GRAFOS – 25/2

Ciência da Computação
Universidade do Vale do Itajaí – UNIVALI

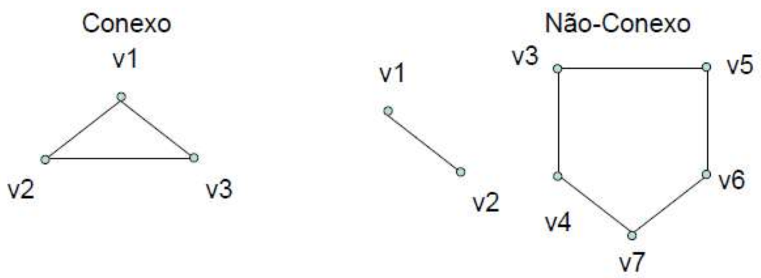
Profª Fernanda dos Santos Cunha
fernanda.cunha@univali.br

1

Grafos: Unidade 3 Conexidade

Grafo Conexo

- Um grafo G=(V,A) *não orientado* é dito ser conexo (ou S_Conexo), se há pelo menos uma sequência de arestas ligando cada par de vértices do grafo G.

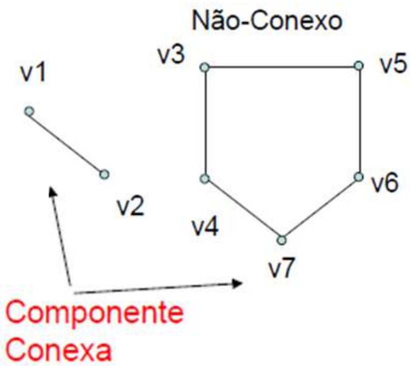


2

Grafos: Unidade 3 Conexidade

Grafo Não Conexo

- Um grafo não conexo consiste de dois ou mais subgrafos conexos (componentes conexas).

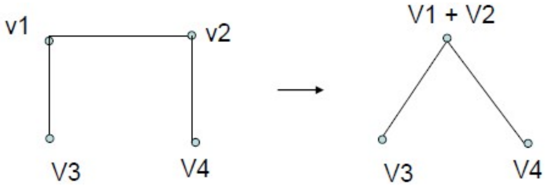


3

Grafos: Unidade 3 Conexidade

Algoritmo para Conexidade

- **Fusão:** dados dois vértices u e v adjacentes pertencentes a um grafo G, a fusão de v com u é realizada eliminando-se a aresta (v,u) que liga os dois vértices e em seguida tornando v e u um único vértice w. O vértice resultante desta fusão é um vértice w adjacente a todos os vértices anteriormente adjacentes a u e/ou v.

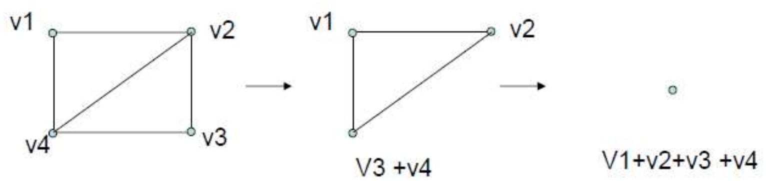


4

Grafos: Unidade 3 Conexidade

Algoritmo para Conexidade

- **Fusão (cont.):** o processo de redução deve ser feito até que todos os vértices adjacentes a um dado vértice são fundidos com ele, obtendo-se uma componente conexa.
- Quando houver apenas uma componente conexa o grafo será conexo.



5

Grafos: Unidade 3 Conexidade

Algoritmo para Conexidade

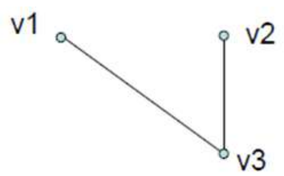
- OBS.: qualquer grafo simples com n vértices e mais que (n-1)(n-2)/2 arestas é conexo.

N=3

Num_arestas > (3-1)*(3-2)/2

Num_arestas > (2*1)/2

Num_arestas **> 1**



6

Algoritmo para Conexidade de Goodman

Verifica se um grafo é conexo e identifica seus componentes quando é desconexo.

