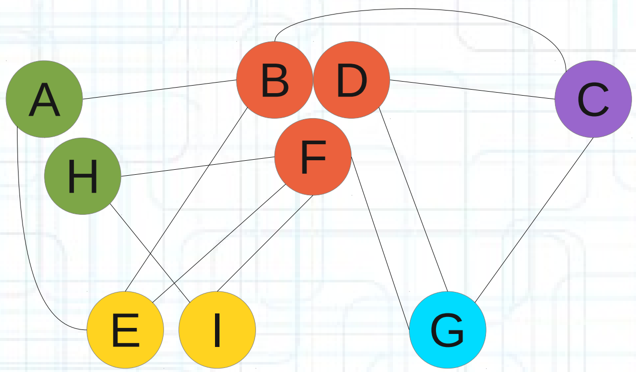
Homomorfismos



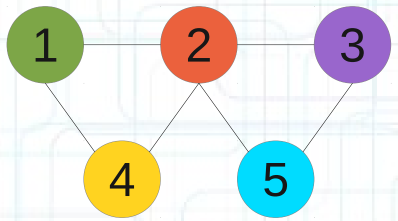


f(A) = f(H) = 1
f(B) = f(D) = f(F) = 2
f(C) = 3
f(E) = f(I) = 4
f(G) = 5

Homomorfismos



G

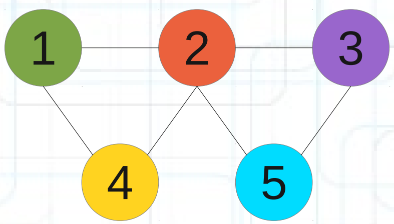


H

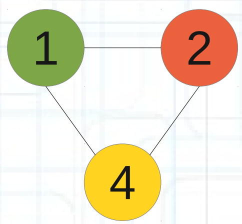
f(A) = f(H) = 1
f(B) = f(D) = f(F) = 2
f(C) = 3
f(E) = f(I) = 4
f(G) = 5

Homomorfismos

- Se H é subgrafo de G, então H é G-colorível
- Todo grafo H possui um subgrafo induzido H' ao qual é *homomorficamente equivalente*
  - H' é o núcleo de H



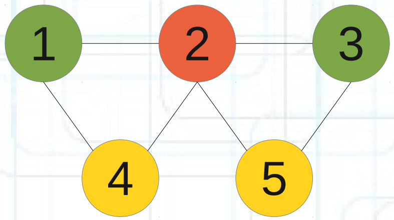
H

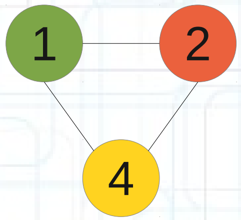


H'

Homomorfismos

- Todo grafo H possui um subgrafo induzido H' ao qual é *homomorficamente equivalente*
  - H' é o núcleo de H





H **H'**

Homomorfismos

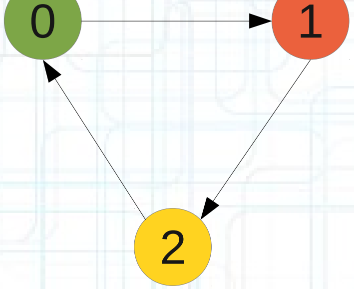
- Problema: fixo um grafo H, determinar, para um grafo de entrada G, se existe um homomorfismo de G em H
  - H-homomorfismo ou H-coloração
  - Satisfação de restrições (com template H)

- Qual a relação entre o grafo H e a complexidade do problema de H-coloração?
  - Depende da estrutura de H (de seu núcleo)

H-coloração

- Caso simples: grafos não-direcionados
- Dicotomia:
  - Se H é bipartido ou tem um loop, H-coloração é um problema em P
    - Núcleo K_1: grafos sem aresta
    - Núcleo K^*_1: todos os grafos
    - Núcleo K_2: grafos bipartidos (sem ciclo impar)
  - Senão, P é NP-completo

