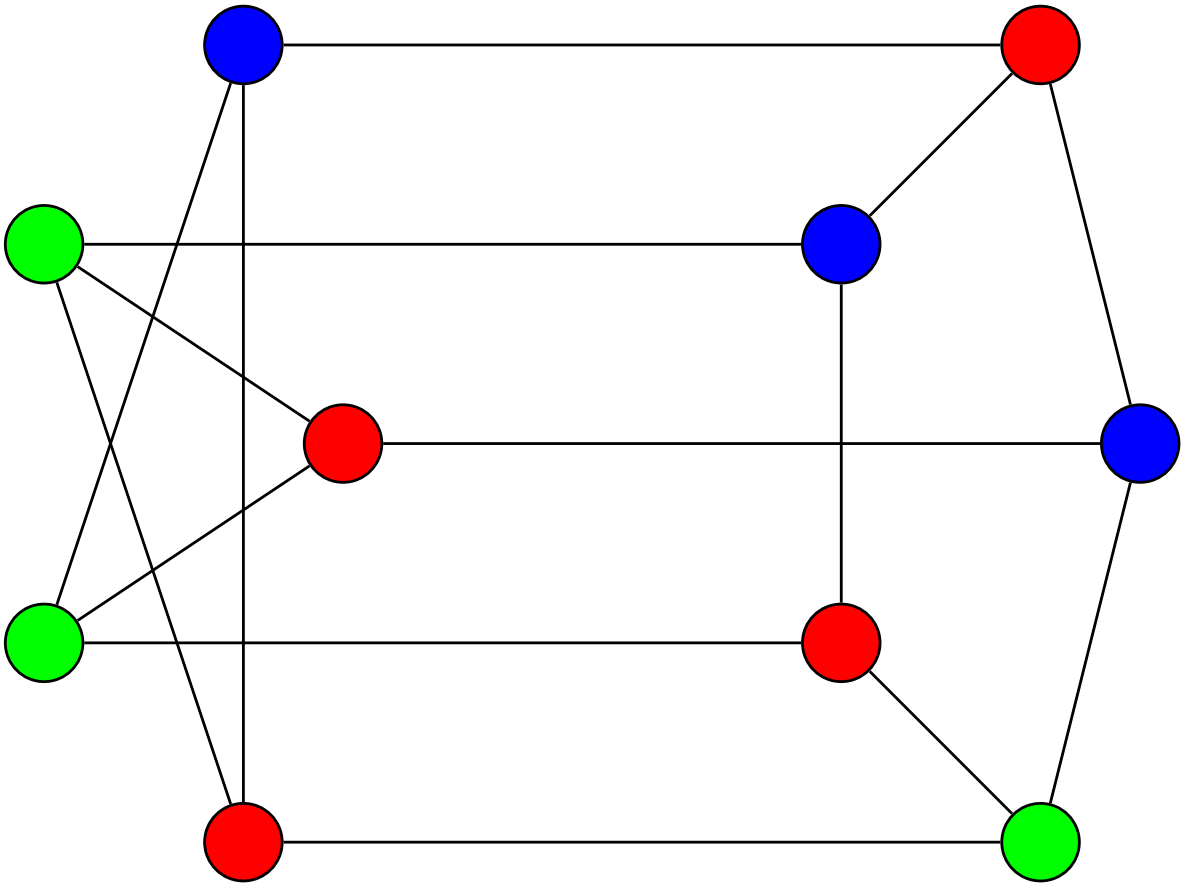
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC
Profa. Carla Negri Lintzmayer

Notas de aula da disciplina Teoria dos Grafos



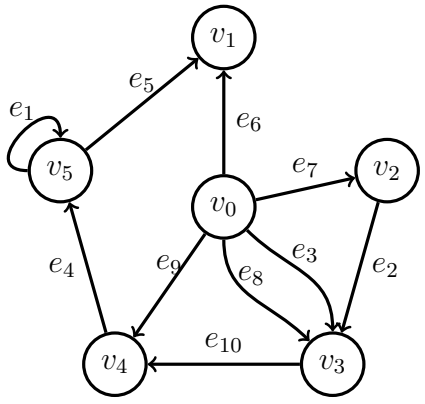
Esse material é um compilado dos seguintes:

- Sedgewick, R.. *Algorithms in C, part 5: graph algorithms*. 3rd ed. Addison-Wesley. 2002.
- Bondy, J. A.; Murty, U. S. R.. *Graph Theory*. Graduate Texts in Mathematics. Springer. New York. 2008.
- Lintzmayer, C. N.; Mota, G. O.. *Notas de aulas - Análise de algoritmos e estruturas de dados*. Em construção.

