AUTARQUIA EDUCACIONAL DO VALE DO SÃO FRANCISCO – AEVSF
FACULDADE DE CIÊNCIAS APLICADAS E SOCIAIS DE PETROLINA – FACAPE
CIÊNCIAS DA COMPUTAÇÃO

APOSTILA DE TEORIA DOS GRAFOS

VERSÃO 1.0

bel. lea. Roberto Tenorio Figueiredo

PETROLINA, 2010

SUMÁRIO

LISTA DE FIGURAS

INTRODUÇÃO À TEORIA DOS GRAFOS

1.1. DEFINIÇÃO

Um grafo G(V, E) é um conjunto V finito, não-vazio de pontos, cujos elementos são chamados de ***vértices*** (nodos ou nós), conectados por um conjunto E de linhas, chamada de ***arestas*** (ou arcos). Uma aresta é um par distinto não-ordenado (v_i, v_j), onde v_i e v_j são elementos distintos de V. Normalmente, utiliza-se uma representação gráfica (geométrica) de um grafo. Eis um exemplo de grafo, com a sua representação gráfica:

$V = \{v_1, v_2, v_3, v_4, v_5\}$ $E = \{a_1, a_2, a_3, a_4, a_5, a_6\}$
onde $a_1 = (v_1, v_2)$, $a_2 = (v_1, v_3)$, $a_3 = (v_2, v_3)$, $a_4 = (v_2, v_5)$, $a_5 = (v_3, v_5)$, $a_6 = (v_3, v_4)$;

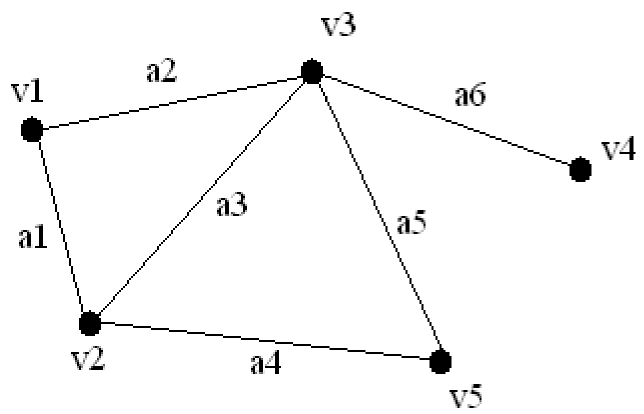


Figura 1: Exemplo de Grafo

Com isso, percebemos que os vértices são as extremidades (ou extremos) das arestas.

A quantidade de vértices de um grafo (densidade de vértices) será denotada pela letra **n** e a quantidade de arestas (densidade de arestas), pela letra **m**.

1.2. CONCEITOS ASSOCIADOS

Neste tópico, veremos todos os conceitos que são diretamente associados ao estudo da Teoria dos Grafos. São eles:

1.2.1. ADJACÊNCIA (VIZINHANÇA)

Duas arestas são consideradas adjacentes, ou vizinhas, se possuírem um vértice em comum. Da mesma forma, dois vértices são adjacentes se possuírem uma aresta em comum.

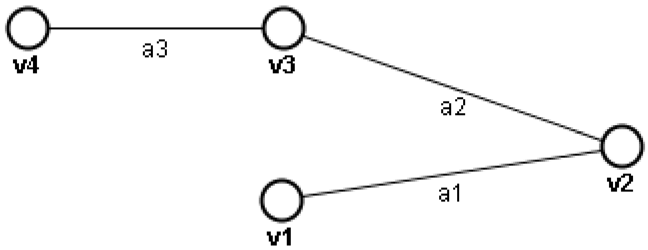


Figura 2: Adjacência de arestas

Na figura 2 as arestas a1 e a2 são adjacentes, pois o vértice v2 está presente em ambas, assim como a2 e a3, com o vértice v3. Na mesma figura, vemos que o vértice v4 é vizinho de v3, devido ao compartilhamento da aresta a3, assim como v1 e v2, com a aresta a1 e v2 e v3, com a aresta a2.

1.2.2. ISOMORFISMO

Dois grafos são isomorfos entre si, se e somente se for possível alterar os nomes dos vértices de um deles de tal modo que os dois grafos fiquem exatamente iguais, mantendo-se as mesmas ligações. Se cada um dos grafos tem n vértices, esse algoritmo de alteração de nomes consome tempo proporcional a n! (n fatorial). Como n! cresce explosivamente com n, esse algoritmo é decididamente insatisfatório na prática. Infelizmente, não se conhece um algoritmo substancialmente melhor.

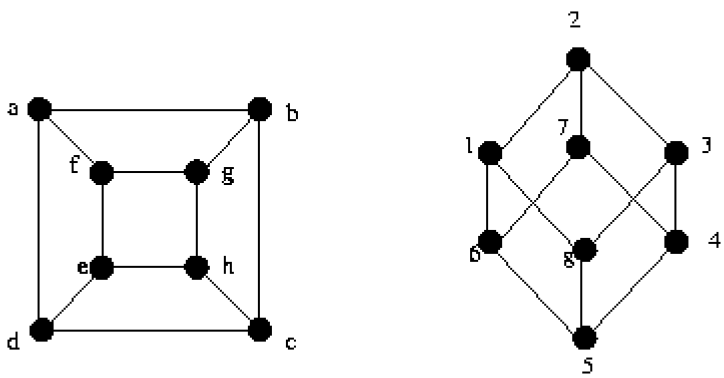


Figura 3: Grafos Isomorfos

Para ver o isomorfismo dos grafos da figura 3, podemos utilizar a seguinte função:

f(a) = 1, f(b) = 2, f(c) = 3, f(d) = 8, f(e) = 5, f(f) = 6, f(g) = 7, f(h) = 4.

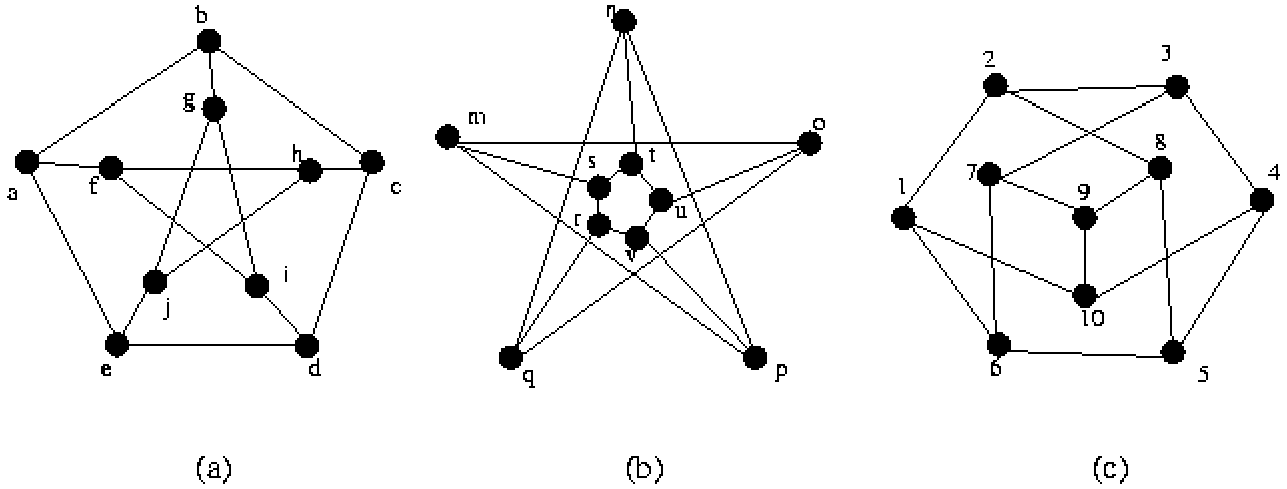


Figura 4: Grafos Isomorfos

Para ver o isomorfismo dos grafos 4(a) e 4(b), utilize a seguinte função:

f(a) = s, f(b) = t, f(c) = u, f(d) = v, f(e) = r, f(f) = m, f(g) = n, f(h) = o, f(i) = p, f(j) = q

Para ver o isomorfismo dos grafos 4(a) e 4(c), utilize a seguinte função:

f(a) = 1, f(b) = 10, f(c) = 4, f(d) = 5, f(e) = 6, f(f) = 2, f(g) = 9, f(h) = 3, f(i) = 8, f(j) = 7.

Esses exemplos devem ser suficientes para mostrar que não é sempre fácil determinar se dois grafos são isomorfos. Para que dois grafos sejam isomorfos, no mínimo, as seguintes condições têm que ser respeitadas:

1. Os dois têm o mesmo número de vértices.
2. Os dois têm o mesmo número de arestas.
3. Os dois têm o mesmo número de vértices de grau *n*, para qualquer valor *n* entre 0 e o número de vértices que o grafo contém.

Note que essas condições não são suficientes para que sejam isomorfos. Por exemplo, os grafos da figura 5 respeitam essas condições e não são isomorfos.

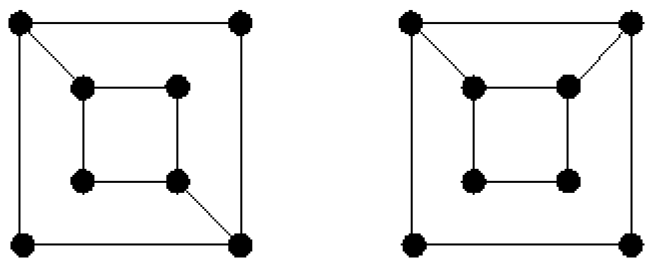


Figura 5: Grafos Não-Isomorfos

1.2.3. TRIVIAL

Um grafo é chamado de Trivial se tiver apenas um vértice e nenhuma aresta.

Figura 6: Grafo Trivial

1.2.4. GRAFO DE ARESTAS

O grafo de arestas de um grafo G, é o grafo formado por vértices gerados a partir das arestas de G e com arestas se formam a partir das adjacências das arestas de G.

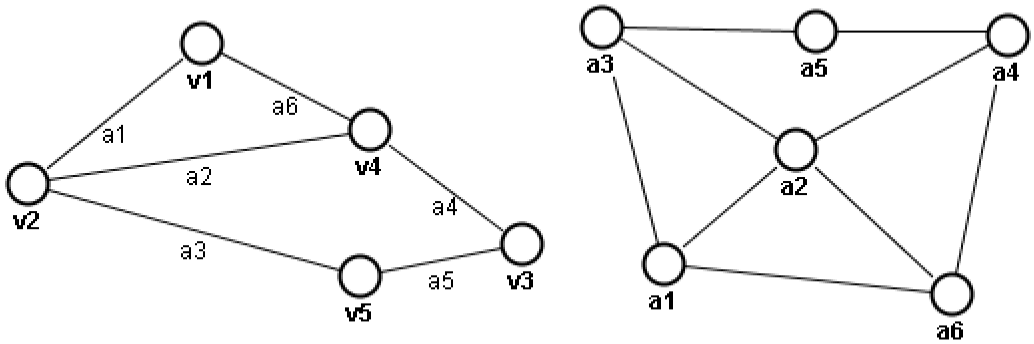


Figura 7: Um grafo e seu respectivo grafo de arestas

1.2.5. LAÇO

Um laço é uma aresta que possui as duas extremidades em um mesmo vértice de G.

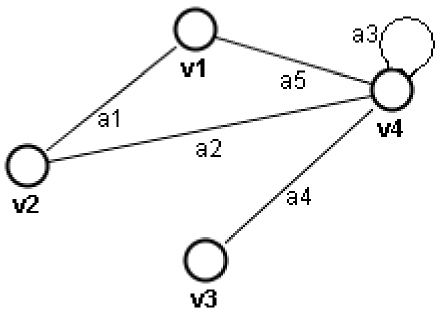


Figura 8: Um grafo com um laço: a aresta a3.

1.2.6. MULTIGRAFO

Um multigrafo é um grafo que possui, pelo menos, um par de arestas com vértices iguais. No exemplo da figura 9, temos um multigrafo pois as arestas a1 e a4 possuem os mesmos vértices. O mesmo acontece com as arestas a5, a7 e a8.

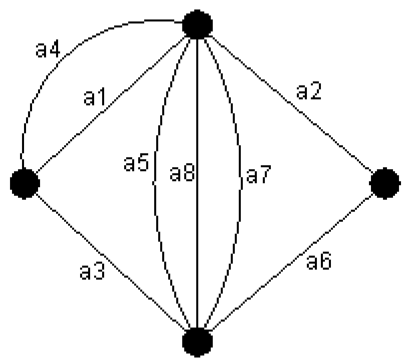


Figura 9: Multigrafo.

1.2.7. GRAU

O grau de um vértice é a quantidade de vértices adjacentes a ele. Dizemos que um grafo é **regular de grau** quando todos os vértices possuem o mesmo grau. Um grafo regular de grau N possui (V * N)/2 arestas, onde V é a quantidade de vértices. Na figura 10, temos um grafo regular de grau com 12 arestas, ou seja, (8*3)/2.

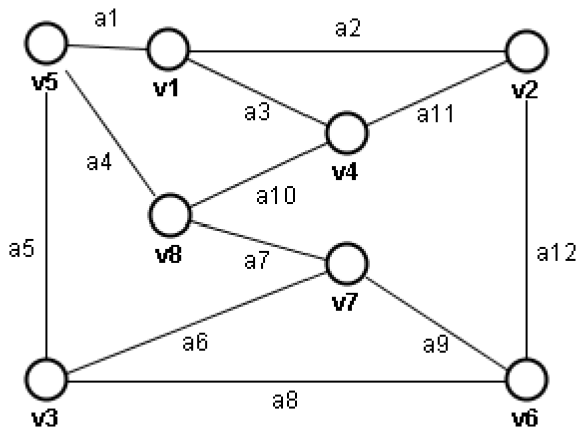


Figura 10: Grafo Regular de Grau 3

Um vértice de grau zero é chamado de isolado. O grau máximo de um grafo, denominado de **delta**, é o grau do maior vértice do grafo. Um grafo regular de grau 3 é chamado de grafo **cúbico**.

1.2.8. CAMINHO

O caminho é uma seqüência de vértices, onde um vértice e seu **sucessor** na seqüência possuem uma aresta ligando-os. O análogo de sucessor é o **antecessor**. Se existe um caminho entre dois vértices v1 e v2, dizemos que v1 **alcança** ou **atinge** v2 e vice-versa. Na figura 10, temos o seguinte exemplo de caminho: v1, v2, v6, v3, v7 e v8. Esse caminho é representado pelas arestas: a2, a12, a8, a6, a7. Evidentemente, a figura 10 possui muitos outros caminhos que podem ser exemplificados. A quantidade de arestas é chamada de **comprimento** do caminho. No exemplo dado, o caminho tem comprimento igual a 5 (cinco). Se todos os vértices do caminho forem distintos, esse caminho recebe o nome de **caminho simples** ou **elementar**. Se todas as arestas forem distintas, esse caminho é chamado de **trajeto** ou **trilha**. O caminho é dito **ímpar** se tiver comprimento ímpar e **par** e se tiver comprimento par.

É chamado de **caminho hamiltoniano**, o caminho simples entre todos os vértices de um grafo. É chamado de **caminho euleriano**, o trajeto que contém todas as arestas de um grafo. Alguns autores trazem o termo caminho pré-hamiltoniano e pré-euleriano, que possui os mesmos conceitos respectivamente, sem a restrição de unicidade.

1.2.9. DISTÂNCIA

A distância entre dois vértices de um grafo é o menor comprimento dos caminhos entre esses dois vértices. Por exemplo, na figura 10, existem vários caminhos que ligam v4 a v6, porém, o menor caminho entre eles, ou seja, à distância, é o caminho v4 – v2 – v6 (distância = 2).
Exemplo de outros caminhos que ligam v4 e v6:

- v4 – v8 – v7 – v6 (três arestas – caminho ímpar);
- v4 – v1 – v5 – v8 – v7 – v6 (cinco arestas – caminho ímpar);
- v4 – v1 – v5 – v3 – v6 (quatro arestas – caminho par);

1.2.10.CICLO

Um ciclo é um caminho que inicia e termina em um mesmo vértice. Na figura 10, podemos exemplificar um ciclo, com o caminho: v1 – v5 – v3 – v6 – v2 – v1. Se todos os vértices do ciclo forem distintos, dizemos que o ciclo é simples ou elementar. Um grafo que não possui ciclo é chamado de acíclico. O menor ciclo do grafo é chamado de **cintura** e o maior ciclo é chamado de **circunferência**. Um grafo é uma **gaiola** se for regular em grau e possuir uma cintura. Uma aresta que une dois vértices não consecutivos de um ciclo é chamada de **corda**. Na figura 12.a, temos o ciclo v1 – v2 – v5 – v6 – v4 – v3 – v1, nele, podemos identificar as cordas: a8, que une v2 e v3 e a7, que une v5 e v4.

É chamado de **ciclo hamiltoniano**, o ciclo simples entre todos os vértices de um grafo. É chamado de **ciclo euleriano**, o ciclo do trajeto que contém todas as arestas de um grafo. Um grafo, que contém um caminho euleriano e não possui um ciclo euleriano, é chamado de **unicursal**.

1.2.11.CONEXO

Um grafo G é conexo se e somente se, existir um caminho entre todos os pares de vértices do grafo G. Caso algum vértice de G não tenha caminho com algum outro vértice de G, chamamos esse grafo de desconexo. A figura 10 mostra um grafo conexo, enquanto que a figura 11 mostra um grafo desconexo, por não existir, por exemplo, caminho entre o vértice v4 e o vértice v6. Neste caso, dizemos que v4 é inalcançável, em relação a v6.

Teorema 01: Um grafo G conexo possui ciclo euleriano se e somente se todos os vértices de G possuírem grau par.

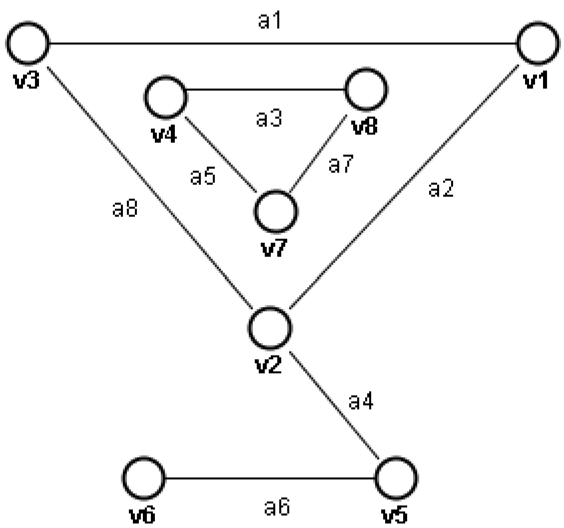
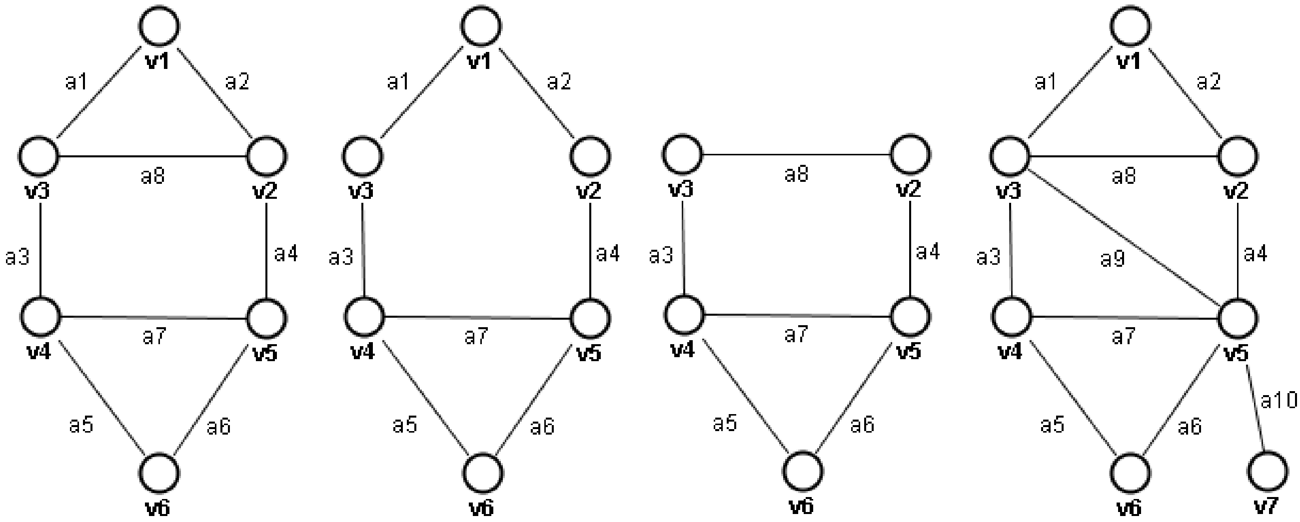


Figura 11: Grafo Desconexo

1.2.12.EXCLUSÃO E INCLUSÃO DE ELEMENTOS

Seja G(V, E) um grafo, $\varepsilon \in E$, denota-se **G – ε** o grafo obtido pela exclusão (subtração) da aresta ε do grafo G. Da mesma forma, $v \in V$, denota-se **G – v** o grafo obtido pela exclusão do vértice v do grafo G. Também é possível realizar operação para inclusão (adição) de elementos, com isso, a notação **G + ε**, significa a adição da aresta ε no grafo G e **G + v** a inclusão do vértice v no grafo G.



(a) Grafo G (b) G – a8 (c) G – v1 (d) G + a9 + a10 + v7

Figura 12: Operações sobre grafos

A operação de exclusão de um vértice, automaticamente exclui todas as arestas a ele associadas, visto que uma aresta só existe se seus vértices extremos existirem. A inclusão de um vértice, não necessariamente, obriga a inclusão de uma nova aresta, pois podem existir elementos desconexos.

1.2.13.SUBGRAFO, GRAFO PARCIAL e SUPERGRAFO

Um subgrafo de um grafo G é o grafo formado pela subtração de um ou mais vértices de G, sendo que o subgrafo deve ter, no mínimo, um vértice. Um exemplo de subgrafo é a figura 12(c), que é subgrafo do grafo G, presente na figura 12 (a). Em algumas literaturas, encontraremos o termo **subgrafo induzido** pelo conjunto X, sendo X o conjunto de vértices excluídos do grafo original.

Chamamos de **grafo parcial** (ou subgrafo gerador) de G, um grafo formado pela subtração de arestas do grafo G. Um exemplo de grafo parcial é a figura 12(b), que é grafo parcial do grafo G, presente na figura 12(a).

O grafo que deu origem a um subgrafo S ou a um grafo parcial P é chamado de **supergrafo** de S ou de P respectivamente.

Dizemos que um subgrafo é **próprio** quando ele possui, pelo menos um vértice ou aresta a menos que seu supergrafo, ou seja, um subgrafo não é próprio quando é exatamente igual a seu supergrafo.

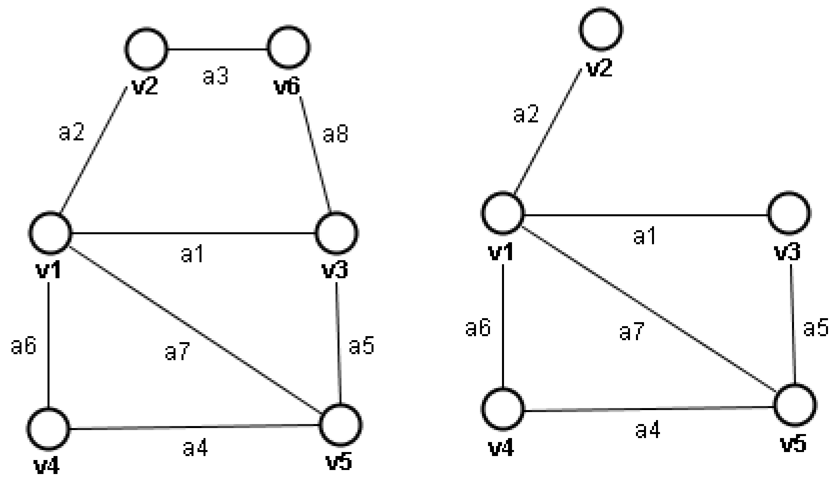
1.2.14.TAMANHO

O tamanho de um grafo é dado pela quantidade de vértices e arestas do mesmo. Por exemplo, o grafo da figura 12.a tem tamanho igual a 6,8, ou seja, possui seis vértices e oito arestas. A quantidade de vértices de um grafo é chamada de **cardinalidade** do grafo.