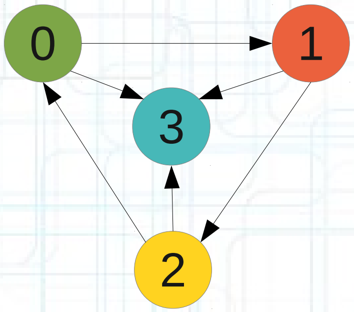
$\overrightarrow{C}_3$-coloração



- Algoritmo eficiente para colorir G:

- Escolha um vértice e dê cor 0
- Se v possui cor i:
  - Dê cor i+1 (mod 3) aos j com $i \rightarrow j$
  - Dê cor i-1 (mod 3) aos j com $j \rightarrow i$
- Algoritmo constrói uma $\overrightarrow{C}_3$-coloração
- Se houver algum conflito de cor, o grafo G não é $\overrightarrow{C}_3$-colorível

H_2-coloração

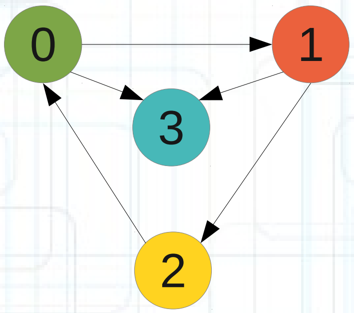


- Algoritmo eficiente para colorir G:

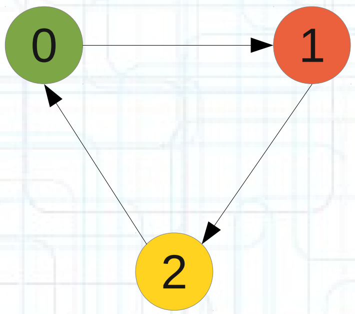
- Dê cor 3 a todos os sumidouros
- Remova os sumidouros
- Use o algoritmo de $\overrightarrow{C_3}$-coloração no grafo restante

H_1-coloração

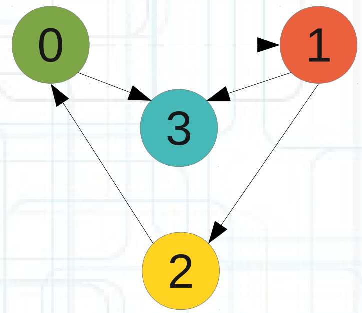
- Problema NP-completo



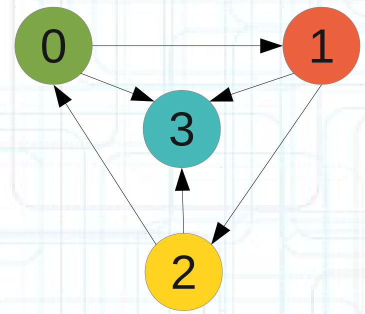
H-coloração



- P



- NP-completo



- P

- Existem grafos H tais que H-coloração está em P, mas para um H' subgrafo induzido, H'-coloração é NP-completo

- Nenhuma conjectura sobre dicotomia foi formulada até hoje

H-coloração

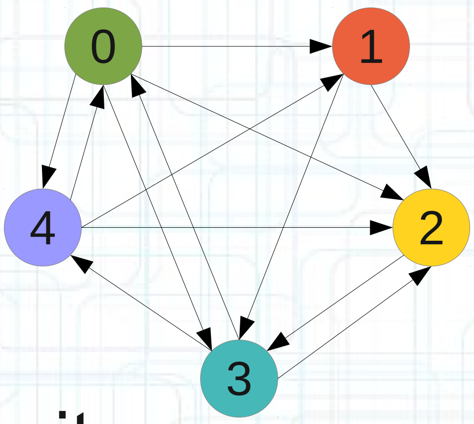
- O estudo de H-colorações é uma grande área de pesquisa em aberto
- Por que é uma área importante?

H-coloração

- O estudo de H-colorações é uma grande área de pesquisa em aberto
- Por que é uma área importante?