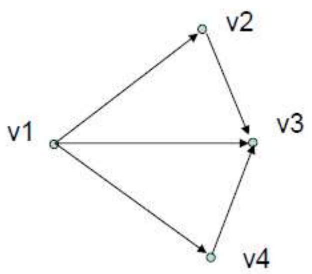
```
P0 [inicialização]
H=Vg; // todos os vértices
c=0;
P1[gere a próxima componente conexa]
Enquanto (H!=vazio){
   selecione um vértice v0 pertencente a H
   Enquanto (v0 for adjacente a algum vértice v pertencente a H){
     H=grafo resultante da fusão de v com v0;
   }
   remova v0, isto é, faça     H=H-v0;
                               c=c+1;
}
P2[Testa o tipo de conexidade]
Se(c>1)
   G é não conexo
senao
   G é conexo
```

7

Grafos: Unidade 3 Conexidade

Sucessor de um vértice

- É todo Vj que seja extremidade final de uma aresta que termina em Vi. Sua notação é $\Gamma^{+}(v_i)$.



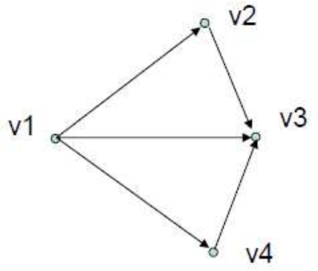
Sucessores de v1= {v2,v3,v4}

8

Grafos: Unidade 3 Conexidade

Antecessor de um vértice

- É todo Vj que seja extremidade inicial de uma aresta que termina em Vi. Sua notação é $\Gamma^{-}(v_i)$.



Antecessores de v3= {v1,v2,v4}

9

Grafos: Unidade 3 Conexidade

Fecho Transitivo Direto

- Um fecho transitivo direto de um vértice Vi é o conjunto de todos os vértices que podem ser atingidos a partir de Vi, em um número qualquer de etapas. Sua notação é $\hat{\Gamma}^{+}(v_i)$

$$\hat{\Gamma}^{+}(v_i) = \{v_i\} \cup \left[\bigcup_{k=1}^{n} \hat{\Gamma}^{+k}(v_i)\right]$$

$$\Gamma^{+}(v_i) = \Gamma^{+}(v_i)$$

$$\Gamma^{+2}(v_i) = \Gamma^{+}(\Gamma^{+}(v_i))$$

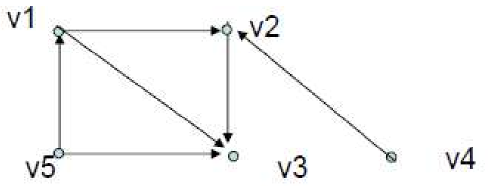
$$\ldots$$

$$\Gamma^{+n}(v_i) = \Gamma^{+}(\Gamma^{+(n-1)}(v_i))$$

10

Grafos: Unidade 3 Conexidade

Fecho Transitivo Direto



$$\hat{\Gamma}^{+}(v_1) = \{v_1, v_2, v_3\}$$

$$\hat{\Gamma}^{+}(v_5) = \{v_5, v_1, v_3, v_2\}$$

11

Grafos: Unidade 3 Conexidade

Fecho Transitivo Inverso

- Um fecho transitivo inverso de um vértice Vi é o conjunto de todos os vértices que podem atingir Vi, em um número qualquer de etapas.

  Sua notação é $\hat{\Gamma}^{-}(v_i)$

$$\hat{\Gamma}^{-}(v_i) = \{v_i\} \cup \left[\bigcup_{k=1}^{n} \Gamma^{-k}(v_i) \right]$$