1.2.15.MAXIMAL E MINIMAL

Um subgrafo S, de um grafo G, é dito maximal em relação a uma certa propriedade P, como por exemplo ser conexo, se S tem a propriedade P, mas nenhum supergrafo próprio de S, subgrafo de G, tem a propriedade P.

Por exemplo, dizer que S é um subgrafo conexo maximal de G equivale a dizer que S é um subgrafo conexo de G e além disso, não existe nenhum supergrafo próprio de S que seja um subgrafo próprio conexo de G. Note que, nada impede que G tenha um outro subgrafo conexo de tamanho maior ou igual ao de S.



(a) Grafo G (b) Subgrafo S de G

Figura 13: Um grafo G e subgrafo de G

Para melhorar o exemplo, vejamos a figura 13. **Pergunta**: Existe algum subgrafo próprio de G, com todas suas arestas, que seja conexo e que contenha S? **Resposta**: Não, portanto, S é subgrafo conexo maximal de G.

De maneira análoga, um subgrafo S, de grafo G é dito minimal em relação a uma certa propriedade P, como por exemplo ser conexo, se G e S tem a propriedade P, mas nenhum subgrafo próprio de S tem a propriedade P. Em relação a ser conexo, apenas o grafo trivial é minimal de um grafo qualquer.

1.2.16.COMPONENTES CONEXOS

Chamamos que componentes conexos de G a quantidade de subgrafos maximais de G, que sejam conexos. Por exemplo, o grafo da figura 14.a, possui três componentes conexos. Esses componentes conexos são mostrados na figura 14.b.

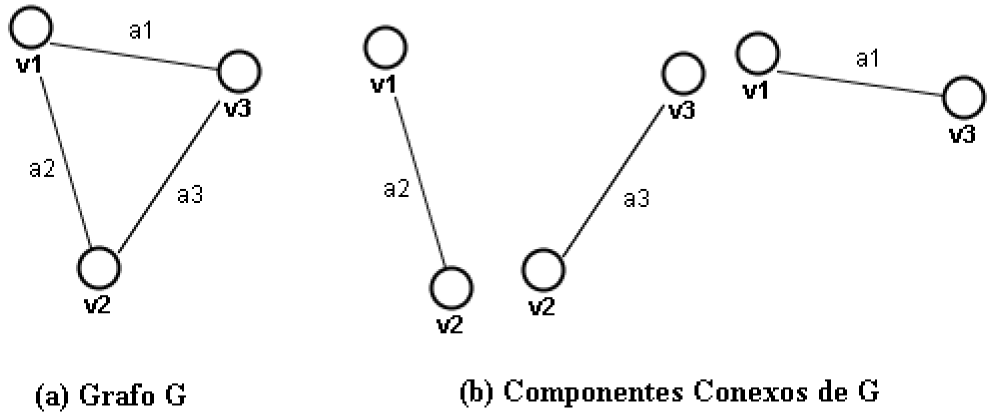


Figura 14: Grafo G e seus componentes conexos

1.2.17.COMPLETO

Um grafo G é completo, se e somente se, para qualquer par de vértices distintos de G, existe uma aresta ligando-os. Exemplo: O grafo da figura 14.a é completo.

Denomina-se **k-conexo** ou **k-regular**, um grafo completo com uma quantidade k de vértices. Exemplo: o grafo completo da figura 14.a é 3-conexo. Essa notação pode ser encontrada na forma K_3.Um grafo K_N, possui $\binom{N}{2}$ vértices.

Obs.: $\binom{N}{2} = \frac{N!}{K!(N-K)!}$ ➔ binômio de Newton

1.2.18.COMPLEMENTO

O complemento de um grafo G é um grafo G’, onde G’ possui exatamente os mesmos vértices de G, porém, possui todas as arestas que não estão presentes em G, sendo que nenhuma aresta de G’ é igual a uma aresta de G.

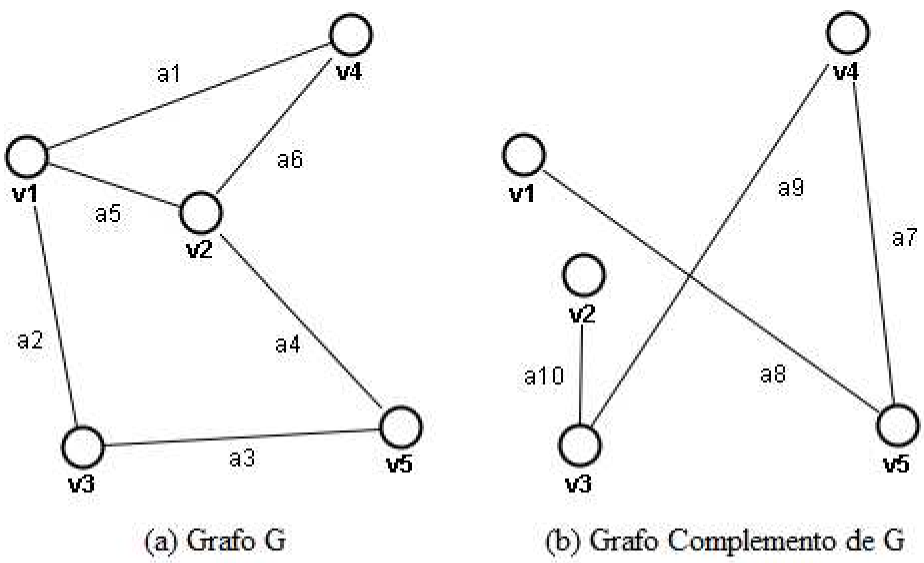


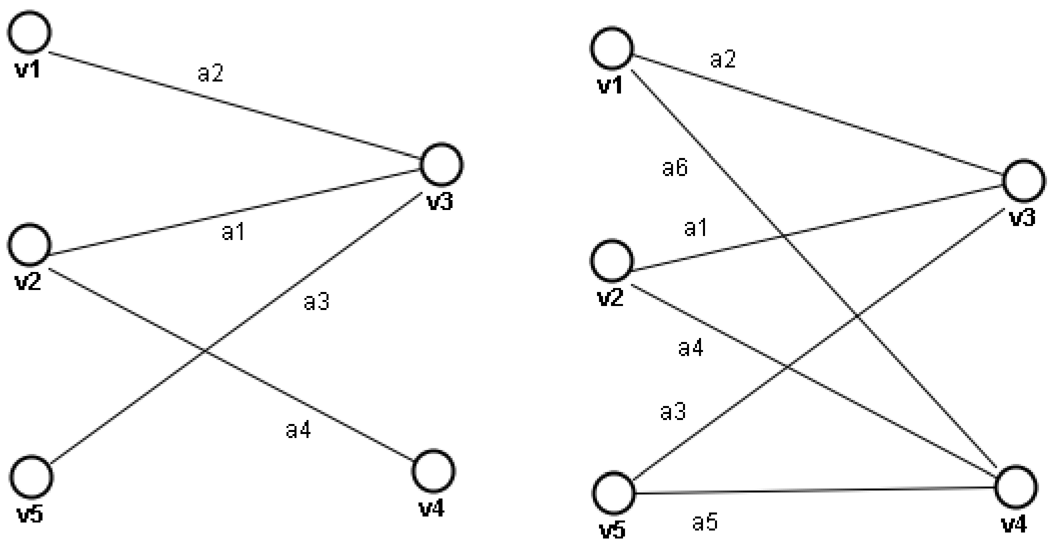
Figura 15: Um grafo e seu complemento

1.2.19.BIPARTIDO

Um grafo é chamado de bipartido quando seu conjunto de vértices V pode ser dividido em dois subconjuntos distintos V_1 e V_2, tais que todas as arestas do grafo, possuam, em suas extremidades, um vértice de cada subconjunto. Isto significa que o grafo bipartido não pode conter uma aresta que ligue dois vértices de um mesmo subconjunto.

Teorema 02: Um grafo é bipartido se e somente se todo ciclo de G possuir comprimento par.

Chamamos de **grafo bipartido completo**, o grafo bipartido que possua uma aresta entre cada par qualquer de vértices de V_1 e V_2. O grafo bipartido completo é denotado por $K_{N1,N2}$, onde N1 é a quantidade de vértices do subconjunto V_1 e N2 a quantidade de vértices do subconjunto V_2.



(a) Grafo Bipartido (b) Grafo Bipartido Completo

Figura 16: Um grafo bipartido e o grafo bipartido completo $K_{3,2}$

O grafo $K_{1,3}$ é conhecido como **Garra**. Um conjunto de vértices qualquer de um grafo G, que não possua arestas ligando os membros deste conjunto, é chamado de conjunto **estável** de vértices de G. Um grafo bipartido é dito **balanceado** se a quantidade vértices de seus conjuntos forem iguais.

1.2.20.CLIQUE

O clique, de um grafo G, é um subgrafo de G que seja completo. Na figura 13.a temos um grafo e os vértices v1, v4 e v5 formam um clique desse grafo. Em contra partida, temos o **conjunto independente de vértices**, que é um subgrafo de G totalmente desconexo, ou seja, é um subgrafo formado apenas de grafos triviais, sem nenhuma aresta.

1.2.21.COBERTURAS

Pode existir cobertura de vértices ou de arestas. A cobertura de vértices de um grafo G, é um conjunto de vértices V', onde cada aresta de G possui, ao menos, uma extremidade no conjunto V'. Como exemplo, vejamos o grafo da figura 10. Uma cobertura de vértices desse grafo é o conjunto {v2, v3, v4, v5, v7}. O valor da menor cobertura de vértices de G é denominado **número de cobertura de vértices** do grafo.

A cobertura de arestas de um grafo G, é um conjunto de arestas E', onde cada vértice de G incide em, ao menos, uma das arestas de E'. Como exemplo, vejamos o grafo da figura 10. Uma cobertura de arestas desse grafo é o conjunto {a3, a5, a7, a12}. O valor da menor cobertura de arestas de G é denominado **número de cobertura de arestas** do grafo.

1.2.22.RÓTULO E VALORAÇÃO

Um rótulo é uma cadeia de caracteres ou número que identifica um vértice ou uma aresta. Um grafo é dito rotulado, se todos os seus vértices e/ou arestas possuírem um rótulo. Os rótulos de um grafo formam o **conjunto de rótulos** do grafo. Vejamos a figura 17.

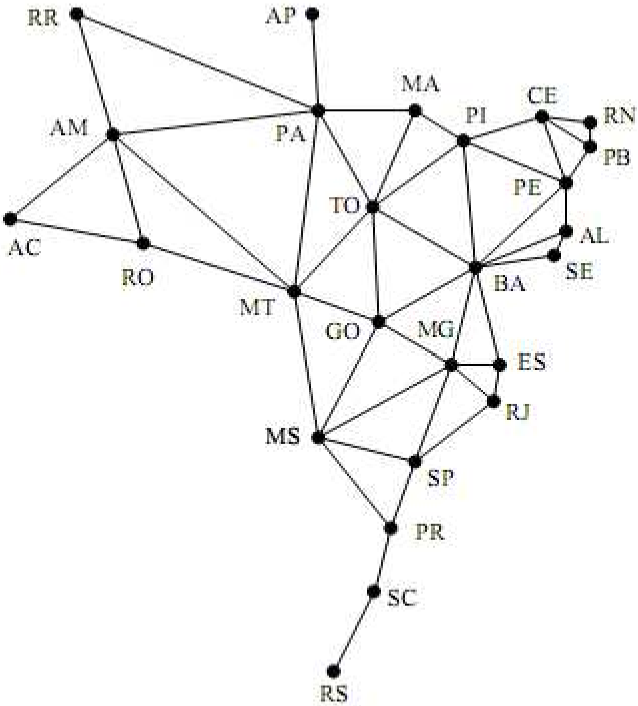


Figura 17: Grafo dos estados do Brasil

Na figura 17, temos o grafo dos estados do Brasil, onde cada vértice é um dos estados da República Federativa do Brasil e dois vértices são adjacentes se os estados que representam possuírem uma fronteira comum. Este é um exemplo de grafo rotulado, onde o conjunto de rótulos do grafo é o conjunto {RR, AP, AM, AC, RO, PA, MT, TO, MA, PI, CE, RN, PB, PE, AL, SE, BA, GO, MS, MG, ES, RJ, SP, PR, SC, RS}.

Um grafo é dito valorado, se seus rótulos forem numéricos e esses números pertencerem a um conjunto devidamente especificado. Como exemplo, vamos vejamos o grafo da figura 18. É o grafo dos estados do nordeste brasileiro, onde cada vértice representa um estado e dois vértices são adjacentes se os estados que representam possuírem uma fronteira comum. Esse grafo é rotulado, sendo que os rótulos das arestas representam a distância entre as capitais dos estados que estão nas extremidades dessas arestas, ou seja, os rótulos das arestas são especificados pela distância real entre as capitais dos estados, isto significa que o grafo da figura 18 é valorado.

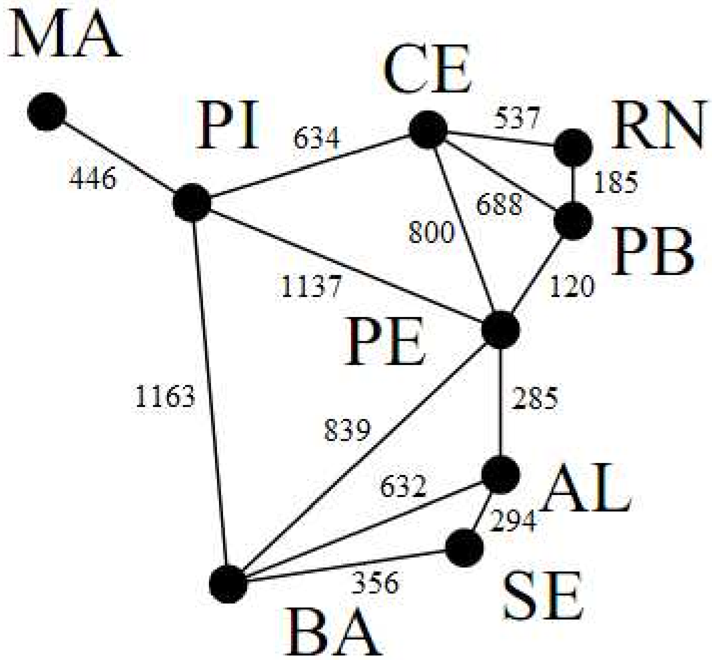


Figura 18: Grafo valorado dos estados do nordeste do Brasil

É definido como **Caminho de Valor Máximo**, o caminho elementar entre dois vértices, em um grafo valorado, onde a soma do valor das arestas é o máximo possível. Como exemplo, na figura 18, temos o caminho de valor máximo entre MA e BA, igual a MA – PI – CE – RN – PB – PE – AL – SE – BA.

É definido como **Caminho de Valor Mínimo**, o caminho elementar entre dois vértices, em um grafo valorado, onde a soma do valor das arestas é o mínimo possível. Como exemplo, na figura 18, temos o caminho de valor mínimo entre MA e BA, igual a MA – PI – BA.

1.2.23.ALTAMENTE IRREGULARES

Um grafo é chamado de **altamente irregular** se cada um dos vértices do grafo é adjacente, apenas, a vértices de graus diferentes do seu. O grafo da figura 13.b é altamente irregular.

1.2.24.ÁRVORES

Um grafo acíclico conexo é chamado de árvore. Se G é uma árvore, um vértice de G com grau menor ou igual a 1 é chamado de folha. Um vértice com grau maior que um é chamado de vértice interior. Um conjunto de árvores é denominado de **floresta**, ou seja, todo grafo acíclico é uma floresta. Toda árvore com n vértices, possui exatamente n-1 arestas. O grafo da figura 19 é uma árvore com seis vértices, portanto, com cinco arestas.

Teorema 03: Um grafo G é uma árvore se e somente se existir um único caminho entre cada par de vértices de G.

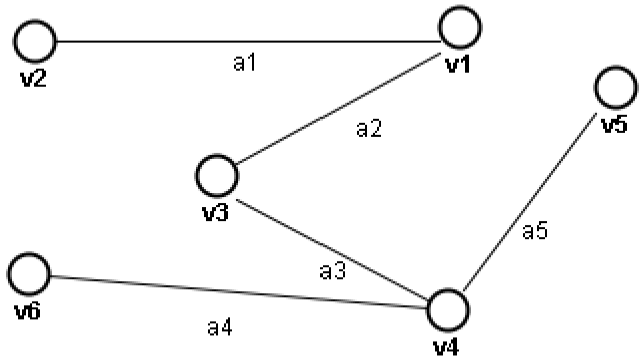


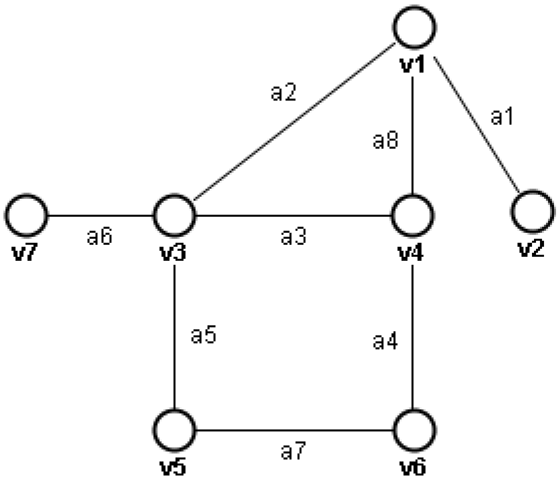
Figura 19: Exemplo de Árvore

Teorema 04: Seja G (V, E) um grafo. As possíveis afirmativas são equivalentes:

(i) G é uma árvore;
(ii) G é conexo e |E| é mínimo;
(iii) G é conexo e |E| = |V| – 1;
(iv) G é acíclico e |E| = |V| – 1;
(v) G é acíclico e para todo v, w ϵ V, a adição da aresta (v,w) produz um grafo contendo exatamente um ciclo.

1.2.25.EXCENTRICIDADE E CENTRO

Em um grafo G(V, E), a excentricidade (ou afastamento) de um vértice v ϵ V é o valor da distância máxima entre v e w, para todo w ϵ V. O centro de um grafo G é o subconjunto dos vértices de excentricidade mínima. Na figura 20.a temos um grafo e na figura 20.b às excentricidades dos vértices do grafo da figura 20.a.



(a) Um grafo G

Vértice	Excentricidade
v 1	2
v 2	3
v 3	2
v 4	2
v 5	3
v 6	3
v 7	3

(b) Excentricidades dos vértices do grafo G

Figura 20: Um grafo e a excentricidade de seus vértices

O centro do grafo da figura 20.a é formado pelo conjunto de vértices {v1, v3, v4}. O centro de um grafo pode possuir, no mínimo, um vértice e no máximo, n vértices, onde n é a quantidade total de vértices do grafo. O centro de uma árvore possui, no máximo, dois vértices.

A menor excentricidade de um grafo é chamada de **raio** do grafo, e a maior, é chamada de **diâmetro** do grafo. Um **vértice periférico** é o vértice cuja excentricidade é igual ao diâmetro do grafo. O vértice **mediano** ou **centróide** é aquele cuja soma das distâncias para os demais vértices de um grafo G, é mínima.

Lema 01: Seja G uma árvore com pelo menos três vértices. Seja G' a árvore obtida de G pela exclusão de todas as suas folhas. Então G e G' possuem o mesmo centro.

Teorema 05: O centro de uma árvore G possui um ou dois vértices.

Se um subgrafo gerador de um grafo G for uma árvore, denominamos esse subgrafo gerador de árvore geradora ou árvore de espalhamento. Todo grafo G conexo possui uma árvore geradora. Caso o grafo G não seja conexo, dizemos que possui uma floresta geradora. Na figura 21 temos um grafo e uma de suas árvores geradoras.

1.2.26.ELO

Seja G (V, E) um grafo conexo e T (V, E) uma árvore geradora de G. Uma aresta que pertença a G e não pertença a T é denominada de elo de G em relação a T. O total de elos de um grafo é denominado **posto** do grafo e é calculado pela equação m – n + 1, onde m é a quantidade de arestas do grafo e n é a quantidade de vértices desse grafo.

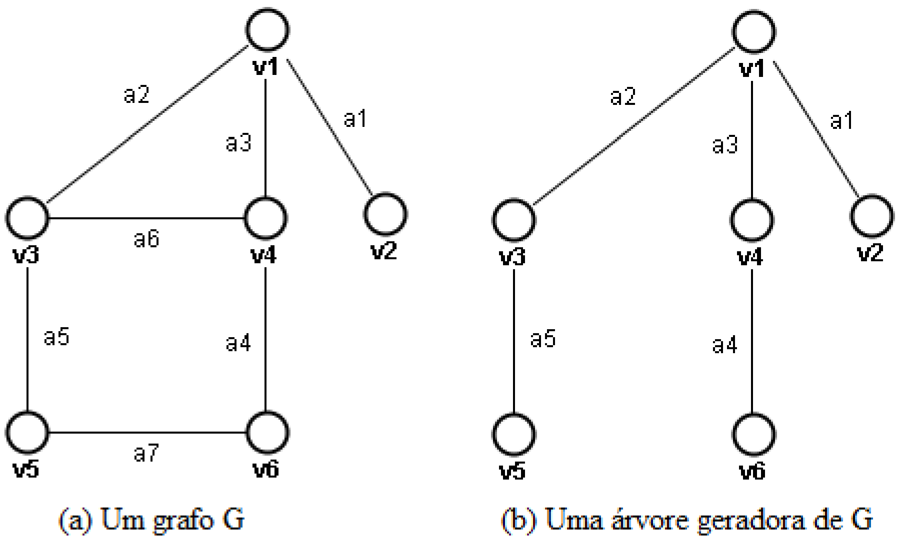


Figura 21: Um grafo G e uma de suas árvores geradoras

Na figura 21, temos um grafo e uma de suas árvores geradoras, neste caso, as arestas a6 e a7 são elos do grafo da figura 21.a, sendo 2 (arestas – vértices + 1 = 7 – 6 + 1) o posto do grafo.

1.2.27.CICLO FUNDAMENTAL

O ciclo fundamental de um grafo G é formado pela adição de um único elo de G, em uma de suas árvores geradoras. Chamamos de conjunto fundamental de ciclos, o conjunto de todos os ciclos fundamentais de G, em relação a uma de suas árvores geradoras. No grafo da figura 21.a, temos um grafo que possui dois ciclos fundamentais, em relação a sua árvore geradora presente na figura 21.b.

1.2.28.ÁRVORES ENRAIZADAS

Uma árvore é denominada enraizada (que possui arborescência), quando um de seus vértices é escolhido como inicial da árvore. Este vértice inicial é então chamado de raiz da árvore. Uma árvore sem raiz é denominada árvore livre.

Em árvores enraizadas, se a ordenação dos filhos for relevante, ou seja, a ordenação tenha que existir, então dizemos que a árvore é uma árvore enraizada ordenada. Árvores enraizadas ordenadas isomórficas devem preservar a ordem de seus filhos, caso contrário, não serão isomórficas, mesmo que preservem os mesmos vértices e arestas da árvore inicial.

1.2.29.NÍVEL EM ÁRVORES

Em árvores enraizadas, é possível determinar níveis para cada um dos vértices da árvore. Assim, dizemos que a raiz está no nível zero. Os vértices, adjacentes a raiz, estão no nível um. Os vértices cuja distância até a raiz for igual a duas arestas, estão no nível dois, e assim por diante. A figura 22 ilustra uma árvore e os níveis de seus vértices.