10 Aula 14: digrafos – grafos direcionados

- Um *digrafo* G é uma tripla (V, E, ψ), onde:
  - V é um conjunto não vazio de elementos chamados *vértices*;
  - E é um conjunto de elementos chamados *arestas* ou *arcos*, disjunto de V; e
  - ψ é a *função de incidência*, que associa um arco a um par ordenado de vértices.
- Por exemplo, $H = (V, E, \psi)$ onde
  - $V = \{v_0, v_1, v_2, v_3, v_4, v_5\}$
  - $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$
  - $\psi(e_1) = (v_5, v_5)$, $\psi(e_2) = (v_2, v_3)$, $\psi(e_3) = (v_0, v_3)$, $\psi(e_4) = (v_4, v_5)$, $\psi(e_5) = (v_5, v_1)$, $\psi(e_6) = (v_0, v_1)$, $\psi(e_7) = (v_0, v_2)$, $\psi(e_8) = (v_0, v_3)$, $\psi(e_9) = (v_0, v_4)$, $\psi(e_{10}) = (v_3, v_4)$

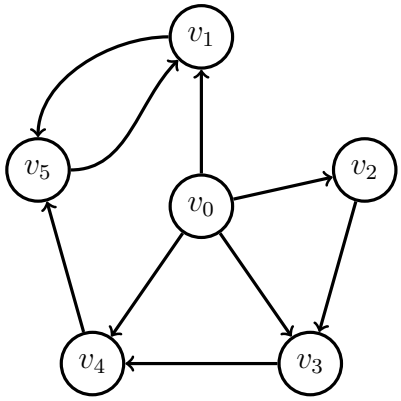
  é um digrafo.
- Digrafos permitem uma representação gráfica amigável:
  - círculos (bolinhas) representam vértices, e
  - uma aresta/arco e é representada por uma flecha que liga os círculos que representam os vértices x e y se $\psi(e) = (x, y)$.
- O digrafo H dado acima pode ser visualizado na figura a seguir (note como V, E e ψ podem ser inferidas pelo desenho):



- Dado um digrafo $G = (V, E, \psi)$:
  - usamos $V(G) = V$, usamos $E(G) = E$ e $\psi(G) = \psi$;
  - a *ordem* de G é $|V(G)|$;
  - arestas e e f são *paralelas* se $\psi(e) = \psi(f)$;
  - aresta e é um *laço* se $\psi(e) = (v, v)$ para algum $v \in V(G)$.

10.1 Digrafos simples

- Um digrafo G é *simples* se não contém laços nem arestas paralelas. Nesse caso, uma aresta pode ser unicamente determinada pelos seus extremos e $\psi(G)$ pode ser definida implicitamente fazendo com que $E(G)$ seja um conjunto de pares ordenados de vértices.
  - Por exemplo, J onde $V(J) = \{v_0, v_1, v_2, v_3, v_4, v_5\}$ e $E(J) = \{(v_0, v_1), (v_0, v_2), (v_0, v_3), (v_0, v_4), (v_1, v_5), (v_4, v_5), (v_2, v_3), (v_3, v_4), (v_5, v_1)\}$ é um digrafo simples.
  - Sua representação gráfica é dada na figura a seguir:



- A partir de agora, o termo *digrafo* será usado exclusivamente para denotar digrafos simples. Se necessário, faremos a diferenciação usando o termo *multidigrafo*.