$$\Gamma^{-}(v_i) = \Gamma^{-}(v_i)$$

$$\Gamma^{-2}(v_i) = \Gamma^{-}(\Gamma^{-}(v_i))$$

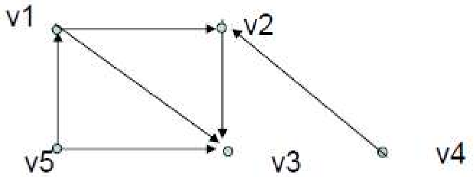
$$\dots$$

$$\Gamma^{-n}(v_i) = \Gamma^{-}(\Gamma^{-(n-1)}(v_i))$$

12

Grafos: Unidade 3 Conexidade

Fecho Transitivo Inverso



$$\hat{\Gamma}^{-}(v_1) = \{v_5, v_1\}$$

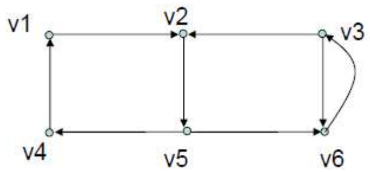
$$\hat{\Gamma}^{-}(v_5) = \{v_5\}$$

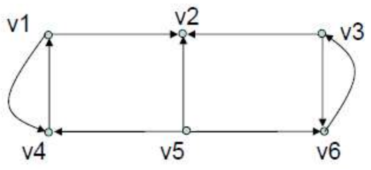
13

Grafos: Unidade 3 Conexidade

Grafo Fortemente Conexo

- No caso de *grafos orientados*, um grafo é dito ser fortemente conexo (F_Conexo) se todo par de vértices ou seja, se cada par de vértices participa de um circuito. Isto significa que cada vértice pode ser alcançável partindo-se de qualquer outro vértice de grafo.



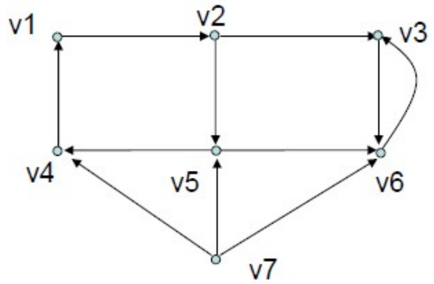


14

Grafos: Unidade 3 Conexidade

Componente Fortemente Conexa

- Um grafo G=(V,A) que **<u>não é</u>** *fortemente conexo* é formado por pelo menos dois subgrafos F_Conexos (fortemente conexo). Cada um desses *subgrafos será uma componente fortemente conexa* de G.



15

Grafos: Unidade 3 Conexidade

Algoritmo de Roy para Componentes Conexas

- Este algoritmo **encontra as componentes fortemente conexas de um grafo G dirigido** através das relações de vizinhança (Roy,1969).
- Objetiva identificar conjuntos de vértices que possuem sucessores e antecessores comuns.
- Se G é um grafo <u>não dirigido conexo</u>, então todos os vértices de G são sucessores e antecessores em G. Logo, este grafo possuirá apenas uma componente fortemente conexa.

16

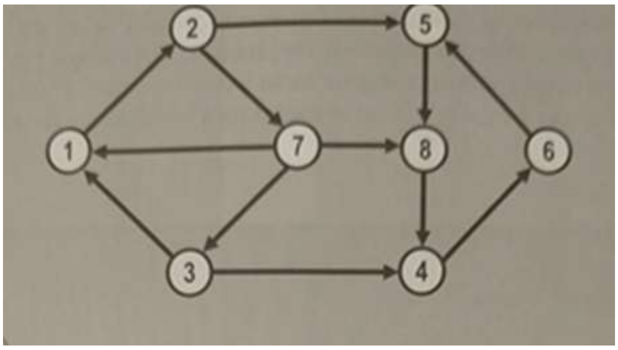
Algoritmo de Roy para Componentes Conexas

```
Ler G=(N,M) {direcionado}      i=0    V=N
Enquanto(V!=Ø)Faça
  Escolher e marcar um vértice v qq, v∈V, com(+)e(-)
  Enquanto for possível marcar com(+) um vértice w não
  marcado com(+) que tenha como sucessor um vértice
  marcado com(+)
     Marcar w com(+)
  Fimenquanto
  Enquanto for possível marcar com(-) um vértice w não
  marcado com(-) que tenha como antecessor um vértice
  marcado com(-)
     Marcar w com(-)
  Fimenquanto
  i=i+1
  Si=vértices marcados com(+)e(-) simultaneamente
  V=V/Si
  Desmarcar todos os vértices de V
Fimenquanto
```

17

Algoritmo de Roy para Componentes Conexas

Exemplo: GRAFO ORIGINAL

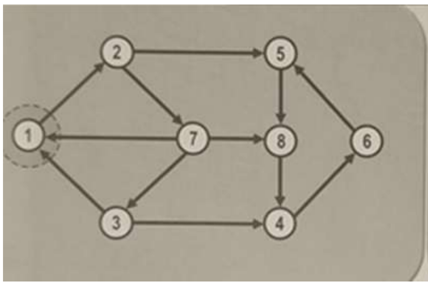


18

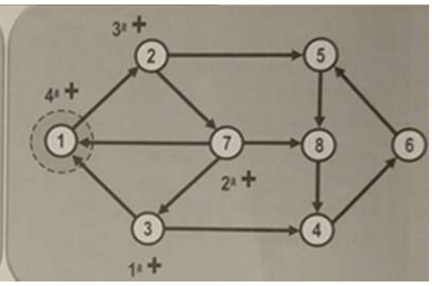
Algoritmo de Roy para Componentes Conexas

O algoritmo inicia no vértice 1, marcando (+) e (–).
E a partir daí, busca os vértices w que tem sucessores marcados (+) para achar a rotulação positiva.