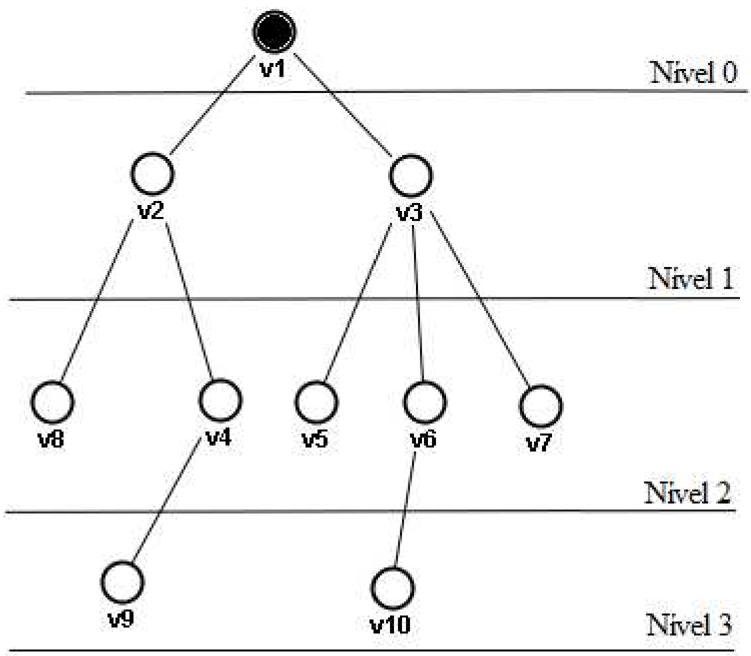


Figura 22: Uma árvore enraizada e os níveis de seus vértices

Dessa forma, supondo v e w dois vértices de uma árvore enraizada G, se v pertence ao caminho entre a raiz e w, dizemos que o vértice v é **ancestral** de w e w é **descendente** de v. Se ainda $v \neq w$, então dizemos que v é ancestral próprio de w e w é descendente próprio de v. Como exemplo, observamos a figura 22, nela temos o vértice v2 que é ancestral de v9. Automaticamente, v9 é descendente de v2. A raiz é ancestral de todos os vértices.

Caso v seja ancestral próprio de w e exista uma aresta entre v e w, dizemos que v é pai de w e w é filho de v. Como exemplo, vejamos a figura 22, nela o vértice v6 é pai do vértice v10 e o vértice v10 é filho do vértice v6. Dois vértices com o mesmo pai são chamados de irmãos. Uma **folha** é um vértice que não possui filhos.

A **altura** de uma árvore é igual ao valor do maior nível da árvore. A árvore da figura 22 possui uma altura igual a três.

1.2.30.SUBÁRVORE

Uma subárvore de uma árvore G, denotada por G', e uma árvore formada por um vértice v de G, e todos os descendentes de v em G, sendo v diferente da raiz de G. Caso algum vértice de G, descente de v, não pertença a G' conexo, dizemos então que G' é uma subárvore parcial de G. Na figura 23.a temos uma subárvore da árvore constante na figura 22. Na figura 23.b, uma subárvore parcial da mesma árvore.

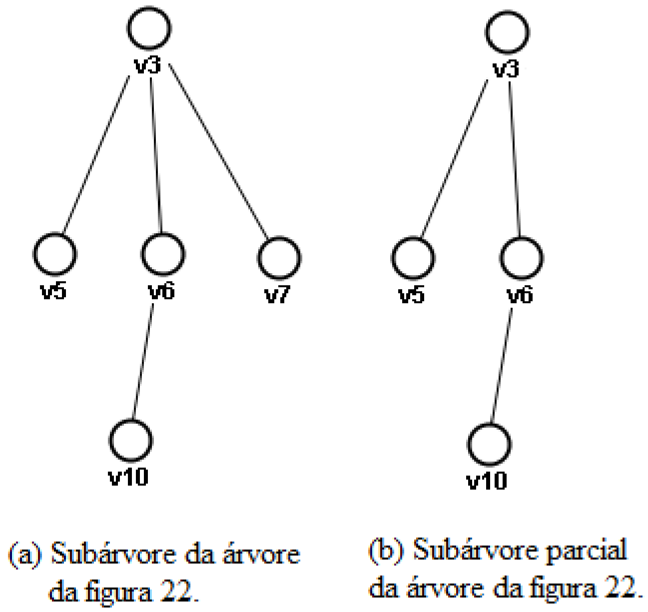


Figura 23: Uma subárvore e uma subárvore parcial do grafo da figura 22.

1.2.31.ÁRVORE ESTRITAMENTE M-ÁRIA

Uma árvore estritamente m-ária G, é uma árvore enraizada ordenada, em que cada vértice não folha, possui exatamente m filhos, onde $m \geq 1$. Quando $m = 2$, dizemos que a árvore é estritamente binária.

Seja G, uma árvore estritamente m-ária rotulada, cujos rótulos são, exatamente, sua posição na ordenação da árvore. Chamamos simplesmente de árvore m-ária, uma subárvore ou uma subárvore parcial dessa árvore G.

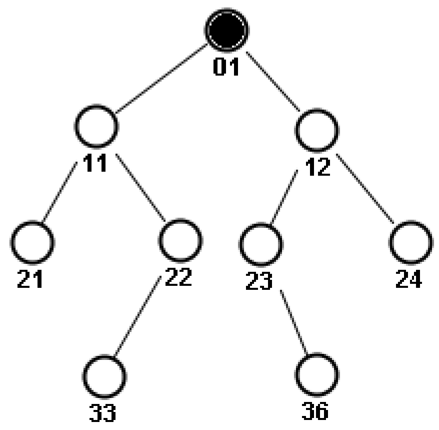


Figura 24: Uma árvore binária

Como exemplo de árvore m-ária, temos a árvore binária da figura 24, nela, podemos identificar o filho esquerdo e direito de cada vértice. Como a subárvore pode ser parcial, o filho esquerdo pode existir sem o direito e vice-versa. No exemplo da figura 24, os rótulos de cada vértice são compostos de dois valores, o primeiro identifica o nível do vértice e o segundo, sua ordem na árvore, assim, podemos dizer que o vértice 36, é o sexto vértice do nível três. Isto significa que, com certeza, ele é o filho direito do terceiro vértice do nível dois. Se excluirmos os vértices 33 e 36 da árvore da figura 24, a árvore passa a ser estritamente binária.

1.2.32.CORTE DE VÉRTICE

Um corte de vértice de G é um subconjunto minimal de vértices, cuja remoção de qualquer vértice desse conjunto, desconecta G. Na figura 21.b, a remoção do vértice v3, desconecta v5 do restante do grafo, tornando-o desconexo, portanto v3 faz parte do corte de vértice do grafo da figura 21.b.

Analogicamente, um corte de arestas é um subconjunto minimal de arestas, cuja remoção de qualquer aresta desse conjunto, desconecta G. Na figura 21.b, a remoção da aresta a3 desconecta o grafo, portanto a3 faz parte do corte de arestas do grafo da figura 21.b.

No grafo da figura 25, podemos facilmente identificar o corte de vértice e de aresta. O corte de vértice pode ser composto pelo vértice v2 ou pelo vértice v6 e o corte de arestas, pode ser composto pelos conjuntos: {a4}, {a1, a3}, {a5, a6}, etc.

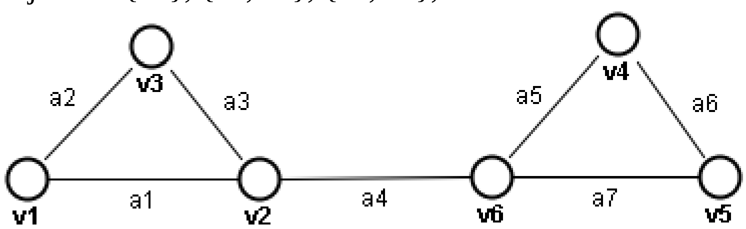


Figura 25: Um grafo conexo

Neste caso, por exemplo, o conjunto {a1, a3} é um corte de arestas do grafo da figura 25, pois juntos desconectam o grafo, mas isolados, não conseguem fazê-lo. Denomina-se **conectividade de vértices**, a quantidade de elementos (cardinalidade) do menor corte de vértices de G. No grafo da figura 25, a conectividade de vértices é igual a 1. Denomina-se **conectividade de arestas**, a quantidade de elementos (cardinalidade) do menor corte de arestas de G. No grafo da figura 25, a conectividade de arestas é igual a 1.

Se G for um grafo completo Kn, com n > 1, então não existe subconjunto próprio de vértices que desconecte G.

1.2.33.ARTICULAÇÃO E PONTE

Uma articulação de um grafo G (V, E) conexo é um vértice de G, que quando removido, torna G desconexo. Na figura 25, o vértice v2 é uma articulação do grafo representado na figura. Uma ponte (ou istmo) de um grafo G (V, E) conexo é uma aresta de G, que quando removida, torna G desconexo. Na figura 25, a aresta a4 é uma ponte do grafo representado na figura.

Dessa forma, podemos definir um grafo **biconexo em vértices** quando, no grafo, não possuir articulações e **biconexo em arestas**, quando não possuir pontes. Quando o grafo não possui articulações, é comum o uso do termo **SCAM** (SubConjunto de Articulação Minimal), que é o conjunto mínimo de vértices capaz de tornar o grafo desconexo.

A **extraconectividade** k de um grafo G é a menor quantidade (cardinalidade) de um SCA (subconjunto de articulação), cujas componentes conexas criadas a partir da exclusão do SCA, possuem uma quantidade de vértices maior que k.

Denominam-se **componentes biconexos** de um grafo G, os subgrafos maximais de G, que sejam biconexos em vértices ou isomorfos a K_2. Denomina-se **bloco** de um grafo G, todo subgrafo 2-conexo maximal de G.

Lema 02: Seja G um grafo. Então:
(i) Cada aresta de G pertence a exatamente um bloco do grafo;
(ii) Um vértice v de G é articulação se e somente se v pertencer a mais de um bloco de G.

Lema 03: Seja G (V, E) um grafo conexo, com |V| > 2. Então:
(i) Um vértice v do grafo é articulação de G se e somente se existirem, no grafo, vértices w, u ≠ v tais que v está contido em todo caminho entre w e u em G.
(ii) Uma aresta (p, q) de G é ponte se e somente se (p, q) for o único caminho simples entre p e q em G.

Lema 04: Um grafo G (V, E), com |V| > 2, é biconexo se e somente se cada par de vértices de G está contido em algum ciclo.

Teorema 06: Seja G um grafo k-conexo, então existe algum ciclo de G passando por cada subconjunto de k vértices.

Teorema 07: Um grafo G(V, E) é k-conexo, se e somente se existissem k caminhos disjuntos (exceto nos extremos) entre cada par de vértices de G.

1.2.34.PLANARIDADE

Um grafo é planar se puder ser escrito em um plano sem o cruzamento de arestas, com exceção nos vértices. Na figura 26, temos um exemplo de grafo planar.

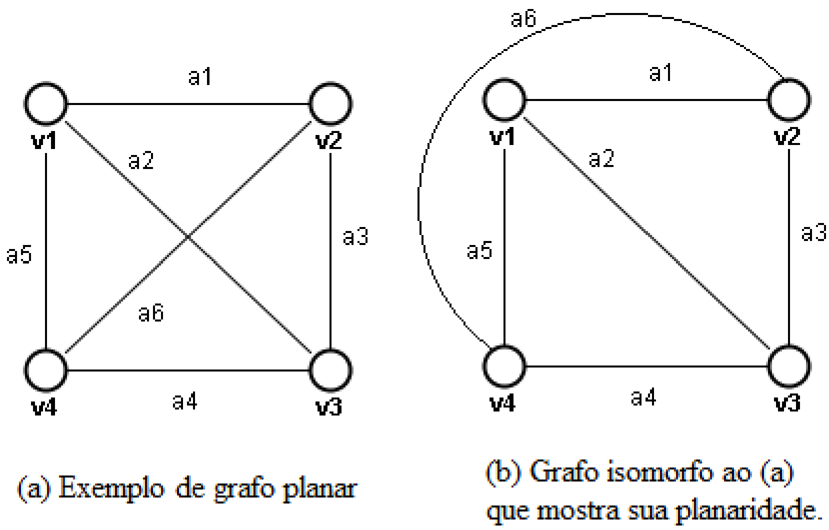


Figura 26: Grafo planar

Dizemos que o grafo é **imersível** em uma superfície S, se puder ser escrito em S sem cruzar suas arestas. Um grafo que não pode ser desenhado em um plano é chamado de **não planar**.

Se qualquer adição de arestas, o grafo passar a ser não planar, dizemos que o grafo é planar maximal. Todo grafo planar possui uma **representação planar** (ou topológica). A representação planar de um grafo é o seu desenho utilizando-se apenas de linhas retas. Na figura 27, podemos verificar a representação planar do grafo da figura 26.

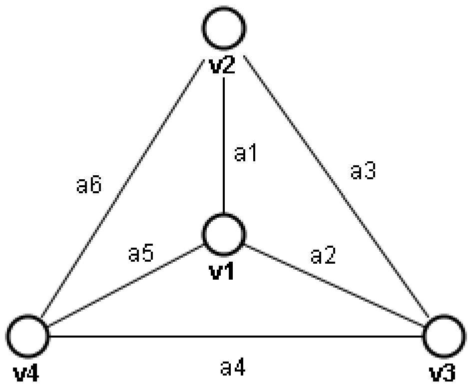


Figura 27: Representação planar do grafo da figura 26

1.2.35.FACES

As faces de um grafo planar são as regiões divididas pelas arestas do grafo. Dentro deste conceito, sempre vai existir uma e somente uma face não limitada pelas arestas do grafo, a essa face, damos o nome de face externa (perímetro do grafo). As fases de um grafo só poderão ser destacadas se o mesmo tiver sido escrito sem o cruzamento de suas arestas.

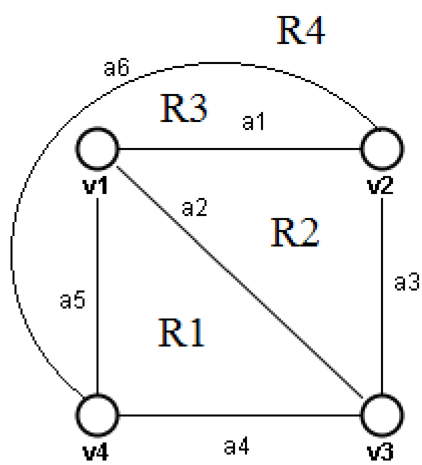


Figura 28: Grafo planar com suas faces identificadas

Na figura 28, podemos observar um grafo planar e as suas quatro faces, identificadas por Rn. A face R4 é a face externa.

Um grafo é chamado de **periplanar** ou planar exterior se for planar e todos os seus vértices estiverem na fronteira da fase externa. O grafo da figura 28 não é periplanar, pois o vértice v1 não faz fronteira com a face externa do grafo.

Teorema 08: Seja G um grafo planar, então n + f = m + 2, onde **n** é a quantidade de vértices, **f** é a quantidade de faces e **m** a quantidade de arestas.

O teorema 08 é conhecido como **fórmula de Euler** para poliedros.

Lema 05: Seja G um grafo planar, então $m \leq 3n - 6$, onde **n** é a quantidade de vértices e **m** a quantidade de arestas.

Lema 06: Os grafos K_5 e $K_{3,3}$ não são planares.

1.2.36.SUBDIVISÃO

A subdivisão de uma aresta (v, w) de um grafo G é uma operação que transforma a aresta em um caminho do tipo v, z1, z2, z3, z4, ..., zn, w, onde z são vértices de grau 2, adicionados a G.

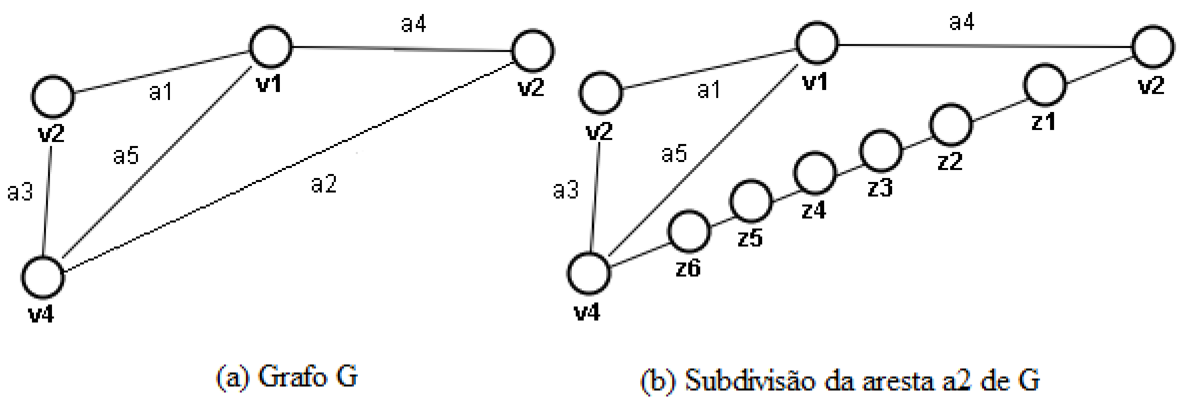


Figura 29: Um grafo e a subdivisão de uma de suas arestas

Na figura 29, vemos um grafo G e a aresta a2 de G, subdividida por seis vértices de grau 2.

Teorema 09: Um grafo é planar se e somente se não contiver, como subgrafo, uma subdivisão de K_5 ou $K_{3,3}$.

1.2.37.CICLO HAMILTONIANO

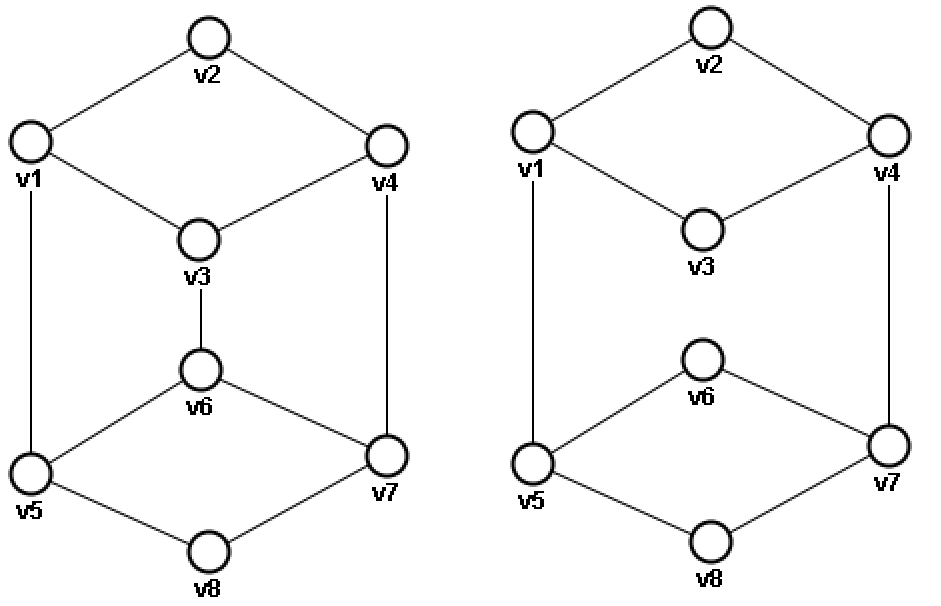
Um grafo hamiltoniano é um grafo que possui um ciclo hamiltoniano. Um ciclo hamiltoniano é um ciclo que contenha, uma única vez (com exceção do fechamento do ciclo), todos os vértices do grafo. Todo grafo hamiltoniano é biconexo em vértices, porém, nem todo grafo biconexo em vértices é hamiltoniano.

Na figura 30.a, podemos observar um grafo hamiltoniano, que possui o ciclo hamiltoniano v1, v5, v8, v7, v6, v3, v4, v2, v1. Na figura 30.b, é mostrado um grafo biconexo em vértices que não possui nenhum ciclo hamiltoniano.

Não existe um algoritmo satisfatório que indique que um grafo possui ou não um ciclo hamiltoniano.

Teorema 10: Seja G(V, E) um grafo hamiltoniano e S um subconjunto próprio de V, então o número de componentes conexos de $G - S \leq |S|$, onde |S| é a quantidade de vértices de S.

Teorema 11: Seja G(V, E) um grafo com pelo menos 3 vértices e tal que o grau $(v) \geq n / 2$, para todo vértice $v \in V$, então G é hamiltoniano.



(a) Grafo hamiltoniano (b) Grabo biconexo não hamiltoniano

Figura 30: Um grafo hamiltoniano e um grafo biconexo não hamiltoniano

1.2.38.COLORAÇÃO

A coloração de um grafo G é a atribuição de uma quantidade de cores diferentes para cada vértice de G, de tal modo que dois vértices adjacentes não possuam a mesma cor.

Chamamos de k-coloração de G, uma colocação do grafo com k cores, diz-se então que G é k-colorível. Por exemplo, um grafo bicolorível, é aquele que pode receber uma coloração com apenas duas cores.

Denomina-se **número cromático** de um grafo G ao menor número de cores k, para o qual existe uma k-coloração de G. Uma coloração que utiliza a quantidade mínima de cores é chamada de **mínima**. Colorir um grafo é fácil, basta colocar uma cor diferente a cada vértice do grafo, o que já garante que dois vértices adjacentes não terão a mesma cor. O difícil é encontrar a coloração mínima do grafo.

Na figura 31, podemos ver o mapa de Portugal com todas as suas regiões, o grafo que representa esse mapa, onde cada vértice é uma região e regiões fronteiriças são adjacentes no grafo e a sua coloração mínima. Neste caso, o número cromático do grafo é igual a 3.

Teorema 12: Todo grafo planar, possui, no máximo, número cromático igual a 4.

Lema 07: Um grafo G(V, E) é bicolorível se e somente se for bipartido.

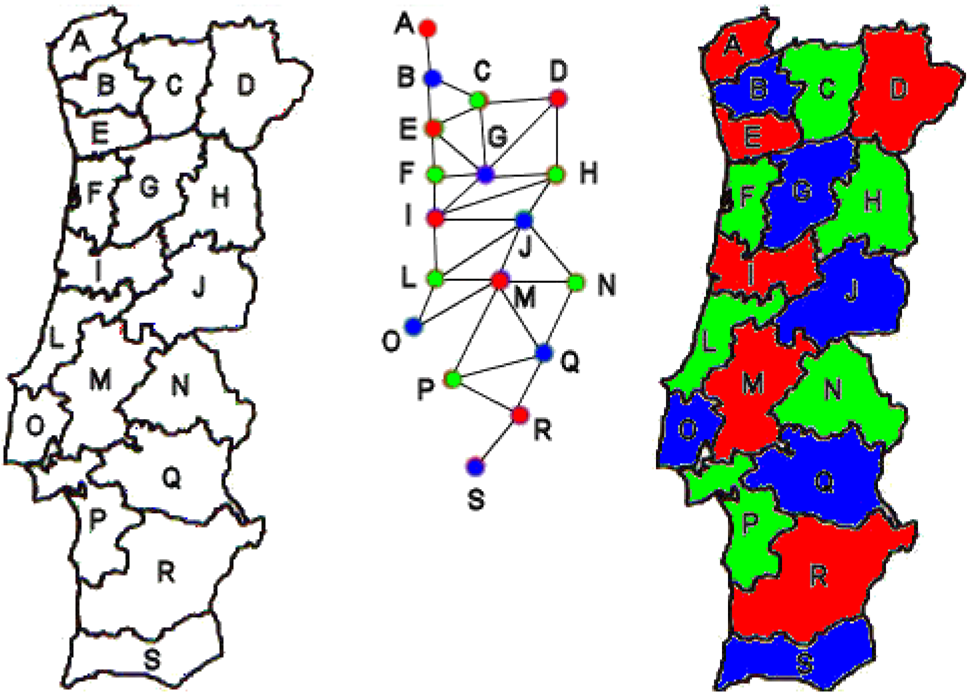


Figura 31: Grafo das regiões de Portugal

1.2.39.K-CRÍTICO

Um grafo G é denominado de k-crítico quando seu número cromático for igual a k e todo subgrafo próprio de G tem número cromático menor que k. Todo grafo possui um subgrafo k-crítico e todo grafo k-crítico é conexo. O grafo da figura 14.a é 3-crítico.