Proposição 10.1. *Se G é um digrafo de ordem n, vale que*

$$|E(G)| \leq 2\binom{n}{2} = n(n-1) \ .$$

10.2 Grafos e digrafos

- Seja D um digrafo qualquer.
- O *grafo subjacente de D* é o grafo $G(D)$ tal que $V(G(D)) = V(D)$ e para cada arco xy de D existe uma aresta xy em $G(D)$.
- Muitas definições sobre grafos podem fazer sentido em digrafos se considerarmos seu grafo subjacente.
- Seja G um grafo qualquer.
- O *digrafo associado a G* é o digrafo $D(G)$ tal que $V(D(G)) = V(G)$ e para cada aresta xy de G existem dois arcos, xy e yx, em $D(G)$.

  - Essa relação mostra que todo grafo é um digrafo.

- Uma *orientação de G* é um digrafo $\vec{G}$ tal que $V(\vec{G}) = V(G)$ e para cada aresta xy de G existe um arco, xy ou yx, em $\vec{G}$.

- Se nos é dado um digrafo D qualquer e ele é uma orientação de algum grafo, então D é chamado de **grafo orientado**.

10.3 Adjacência e vizinhança

- Seja G um digrafo.

- Se $e \in E(G)$ e $e = (u, v)$, então:

  - e *sai* de u e *entra* em v;
  - u é *cauda* de e e v é *cabeça* de e;
  - u *domina* v e v é *dominado* por u;
  - u é *vizinho de entrada* de v;
  - v é *vizinho de saída* de u;
  - denotaremos (u, v) simplesmente por uv (e note que $uv \neq vu$).

- A *vizinhança de saída* de $u \in V(G)$, denotada $N^+_G(u)$ ou apenas $N^+(u)$, é o conjunto que contém os de vizinhos de saída de u, isto é, $\{v \in V(G) \colon uv \in E(G)\}$.

- A *vizinhança de entrada* de $u \in V(G)$, denotada $N^-_G(u)$ ou apenas $N^-(u)$, é o conjunto que contém os de vizinhos de entrada de u, isto é, $\{v \in V(G) \colon vu \in E(G)\}$.

10.4 Grau

- Seja G um digrafo.

- O *grau de saída* de um vértice $v \in V(G)$, denotado $d^+_G(v)$ ou $d^+(v)$, é o número de arestas que saem de v.

- O *grau de entrada* de um vértice $v \in V(G)$, denotado $d^-_G(v)$ ou $d^-(v)$, é o número de arestas que entram em v.

- Note que $d^+_G(v) = |N^+_G(v)|$ e $d^-_G(v) = |N^-_G(v)|$.

  - se G é um multidigrafo, podemos ter $d^+_G(v) \neq |N^+_G(v)|$ ou $d^-_G(v) \neq |N^-G(v)|$ para algum $v \in V(G)$.

- Um vértice v é uma *fonte* se $d^-(v) = 0$.
- Um vértice v é um *ralo/sorvedouro* se $d^+(v) = 0$.
- O *grau de saída mínimo* de G, denotado $\delta^+(G)$, é o menor grau de saída dentre os graus de saída de todos os vértices de G, isto é,

$$\delta^+(G) = \min\{d_G^+(u) \colon u \in V(G)\} \ .$$

  Analogamente podemos definir o *grau de entrada mínimo* de G, que é denotado $\delta^-(G)$.
- O *grau de saída máximo* de G, denotado $\Delta^+(G)$, é o maior grau de saída dentre os graus de saída de todos os vértices de G, isto é,

$$\Delta^+(G) = \max\{d_G^+(u) \colon u \in V(G)\} \ .$$

  Analogamente podemos definir o *grau de entrada máximo* de G, que é denotado $\Delta^-(G)$.

10.5 Isomorfismo

O conceito de isomorfismo é igual ao definido para grafos.

10.6 Algumas classes de digrafos

Um digrafo D é:

- *arborescência* se o grafo subjacente é uma árvore e se todo vértice tem grau de entrada 1, exceto por um vértice, que é a raiz.
- *torneio* se ele é a orientação de um grafo completo.
- *DAG* se ele é acíclico (DAG significa *directed acyclic graph*).

O *digrafo reverso* de D é o digrafo $\overleftarrow{D}$ tal que $V(\overleftarrow{D}) = V(D)$ e $xy \in E(\overleftarrow{D})$ se e somente se $yx \in E(D)$.