- Todo problema de satisfação de restrições é polinomialmente equivalente ao problema de H-coloração, para algum grafo H
  - Estudo de classes de grafos

H-coloração



- Exemplo 1: H digrafo semi-completo
  - Se H tem no máximo um circuito: problema em P
  - Senão: problema NP-completo
- Exemplo 2: H sem fontes e sumidouros
  - Se o núcleo de H é um ciclo orientado: problema em P
  - Senão: problema NP-completo

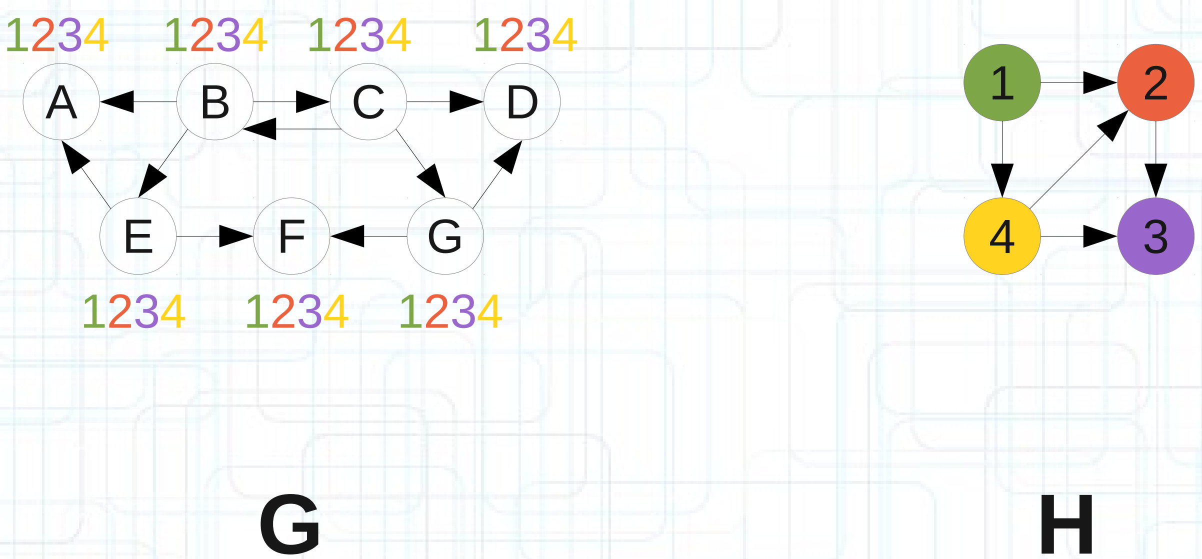
Dualidade

- Alguns dos algoritmos evidenciam uma dualidade na H-coloração
- Exemplo: se G não é $\overrightarrow{C_3}$-colorível, então o algoritmo encontra algum circuito de tamanho n (com n = 1,2 (mod 3)) – logo, $\overrightarrow{C_n}$ é G-colorível