Teorema 13: Se G(V, E) é k-crítico, então grau (v) $\geq$ k – 1, para todo v $\in$ V.

1.2.40.COLORAÇÃO DE ARESTAS

De maneira análoga a coloração de vértices, a coloração de arestas visa colorir as arestas de maneira que duas arestas adjacentes tenham cores distintas. Denominamos de k-coloração de arestas, se k cores foram utilizadas com essa restrição. O menor valor possível de k, para um grafo G, é chamado de **índice cromático**.

Teorema 14: O índice cromático de um grafo é igual a delta ou a delta+1 (onde delta é o grau máximo do grafo).

Grafos com índice cromático igual a delta são classificados como Classe 1, os demais, são Classe 2. Uma coloração é dita **harmoniosa**, se o grafo G for colorido em vértices e arestas e cada aresta possui uma cor diferente de todas as demais arestas e vértices. O número cromático harmonioso é a quantidade mínima de cores para uma coloração harmoniosa em G.

1.2.41.EMPARELHAMENTO

Um emparelhamento M (matching ou acoplamento) de um grafo G(V, E) é um conjunto de arestas onde todo vértice de G incide em no máximo um elemento de M, ou seja, um vértice de G não pode ser extremo de duas arestas distintas de um emparelhamento. Um laço não pode fazer parte de um emparelhamento porque incide duas vezes em um mesmo vértice. Como exemplo, vejamos o grafo da figura 27, nele, são exemplos de emparelhamento os conjuntos {a2, a6} e {a3, a5}. Se um acoplamento envolver todos os vértices do grafo, então ele é chamado de acoplamento perfeito.

1.2.42.CAMINHO ALTERNANTE

Seja M um emparelhamento qualquer. Um caminho M-alternante é um caminho simples que alterna arestas em M e arestas que não estão em M. Um caminho M-aumentante é um caminho M-alternante, onde os extremos não são arestas de M. Pela definição, podemos observar que um caminho M-aumentante é necessariamente de comprimento ímpar. Um ciclo M-alternante é um ciclo formado por um caminho M-alternante de comprimento par. Um **broto** é um ciclo ímpar que é um caminho M-alternante. O único vértice do broto que não faz parte de M é chamado de **base do broto** e as arestas que compõe o caminho alternante chamados de **talo**. Um **cociclo** é o conjunto de arestas que compõe um ciclo M-alternante ou um broto.

1.2.43.ARBORICIDADE

A arboricidade de vértices de um grafo G é o menor número de subconjuntos de vértices nos quais se pode particionar um grafo, de tal forma que cada um deles induza um grafo sem ciclos. O grafo da figura 44 tem arboricidade de vértices igual a 2, com os conjuntos {v1, v2, v3, v4, v6} e {v5}. A arboricidade de arestas de um grafo G é o menor número de subconjuntos de arestas nos quais se pode particionar um grafo, de tal forma que cada um deles induza um grafo sem ciclos. O grafo da figura 27 tem arboricidade de arestas igual a 2, com os conjuntos {a2, a4, a6} e {a1, a3, a5}.

1.2.44.DÍGRAFO

Os grafos mostrados até o momento são chamados de grafos não direcionados, pois, se uma aresta (v, w) pertence a G, então v é adjacente de w e w é adjacente de v.

Um grafo direcionado, mais conhecido como dígrafo, é um grafo onde as arestas são pares ordenados, ou seja, a aresta (v, w) é diferente da aresta (w, v). Dessa forma, uma aresta (v, w) é **divergente** de v (incidência exterior da aresta em v) e **convergente** a w (incidência interior da aresta em w).

Os conceitos estudados até agora também servem, de forma análoga, aos dígrafos (exemplo: caminho simples, trajeto, etc.), porém, alguns sofrem ligeiras modificações, como por exemplo, os dígrafos podem possuir um **circuito** (ciclo em dígrafos) de comprimento 2, no caso de ambas as arestas (v, w) e (w, v) pertencerem a esse dígrafo. A figura 32 ilustra um dígrafo.

No dígrafo da figura 32, podemos perceber a diferença de um grafo normal e um dígrafo. Como exemplo desta afirmação, vejamos que existe um caminho entre v2 e v3: {a2, a3}, porém, não existe caminho entre v3 e v2.

Um **cocircuito** é um cociclo em dígrafos, onde o grau de entrada ou de saída é igual a zero.

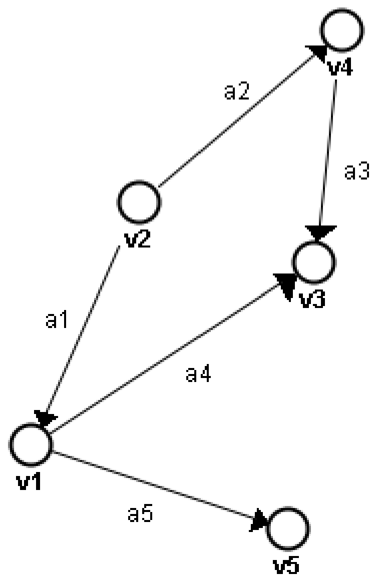


Figura 32: Um dígrafo

1.2.45.GRAU DE ENTRADA E SAÍDA

Denominamos de **grau de entrada** (ou semigrau exterior) de um vértice v', a quantidade de arestas que chegam até v'. Denominamos ainda de **grau de saída** (ou semigrau interior), a quantidade de arestas que partem de v'. Como exemplo, observemos a figura 32, nela temos:

Vértices	Grau de Entrada	Grau de Saída
v1	1	2
v2	0	2
v3	2	0
v4	1	1
v5	1	0

1.2.46.FONTE E SUMIDOURO

Uma fonte de um dígrafo é um vértice cujo grau de entrada é igual zero. Um sumidouro (ou poço) de um dígrafo é um vértice cujo grau de saída é igual a zero. Na figura 32, o vértice v2 é uma fonte e o vértice v3 é um sumidouro, por exemplo.

1.2.47.SUBJACENTE

Se forem retiradas as direções das arestas de um dígrafo D, obtém-se um multigrafo não direcionado, chamado de grafo subjacente a D. Um grafo não direcionado pode ser subjacente a até 2^m dígrafos, onde m é a quantidade de arestas.

1.2.48.FORÇAS DE CONEXÃO

Um dígrafo D(V, E) é **fortemente conexo** (ou f-conexo) quando para todo vértice v, w ϵ V, existir um caminho em D de v para w e de w para v. Caso exista, ao menos, um desses caminhos, para todo v, w ϵ V, então D é **unilateralmente conexo** (ou semi-fortemente conexo, ou ainda sf-conexo). D é chamado de **fracamente conexo** (ou simplesmente conexo /

s-conexo) se seu grafo subjacente for conexo. D é chamado de desconexo se seu grafo subjacente for desconexo.

Com esse conceito, faz-se existir as categorias de conectividade. Diz-se então que um dígrafo:

- pertence a categoria C_3, se for fortemente conexo;
- pertence a categoria C_2, se for unilateralmente conexo e não for fortemente conexo;
- pertence a categoria C_1, se for simplesmente conexo e não for unilateralmente conexo;
- pertence a categoria C_0, se for desconexo.

Obviamente, todo grafo fortemente conexo é unilateralmente conexo e simplesmente conexo, porém, nem todo grafo unilateralmente conexo é fortemente conexo. Todo grafo unilateralmente conexo é simplesmente conexo, porém, nem todo grafo simplesmente conexo é unilateralmente conexo.

Se existir algum caminho em um dígrafo D de um vértice v, para um vértice w, então diz-se que v alcança ou atinge w, sendo este **alcançável** ou **atingível** por v.

Na figura 33 podemos ver exemplos de dígrafos. No dígrafo da figura 33.a, de qualquer vértice é possível alcançar qualquer outro vértice, portanto este dígrafo é fortemente conexo. No dígrafo da figura 33.b, partindo de v3, é possível alcançar ou atingir qualquer outro vértice, porém, essa afirmação não se repete para os demais vértices, portanto, este dígrafo é unilateralmente conexo. No dígrafo da figura 33.c, partindo de qualquer vértice, é impossível atingir todos os outros vértices, portanto, este dígrafo é apenas fracamente conexo.

Em um dígrafo com fontes e sumidouros, se forem acrescentadas arestas (arestas de retorno) ligando cada um dos sumidouros a cada uma das fontes, e esta operação tornar o grafo fortemente conexo, dizemos que o grafo está em sua forma **canônica**.

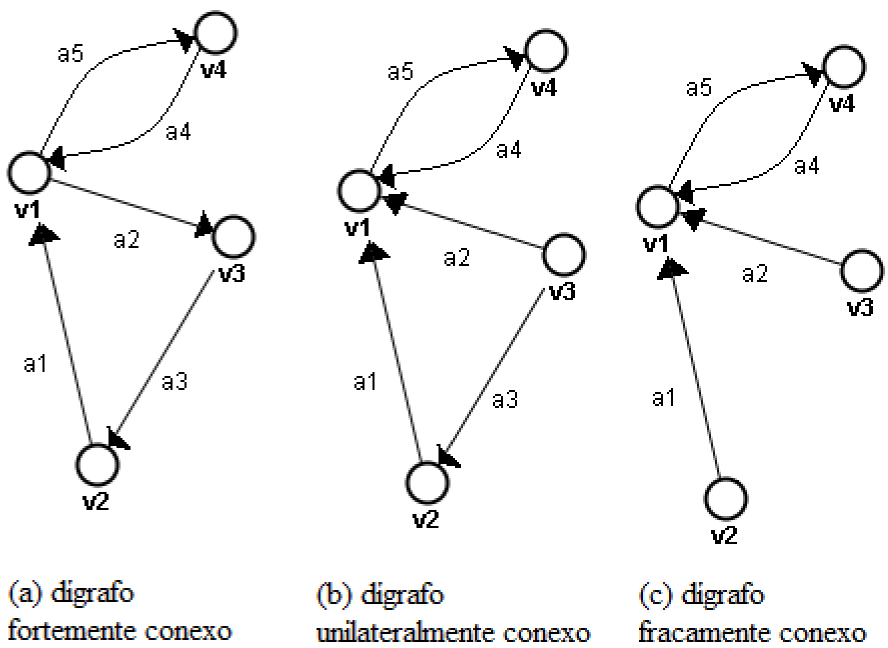


Figura 33: Dígrafos e suas forças de conexão

1.2.49.DÍGRAFOS ACÍCLICOS

Um dígrafo é acíclico quando não possuir ciclos. Um dígrafo acíclico pode ter um grafo subjacente acíclico ou cíclico, porém, se o grafo subjacente for acíclico, o seu dígrafo também o será. O dígrafo da figura 32 é acíclico e possui um grafo subjacente cíclico.

Lema 08: Todo dígrafo acíclico possui, pelo menos, uma fonte e um sumidouro.

1.2.50.FECHO TRANSITIVO E REDUÇÃO TRANSITIVA

Seja D(V, E) um dígrafo acíclico. Denomina-se **fechamento transitivo** de D, ao maior dígrafo D'(V, E') que preserva a alcançabilidade de D, ou seja, para todo v, w ϵ V, se v alcança w em D, então a aresta (v, w) ϵ E'. Denomina-se **redução transitiva** de D, ao menor dígrafo D''(V, E'') que preserva a alcançabilidade de D, ou seja, se (v, w) ϵ E'' então v não mais alcança w se retirarmos a aresta (v, w) de D. A redução transitiva também é conhecida como diagrama de Hasse.

Na figura 34.a temos um exemplo de dígrafo. Na figura 34.b temos o fecho transitivo do dígrafo da figura 34.a e na figura 34.c temos a redução transitiva do dígrafo da figura 34.a.

Seja D(V, E) um dígrafo e v um vértice de G. O **fecho transitivo direto** de um vértice v é o conjunto de vértices de G atingíveis a partir de v. O **fecho transitivo inverso** de um vértice v é o conjunto de vértices de G que atingem v.

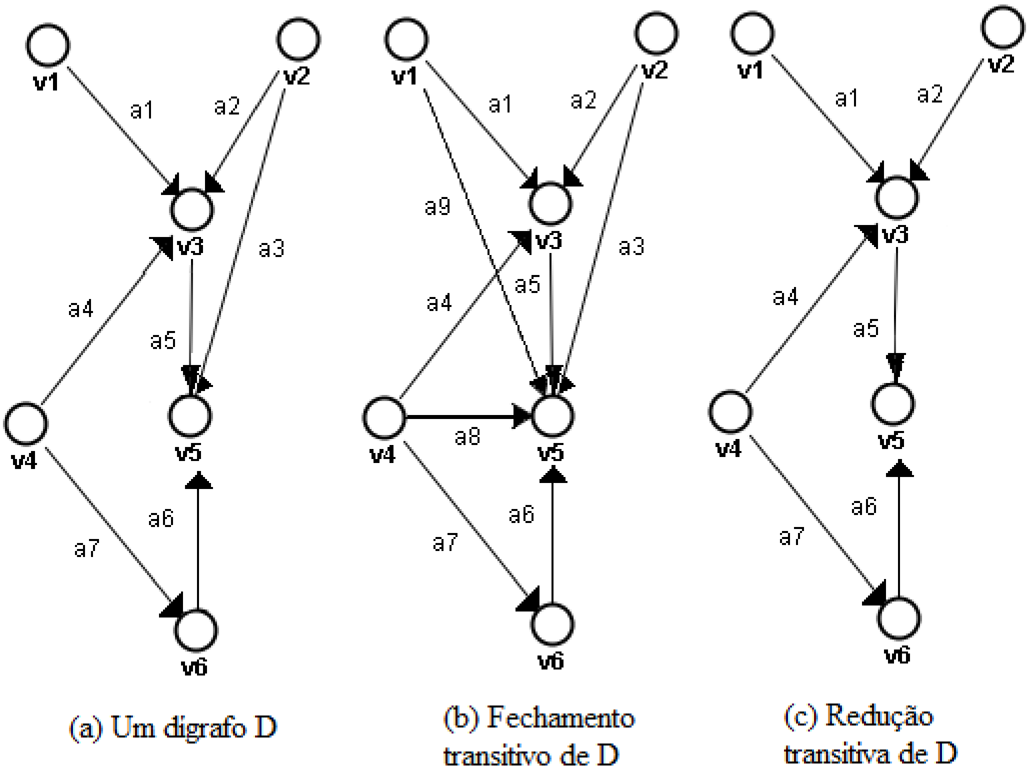


Figura 34: Um dígrafo, seu fecho transitivo e sua redução transitiva.

1.2.51.ORDENAÇÃO PARCIAL

Um conjunto parcialmente ordenado ou ordenação parcial (S, ≤) é um conjunto não vazio S e uma relação binária ≤ em S, satisfazendo às seguintes condições:

(i) $s1 \leq s1$, para $s1 \in S$ (≤ é reflexivo)
(ii) $s1 \leq s2$ e $s2 \leq s1 \rightarrow s1 = s2$, para s1 e $s2 \in S$ (≤ é anti-simétrica)
(iii) $s1 \leq s2$ e $s2 \leq s3 \rightarrow s1 \leq s3$, para s1, s2 e $s3 \in S$ (≤ é transitiva)

Um conjunto parcialmente ordenado ou ordenação parcial (S, ≤) pode ser também caracterizado pelo par (S, <), onde < é uma relação binária em S definida por:

$$s1 < s2 \leftrightarrow s1 \leq s2 \text{ e } s1 \neq s2$$

A relação < satisfaz:

(i) $s1 \nless s1$, para $s1 \in S$ (< é irreflexivo)
(ii) $s1 < s2 \rightarrow s2 \nless s1$, para s1 e $s2 \in S$ (< é assimétrica)
(iii) $s1 < s2$ e $s2 < s3 \rightarrow s1 < s3$, para s1, s2 e $s3 \in S$ (< é transitiva)

Agora seja D’(V, E’), o fecho transitivo de um dígrafo acíclico D(V, E). Pode-se constatar que E’ é uma relação irreflexiva, assimétrica e transitiva. Conseqüentemente, fechamentos transitivos de dígrafos acíclicos e conjuntos parcialmente ordenados são conceitos equivalentes, ou seja, P(S, <) caracteriza um conjunto parcialmente ordenado se e somente se o dígrafo D’ (S, <) for o fechamento transitivo de algum dígrafo acíclico D. Diz-se então que D **induz** o conjunto parcialmente ordenado P(S, <). Note que s1 < s2 em P se e somente se existir um caminho em D de s1 para s2. Utiliza-se então a notação v < w para indicar que v alcança w no dígrafo acíclico D(V, E), sendo v, w ϵ V.

1.2.52.ÁRVORE DIRECIONADA ENRAIZADA

Uma árvore direcionada enraizada é um dígrafo D, que possui um único vértice R, denominado de raiz, com grau de entrada igual a zero, e todos os demais vértices tem grau de entrada igual a 1. Obviamente, D é acíclico com exatamente uma única raiz. Assim, dizemos que D tem arborescência. Podemos observar um exemplo de árvore direcionada enraizada na figura 35.

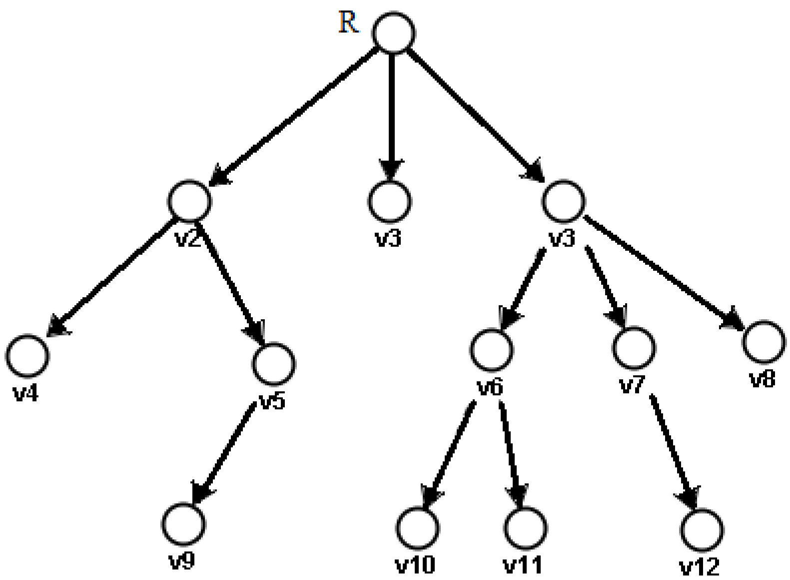


Figura 35: Árvore direcionada enraizada.

1.2.53. OPERAÇÕES COM GRAFOS

Operações são equações feitas nos grafos. Algumas operações não são aplicadas a dígrafos, outras são mais genéricas. Algumas são aplicadas a apenas um grafo, essas são chamadas de unárias, outras são aplicadas a pares de grafos, a essas damos o nome de binárias.

Nas operações mostradas aqui, vamos considerar os grafos G1(V1, E1) e G2(V2, E2) operandos e G(V, E) o resultado.

1.2.53.1. UNIÃO (U)

Dois grafos passam a ser um único grafo. Vértices iguais, passam a ser um único vértice.

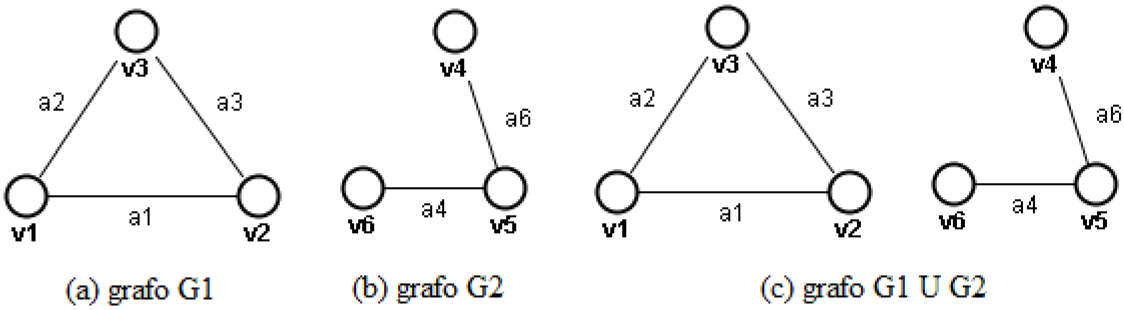


Figura 36: União de dois grafos.

1.2.53.2. SOMA (+)

Para todo vértice de G1, cria-se uma aresta para todo vértice de G2.

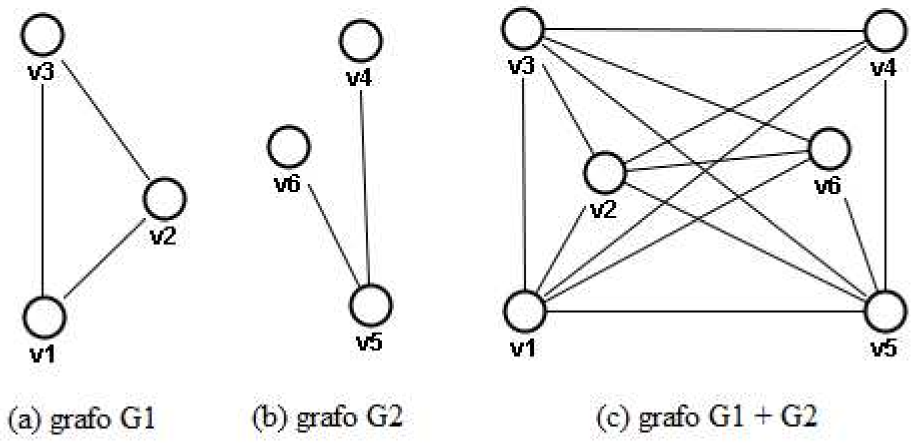


Figura 37: Soma de grafos

1.2.53.3. PRODUTO CARTESIANO (X)

No produto cartesiano, cada vértice de G1 é combinado com um vértice de G2, sendo cada vértice de G um par ordenado. Os vértices (v, w) e (p, q) são adjacentes se tiverem uma das seguintes condições:

1: v = p e (w, q) ϵ E2;
2: w = q e (v, p) ϵ E1;

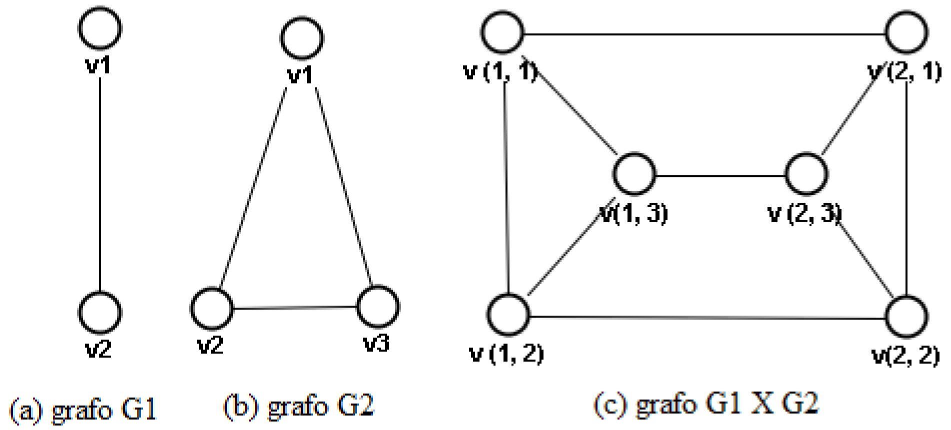


Figura 38: Produto cartesiano em grafos

1.2.53.4. COMPOSIÇÃO OU PRODUTO LEXOGRÁFICO (⊕)

Na composição lexográfica, cada vértice de G1 é combinado com um vértice de G2, sendo cada vértice de G um par ordenado. Os vértices (v, w) e (p, q) são adjacentes se tiverem uma das seguintes condições:

1: (v, p) ϵ E1;
2: v = p e (w, q) ϵ E2;

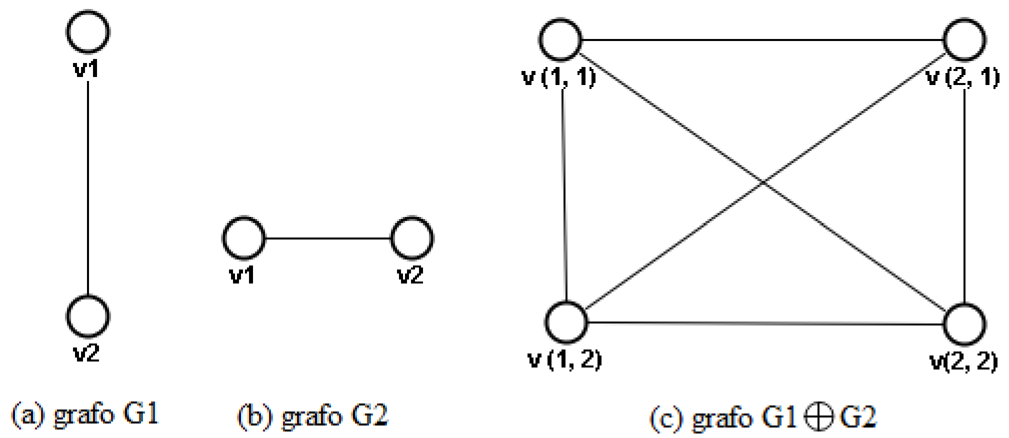
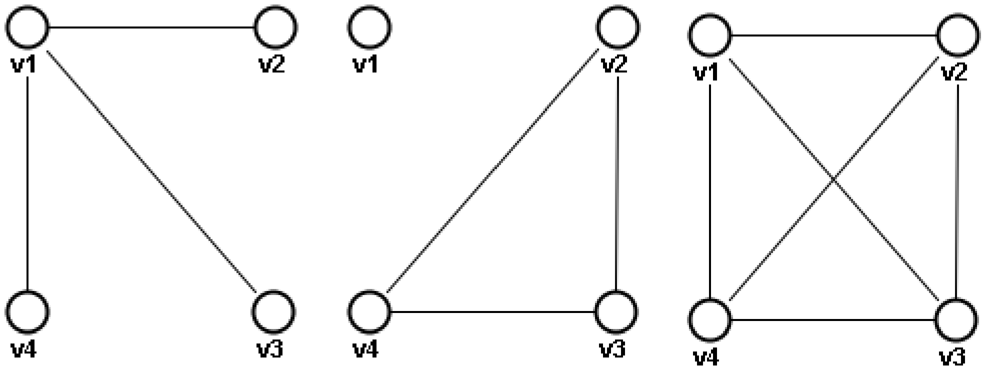


Figura 39: Composição lexicográfica em grafos

1.2.53.5. SOMA DE ARESTAS (++)

Na soma de arestas, a condição V1 = V2 deve ser satisfeita. G é um grafo que possui as arestas E1 e E2.

