

UNIVERSIDADE FEDERAL DO AMAZONAS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA DE INICIAÇÃO CIENTÍFICA

FUNDAMENTOS MATEMÁTICOS DA TEORIA DOS GRAFOS E SUAS APLICAÇÕES

Bolsista: Vinicius Yassushi Yang Arashiro, FAPEAM

Manaus - Amazonas
2013

UNIVERSIDADE FEDERAL DO AMAZONAS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA DE INICIAÇÃO CIENTÍFICA

RELATÓRIO PARCIAL
PIB - E / 0047 / 2013-2014
FUNDAMENTOS MATEMÁTICOS DA TEORIA DOS GRAFOS E SUAS APLICAÇÕES

Bolsista: Vinicius Yassushi Yang Arashiro, FAPEAM
Orientador: Profº Dr. Sandro Dimy Barbosa Bitar

Manaus - Amazonas
2013

Resumo

Neste relatório são apresentados resultados relacionados à Teoria dos Grafos, uma importante ferramenta que pode ser aplicada em diversas áreas, dentre elas se destacam a Matemática e a Engenharia Elétrica e da Computação.

São abordados os conceitos básicos a respeito dos grafos, caminhos e árvores, e a capacitação matemática para a teoria. Esta abordagem inicial é de suma importância para a compreensão dos Teoremas clássicos, como o Teorema de Euler, estudos de modelos e o Problema das 4 Cores.

Abstract

In this report we present the results related to the Graph Theory, an important tool that can be useful in different areas, among them Mathematics, Electrical Engineering and Computer Engineering.

Basic concepts, paths and trees are discussed like the mathematical capacitation for the theory. This initial approach is considered to be very important to the comprehension of the classic Theorems, as Euler Theorem, models studies and the Four Color Theorem.

Conteúdo

1	Introdução	7
2	Desenvolvimento	8
2.1	Grafos	8
2.2	Árvores	16
2.2.1	Armazenamento de árvores	19
2.2.2	Árvores rotuladas e não-rotuladas	21
2.2.3	Encontrando a melhor árvore	23
2.2.4	Caixeiro viajante	24
2.3	Emparelhamento de grafos	24
2.4	Coloração	25
2.4.1	Teorema das quatro cores	26
2.4.2	Aplicação de coloração	27
	Conclusão	30
	Referências Bibliográficas	31

Lista de Figuras

2.1	(a) Exemplo de um grafo com 7 arestas e 6 vértices.	8
2.2	(a) Multigrafo com laço no vértice E e arestas paralelas entre os vértices A e F .	9
2.3	(a) Exemplo de um grafo com com um caminho. (b) Ciclo C_5	10
2.4	Um grafo G e seu subgrafo H	10
2.5	(a) Grafo conexo. (b) Grafo desconexo	11
2.6	Os grafos (a) e (b) apresentam isomorfismo.	11
2.7	(a) mostra o grafo completo K_6 . (b) mostra o grafo completo K_5	11
2.8	Grafo G à esquerda e seu Grafo complementar à direita.	12
2.9	(a) Grafo bipartido. (b) Grafo bipartido completo $K_{4,5}$	12
2.10	Grafos planares	13
2.11	Grafo representando o problema da ponte de Königsberg.	14
2.12	(a) É um grafo euleriano (trilha pode ser A-B-C-D-E-F-A-D-B-E-A), (b) é um grafo semieuleriano (a trilha pode ser A-E-B-D-C-B-A-D-E) e (c) não é euleriano, nem semieuleriano.	16
2.13	Exemplos de grafos e seus ciclos hamiltonianos.	16
2.14	Três exemplos de árvores.	17
2.15	A árvore de descendência de árvores.	19
2.16	Árvore rotulada.	19
2.17	Essas árvores são iguais?	22
2.18	Um grafo bipartido com emparelhamento perfeito.	24
2.19	Prova do Teorema das 5-cores	27
2.20	Tabela de alocação de alunos nos exames finais	28
2.21	Dois conjuntos disjuntos independentes	28
2.22	Cruzamento de vias de uma cidade	28
2.23	Cruzamento com direção incompatível à esquerda. Representação do grafo de direções incompatíveis.	29

Capítulo 1

Introdução

Este relatório expõe assuntos relacionados à Teoria dos Grafos. Grafos compõem-se basicamente de vértices e arestas, que correspondem, respectivamente, às entidades e às relações entre elas. Por exemplo, em um sistema viário de uma cidade, as arestas seriam as ruas e avenidas, enquanto os vértices seriam os cruzamentos entre elas.