10.7 Subdigrafos

Os conceitos de subdigrafos são iguais aos de subgrafos.

10.8 "Aperto de mãos"

O teorema a seguir estabelece uma relação identidade fundamental que relaciona os graus dos vértices com o número de arestas em um digrafo.

Teorema 10.2 (Aperto de mãos)**.** *Para todo digrafo G temos que* $\sum_{v \in V(G)} d^+_G(v) = \sum_{v \in V(G)} d^-_G(v) = |E(G)|$.

10.9 Passeios, trilhas, caminhos e ciclos

Os conceitos de passeios, trilhas, caminhos e ciclos, são os mesmos definidos para grafos. O conceito de distância também.

Não é incomum falar "passeio/trilha/caminho/ciclo orientado/direcionado" para reforçar que a direção das arestas importa!

10.10 Conexidade

- Um digrafo G é dito *fortemente conexo* se existe um uv-caminho para todo par $u, v \in V(G)$.
- Um digrafo que não é fortemente conexo consiste de um conjunto de *componentes fortemente conexas*, que são subdigrafos fortemente conexos maximais.
- Note que nem todo arco de um digrafo está em alguma componente fortemente conexa, portanto.

10.11 Representação de digrafos

- Também usamos as duas implementações vistas para grafos:
  - Matriz de adjacências
  - Listas de adjacências
- Quais devem ser as adaptações?

10.12 Buscas em digrafos

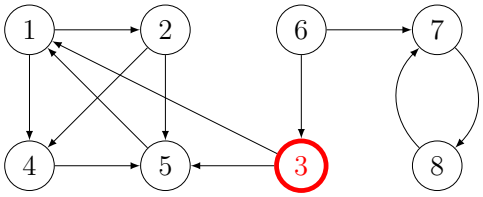
- Essencialmente são os mesmos algoritmos, exceto que devemos considerar a vizinhança de saída do vértice que está sendo explorado.

- Assim, vamos crescer uma arborescência durante a exploração.

```
1:  Função BFS(D, s)
2:      visitado[s] ← 1
3:      dist[s] ← 0
4:      crie uma fila vazia F
5:      ENFILEIRA(F, s)
6:      Enquanto F ≠ ∅ faça
7:          u ← DESENFILEIRA(F)
8:          Para cada v ∈ N⁺(u) faça
9:              Se visitado[v] = 0 então
10:                 visitado[v] ← 1
11:                 pred[v] ← u
12:                 dist[v] ← D[u] + 1
13:                 ENFILEIRA(F, v)
```

```
1:  Função DFS(D, s)
2:      visitado[s] ← 1
3:      tempo ← tempo + 1
4:      preordem[s] ← tempo
5:      Para cada v ∈ N⁺(s) faça
6:          Se visitado[v] = 0 então
7:              pred[v] ← s
8:              DFS(D, v)
9:      tempo ← tempo + 1
10:     posordem[s] ← tempo
```

- BFS continua encontrando distâncias mínimas.
- Não conseguimos informações sobre caminhos que não comecem na raiz das buscas.
- Não necessariamente é possível obter uma arborescência geradora do digrafo, ou mesmo detectar componentes conexas ou fortemente conexas diretamente.
  - Mas podem ser adaptadas.



- Considerando uma arborescência de DFS correspondente a um dado digrafo D, uma aresta $xy \in E(D)$ é:
  - de *retorno* se y é ancestral de x na árvore; note que $posordem[y] > posordem[x]$;
  - de *descida* se y é descendente de x na árvore; note que $preordem[y] > preordem[x]$;
  - de *cruzamento* se y e x não têm relação de parentesco; note que $preordem[y] < preordem[x]$.

Lema 10.3. *Um digrafo D é um DAG se e somente se não são encontradas arestas de retorno em nenhuma chamada da DFS sobre D.*

10.13 Digrafos acíclicos – DAGs

- Diversas aplicações em problemas de escalonamento: realização de tarefas que têm dependência entre si.