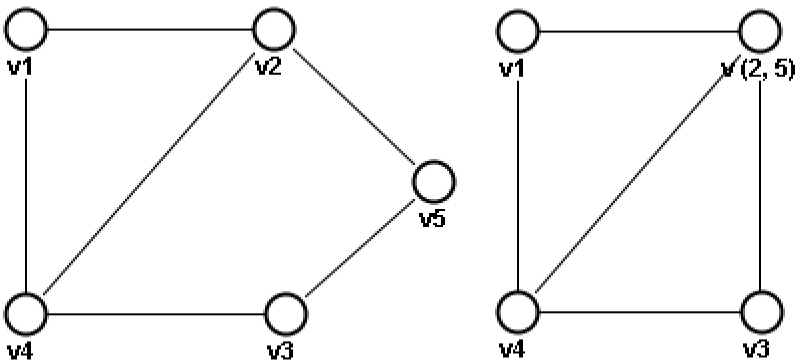


(a) grafo G1 (b) grafo G2 (c) grafo G1 ++ G2

Figura 40: Soma de arestas em grafos

1.2.53.6. CONTRAÇÃO DE DOIS VÉRTICES

Dois vértices v, w de G se transformam em um único vértice. Qualquer ligação existente para um dos vértices em separado ou para ambos existe para o vértice combinado. Uma aresta (v, z) e uma aresta (w, z) viram uma única aresta (vw, z) na contração de x com w. Essa operação é chamada de contração em arestas. A contração de vértices é uma operação é unária e pode ser feita em dígrafos. Se sucessivas contrações forem feitas, a partir de uma regra específica e pré-determinada, o resultado das contrações é chamado de **grafo reduzido**.



(a) grafo G1 (b) grafo G, contração entre os vértices v2 e v5

Figura 41: Contração de vértices

1.2.53.7. DESDOBRAMENTO DE VÉRTICES

Um vértice do grafo se transforma em dois com uma aresta ligando-os. Esta operação é unária e pode ser feita em dígrafos. Caso seja feita em dígrafos, o desdobramento não pode alterar a adjacência do dígrafo.

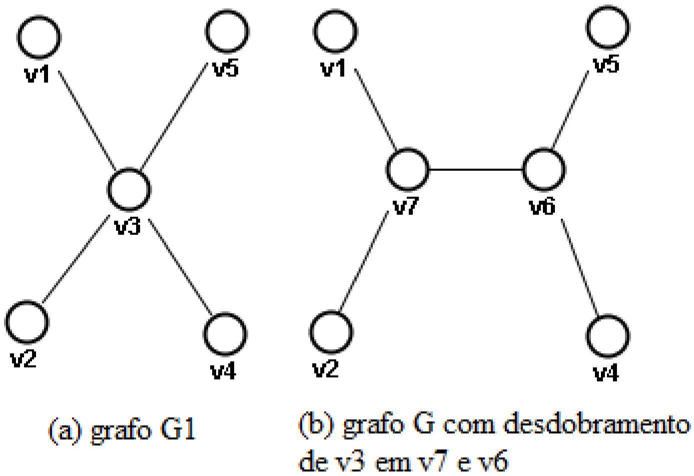


Figura 42: Desdobramento do vértice v3 em v7 e v6

1.2.53.8. INCLUSÃO OU INSERÇÃO

É a operação de se incluir um novo vértice e/ou aresta em um grafo. O grafo resultante é chamado de homeomorfo. É uma operação unária e pode ser feira em dígrafos. A inserção de apenas uma aresta é chamada de **religação**. O inverso da inclusão é a **remoção**, que também pode ser de vértices ou arestas.

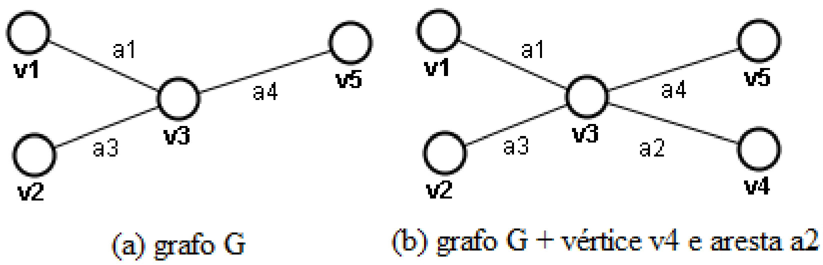


Figura 43: Inserção de vértices e arestas

Para mais operações, consultar a bibliografia contida no final deste escrito.

1.2.54.SUBCONJUNTOS ESTÁVEIS

Seja G(V, E) um grafo não orientado. Diz-se que S, subconjunto de V, é um **subconjunto internamente estável** (SCIE – conjunto independente/dominante), se dois vértices quaisquer de S nunca são adjacentes entre si.

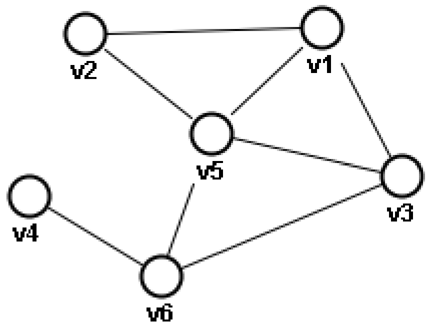


Figura 44: Um grafo G.

Para o grafo da figura 44, são exemplos de subconjunto internamente estável os conjuntos: {v2, v3}, {v1, v4} e {v2, v3, v4}. Agora, se para um subconjunto internamente estável S, não existir outro subconjunto internamente estável S' tal que S' ⊃ S, então S é um subconjunto internamente estável maximal. Para o grafo da figura 44, são exemplos de subconjunto internamente estáveis maximais os conjuntos: {v2, v3, v4}, {v1, v6} e {v4, v5}

Seja D(V, E) um dígrafo. Diz-se que T, subconjunto de V, é um **subconjunto externamente estável** (SCEE – conjunto absorvente) se todo vértice que não pertencente a T tiver pelo menos um vértice de T como sucessor. Agora, se dada um subconjunto externamente estável S não existe outro subconjunto externamente estável S' tal que S' ⊃ S, então S é dito ser um subconjunto externamente estável minimal.

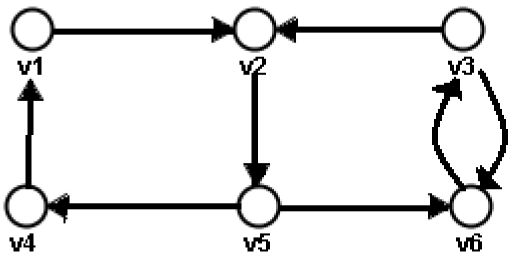


Figura 45: Um dígrafo D

No grafo da figura 45, são exemplos de subconjunto externamente estável minimais os conjuntos: {v2, v4, v6} e {v1, v5, v3}. Este conceito também pode ser aplicado a grafos não orientados, bastando que consideremos que todo vértice exterior a T deva ter como adjacente pelo menos um vértice de T.

O **núcleo** de um grafo é um subconjunto de seus vértices que sejam, simultaneamente, internamente e externamente estáveis. Se o grafo contiver um SCAM e for SCIE é conhecido como **grafo frágil**.

Teorema 15: Todo núcleo é, ao mesmo tempo, um SCIE maximal e um SCEE minimal.

Teorema 16: Todo dígrafo, sem circuitos, possui um núcleo único.

1.2.55.ESPESSURA

A espessura (thickness) de um grafo G é o menor número de subgrafos planares, que não possuem arestas em comum (disjuntos em relação a arestas), cuja união produz G. G só será planar se sua espessura for, no máximo, igual a 1.

A quantidade mínima da espessura de um grafo é dada pela equação m / (3n – 6), onde m é a quantidade de arestas e n a quantidade de vértices.

1.2.56.SIMETRIA

Um dígrafo D(V, E) é dito simétrico se e somente se para toda aresta (v, w) em D, existir a aresta (w, v) em D. Um grafo não orientado é sempre simétrico. Um dígrafo D(V, E) é dito anti-simétrico se e somente se existir a aresta (v, w) em D e não existir a aresta (w, v) em D, para qualquer aresta de D. O dígrafo é pseudo-simétrico quando todos os seus vértices possuírem grau de entrada igual ao seu grau de saída.

Teorema 17: Um dígrafo conexo possui um circuito euleriano se e somente se for pseudo-simétrico.

1.2.57.BASE E ANTI-BASE

Uma base de um dígrafo D(V, E) é um subconjunto B de V, tal que:

- dois vértices quaisquer de B não são ligados por nenhum caminho;
- todo vértice não pertencente a B pode ser atingido por um caminho partindo de B.

No exemplo da figura 46, temos um exemplo de base no conjunto B = {v1, v4}

Uma anti-base de um dígrafo D(V, E) é um subconjunto A de V, tal que:

- dois vértices quaisquer de A não são ligados por nenhum caminho;
- de todo vértice não pertencente a A, pode-se atingir A por algum caminho.

No exemplo da figura 46, temos um exemplo de anti-base no conjunto {v3, v6, v8}

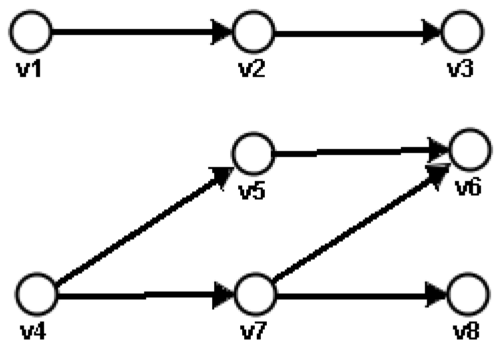


Figura 46: Um dígrafo

1.2.58.CONJUNTO FUNDAMENTAL E ANTI-FUNDAMENTAL

Um conjunto fundamental é o fecho transitivo direto de um vértice de uma base. Um vértice, que define um conjunto fundamental, é uma raiz do subgrafo induzido pelos vértices desse conjunto. Um conjunto anti-fundamental é o fecho transitivo inverso de um vértice de uma anti-base.

1.2.59.GÊNERO

O gênero de um grafo é o gênero da menor superfície na qual o grafo é imersível. O gênero de uma superfície é a quantidade de furos (ou alças) que a superfície possui. Por exemplo, um toro, mostrado na figura 47, é uma superfície de grau 1 pois possui um único furo.

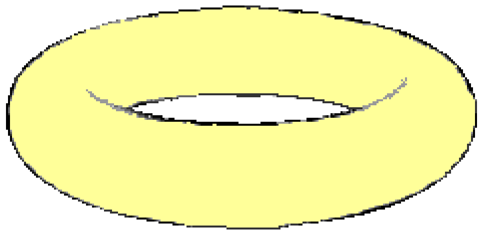
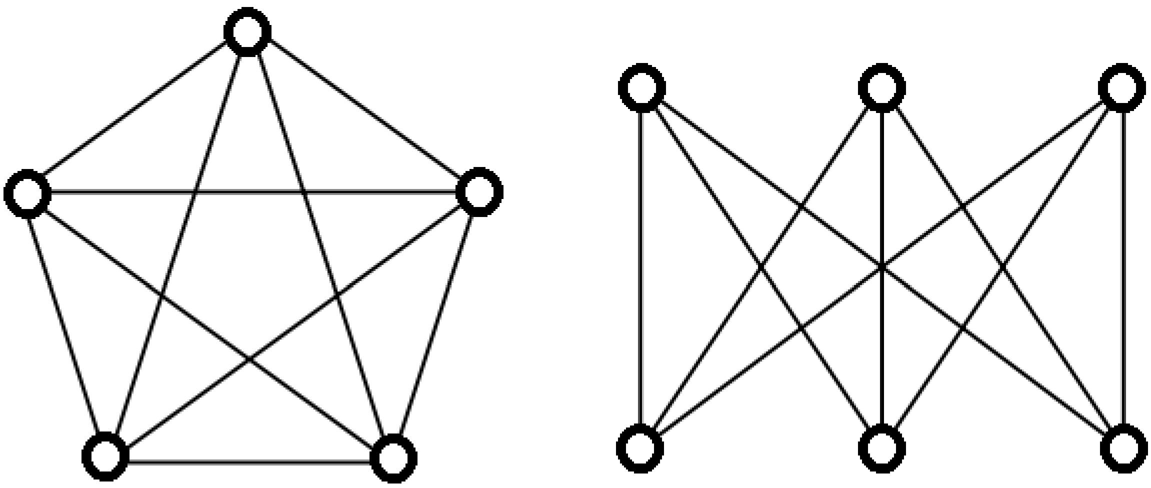


Figura 47: Um Toro

Um grafo planar, que é imersível em um plano, tem gênero igual a zero, pois o plano não possui furos. Os grafos K_5 e $K_{3,3}$, mostrados na figura 48, possuem gênero 1, pois não são imersíveis no plano, mas são imersíveis em um toro, pois a única aresta que os torna não planares, pode passar pelo furo do toro.



(a) Grafo K_5 (b) Grafo $K_{3,3}$

Figura 48: Grafos de gênero igual a 1

1.2.60.TRIANGULAÇÃO

Uma triangulação é um grafo planar conexo no qual todas as fases, exceto a face externa são compostas por exatamente três arestas, ou seja, por triângulos. Uma triangulação pode ser vista na figura 49. Uma **cápsula** é uma triangulação na qual o número de vértices de grau 3 não adjacentes a face externa é maior ou igual ao número de vértices restantes.

Teorema 18: Em um grafo planar maximal mais de 4 vértices, o conjunto de vértices de grau 3 é um SCIE.

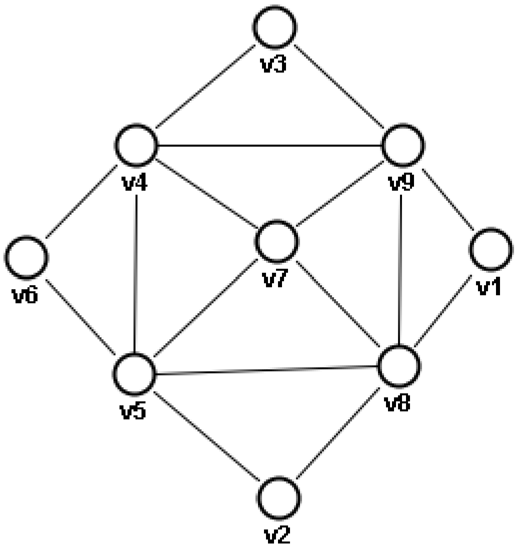


Figura 49: Uma triangulação

Um **triângulo separador**, em uma triangulação, é um triângulo que não seja fase. Um triângulo separador pode ser visto na figura 50, entre os vértices v1, v2 e v4. Uma triangulação que não possui triângulos separadores é chamada de NST-**Triangulação**. Todo grafo planar maximal, que seja uma triangulação e que não tenha um triângulo separador, possui ciclo hamiltoniano.

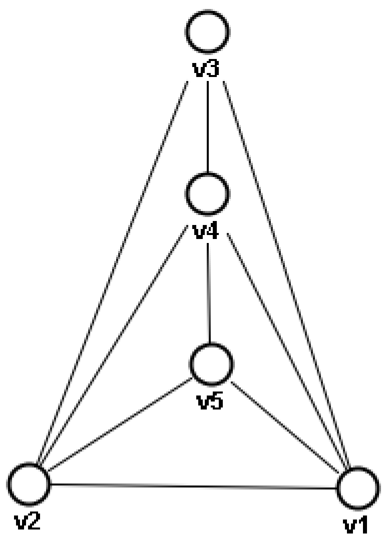
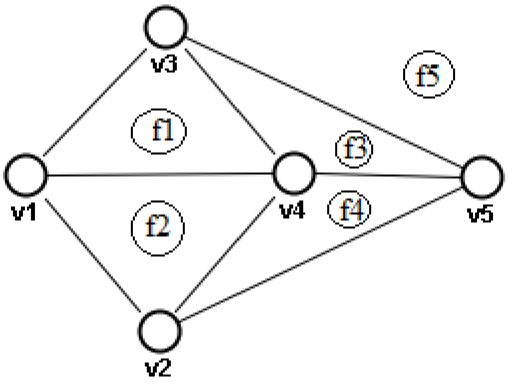


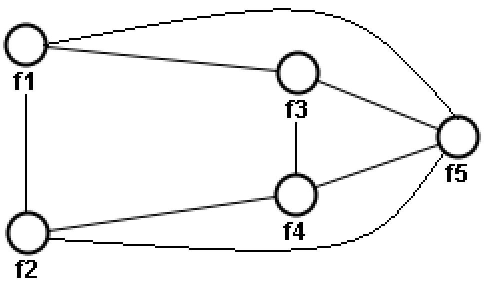
Figura 50: Uma triangulação com triângulo separador

1.2.61.DUAL

O grafo dual H, de um grafo G planar conexo, é o grafo onde cada vértice de H corresponde a uma fase de G e as arestas de H ligam os vértices que representam faces fronteiriças de G.



(a) um grafo G



(b) grafo H, dual ao grafo da figura (a)

Figura 51: Um grafo e seu dual

1.2.62.GRAFOS PERFEITOS

Um grafo G é dito perfeito se e somente se nem G, nem o complemento de G contiverem um ciclo ímpar de comprimento maior que 3 como subgrafo. Um grafo é criticamente imperfeito se não for perfeito, mas todos os seus subgrafos próprios forem perfeitos.

1.2.63.PANCÍCLICOS

Um grafo G é pancíclico se possuir ciclos de todos os comprimentos possíveis a partir de 3, dentro de sua quantidade de vértices. Por exemplo, um grafo com 5 vértices para ser pancíclico, precisa ter, pelo menos, um ciclo de tamanho 5, outro de tamanho 4 e outro de tamanho 3. Um grafo G é pancíclico em vértices se cada vértice pertencer a, pelo menos, um ciclo de cada tamanho. Um grafo G é pancíclico em arestas se cada aresta pertencer a, pelo menos, um ciclo de cada tamanho.

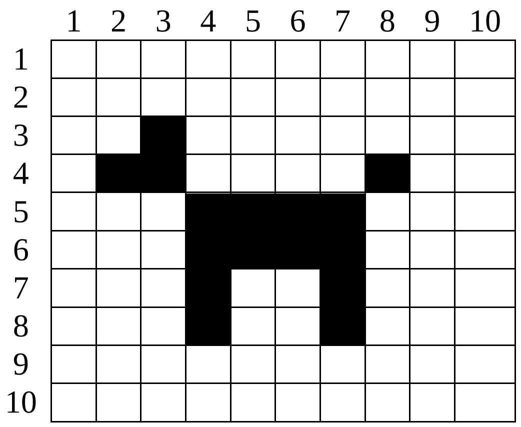
Um grafo G é fracamente pancíclico se possuir um ciclo de cada tamanho, entre sua cintura e sua circunferência.

1.2.64.TRAÇADO DE GRAFOS

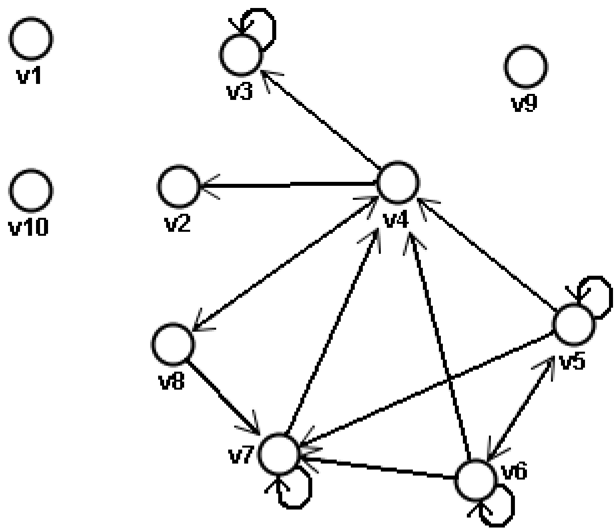
Uma dificuldade que aparece ao se lidar com grafos está na desconexão entre um determinado nível de conteúdo informativo presente nos esquemas e o restante desse conteúdo, comum a estes e a todas as outras formas de representação. Essa dificuldade fala da associação da estrutura representada no esquema de um grafo a algum suporte do mundo real. Um exemplo disso é um mapa, onde os pontos ou regiões possuem relações de posição bem definidas, as quais desaparecem (embora permaneçam seus dados quantitativos) ao se construir o modelo do grafo.

O problema existe porque muitas representações gráficas, através da forma como são construídas, fazem apelo a percepção global que é uma capacidade dos seres humanos e, simultaneamente essas representações fazem sentido ao se processarem os modelos respectivos. O conteúdo acessível à percepção global não está, porém, ao alcance do modelo lógico, que não tem como levá-la em conta.

Um exemplo claro dessa dificuldade é a do “caniche” de Kaufmann. Ele é visível sobre uma forma de **matriz figurativa de adjacência**, de significado evidente, conforme mostra a figura 52.a. A matriz é um grafo, o esquema ao lado (figura 52.b) também, no entanto a não apreensão do significado pictório da forma pela estrutura do grafo salta aos olhos.