Grafos estão presentes em inúmeras áreas, como escalonamento de tarefas, sistemas geográficos, projeto de circuitos integrados, geometria computacional ou desenvolvimento de compiladores. Para facilitar a resolução dos problemas práticos, teoremas acerca da Teoria dos Grafos foram desenvolvidos por grandes estudiosos matemáticos, entre eles, Leonhard Euler, considerado o primeiro a propor um problema, porém sem intenção, envolvendo grafos, o problema da Ponte de Königsberg.

Apesar de estudos sobre grafos serem feitos desde meados do século *XVIII*, muitos estudiosos tentam encontrar algoritmos que facilitem o uso desta poderosa ferramenta matemática. Por exemplo, atualmente não existem condições de contorno para que tal grafo seja hamiltoniano. Termos como grafo hamiltoniano serão vistos no relatório a seguir.

Capítulo 2

Desenvolvimento

2.1 Grafos

Nesta seção, será feito embasamento teórico sobre a *Teoria dos Grafos*, com o intuito de ver que vários problemas de nosso cotidiano podem ser modelados como aplicação prática da Teoria dos Grafos.

Definição 2.1.1 Um grafo G consiste de um conjunto V de elementos, os vértices, e um conjunto A de elementos, as arestas, e uma função de incidência ψ que associa a cada aresta α de G um par não ordenado de vértices (não necessariamente distintos) de G , chamados de extremos de α .

O número de vértices de um grafo G é denotado por $|V|$ ou por n , e o número de arestas de um grafo G é denotado por $|A|$ ou por m . O seu conjunto de vértices é simbolizado por $V(G)$, e por $A(G)$ o conjunto de arestas.

A única coisa que importa sobre uma aresta é o par de vértices que ela conecta, assim elas podem ser consideradas como subconjuntos de V . Isso significa que a aresta conectando os vértices u e v é simplesmente o conjunto $\{u, v\}$, mais simplificada podemos denotar a aresta por uv .

Grafos podem ser representados por diagramas, onde cada vértice é representado por um ponto e cada aresta por uma linha ligando os pontos que representam seus extremos.

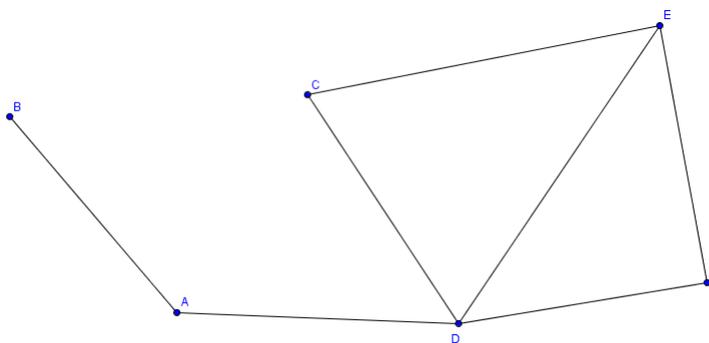


Figura 2.1: (a) Exemplo de um grafo com 7 arestas e 6 vértices.

Definição 2.1.2 Laço é uma aresta que conecta dois pares de vértices não distintos, isto é, seus extremos coincidem.

Definição 2.1.3 Arestas paralelas são aquelas que conectam o mesmo par de vértices.

Definição 2.1.4 *Multigrafos são grafos que contém arestas paralelas e laços.*

Paralelamente à definição (2.1.4), surge uma outra definição de grafos.

Definição 2.1.5 *Grafos simples permitem no máximo uma aresta conectando um par de vértices, e nenhum vértice conectado a ele próprio.*