Primeiro vértice de exame



Rotulação positiva v1

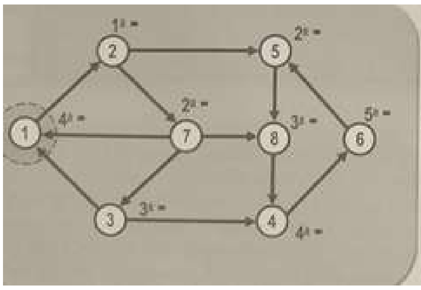


19

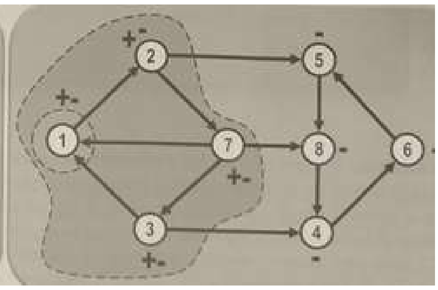
Algoritmo de Roy para Componentes Conexas

Feito a rotulação positiva, o algoritmo inicia também pelo 1 a busca de vértices w q tem antecessores marcados (-) para achar a rotulação negativa.
Feitos estes processos, chega-se a componente fortemente conexa após a 1ª iteração do laço mais externo, tal componente é dada por S1={1,2,3,7}, em destaque na 2ª fig.

Rotulação negativa v1



Vértices com as 2 rotulações

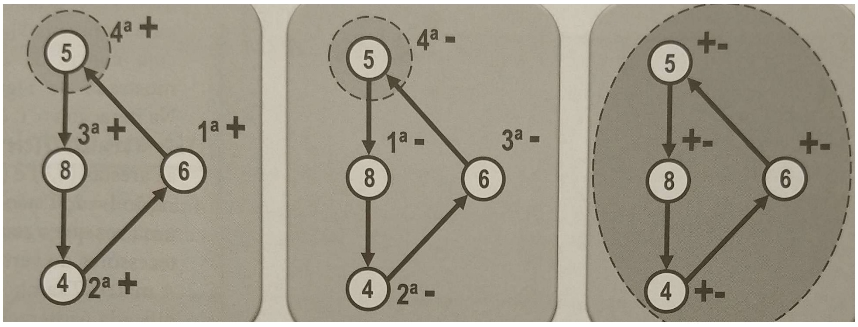


20

Algoritmo de Roy para Componentes Conexas

Os vértices restantes são desmarcados e o algoritmo reinicia o laço externo com V={4,5,6,8} e parte em busca de novas rotulações positivas e negativas para este subgrafo, iniciando pelo vértice 5. Após a 2ª iteração do laço mais externo, a 2ª componente é dada por S2={4,5,6,8}, em destaque na 3ª fig.

Rotulação positiva | **Rotulação negativa** | **Segunda componente**

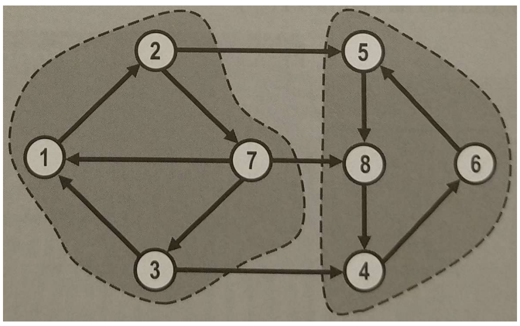


21

Algoritmo de Roy para Componentes Conexas

A solução final do algoritmo está na figura abaixo, onde estão destacadas as duas componentes conexas S1 e S2 do grafo. Em cada iteração determinou-se uma delas. Portanto, se aplicado a um grafo não direcionado, o algoritmo determina que o grafo é conexo com apenas 1 iteração do laço externo.

Componentes fortemente conexas do grafo



22

Grafos: Unidade 3 Conexidade

BIBLIOGRAFIA

GOLDBARG, M; GOLDBARG, E. **Grafos – Conceitos, algoritmos e aplicações**. Campus, 2012.

23