(a) Matriz Figurativa do “caniche” de Kaufmann



(b) grafo do "caniche" de Kaufmann

Figura 52: "Caniche" de Kaufmann

Explicando a construção da Matriz Figurativa de Adjacência: a matriz figurativa é uma forma de representação de grafos que leva em consideração seu caráter pictório, ou seja, a visualização digital de seu conteúdo pelo usuário do sistema. Ela é construída da seguinte forma: os elementos constantes na coluna indicativa são os vértices de onde as arestas nascem e os elementos na linha indicativa, são os vértices para onde as arestas apontam. Por exemplo, se existir uma aresta partindo (divergindo) do vértice 4 e apontando (convergindo) para o vértice 2, a posição 4,2 (linha 4 – coluna 2) da matriz será escurecida, caso contrário, permanecerá branca. Na figura 52, temos a matriz figurativa de adjacência do “caniche” de kaufmann e sua representação no diagrama dos grafos, ponto de base para a criação da matriz. Note, na figura 52, que os vértices 1, 9 e 10 não possuem arestas, portanto, toda a sua linha e coluna na matriz figurativa de adjacência permaneceram brancas.

1.2.65.JOGOS EM GRAFOS

Diversos tipos de jogos podem ser definidos para execução em estruturas de grafo, o que fornece uma interessante extensão para a teoria dos jogos. Falamos aqui de jogos com dois jogadores, usando recursos iguais ou diferentes, conforme o caso.

Um jogo pode ser jogado sobre os vértices, aproveitando-se as ligações entre eles. Vale à pena lembrar que o conhecido **WAR** pertence a essa categoria. Pode também ser jogado sobre o conjunto de arestas e, nos exemplos que aqui apresentamos, cada jogada consistem em orientar uma aresta.

Neufeld e Nowakowki definiram um jogo "**polícia-e-ladrão**" e seu trabalho o estuda quando realizado em produtos de grafos. O jogador A manipula um ou mais policiai, que são colocados sobre os vértices, um ou mais em cada vértice escolhido. O jogador B move o ladrão. As jogadas são feitas entre vértices adjacentes e o jogador A vence se conseguir colocar um policial no mesmo vértice do ladrão, enquanto que o jogador B vence se conseguir criar uma situação de fuga indefinida (pode-se concluir que o jogador B vencerá sempre que o grafo for, por exemplo, constituído por um único caminho simples e fechado e houver um só policial, e perderá sempre, nesse mesmo grafo, se houverem dois policiais). O jogo pode ser ativo (alguém deve se movimentar em cada jogada) ou passivo (um jogador pode "passar" a jogada).

Dois jogos de orientação de arestas são o jogo **circuito** e o jogo da **arborescência**. No primeiro, estudado por Bollobás e Szabó, o jogador A tem por objetivo construir um circuito, enquanto que o jogador B tenta evitar essa construção. O artigo discute, entre outras questões, situações e estruturas que garantem a vitória de A. O jogo a arborescência foi discutido por Hamidoune e Las Vergnas e, nele, o objetivo de A é construir uma arborescência, enquanto que o jogador B tenta evitar que ela seja construída.

Berge apresentou diversos problemas relacionados com jogos combinatórios, discutindo suas relações com grafos. Kano definiu um jogo sobre um grafo, no qual cada jogador remove um conjunto de arestas que induza uma estrela, vencendo a partida o jogador que remover a última aresta.

1.2.66.GRAFOS DE CENA

Podemos definir um grafo de cena como uma coleção de objetos 3D arranjados em estrutura de árvore. Esta estrutura hierárquica encoraja um agrupamento natural dos objetos, facilitando transformações e permitindo a implementação mais eficiente de várias operações tais como detecção de proximidade, colisão, oclusão (obstrução da visão de um objeto por outro que está mais próximo do observador e sobre uma mesma direção de visão), entre outros.

Grafos de cenas são ferramentas conceituais utilizadas na representação de ambientes virtuais tridimensionais nas aplicações de computação gráfica. Um ambiente virtual é uma representação de diversos aspectos do mundo real ou abstrato. Os aspectos que são considerados em uma aplicação de computação gráfica são: posição do objeto, forma, textura da superfície, iluminação, descrição geométrica, câmera, transformação, aparência e comportamento.

Cada um desses aspectos deve ser inserido em um grafo de cena para representar o ambiente virtual. O grafo de cena é formado, portanto, por nós conectados por arestas compondo um grafo acíclico direcionado. Cada nó possui um conjunto de atributos que podem, ou não, influenciar seus nós conectados. Os nós são organizados de uma maneira hierárquica correspondendo semântica e espacialmente com o mundo modelado.

Os nós podem ser divididos em três categorias: nó raiz, nós intermediários que são chamados de nós internos ou nós de agrupamento e os nós folha que estão localizados no final de um ramo. O nó raiz é o primeiro nó do grafo e todos os outros nós estão ligados a ele direta ou indiretamente. O nós internos possuem várias propriedades, sendo o uso mais comum o de representar transformações 3D (rotação, translação e escala). Os nós folha contêm, geralmente, a representação geométrica de um objeto (ou dados de áudio, quando o grafo de cena possuir esse recurso).

Construindo um Grafo de Cena: Os nós devem ser conectados formando um grafo acíclico e direcionado (dígrafo). As conexões entre eles devem satisfazer as relações espaciais e semânticas existentes no mundo real. A figura 53 mostra um exemplo de um grafo de cena representando uma casa. A casa é dividida em quartos e os quartos possuem diversos objetos (nós folha), como cama e cadeira, por exemplo. O grafo foi construído de maneira a manter a noção intuitiva dos relacionamentos entre os objetos citados.

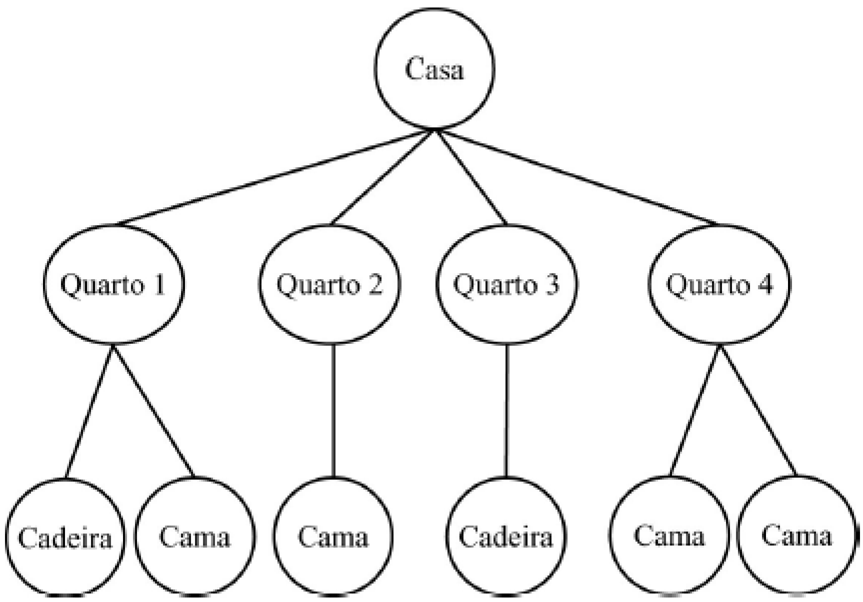


Figura 53: Grafo de Cena bem construído de uma casa

O mesmo grafo poderia ser construído de outra maneira, agrupando-se as cadeiras e camas em nós separados, logo, ao invés de uma casa com quatro quartos, teríamos uma casa e, logo abaixo, dois nós agrupando cadeiras e camas (figura 54). Esse modelo está semanticamente relacionado com a casa, pois a mesma possui camas e cadeiras, porém os objetos geométricos não estão agrupados espacialmente. Esse resultado indesejado é contrário ao uso eficiente do grafo de cena, como veremos adiante.

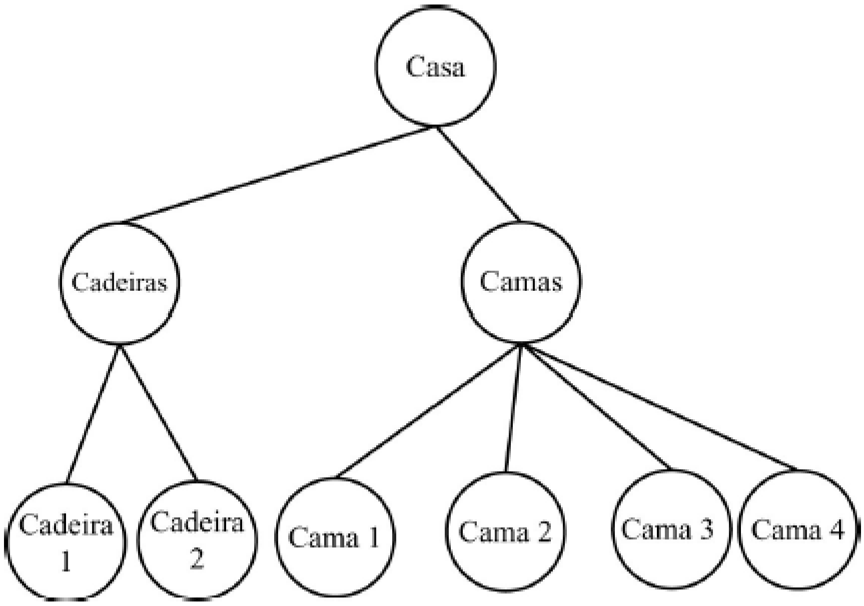


Figura 54: Estratégia de uso não eficiente de um grafo de cena

Propriedades: Os grafos de cena implementam um princípio chamado de **herança** de estado. Os nós internos armazenam o estado do sistema, onde estado significa a posição e a orientação dos objetos no ambiente virtual e seus atributos de aparência. A herança de estado é uma propriedade dos grafos de cena que determina que cada nó deve herdar as propriedades de estado de todos os seus ancestrais no grafo até a raiz.

Analisemos, novamente, o modelo da casa e admitamos que seus objetos estejam modelados em relação à origem do sistema de coordenada. Em cada quarto acrescentamos uma translação que irá posicionar seus quartos corretamente em relação à casa. A casa pode ser ainda rotacionada, por exemplo, a fim de ficar voltada para uma determinada direção. Essa cena é ilustrada na figura 55. Devido à herança de estado, todas as geometrias identificadas pelos nós raízes herdarão as propriedades dos seus ancestrais e serão posicionadas corretamente. A herança de estado é, portanto, uma ferramenta bastante útil para organização da cena 3D.

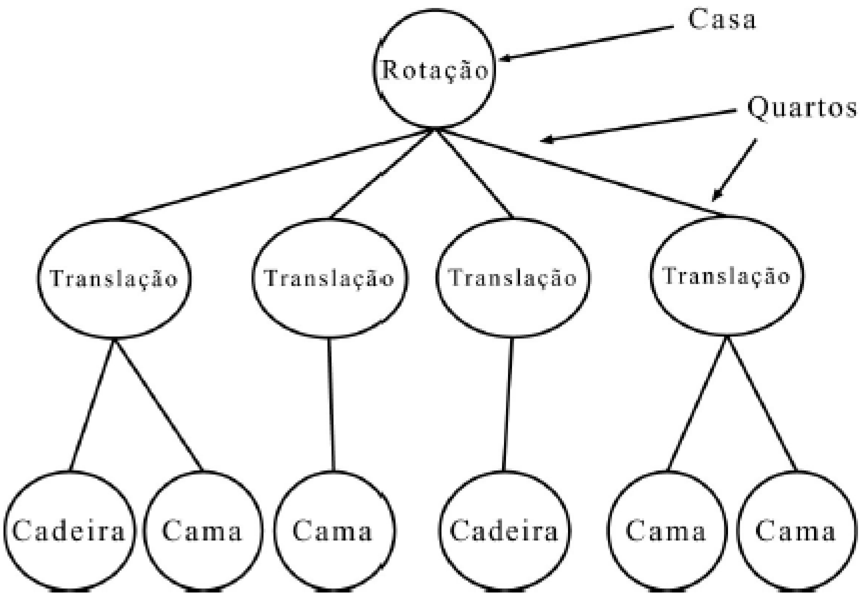


Figura 55: Grafo de cena usando transformações (métodos) para posicionar quartos e sala

Todos os nós de um grafo de cena podem possuir atributos como material, textura, transparência, entre outros. Todos esses atributos são herdados dos nós ancestrais. Um determinado nó pode sobrescrever um determinado atributo e, sendo assim, toda sua subárvore será modificada. A figura 56 ilustra um exemplo de um conjunto de objetos com um determinado material (cor) e alguns nós sobrescrevendo esse atributo.

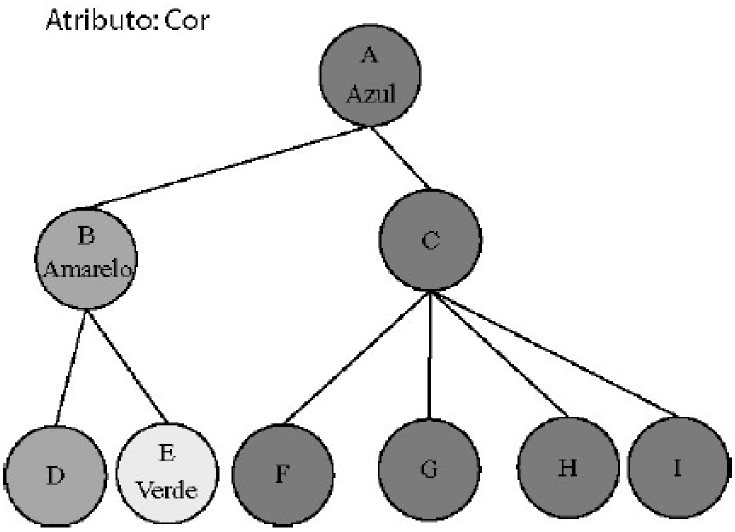


Figura 56: Materiais redefinidos ao longo do grafo. Os nós herdam a cor do ancestral.

1.2.67.ONORÍFICOS

São chamados de onoríficos, os dígrafos conexos, de até oito vértices, com exatamente sete arestas (contanto com os laços), que possuem um vértice inicial, chamado de raiz e caminho euleriano partindo dessa raiz. Na figura 57 temos um dígrafo onorífico, com raiz em v7.

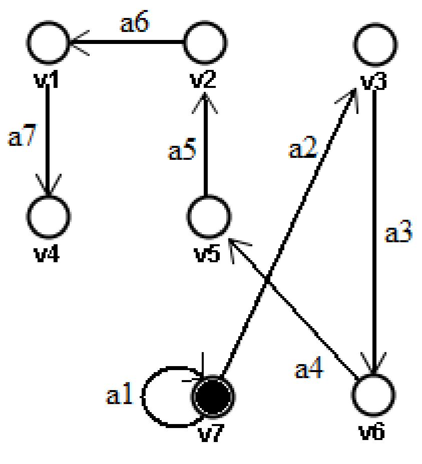


Figura 57: Grafo onorífico

1.2.68.ÁRVORE DE EXTENSÃO

Dado um grafo G, não orientado conexo, uma árvore de extensão deste grafo é um subgrafo o qual é uma árvore que conecta todos os vértices de G. Um único grafo pode ter diferentes árvores de extensão. Um exemplo de árvore de extensão é a árvore formada pelas arestas {a1, a2, a6, a7, a9, a10, a11} do grafo da figura 10.

1.2.69.CADEIA

Uma cadeia é uma seqüência qualquer de arestas adjacentes que ligam dois vértices, tornando-se um caminho entre estes dois vértices. Por exemplo, a seqüência {a4 – a5 – a6} compõe uma cadeia entre os vértices v6 e v1 no dígrafo da figura 57. É mais comum, no estudo das cadeias, utilizar-se de grafos ou dígrafos enraizados, sendo que as cadeias sempre iniciam na raiz.

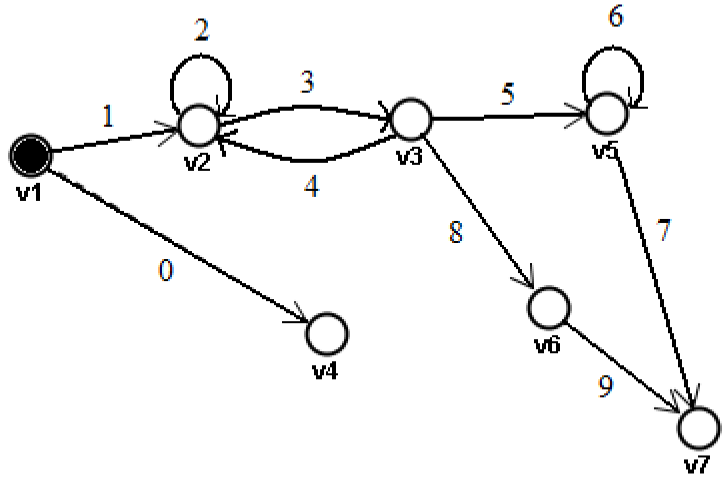


Figura 58: Dígrafo enraizado

➔Propriedades da cadeia:

Neste subitem, serão mostradas as propriedades utilizadas no estudo das cadeias. Para exemplificar as propriedades, vamos utilizar o dígrafo da figura 58, com raiz em v1 e valorado de 0 a 9, para facilitar a exemplificação.

- **Elementar**: é uma cadeia que não passar duas vezes pelo mesmo vértice.
  Exemplos de cadeias elementares: {1 3 5 7}, {0}, {1 3 8 9}, {1 3 5}
  Exemplos de cadeias não elementares: {1 2}, {1 3 4}, {1 3 5 6}, {1 3 4 3 5 7}

- **Simples**: é uma cadeia que não passa duas vezes pela mesma aresta.
  Exemplos de cadeias simples: {1 2 3 5 7}, {0}, {1 3 5 6 7}, {1 3 4}, {1 3 8 9}
  Exemplos de cadeias não simples: {1 2 2 3}, {1 2 3 4 3 5 6 7}, {1 2 3 4 3 5 6 6 7}

- **Tamanho ou Comprimento |c|**: é a quantidade de arestas que compõe a cadeia. Exemplos:

c = {1 2 3 5 7} → |c| = 5;	c = {0} → |c| = 1
c = {1 3 8 9} → |c| = 4;	c = {1 2 2 3} → |c| = 4;
c = {1 2 3 4 3 5 6 7} → |c| = 8;	c = {1 2 3 4 3 5 6 6 7} → |c| = 9;

- **Concatenação //**: é a unificação de cadeias. Só será permitido concatenar uma cadeia c1 com uma cadeia c2 se a última aresta da cadeia c1 for adjacente a primeira aresta da cadeia c2. Exemplo:
  c1 = {1 2 3 4 3}, c2 = {5 6 6 6 7}
  c1 // c2 = {1 2 3 4 3 5 6 6 6 7}

Atenção: a concatenação não é uma propriedade comutativa, ou seja: c1 // c2 ≠ c2 // c1.

- **Sucessor**: é o elemento seguinte a um elemento qualquer da cadeia.
  Exemplo:
  Na cadeia c = {1 2 3 5 6 6 7}, podemos dizer que 3 é sucessor de 2, mas não é sucessor de 1, pois 3 aparece logo após 2 e não logo após 1.

- **Antecessor**: é o elemento anterior a um elemento qualquer da cadeia.
  Exemplo:
  Na cadeia c = {1 2 3 5 6 6 7}, podemos dizer que 3 é antecessor de 5, mas não é antecessor de 6, pois 3 aparece logo antes de 5 e não de 6.

- **Reversa** $\overline{}$: é a cadeia escrita de trás para frente. Esta propriedade é comum apenas em grafos não orientados, pois, em dígrafos, a reversão da cadeia entre os vértices v e w, poderá não resultar em um caminho entre w e v. Para o exemplo, vamos considerar o grafo subjacente ao grafo da figura 58.
  Exemplo:
  c = {1 2 3 5 6 6 7} → $\overline{c}$ = {7 6 6 5 3 2 1}

  No exemplo, a cadeia c liga o vértice v1 ao vértice v7 e a cadeia $\overline{c}$, liga o vértice v7 ao vértice v1, no grafo subjacente ao dígrafo da figura 58.

- **Elemento Neutro** λ: é um elemento que concatenado com uma cadeia, resulta na própria cadeia, ou seja: c // λ = λ // c = c.
  Exemplo:
  c = {1 2 3 5 6 6 7} → c // λ = {1 2 3 5 6 6 7} // λ = {1 2 3 5 6 6 7}

- **Validação**: sejam dois vértices v, w de um grafo, onde v é a raiz e w um vértice marcado como validador (ou final). Uma cadeia é dita válida em um grafo, se for um caminho entre v e w.

  Exemplo (considerando v7 como vértice validador):

  c = {1 2 2 2 3 4 2 2 3 5 6 6 6 7} é uma cadeia válida entre os vértices v1 e v7.

  Neste exemplo, o vértice v1 é chamado de vértice **inicial** da cadeia e o vértice v7 é chamado de vértice **final** da cadeia.

- **Fecho**: o fecho é um ciclo, em um grafo, que permite cadeias de comprimento que tende ao infinito.
  Exemplo: no dígrafo da figura 58, temos os fechos: {2}, {3, 4} e {6}, permitindo a existência de cadeias, como:
  c = {1 2 2 2 ... 2 3 5 6 7} c = {1 3 4 3 4 3 4 3 4 ... 3 4 3 5 6 6 ... 6 6 7}

  Quando as arestas que compõe um fecho puderem ser totalmente excluídas da cadeia, de maneira que mantenha a validação da mesma, chamamos esse fecho de **fecho estrela** *. Caso contrário, temos um **fecho positivo** +. Os fechos {2} e {6} são fechos estrela e o fecho {3, 4} é positivo. Com isso, podemos representar as cadeias do exemplo, da seguinte maneira:
  c = {1 2* 3 5 6 7} e c = {1 $(3\ 4)^+$ 3 5 6* 7}

Lema 9: Todo laço é um fecho estrela, mas nem todo fecho estrela é um laço.

1.2.70.REPRESENTAÇÃO ALGORÍTMICA

A representação esquemática de grafos é necessária para a compreensão das explicações e por isso foram utilizadas deste o início deste escrito, porém, a representação esquemática não é adequada para fornecer a um computador dados sobre a estrutura de um grafo, por isso, pesquisadores desenvolveram formas de representação que atendem a necessidades algébricas ou combinatórias, dentro de estruturas de dados, para armazenamento e uso em algoritmos. A seguir, veremos as principais formas de se representar um grafo algoritmicamente.

Para exemplificar estas formas, vamos utilizar do grafo da figura 58. Considere **n** igual ao número de vértices do grafo e **m** o numero de arestas.

- **Lista de Adjacência ou dicionário**: composta por uma tabela com n linhas e 2 colunas. A primeira coluna lista a seqüência de vértices e a segunda, mostra os vértices adjacentes ao vértice correspondente na coluna 1. Em dígrafos é possível ter um segundo caso, onde a lista de adjacência possui, na primeira coluna, os vértices do grafo e na segunda, os vértices que atinge o vértice correspondente na coluna 1.

Vértices	
Origem	**Destino**
v1	v2 v4
v2	v2 v3
v3	v2 v5 v6
v4	
v5	v5 v7
v6	v7
v7	

Lista de Adjacência

Vértices	
Atingíveis	**Atingem**
v1	
v2	v1 v2 v3
v3	v2
v4	v1
v5	v3 v5
v6	v3
v7	v5 v6

Lista de Adjacência: Segundo Caso

- **Matriz de Adjacência:** composta por uma matriz n x n, onde as linhas representam o vértice de origem e as colunas o vértice de destino. Se existir uma aresta entre os vértices v, w, a linha v, coluna w terá um valor igual a 1 (ou o valor da aresta em casos de grafos valorados), caso contrário, terá um valor igual a 0.

	v1	v2	v3	v4	v5	v6	v7
v1	0	1	0	1	0	0	0
v2	0	1	1	0	0	0	0
v3	0	1	0	0	1	1	0
v4	0	0	0	0	0	0	0
v5	0	0	0	0	1	0	1
v6	0	0	0	0	0	0	1
v7	0	0	0	0	0	0	0

Matriz de Adjacência

- **Matriz de Incidência**: composta por uma matriz n x m, onde cada linha representa um vértice e cada coluna, uma aresta. Caso exista uma aresta **A** saindo do vértice V, a matriz terá o valor de +1 (ou o valor positivo, no caso de grafos valorados) na linha V, coluna A. Caso a aresta A esteja chegando em v, a matriz terá o valor de – 1 na linha V, coluna A.