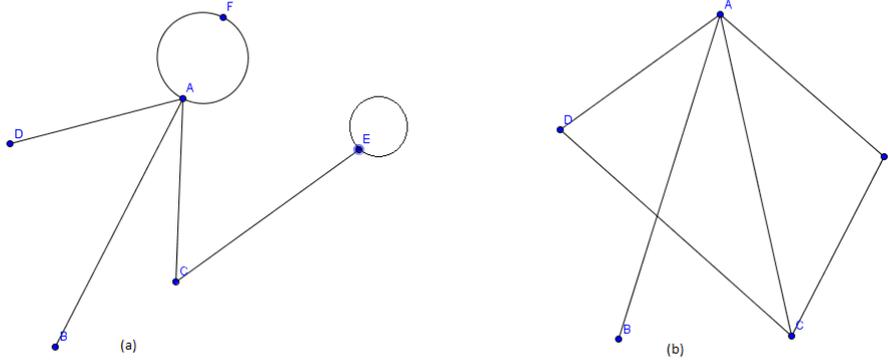


Figura 2.2: (a) Multigrafo com laço no vértice E e arestas paralelas entre os vértices A e F
b) Grafo Simples

Definição 2.1.6 *Vértices adjacentes são dois vértices conectados por uma aresta.*

Definição 2.1.7 *O grau de um vértice é a quantidade de arestas incidentes sobre ele, simbolizado por $d(v)$*

Quando calculamos o grau dos vértices de G , devemos contar duas vezes no caso se houver laços em G .

Teorema 2.1.1 *Para todo grafo G , temos que*

$$\sum_{v \in V(G)} d(v) = 2|A|$$

A relação nos diz que a soma dos graus dos vértices de um grafo G é igual ao dobro do número de arestas de G .

Demonstração: Quando contamos os graus dos vértices estamos contando as extremidades das arestas uma vez. Como cada aresta tem duas extremidades, cada aresta foi contada duas vezes.

□

Corolário 2.1.1 *Todo grafo G possui um número par de vértices com grau ímpar.*

Demonstração: Se tivéssemos um número ímpar de vértices de grau ímpar, a soma dos graus seria ímpar. Mas a soma dos graus é o dobro do número de arestas e, portanto é um número par.

□

Definição 2.1.8 Dizemos que um passeio de um grafo G é uma sequência de vértices e arestas, que começa e termina com vértices, e cada aresta une vértices que ocorrem imediatamente antes e depois na sequência. Chamamos de extremidades do passeio o primeiro e o último vértices da sequência. Se as extremidades do passeio coincidem, isto é, se forem o mesmo vértice, dizemos que trata-se de um passeio fechado, caso contrário é chamado de passeio aberto. Seu comprimento s é a quantidade de arestas presentes no passeio. Quando as arestas do passeio são distintas, o passeio passa a ser chamado de trilha. Se suas extremidades coincidirem no mesmo vértice, chama-se de trilha fechada.

Definição 2.1.9 Caminho de um grafo G é um passeio onde não há a repetição de vértices e arestas. Se for conectada o último e o primeiro vértice, obtemos um ciclo. O comprimento do caminho ou do ciclo é a quantidade de arestas da sequência. Um ciclo de comprimento k é denotado por k -ciclo. O ciclo também pode ser definido como um grafo conexo regular de grau 2, e denotado por C_n , onde n é o número de vértices.

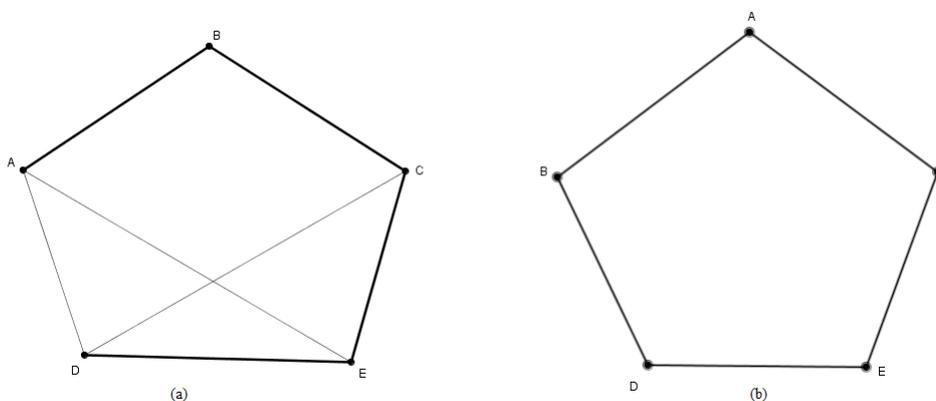


Figura 2.3: (a) Exemplo de um grafo com com um caminho. (b) Ciclo C_5

Definição 2.1.10 Um grafo H é dito um subgrafo de G se ele pode ser obtido de G se removendo-se algumas de suas arestas e alguns de seus vértices. Mais precisamente, um grafo $H = (U, B)$ é dito subgrafo de $G = (V, A)$ se $U \subseteq V$ e $B \subseteq A$, onde U representa o conjunto de vértices do subgrafo H e B o conjunto de arestas de H .