Consistência

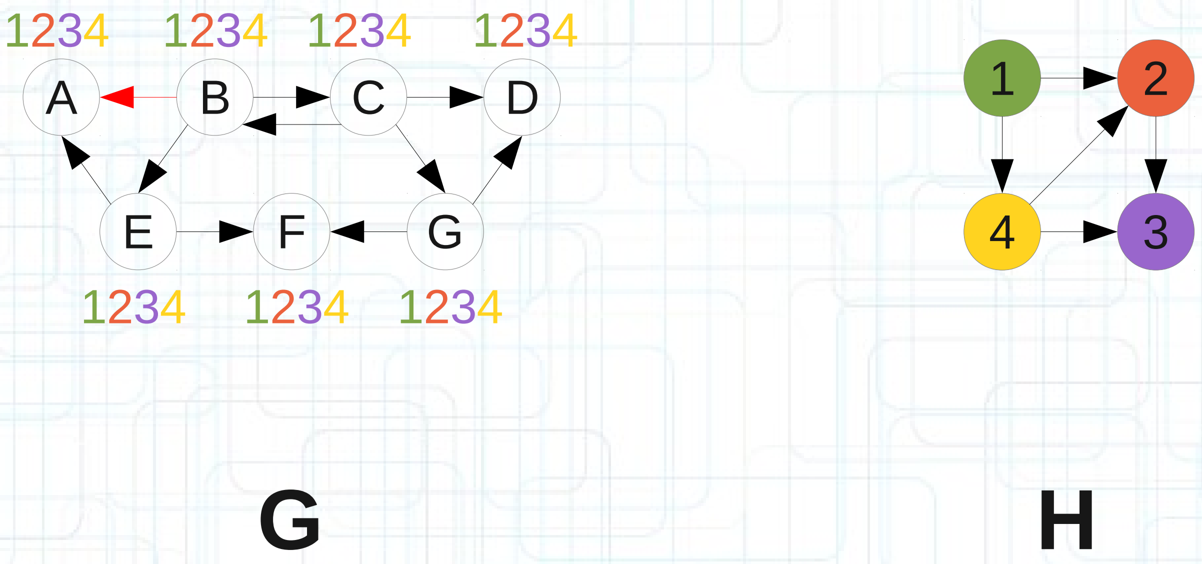
- Uma maneira de mostrar que G não é H-colorível é argumentar que algum vértice de G, em qualquer H-coloração, tem que ficar “sem cor”
- Para isto, daremos a cada vértice de G uma lista de cores aceitáveis
- Algoritmo: para cada aresta, remover cores conflitantes até que o conjunto seja consistente
  - Sudoku-style :)

Consistência



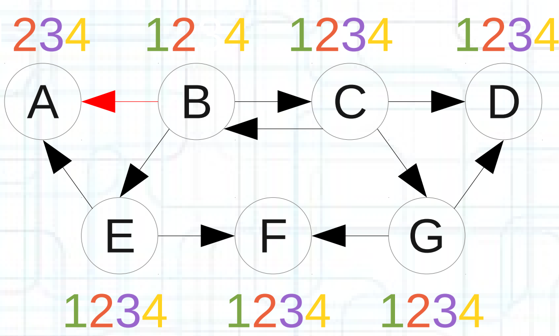
- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência

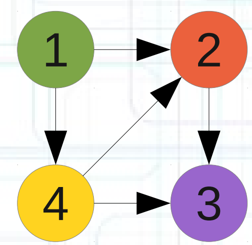


- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



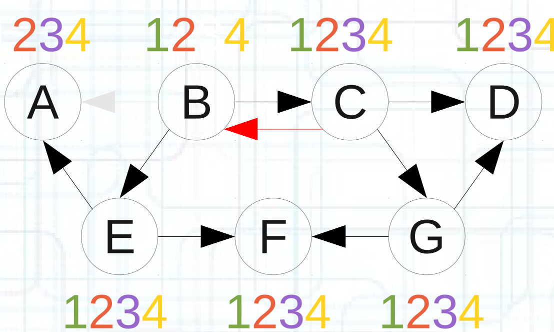
G



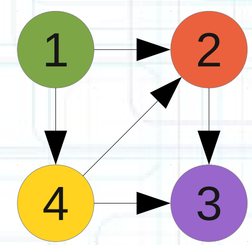
H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



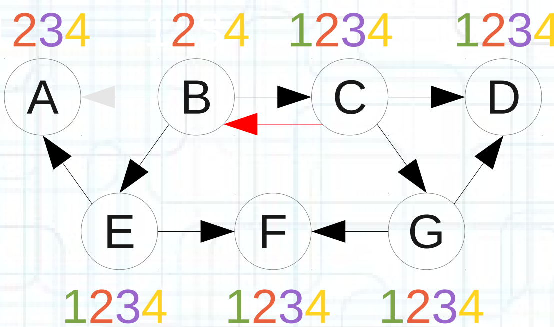
G



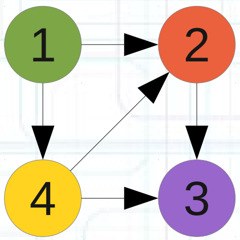
H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