	0	1	2	3	4	5	6	7	8	9
v1	+1	+1								
v2			+1/ – 1	+1	– 1					
v3				– 1	+1	+1			+1	
v4	– 1									
v5						– 1	+1 / – 1	+1		
v6									– 1	+1
v7								– 1		– 1

Matriz de Incidência

- **Matriz Figurativa**: composta por uma matriz n x n, onde se existir uma aresta entre os vértices v, w, a linha v, coluna w terá uma informação, que depende da finalidade do grafo. Vejamos alguns exemplos:

	v1	v2	v3	v4	v5	v6	v7
v1		v1v2		v1v4			
v2		v2v2	v2v3				
v3		v3v2			v3v5	v3v6	
v4							
v5					v5v5		v5 v7
v6							v6 v7
v7							

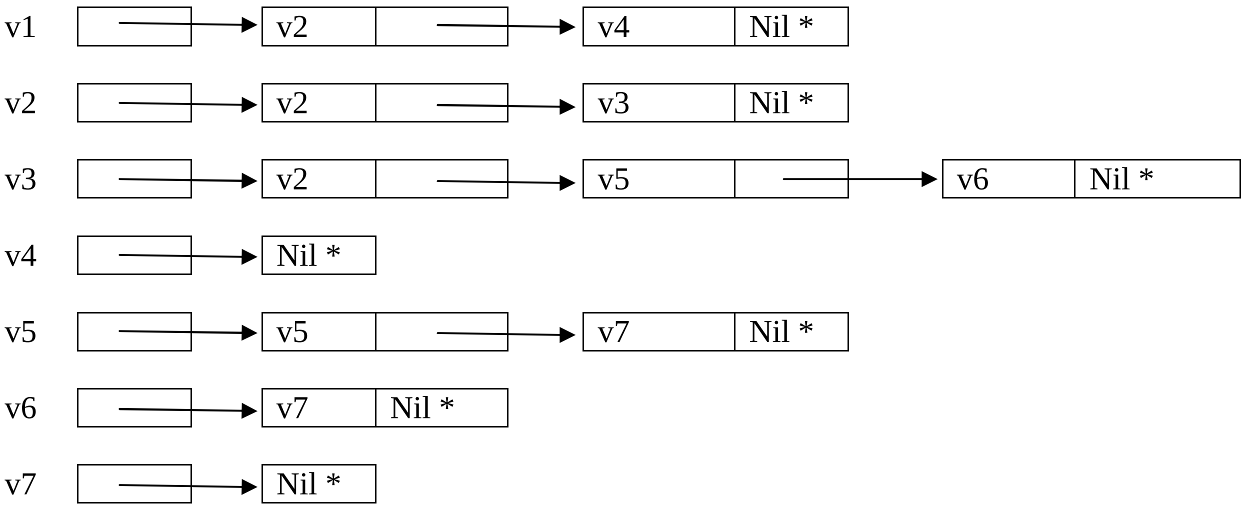
Matriz Figurativa por vértices

	v1	v2	v3	v4	v5	v6	v7
v1		1		0			
v2		2	3				
v3		4			5	8	
v4							
v5					6		7
v6							9
v7							

Matriz figurativa por arestas

Uma outra possibilidade para a matriz figurativa pode ser vista na figura 52. Nela, utilizam-se as cores pretas e brancas para marcar a presença ou não de arestas respectivamente.

- **Estrutura de Adjacência**: composta por um vetor, cujos índices são os vértices do grafo. A informação contida no vetor é um ponteiro para um registro que guarda uma única informação, indicando uma aresta adjacente e um ponteiro para o próximo registro, formando assim uma lista encadeada de adjacências, onde os vértices sucessores estão presentes na lista indicada pelo índice do vetor correspondente ao vértice de origem.



Estrutura de Adjacência

- **Matriz Laplaciana**: para formar a matriz laplaciana, utiliza-se a matriz de adjacência. Com a matriz de adjacência pronta, basta inverter o sinal dos valores e substituir a diagonal principal pelo grau de saída de cada vértice indicado na linha.

	v1	v2	v3	v4	v5	v6	v7
v1	2	-1	0	-1	0	0	0
v2	0	2	-1	0	0	0	0
v3	0	-1	3	0	-1	-1	0
v4	0	0	0	0	0	0	0
v5	0	0	0	0	2	0	-1
v6	0	0	0	0	0	1	-1
v7	0	0	0	0	0	0	0

Matriz Laplaciana

Obs.: Existem outras diversas formas de se representar um grafo. De acordo com a necessidade de cada um, vários pesquisadores criam suas próprias representações. As representações, mostradas aqui, são as mais utilizadas pela literatura corrente sobre grafos.

1.2.71. AUTOMORFISMO

Um automorfismo de um grafo G é um isomorfismo de G para si próprio, ou seja, os automorfismos de G são as permutações que podem ser aplicadas a ambas as linhas e colunas da matriz de adjacência sem mudar a adjacência entre os vértices de G. Isto significa que, na matriz de adjacência, os vértices podem ter sua ordem alterada, sem nenhuma alteração nos valores apresentados pela matriz. Na figura 59 temos um grafo e duas matrizes de adjacências desse grafo. Note que os valores presentes nas duas matrizes são exatamente iguais, porém, a seqüência dos vértices é diferente.

	v1	v2	v3	v4
v1	0	1	0	0
v2	1	1	1	0
v3	0	1	1	1
v4	0	0	1	0

	v4	v3	v2	v1
v4	0	1	0	0
v3	1	1	1	0
v2	0	1	1	1
v1	0	0	1	0

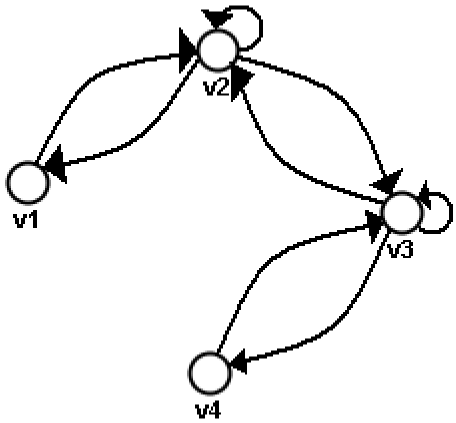


Figura 59: Grafo automórfico

1.2.72.PROBLEMAS EM GRAFOS

Nesta seção, veremos alguns dos problemas que circundam a Teoria dos Grafos. A respeito da classificação da complexidade, veremos em tópicos posteriores.

1 – Problema do SCIE. Consiste em encontrar o subconjunto internamente estável maximal de um grafo G. É um problema NP-Completo.

2 – Problema do Isomorfismo em Grafos. Consiste em saber se dois grafos são ou não isomórficos entre si.

3 – Problema da Coloração em Grafos. Consiste em descobrir qual é a colocação mínima de um grafo G.

4 – Problema do Clique. Consiste em, dado um grafo G e uma constante natural positiva k, decidir se existe um clique de tamanho ≥ k em G.

5 – Problema da Satisfabilidade (SAT). É um problema de lógica que envolve expressões booleanas. Com uma expressão booleana na sua forma normal conjuntiva (FNC), decidir se a expressão é satisfatível, ou seja, verificar se existe uma atribuição de valores às variáveis da expressão de tal modo que a expressão seja avaliada verdadeira.

6 – Problema da Cobertura. Consiste em, dado um grafo G e uma constante natural positiva k, decidir se existe uma cobertura de tamanho ≤ k em G.

7 – Problema do Ciclo Hamiltoniano. Saber se um grafo G possui ou não um ciclo hamiltoniano.

8 – Problema da Parada. Consiste em se decidir, para qualquer algoritmo A e qualquer entrada de A, se A vai terminar ou entrar em loop infinito.

9 – Problema da Árvore de Extensão Mínima. Em um grafo valorado, consiste em descobrir qual é a árvore de extensão com menor valor no grafo.

10 – Problema do Caminho Mínimo. Em um grafo valorado G, consiste em descobrir o menor caminho entre dois vértices qualquer de G.

11 – Problema do Carteiro Chinês. Um carteiro que deve percorrer um roteiro todo dia. O problema é de identificar esse roteiro de maneira a minimizar a distância total percorrida. Essa situação pode ser representada por um grafo onde as arestas correspondem às ruas e os vértices correspondem aos cruzamentos.

12 – Problema do Caixeiro Viajante. O problema do caixeiro-viajante (PCV/TSP – Travelling Salesman Problem) é um problema de otimização que, apesar de parecer modesto é, na realidade, um dos mais investigados, por cientistas, matemáticos e investigadores de diversas áreas, tais como: logística, genética e produção, entre outros. Consiste na procura de um ciclo ou circuito que possua a menor distância, começando numa cidade (vértice) qualquer, entre várias, visitando cada cidade precisamente uma vez e regressando à cidade inicial. É um problema NP-Difícil.

13 – Problema do Passeio do Cavalo do Xadrez. O Cavalo é a única peça do xadrez que pode saltar sobre outras peças. Ele tem um movimento bem peculiar em formato de "L": duas casa no sentido vertical ou horizontal e uma casa no outro sentido. Seguindo as regras de movimentação do cavalo, é possível que um cavalo parta de uma casa qualquer, do tabuleiro de xadrez, percorra todo o tabuleiro visitando cada casa uma e somente uma única vez e retorne à casa inicial?

14 – Problema das Cinco Damas. A dama é a peça mais poderosa do jogo de Xadrez. Numa jogada ela pode mover-se tantas casas quantas quiser em qualquer direção vertical, horizontal ou diagonal, desde que não haja nenhuma outra peça que obstrua sua passagem. Todas as casas que a dama alcança estão sob seu domínio. Qualquer outra peça que estiver numa casa sob domínio da dama estará sob seu ataque. Este problema consiste em procurar as posições a serem ocupadas por damas em um tabuleiro de xadrez vazio, de modo que elas possam controlar todas as casas do tabuleiro com o menor número de damas possível.

15 – Problema das Pontes de Königsberg. Este problema é um famoso problema histórico da matemática que foi uma das principais fundações da teoria dos grafos. O problema é baseado na cidade de Königsberg (fundada em 1255, pelos Cavaleiros Teutónicos. Foi território da Prússia até 1945, atualmente é chamada de Kaliningrado e pertence à Rússia), que é cortada pelo Rio Pregolia, onde há duas grandes ilhas que, juntas, formam um complexo que na época da formulação do problema, continha sete pontes [Kraemer (lojista), Schmiede (ferreiro),

Holz (madeira), Honig (mel), Greune (verde), Koettel (entranha) e Hohe (alto)], conforme mostra a figura 60.c. Das sete pontes originais, uma foi demolida e reconstruída em 1935, duas foram destruídas durante a Segunda Guerra Mundial e outras duas foram demolidas para dar lugar a uma única via expressa. Atualmente apenas duas pontes são da época de Leonard Euler, o matemático que demonstrou o problema.

Discutia-se nas ruas da cidade a possibilidade de atravessar todas as pontes sem repetir nenhuma. Havia-se tornado uma lenda popular a possibilidade da façanha quando Euler, em 1736, provou que não existia caminho que possibilitasse tais restrições.

Euler usou um raciocínio muito simples. Transformou os caminhos em retas e suas intersecções em pontos, conforme a figura 60.d, criando possivelmente o primeiro grafo da história. Então percebeu que só seria possível atravessar o caminho inteiro passando uma única vez em cada ponte se houvesse no máximo dois pontos de onde saísse um número ímpar de caminhos. A razão de tal coisa é que de cada ponto deve haver um número par de caminhos, pois será preciso um caminho para "entrar" e outro para "sair". Os dois pontos com caminhos ímpares referem-se ao início e ao final do percurso, pois estes não precisam de um para entrar e um para sair, respectivamente.

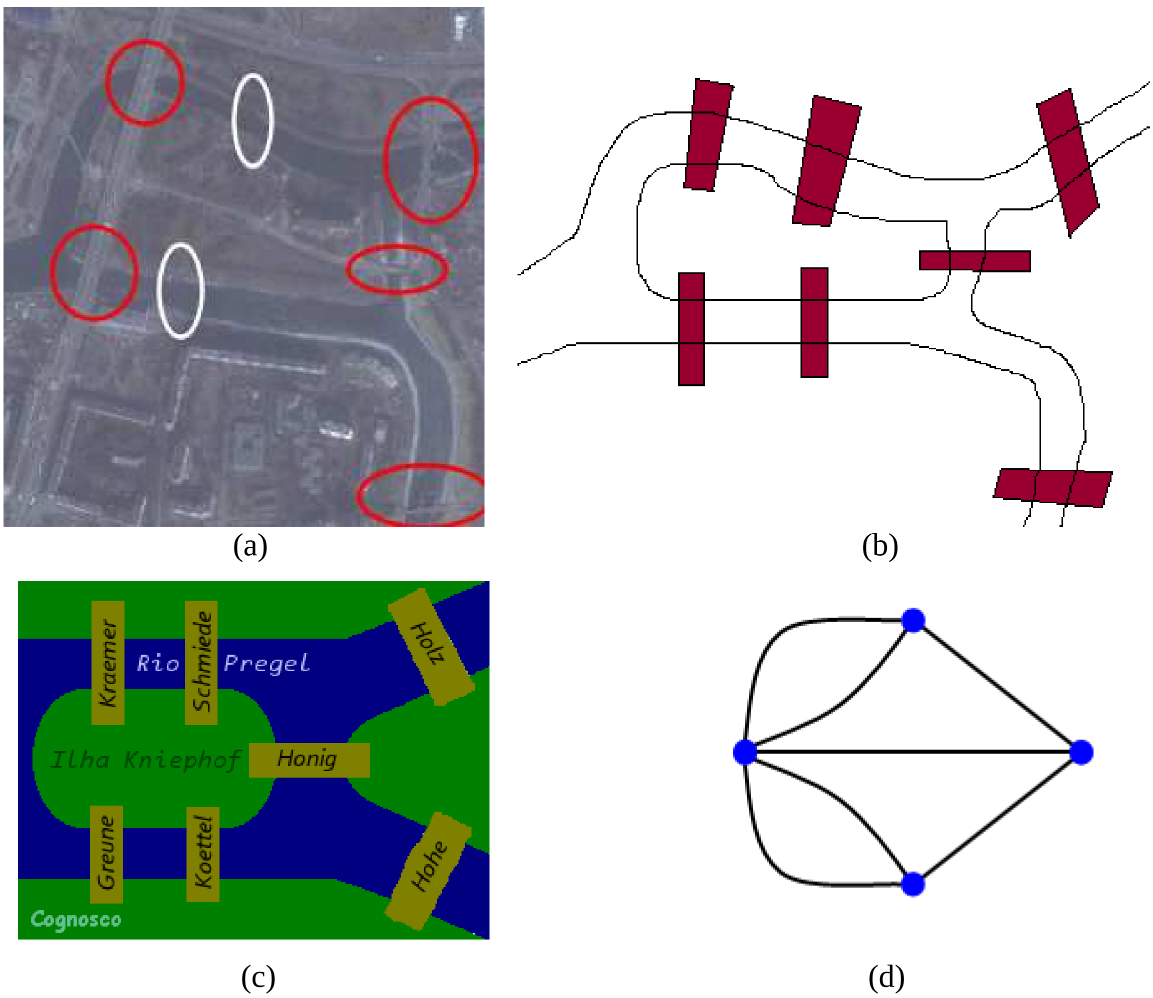


Figura 60: Pontes de Königsberg

16 – Problema do Desenho da Casa. No desenho da figura 61, uma criança diz ter posto a ponta do lápis numa das bolinhas e com movimentos contínuos (sem levantar o lápis e sem sobrepor linhas) traçou as linhas que formam o desenho da casa, traçando cada linha uma única vez. Isso é possível?

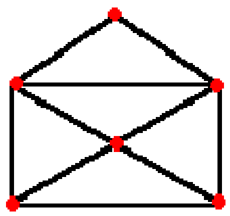


Figura 61: Problema do Desenho da Casa

17 – Problema das Oito Damas. A dama é a peça mais poderosa do jogo de Xadrez. Numa jogada ela pode mover-se tantas casas quantas quiser em qualquer direção vertical, horizontal ou diagonal, desde que não haja nenhuma outra peça que obstrua sua passagem. Todas as casas que a dama alcança estão sob seu domínio. Qualquer outra peça que estiver numa casa sob domínio da dama estará sob seu ataque. Este problema consiste em descobrir quantas maneiras há de se colocar 8 damas em um tabuleiro de xadrez de forma que nenhuma dama ameace a outra.

18 – Problema dos Conspiradores Políticos. Os agentes A, B, C, D, E, F, G e H são conspiradores políticos. De forma a coordenar seus esforços, é vital que cada agente seja capaz de comunicar-se direta ou indiretamente com todos os outros conspiradores. Esta comunicação, contudo, envolve um certo risco para cada um. Os fatores de risco associados à comunicação direta entre cada par de conspiradores é dado por:

A	A	A	A	A	B	B	C	C	C	C	D	D	E
B	C	E	F	G	C	F	D	F	G	H	E	H	H
9	3	8	3	4	10	6	6	4	5	7	6	3	5

Todas as outras comunicações diretas são impraticáveis, pois exporiam todo o esquema de disfarce. Qual é o menor risco total envolvido neste sistema de conexão, ou seja, o menor risco para que uma mensagem seja repassada para todos os conspiradores?

19 – Problema do Metrô. Considere a rede de metrô de uma cidade como Paris, na França. Esta rede cobre boa parte da cidade, sendo composta por várias linhas que se cruzam em estações específicas. Nestes pontos de cruzamento um usuário pode livremente sair de uma composição e passar para uma composição de outra linha. Sendo assim, em geral o usuário tem mais de uma opção de rota quando deseja deslocar-se de uma parte a outra da cidade. Escolher a melhor rota passa, então, a ser fundamental para que o deslocamento seja o mais rápido possível.

As distâncias entre estações vizinhas do metrô não são iguais e, conseqüentemente, o tempo de deslocamento entre estações vizinhas não é constante. Estas diferenças de tempo não são, contudo, significativas quando comparadas com os tempos de parada das composições nas estações e de troca de composição (de linha).

Em redes de metrô de várias cidades do mundo há sistemas computacionais, que utilizam grafos, para auxiliar os usuários a escolher a melhor rota para um deslocamento particular. Mais do que simplesmente indicar uma rota possível, este problema consistem em identificar aquela rota que conduz o usuário o mais rapidamente ao seu destino. Dois critérios básicos para esta escolha são (em ordem de importância): procurar minimizar o número de trocas de composição (de linhas) e procurar minimizar o número de paradas em estações.

20 – Problema do Organização dos Horários. Em uma universidade, temos vários cursos que funcionam de segunda a sexta, nos turnos de manhã, tarde e noite. Alguns cursos possuem quatro anos de duração, outros cinco anos. Os professores, que dão aulas em vários cursos, também trabalham em outras instituições, porém, possuem uma disponibilidade de horário suficiente para atender suas disciplinas com folga naquela universidade. Os cursos são divididos em semestres, e, para cada semestre, os cursos têm cinco disciplinas específicas para disponibilizar. O problema consiste em montar o horário escolar para atender todos os cursos, de maneira que não haja choque de horário em disciplinas de um mesmo semestre e turno, e que atenda a disponibilidade dos professores, sem que haja choque de horário entre as disciplinas que irá ministrar.

21 – Problema dos Canibais e dos Missionários. Três canibais e três missionários estão viajando juntos e chegam a um rio. Eles desejam atravessar o rio, sendo que o único meio de transporte disponível é um barco que comporta no máximo duas pessoas. Há uma outra dificuldade: em nenhum momento, o número de canibais pode ser superior ao número de missionários, pois desta forma, os missionários estariam em grande perigo de vida. O problema consiste em administrar a travessia do rio, de maneira que todos atravessem sem deixar algum missionário correndo risco de vida.

22 – Problema dos Três Maridos Ciumentos. Três esposas e seus respectivos maridos desejam ir ao centro da cidade em um Corvette, o qual comporta apenas duas pessoas. Como eles poderiam deslocar-se até o centro considerando que nenhuma esposa deveria estar com um ou ambos os outros maridos a menos que seu marido também esteja presente?

23 – Problema da Telefonia Celular. A empresa de telefonia celular Tabajara pretende se instalar em Petrolina. A empresa considera importante que todo o perímetro a cidade esteja coberta por sinal de telefonia celular, de forma que em qualquer ponto do território municipal (urbano, rural, projetos e distritos) um usuário possa obter sinal e falar normalmente. Isto significa distribuir diversas antenas de telefonia celular, porém, dado que o custo de instalação de uma antena é relativamente alto, a empresa solicita ao departamento técnico um estudo para identificar o menor número de antenas (com as correspondentes localizações aproximadas) necessárias para realizar esta cobertura.

24 – Problema dos Potes de Vinho. Considere que temos três potes com capacidades de 8, 5 e 3 litros, respectivamente, os quais não possuem qualquer marcação. O maior deles está completamente cheio enquanto que os outros dois estão vazios. O problema consiste em dividir o vinho em duas porções iguais de 4 litros, tarefa esta que pode ser realizada por transvasos sucessivos de um vaso no outro. Qual o menor número de transvasos necessários para completar a divisão?

Outros problemas sobre a Teoria dos Grafos podem ser vistos em nossa bibliografia ou no site http://www.inf.ufsc.br/grafos/problemas/index.htm.

1.3. COMPLEXIDADE ALGORÍTMICA

A teoria da complexidade computacional é a parte da teoria da computação que estuda os recursos necessários durante o cálculo, para resolver um problema, ou seja, a complexidade de um algoritmo é a quantidade de "trabalho" necessária para a sua execução, expressa em função das operações fundamentais, as quais variam de acordo com o algoritmo, e em função do volume de dados.

Os recursos comumente estudados são:

- Tempo, que pode ser o tempo real, em segundos, para execução de um algoritmo ou o número de passos de execução realizados por um algoritmo para resolver um problema. Esse estudo é chamado de **complexidade temporal**;
- Espaço, que mede a quantidade de memória utilizada para resolver um problema. Este estudo é chamado de **complexidade espacial**;

Podem-se estudar igualmente outros parâmetros, tais como o número de processadores necessários para resolver o problema em paralelo. A teoria da complexidade difere da teoria da computabilidade a qual se ocupa simplesmente em solucionar um problema, com algoritmos efetivos, sem levar em conta os recursos necessários para ele ser executado.

Os problemas que têm uma solução com ordem de complexidade linear são os problemas que se resolvem em um tempo que se relaciona linearmente com seu tamanho.

A importância da complexidade pode ser observada no exemplo da figura 62, que mostra cinco algoritmos A1 a A5 para resolver um mesmo problema, de complexidades diferentes. Supomos que uma operação leva 1 milisegundo para ser efetuada, a figura 62 mostra o tempo necessário para cada um desses cinco algoritmos serem executados, onde Tk(n) é a complexidade do algoritmo.

	A1	A2	A3	A4	A5
n	T1(n) = n	T2(n) = nlogn	T3(n) = n^2	T4(n) = n^3	T5(n) = 2^n
16	0,016s	0,064s	0,256s	4s	1m4s
32	0,032s	0,16s	1s	33s	46 dias
512	0,512s	9s	4m22s	1 dia 13h	10^{137} séculos

Figura 62: Tabela de complexidade algorítmica

A complexidade do tempo de um problema é o número de passos que se toma para resolver uma instância de um problema, a partir do tamanho da entrada utilizando o algoritmo mais eficiente à disposição. Intuitivamente, caso se tome uma instância com entrada de longitude n que pode resolver-se em n^2 passos, se diz que esse problema tem uma complexidade em tempo de n^2. Supostamente, o número exato de passos depende da máquina em que se programa, da linguagem utilizada e de outros fatores. Para não ter que falar do custo exato de um cálculo se utiliza a **notação assintótica**. Quando um problema tem custo dado em tempo $O(n^2)$ em uma configuração de computador e linguagem, este custo será o mesmo em todos os computadores, de maneira que esta notação generaliza a noção de custo independentemente do equipamento utilizado.