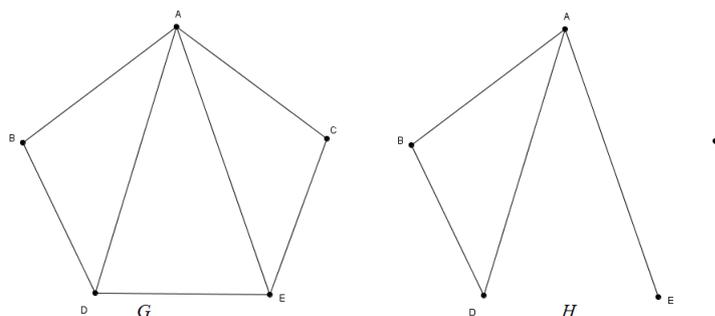


Figura 2.4: Um grafo G e seu subgrafo H .

Definição 2.1.11 Um grafo G é dito conexo quando dois vértices quaisquer do grafo G podem ser unidos por pelo menos um caminho. Para ser mais preciso, um grafo G é conexo se para cada dois vértices u e v existe um caminho com extremidades u e v que é um subgrafo de G .

Se um grafo G é desconexo, então ele terá dois ou mais subgrafos conexos, também chamados de componentes conexas do grafo.

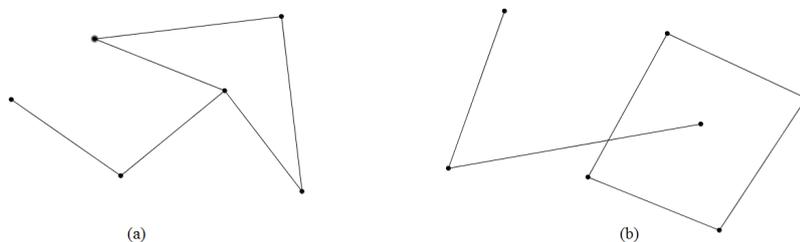


Figura 2.5: (a) Grafo conexo. (b) Grafo desconexo

Definição 2.1.12 Dois grafos G_1 e G_2 são isomorfos se existe uma correspondência 1-a-1 entre seus conjuntos de vértices que preserve as adjacências.

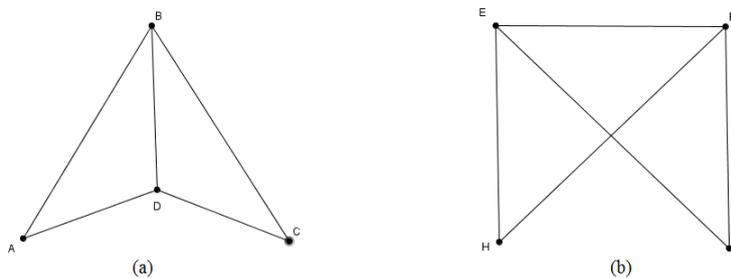


Figura 2.6: Os grafos (a) e (b) apresentam isomorfismo.

Definição 2.1.13 Grafo completo ocorre quando qualquer par de vértices do grafo é ligado por uma aresta. Este grafo também é chamado de clique. Um grafo completo com n vértices é representado por K_n e o número total de arestas do grafo completo é $\binom{n}{2}$ arestas.

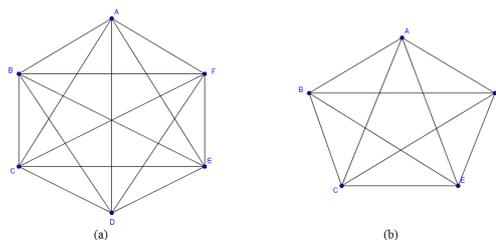


Figura 2.7: (a) mostra o grafo completo K_6 . (b) mostra o grafo completo K_5 .

Definição 2.1.14 Chamamos de grafo complementar \overline{G} de um grafo G , quando \overline{G} é formado pelo mesmo conjunto de vértices de G e pelas arestas que faltam em G para se tornar completo. É fácil perceber que $V(G) = V(\overline{G})$ e que $A(G) \cup A(\overline{G})$ inclui todas as arestas do grafo completo.