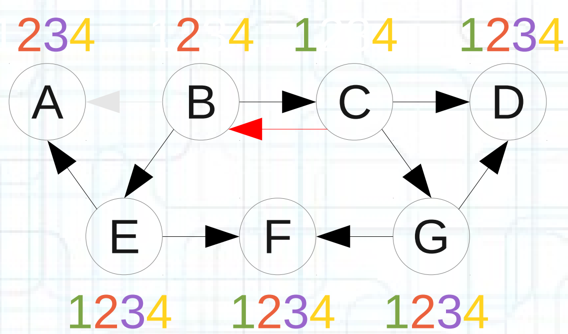
G

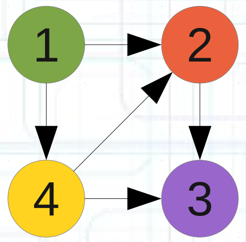


H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência

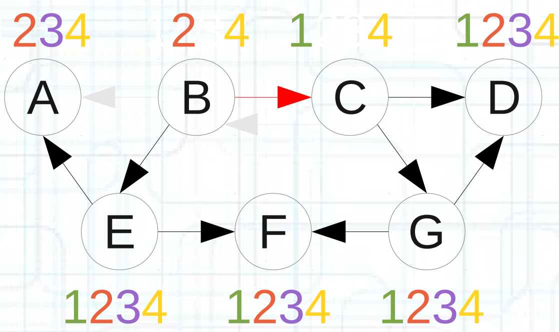




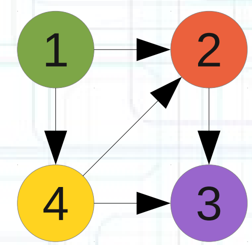
G **H**

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



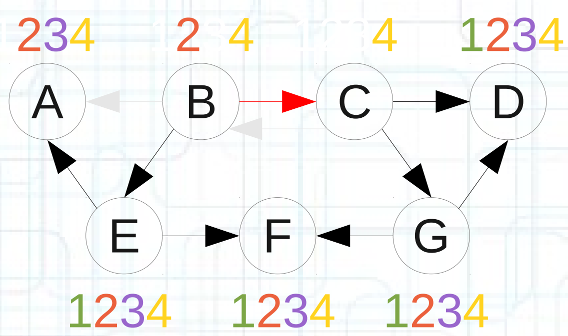
G



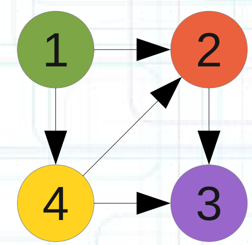
H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



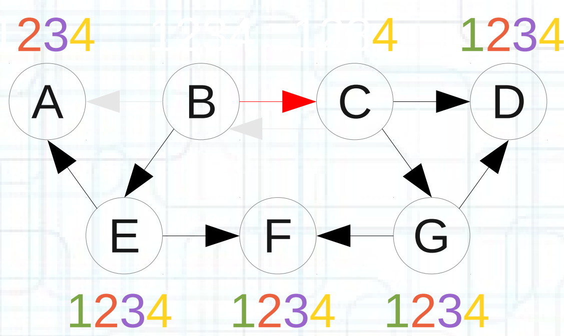
G



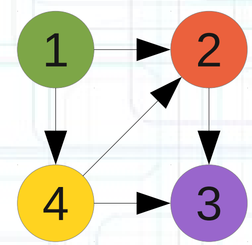
H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência



G



H

- Para cada aresta uv de G
  - Remover da lista de u as cores que não são antecedentes de cores de v
  - Remover da lista de v as cores que não são consequentes de cores de u

Consistência

- Lista vazia implica G não-H-colorível
  - Recíproca não é verdadeira no caso geral
- Para determinadas classes de grafos H, a recíproca é verdadeira
  - Exemplo: H possui a propriedade X-underbar
    - Se $i \to j$ em H e $i' \to j'$ é aresta de H, então $\min(i,i') \to \min(j,j')$ em H
  - Basta escolher as menores cores disponíveis para cada vértice de G