São usadas três perspectivas no estudo do caso da complexidade. São elas:

Melhor Caso: Representado por Ω( ), consiste em assumir que vai acontecer o melhor. Pouco utilizado e de baixa aplicabilidade.

Exemplo 1: Em uma lista telefônica queremos encontrar um número, assume-se que a complexidade do caso melhor é Ω (1), pois está pressupondo-se o número desejado está no topo da lista.

Exemplo 2: Extrair qualquer elemento de um vetor. A indexação em um vetor ou array, leva o mesmo tempo seja qual for o índice que se queira buscar. Portanto é uma operação de complexidade constante Ω (1).

Caso Médio: Representado por θ( ). Este é o caso que é o mais difícil de ser determinado, pois, necessita de análise estatística e em conseqüência de muitos testes, contudo é muito utilizado, pois é o que representa mais corretamente a complexidade do algoritmo.

Exemplo: Procurar uma palavra em um dicionário. Pode-se iniciar a busca de uma palavra na metade do dicionário. Imediatamente se sabe se foi encontrada a palavra ou, no caso contrário, em qual das duas metades deve se repetir o processo (é um processo recursivo) até se chegar ao resultado. Em cada busca (ou sub-busca), o problema (as páginas em que a palavra pode estar) vai se reduzindo à metade, o que corresponde com a função logarítmica. Este procedimento de busca (conhecido como busca binária) em uma estrutura ordenada têm complexidade logarítmica $\theta(\log_2 n)$.

Pior Caso: Representado normalmente por O( ). Se dissermos que um determinado algoritmo é representado por g(x) e a sua complexidade Caso Pior é n, será representada por g(x) = O(n). Consiste em assumir que o pior dos casos pode acontecer, sendo de grande utilização e também normalmente o mais fácil de ser determinado.

Exemplo 1: Será tomado como exemplo o jogo de azar com 3 copos, deve descobrir-se qual deles possui uma moeda debaixo dele, a complexidade caso pior será O(3) pois no pior dos casos a moeda será encontrada debaixo do terceiro copo, ou seja, será encontrada apenas na terceira tentativa.

Exemplo 2: O processo mais comum para ordenar um conjunto de elementos têm complexidade quadrática. O procedimento consiste em criar uma coleção vazia de elementos. A ela se acrescenta, em ordem, o menor elemento do conjunto original que ainda não tenha sido eleito, o que implica percorrer completamente o conjunto original (O(n), sendo n o número de elementos do conjunto). Esta percorrida sobre o conjunto original se realiza até que se tenha todos seus elementos na seqüência de resultado. Pode-se ver que há de se fazer n seleções (se ordena todo o conjunto) cada uma com um custo n de execução: o procedimento é de ordem quadrática $O(n^2)$. Deve esclarecer que existem diversos algoritmos de ordenação com melhores resultados.

1.3.1. NOTAÇÃO ASSINTÓTICA

Também chamada de notação Grande-O, a notação assintótica é uma notação matemática utilizada para analisar o comportamento assintótico[1] de funções. Essa notação é bastante utilizada para a análise de algoritmos em ciência da computação.

A criação dessa notação é creditada ao matemático alemão Paul Bachmann, em 1894, na publicação da segunda versão da sua obra Analytische Zahlentheorie, sendo popularizada por outro alemão, Edmund Landau, e por causa disso, essa notação é também conhecida como "Símbolo de Landau". Essa notação é padronizada como "ordem de".

De um modo informal, dizemos que uma função f pode ser representada como O(g(x)) se ambas funções, para x suficientemente grande, "crescem" da mesma forma, ou seja, são "proporcionais". Se f é O(g(x)), então, para x grande o suficiente, se num dado intervalo f dobra, então g "dobra" também (com pequena margem de erro), e vice-versa. Por exemplo, $f(x) = 10x^5 + 3x^3$ é $O(x^5)$.

➔Definição

Sejam duas funções reais f(x) e g(x), a notação é definida da seguinte forma:

$$O(g(x)) = \left\{ \begin{matrix} f(x) : \text{existem } c, x_0 \text{ constantes positivas tais que} \\ \forall x : x_0 \leq x : 0 \leq f(x) \leq cg(x) \end{matrix} \right\}$$

A definição acima diz que a função f(x) é superiormente limitada pela função g(x). Na definição foi utilizado f(x) = O(g(x)), mas é mais correto escrever f(x) é O(g(x)), ou f(x) ∈ O(g(x)), já que O(g(x)) define um conjunto de funções (ou classes de funções).

Para demonstrar que f(x) é O(g(x)), não basta encontrar dois números (c e x0) tal que a definição acima seja satisfeita, já que pode existir funções que satisfaz só para alguns valores. As provas usualmente são feitas com provas por indução. Vejamos alguns exemplos:

A função f(x) = 10x + 11 é O(x). Como exemplo, basta utilizar c = 11 e x0 = 10.
A função $f(x) = x^2 + 100$ é $O(x^2)$. Como exemplo, basta utilizar c = 2 e x0 = 10.

➔ Propriedades

- **Soma:** $O(f(n)) + O(g(n)) = O(\max\{f(n), g(n)\})$
- **Produto:** $O(f(n))O(g(n)) = O(f(n)g(n))$
- **Multiplicação por uma constante:** $O(kg(n)) = O(g(n))$, desde que k≠0

[1] Chamamos de comportamento assintótico o comportamento a ser observado em uma função f(n), quando n tende ao infinito.

1.3.2. CLASSES DE ALGORITMOS

Vimos o conceito de complexidade e a notação mais utilizada para medir essa complexidade, mas como saber qual é a complexidade de um determinado algoritmo implementado?

Para resolver esse problema, dividiu-se os algoritmos em “Classes de Problemas”, de acordo com o parâmetro que afeta o algoritmo de forma mais significativa

Abaixo há uma lista de classes de funções que são bastante utilizadas para análise de algoritmos, por ordem crescente de crescimento de funções (as funções que têm um crescimento mais lento são as primeiras). A letra c denota uma constante qualquer.

Notação	**Nome**	**Notação**	**Nome**
$O(1)$	Ordem constante	$O(x^3)$	Ordem cúbica
$O(\log x)$	Ordem logarítmica	$O(x^c)$	Ordem polinomial
$O([\log x]^c)$	Ordem poli-logarítmica	$O(c^x)$	Ordem exponencial
$O(x)$	Ordem linear	$O(x!)$	Ordem fatorial
$O(x \cdot \log x)$	Ordem linear-logarítmica	$O(x^x)$	Ordem exponencial
$O(x^2)$	Ordem quadrática		

Vejamos agora, o detalhamento das principais classes de algoritmos, conforme sua complexidade.

- **Complexidade Constante**
  São os algoritmos de complexidade O(1). É a única classe em que as instruções dos algoritmos são executadas num tamanho fixo de vezes, independe do tamanho N de entradas. Vejamos o exemplo a seguir:

```
// procedimento para esvaziar uma lista de dados
Function Vazia(Lista: TipoLista): Boolean;
Begin
        Vazia := Lista.Primeiro = Lista.Ultimo;
End;
```

- **Complexidade Linear**
  São os algoritmos de complexidade O(N). Uma operação é realizada para cada um dos elementos de entrada.

```
//pesquisa de elementos em uma lista
Procedure Busca(Lista: TipoLista; x: TipoElem; Var pos: integer)
Var i: integer;
Begin
        i:=1;
        while Lista.Elemento[i] <> x do
                i := i+1;
        if i >= Lista.MaxTam then
                pos := -1
        else
                pos := i;
End;
```

- **Complexidade Logarítmica**
  São os algoritmos de complexidade O(logN). Ocorre tipicamente em algoritmos que dividem o problema em problemas menores, como por exemplo, o algoritmo de Busca Binária.

```
Function buscaBinaria (x,n: integer; v: TipoVetorDeInteiros): Integer;
Var e, m, d: integer;
Begin
        e := 0;
        d := n-1;
        while (e <= d) do
        begin
            m := (e + d)/2;

            if (v[m] = x) then
                buscaBinaria := m;

            if (v[m] < x) then
                e := m + 1
            else
                d := m - 1;
        end;
        buscaBinaria := -1;
End;
```

- **Complexidade Linear-Logarítmica**
  São os algoritmos de complexidade O(NlogN). Ocorre tipicamente em algoritmos que dividem o problema em problemas menores, porém juntando posteriormente a solução dos problemas menores. A maioria dos algoritmos de ordenação externa é de complexidade logarítmica ou NLog N. O QuickSort tem complexidade NLog N.

- **Complexidade Quadrática**
  São os algoritmos de complexidade O(N^2). Ocorre quando itens são processados aos pares, geralmente com um *loop* dentro do outro. Vejamos o exemplo:

```
Procedure SomaMatriz(Mat1, Mat2, MatRes: Matriz);
Var i, j: integer;
Begin
        for i:=1 to n do
                for j:=1 to n do
                        MatRes[i,j] := Mat1[i, j] + Mat2[i,j];
End;
```

- **Complexidade Cúbica**
  São os algoritmos de complexidade $O(N^3)$. Ocorre quando itens são processados três a três, geralmente com um *loop* dentro do outros dois. Vejamos um exemplo:

```
Procedure SomaElementos_Vetor_Indices_Matriz (mat: Matriz, vet: Vetor);
Var i, j: integer;
Begin
        for i:=1 to n do
                for j:=1 to n do
                        for k:=1 to n do
                                mat[i, j] := mat[i, j] + vet[k];
End;
```

- **Complexidade Exponencial**
  São os algoritmos de complexidade $O(2^N)$. Utiliza a "força bruta" para resolver os problemas. É uma abordagem simples para resolver um determinado problema, geralmente baseada diretamente no enunciado do problema e nas definições dos conceitos envolvidos. Na grande maioria dos casos não são úteis sob o ponto de vista prático. Um exemplo seria um programa para tentar quebrar uma senha, aonde o algoritmo iria simplesmente testar todas as combinações possíveis de caracteres, até descobrir a senha correta.

Até agora, conhecemos as classes de algoritmos conforme sua complexidade. Mas a dúvida persiste: como saber qual é a complexidade de um determinado algoritmo implementado?
Para calcular a complexidade de um algoritmo, deve-se analisar o pior caso. A análise deve ser feita no programa todo, de acordo com os dados da figura 63.

$$
\begin{aligned}
f(n) &= O(f(n)) \\
c \times O(f(n)) &= O(f(n)) \qquad c = constante \\
O(f(n)) + O(f(n)) &= O(f(n)) \\
O(O(f(n))) &= O(f(n)) \\
O(f(n)) + O(g(n)) &= O(max(f(n), g(n))) \\
O(f(n))O(g(n)) &= O(f(n)g(n)) \\
f(n)O(g(n)) &= O(f(n)g(n))
\end{aligned}
$$

Figura 63: Tabuada da Complexidade

Exemplo 1:

```
Procedure Verifica_Item_Lista
                (Lista: TipoLista; x: TipoItem; pos: integer);
Var i: integer;
Begin
   i:=1;                                              O(1)
   achou := false;
   while (i <= Lista.Tamanho) and not achou do begin     O(N)
        inc(i);
        if Lista.Item[i] = x then
                achou := true;
   end;
   if achou then                                       O(1)
        pos := i
   else
        pos := -1;
```

$$f(N) = O(9 * O(1) + O(N)) = O(O(1) + (O(N)) = O(N)$$

Exemplo 2:

```
Procedure Verifica_Item(Lista: TipoLista; x: TipoItem; pos: integer);
Var i: integer;
Begin
   i:=1;                                              O(1)
   achou := false;
   while (i <= Lista.Tamanho) and not achou do      O(N)
        if Lista.Item[i] = x then
                achou := true;
   if achou then                                       O(1)
        pos := i
   else
        for i:= Lista.Tamanho +1 to MaxTam;           O(N)
                Lista.Item[i] := x;                    O(1)
```

$$f(N) = O(7 * O(1) + 2*O(N)) = O(O(1) + (O(N)) = O(N)$$

1.3.3. CLASSES DE PROBLEMAS

As classes de problemas os dividem de acordo com sua complexidade computacional. Antes de conhecermos as classes, precisamos conhecer os tipos de problemas. São eles:

- **Problema de Decisão**: Consiste na verificação (decisão) da veracidade ou não de determinada questão para o problema, ou seja, a solução desse tipo de problema é SIM ou NÃO;

- **Problema de Localização:** Consiste na verificação da existência e identificação (localização) de uma solução (segundo algum critério) para o problema.

- **Problema de Otimização**: Consiste na verificação da existência e identificação da melhor (otimização) solução possível, dentro as soluções factíveis para o problema.

Hoje em dia, as máquinas resolvem com eficiência problemas mediante algoritmos que têm como máximo uma complexidade ou custo computacional polinômico, significa dizer que, a relação entre o tamanho do problema e seu tempo de execução é polinômico. Os problemas com custo fatorial ou combinatório estão agrupados em NP. Estes problemas não têm uma solução prática, significa dizer que uma máquina não pode resolver-los em um tempo razoável.

As principais classes de problemas são:

1.3.3.1. CLASSE P

Dizemos que um algoritmo resolve um dado problema (computável) se, ao receber qualquer instância do problema, devolve uma solução da instância ou diz que a instância não tem solução.

Um algoritmo que resolve um dado problema é classificado como polinomial se o seu consumo de tempo é limitado por uma função polinomial dos tamanhos das instâncias do problema.

É polinomial, por exemplo, todo algoritmo que consome no máximo $100N^4 + 300\,N^2 + 5000$ unidades de tempo, sendo N o tamanho da instância. Também é polinomial todo algoritmo que consome no máximo $200\,N^9 \log N$ unidades de tempo, por exemplo.

Algoritmos polinomiais são considerados rápidos (ainda que seja difícil aceitar como rápido um algoritmo que consome tempo proporcional a N^{500}, por exemplo).

Algoritmos não-polinomiais como, por exemplo, os que consomem tempo proporcional a 2^N, são considerados inaceitavelmente lentos, ainda que possam ser úteis para valores muito modestos de N.

Um problema computacional é polinomial se existe um algoritmo polinomial para o problema. Problemas desse tipo são considerados tratáveis do ponto de vista computacional. Exemplos de problemas polinomiais: o problema da equação do segundo grau, o problema do máximo divisor comum, o problema do caminho mínimo, etc.

A classe P (Tempo Polinomial) de problemas é o conjunto de todos os problemas polinomiais de decisão. Um problema é não-polinomial se nenhum algoritmo polinomial resolve o problema.

Atenção: não é classificado como não-polinomial problemas para os quais algoritmos polinomiais não são conhecidos atualmente, pois tais algoritmos podem vir a ser descobertos no futuro. Problemas desse tipo são considerados computacionalmente intratáveis.

Um algoritmo eficiente é aquele cuja complexidade é uma função polinomial nos tamanhos de entrada e saída. Para um problema de decisão, o tamanho da saída é constante, o que possibilita ignorá-lo. Como exemplo, se n for o tamanho da entrada, algoritmos cujas complexidades sejam O(1), O(n), $O(n^2)$, $O(n^2 logn)$, $O(n^6)$, $O(n^{10})$ seriam todos classificados como eficientes. Por outro lado, complexidades, como por exemplo, $O(2^n)$, $O(n^n)$, $O(n^{logn})$, O(n!) corresponderiam a algoritmos classificados como não eficientes.

Observe que esse critério de avaliação certamente não é absoluto. Por exemplo, considere dois algoritmos A e B, de complexidade $O(n^{10})$ e $O(2^n)$, respectivamente, para um mesmo problema X. O algoritmo A seria classificado como eficiente e B não eficiente. Suponha que A e B utilizem exatamente n^{10} e 2^n passos respectivamente, e ainda que ambos sejam implementados em um computador que efetua um passo em cada milissegundo. Uma instância de X em que n = 10, o algoritmo "eficiente" levaria 1010 milissegundos, enquanto o "não eficiente" terminaria em 210 = 1024 milissegundos. Contudo, é fácil verificar que quando n cresce o algoritmo A melhora seu desempenho em relação a B e a partir de certo ponto se torna, de fato, mais eficiente. Para grande maioria dos casos práticos, os algoritmos de complexidade polinomial são eficientes, enquanto os de complexidade não polinomial não o são.

Simplificando: o algoritmo é denominado polinomial quando sua complexidade for uma função polinomial nos tamanhos dos dados de entrada. Quando conveniente, utiliza-se o termo **intratável** para designar um problema que não admite algoritmo polinomial. Há pelo menos mais dois argumentos para justificar a adoção desse critério. Observe inicialmente que a complexidade da grande maioria dos algoritmos polinomiais constitui uma função polinomial de baixo grau. Isto é, são mais raras complexidades como $O(n^{10})$, sendo comuns O(n), O(nlgn) ou $O(n^2)$. O argumento adicional para justificar o critério, diz respeito ao modelo de computação correspondente à medida de complexidade. Com essa motivação, defini-se a classe P de problemas, como sendo aquela que compreende precisamente aqueles que admitem algoritmo polinomial.

Observe que se os algoritmos conhecidos para um certo problema X forem exponenciais, este fato por si só, não implica em que X não pertença a P. Se o problema X não pertence à classe P então necessariamente deve existir alguma prova de que todo possível algoritmo para resolver X não possui complexidade polinomial. Por exemplo, os algoritmos conhecidos, até agora para o problema do caixeiro viajante são todos exponenciais. Contudo, não é conhecida prova de que seja impossível a formulação de algoritmo polinomial para o problema. Portanto, se desconhece de o caixeiro viajante pertence ou não a P. Pode-se observar das seções que esta incerteza quanto à pertinência a classe P é compartilhada por um grande número de problemas.

1.3.3.2. CLASSE NP

A classe de problemas NP (Tempo polinomial não determinístico) denota o conjunto de problemas que são decidíveis em tempo polinomial por um algoritmo não-determinístico, ou seja, por um algoritmo que tenha várias possibilidades (fluxos de execução) de resposta e apenas algumas estão certas. Nesses problemas, em geral, o número de computações do melhor algoritmo conhecido cresce exponencialmente em função do tamanho da instância e não existe a garantia da existência de algoritmos melhores para a solução do problema.

Simplificando: Seja um problema X. Se achar a solução de X, possuir custo computacional mínimo conhecido exponencial ou fatorial, porém, testar se uma informação qualquer é solução de X, tiver custo computacional polinomial, dizemos que X é NP. Problemas da classe NP possuem soluções de ordem, por exemplo, $O(2^n)$, $O(n^n)$, $O(n^{\log n})$, $O(n!)$.

Exemplos:

1 – Problema do Ciclo Hamiltoniano. Saber se um grafo G possui ou não um ciclo hamiltoniano é um problema de solução mínima conhecida exponencial, porém, saber se um ciclo dado é hamiltoniano em G, é um problema polinomial.

2 – Problema do Isomorfismo em Grafos. Saber se um grafo G' é isomorfo de um grafo G é um problema de solução mínima conhecida exponencial, porém, saber se um isomorfismo dado é isomorfismo de G' em G, é um problema polinomial.

➔PROBLEMA: P = NP?

O problema "P versus NP" é o principal problema aberto da Ciência da Computação. Possui também enorme relevância em campos que vão desde a Engenharia até a criptografia aplicada aos serviços militares e às transações comerciais e financeiras via Internet.

Não é difícil entender que a classe NP inclui a classe P, ou seja, que todo problema polinomial é polinomial em uma máquina não determinística. O consenso sugere que P é apenas uma pequena parte de NP. Surpreendentemente, ninguém conseguiu provar que um problema NP não esteja em P.

De modo simplificado, o problema pergunta se existem problemas matemáticos cuja resposta pode ser verificada em tempo polinomial, que não possam ser resolvidos (diretamente, sem se ter um candidato à solução) em tempo polinomial. De outra forma: É verdade que todo problema cujas soluções podem ser conferidas por um algoritmo polinomial pode também ser resolvido por um algoritmo polinomial?

A solução para este problema tenta provar que um problema considerado NP, como por exemplo, o problema o ciclo hamiltoniano, tem algoritmo polinomial que o resolva ou provar que é impossível se ter um algoritmo polinomial que o resolva. Se descobrirmos como resolver o problema do ciclo hamiltoniano em tempo polinomial, ficaremos sendo capazes de resolver, também em tempo polinomial, uma grande quantidade de outros problemas matemáticos importantes; por outro lado, se um dia alguém provar que é impossível resolver o problema do ciclo hamiltoniano em tempo polinomial no número de vértices, também se terá estabelecido que uma grande quantidade de problemas importantes não tem solução prática. Por isso, o Instituto Clay oferece um prêmio de um milhão de dólares a quem conseguir resolver este problema matemático.

1.3.3.3. CLASSE NP-DIFÍCIL

Um problema é NP-Difícil se todos os problemas da classe NP forem redutíveis a ele, ou seja, é um problema que, quando resolvido, resolve automaticamente todos os problemas NP.

Para se provar que R é um problema NP-Difícil, deve-se encontrar um problema Q em NP que seja polinomialmente redutível a R, ou seja, deve-se elaborar um algoritmo que transforme a instância do problema Q em uma instância de R, de tal forma que a solução de R implique na solução de Q (cuja transformação também deve ser em tempo polinomial).

Exemplo:

1 – O Problema da Parada. É um problema NP-Difícil que não é NP. O problema consiste em se decidir, para qualquer algoritmo A e qualquer entrada de A, se A vai terminar ou entrar em loop infinito. A solução desse problema seria muito útil na construção de compiladores, porém, ele é indecidível para qualquer algoritmo de qualquer complexidade.

2 – Problema do Ciclo Hamiltoniano. Já explicado.

1.3.3.4. CLASSE NP-COMPLETO

Um problema é NP-Completo, se for ao mesmo tempo NP e NP-Difícil. Nenhum problema NP possui ordem de complexidade superior a um problema NP-Completo.

Talvez a razão mais forte pela qual os cientistas da computação acreditam que P $\neq$ NP seja a existência da classe de problemas NP – Completos. Essa classe tem a surpreendente propriedade de que, se qualquer problema NP – Completo pode ser resolvido em tempo polinomial, então todo problema em NP tem uma solução polinomial, isto é, P = NP. Entretanto, apesar de anos de estudo, nenhum algoritmo de tempo polinomial jamais foi descoberto para qualquer problema NP – Completo.

Exemplo:

1 – Problema do Ciclo Hamiltoniano. Já explicado.

Na figura 64, temos um resumo sobre as classes de problemas estudadas neste escrito.

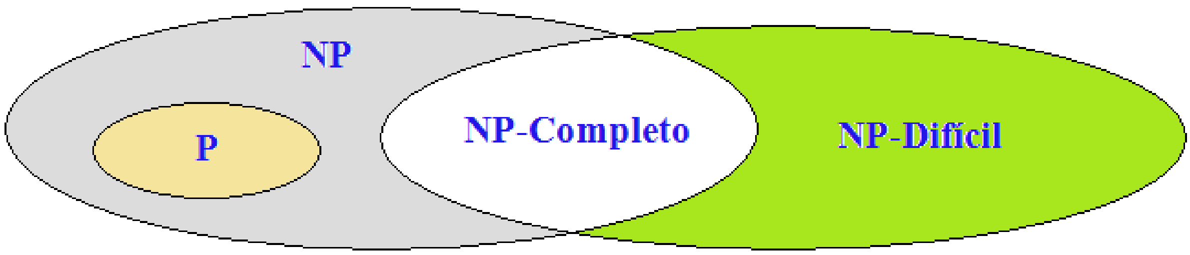


Figura 64: Resumo das classes de problemas.

BIBLIOGRAFIA

Livros:

BOAVENTURA NETTO, Paulo Oswaldo. **Grafos. Teoria, Modelos, Algoritmos.** Editora Edgard Blücher Ltda, São Paulo, 2003.

SZWARCFITER, Jayme Luiz. **Grafos e Algoritmos Computacionais**. Editora Campus Ltda, Rio de Janeiro, 1984.

Apostilas de Professores:

USP – Universidade de São Paulo
Prof. Paulo Feofiloff

UNICAP – Universidade Católica de Pernambuco
Prof. Wilson Rosa de Oliveira Junior