Lema 10.4. *Se D é um DAG, então D contém uma fonte.*

Demonstração. Suponha, para fins de contradição, que D não contém nenhum vértice v com $d^-(v) = 0$. Tome um caminho maximal $P = (v_0, v_1, \ldots, v_k)$ em D. Como $d^-(v_0) > 0$, seja $x \in N^-(v)$. Claramente $x \in V(P)$, pois caso contrário o caminho não seria maximal. Então $x = v_i$ para algum $1 \leq i \leq k$. Mas então $(v_0, v_1, \ldots, v_i, v_0)$ é um ciclo em D, o que é uma contradição. □

Lema 10.5. *Se D é um DAG, então D contém um ralo/sorvedouro.*

- Uma *ordenação topológica* de um digrafo D de ordem n é uma rotulação $f\colon V(D) \to \{1, 2, \ldots, n\}$ dos vértices de D tal que $f(u) \neq f(v)$ se $u \neq v$ e, se $uv \in E(D)$, então $f(u) < f(v)$.
- Alternativamente: uma *ordenação topológica* é uma sequência $(v_1, \ldots, v_n)$ dos n vértices de D tal que se $v_iv_j \in E(D)$, então $i < j$.

Teorema 10.6. *Se D é um digrafo, então D admite uma ordenação topológica se e somente se D é acíclico.*

Demonstração. Vamos provar primeiro que se D admite uma ordenação topológica, então ele é acíclico. Seja $f\colon V(D) \to \{1, \ldots, n\}$ uma ordenação topológica de D

em que $n = |V(D)|$ e suponha, para fins de contradição, que D contém um ciclo $C = (u_1, \ldots, u_k, u_1)$. Para $i \in \{1, \ldots, k-1\}$, note que, por causa do arco $u_i u_{i+1}$, vale que $f(u_i) < f(u_{i+1})$. Isso implica que $f(u_1) < f(u_k)$. Mas o arco $u_k u_1$ existe, o que contradiz f ser uma ordenação topológica.

Agora vamos provar que se D é acíclico, então D admite uma ordenação topológica. Vamos provar por indução em $n = |V(D)|$.

Se $n = 1$, então claramente D admite uma ordenação topológica.

Suponha então que $n > 1$ e que qualquer digrafo acíclico D' tal que $1 \leq |V(D')| < n$ admite uma ordenação topológica.

Pelo Lema 10.4, seja $u \in V(D)$ uma fonte. Seja $D' = D - u$. Note que D' é um digrafo e é acíclico, pois caso contrário D teria um ciclo. Como $|V(D')| < n$, então por hipótese de indução D' admite uma ordenação topológica $S' = (u_1, \ldots, u_{n-1})$.

Vamos provar que $S = (u, u_1, \ldots, u_{n-1})$ é uma ordenação topológica de D. De fato, S contém todos os vértices de D. Como S' é ordenação topológica de D', então $i < j$ para toda $u_i u_j \in E(D)$. Como u é fonte em D, então não existe arco do da forma $u_i u$ em D. Então S é de fato uma ordenação topológica de D. □

Teorema 10.7. *Seja D um DAG de ordem n. Aplique a DFS em D a partir de qualquer vértice e seja $S = (v_1, v_2, \ldots, v_n)$ a sequência dos vértices de D em ordem decrescente do valor de pós-ordem, isto é, $posordem[v_i] > posordem[v_{i+1}]$ para todo $1 \leq i < n$. Então S é uma ordenação topológica de D.*

Demonstração. Seja $xy \in E(D)$. Precisamos mostrar que $posordem[x] > posordem[y]$, o que implica que x aparece em S à esquerda de y.

Quando DFS(D, x) testou o vizinho de saída y de x, apenas um dos dois seguintes casos ocorreu.

Se y não estava visitado, então ele é visitado nesse momento. Como a exploração de x só pode terminar depois que todos os seus vizinhos forem explorados, claramente teremos $posordem[x] > posordem[y]$.

Agora assuma que y já estava visitado. Como D é um DAG, então note que não existe yx-caminho. Isso significa que DFS(G, y) já terminou. Logo, claramente teremos $posordem[y] < posordem[x]$. □