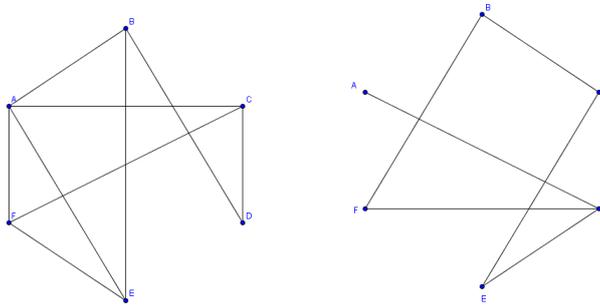


Figura 2.8: Grafo G à esquerda e seu Grafo complementar à direita.

Definição 2.1.15 Um grafo é dito regular (de grau k , ou ainda k -regular) quando todos os seus vértices têm o mesmo grau(k).

Definição 2.1.16 Um grafo é bipartido quando o conjunto de vértices V pode ser particionado em dois subconjuntos disjuntos V_1 e V_2 , e toda aresta de G tem uma extremidade em V_1 e outra em V_2 . O subconjunto V_1 é dito um subconjunto independente de vértices do grafo G pois não há arestas ligando dois vértices de V_1 . Temos também que V_2 é um subconjunto de vértices de G .

O grafo é bipartido completo quando todo vértice de V_1 é ligado a todos os vértices de V_2 . É denotado por $K_{p,q}$.

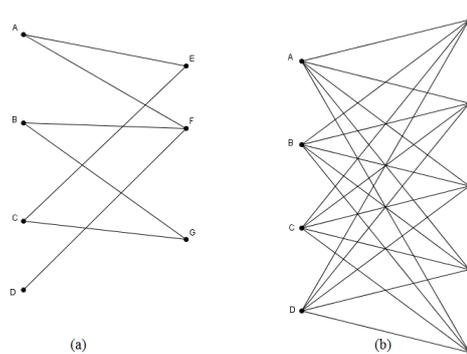


Figura 2.9: (a) Grafo bipartido. (b) Grafo bipartido completo $K_{4,5}$.

Teorema 2.1.2 Um grafo G é bipartido se, e somente se, não contém ciclos de comprimento ímpar.

Demonstração: (\Rightarrow) Seja G bipartido. Se não houver um ciclo em G , não há o que mostrar. Se há um ciclo em G est alterna entre vértices de V_1 e V_2 , dois subconjuntos independentes e disjuntos. Partindo de V_1 (por exemplo), para retornar ao ponto de partida teremos que utilizar um número par de arestas. O ciclo é, portanto, de comprimento par.

(\Leftarrow) Podemos considerar apenas grafos conexos. Seja G um grafo sem ciclos ímpares. Vamos particionar seu conjunto de vértices em dois subconjuntos V_1 e V_2 , independentes e disjuntos. Tomamos primeiramente um vértice qualquer v . O subconjunto V_1 será formado por todos os vértices w tais que exista um caminho de comprimento par entre v e w . O subconjunto V_2 será formado por todos os vértices w tais que exista um caminho de comprimento ímpar entre v e w . Os conjuntos V_1 e V_2 são disjuntos, pois se w estivesse em V_1 e V_2 ao mesmo tempo, haveria um caminho de comprimento par e um caminho de comprimento ímpar ligando v a w . Esses dois caminhos podem se cruzar (ou não) antes de chegar em w , produzindo alguns ciclos. Como o número de arestas usado nestes ciclos é ímpar (é a soma do número de arestas dos dois caminhos) isso produziria pelo menos um ciclo ímpar em G , contrariando a hipótese.

□

Lema. *Se todo vértice de um grafo (não necessariamente simples) G tem grau maior ou igual a 2, então G contém um ciclo.*

Demonstração: Se G contém laços ou arestas múltiplas, não há o que provar, pois, automaticamente, G contém um ciclo. Consideramos, portanto, apenas os grafos simples. À partir de um vértice v_0 , qualquer, iniciamos nossa trilha. Quando chegamos a um vértice qualquer, ou o estamos visitando pela primeira vez e podemos continuar, ou chegamos a um vértice já visitado, produzindo um ciclo. Como o número de vértices é finito, o lema está provado.

□

Definição 2.1.17 *Um grafo planar é um grafo que é desenhado no plano de modo que suas arestas são curvas contínuas não-intersectantes. Assumimos também que o grafo é conexo. Tal grafo divide o plano em certas partes, chamadas de países ou faces. Exatamente um país é infinito, os outros são finitos.*

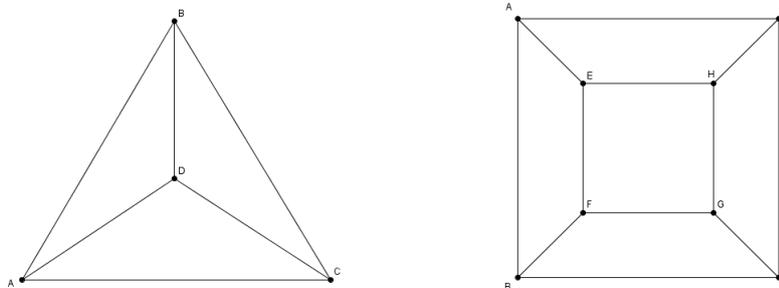


Figura 2.10: Grafos planares

Um resultado muito importante, descoberto por Euler, nos diz que podemos determinar o número de países em um mapa planar conexo, se sabemos o número de nós e arestas do grafo. A fórmula de Euler é a seguinte:

Teorema 2.1.3 *Num grafo planar conexo vale $f - m + n = 2$.*

Demonstração: Demonstraremos o teorema por indução sobre o número de arestas. Tomemos um grafo conexo qualquer. Se for uma árvore, temos $f - m + n = 1 - (n - 1) + n = 2$. Se houver um ciclo, retiramos uma aresta do ciclo, e o grafo fica com uma face a menos, mas pela hipótese de indução a relação vale para o novo grafo. Temos então $(f - 1) - (m - 1) + n = 2$ e, portanto, $f - m + n = 2$.

□

Observamos que podemos acrescentar arestas a um grafo planar sempre que uma porção do plano estiver limitada por um ciclo de comprimento maior do que 3. Logo, um grafo maximal planar (i.e., um grafo ao qual não poderemos acrescentar arestas sem comprometer a planaridade) tem uma representação composta por ciclos de comprimento 3. Isto nos dá outra relação importante.

Teorema 2.1.4 *Num grafo planar conexo G vale $m \leq 3n - 6$; a igualdade vale se G é maximal planar.*

Demonstração: Se formos contar as arestas de cada face, contaremos duas vezes cada aresta do grafo. Como cada face tem no mínimo 3 arestas (a igualdade valendo no caso maximal) temos:

$$3.f \leq 2.m$$

Substituindo na fórmula de Euler:

$$f - m + n = 2,$$

$$3.f - 3.m + 3.n = 6,$$

$$2.m - 3.m + 3.n \geq 6,$$

$$m \leq 3.n - 6.$$

□

Este teorema nos dá outra demonstração de que K_5 não é planar. De fato, K_5 (e de resto todos os grafos completos com mais do que 4 vértices) não obedece à relação acima: $10 > 3 \cdot 5 - 6$.

Talvez o mais antigo resultado em teoria dos grafos tenha sido descoberto por Euler, o maior matemático do século XVIII. Tudo começou com um desafio recreativo que os cidadãos de Königsberg (hoje Kaliningrado) propuseram. A cidade era dividida em quatro distritos por braços do rio Pregel, que eram conectados por sete pontes. Era agradável caminhar por ali, atravessando essas pontes, e daí surgiu a questão: é possível fazer um passeio de modo que se atravessasse toda ponte exatamente uma vez?

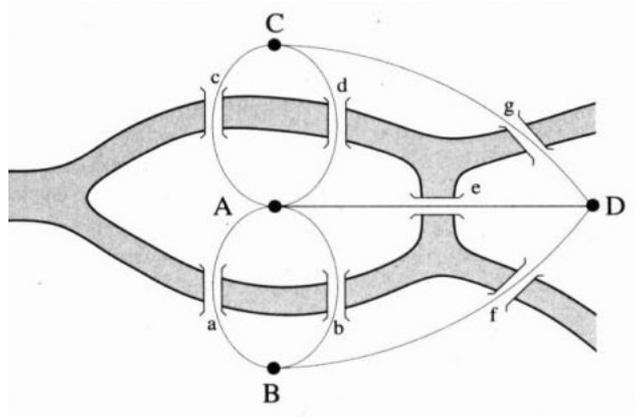


Figura 2.11: Grafo representando o problema da ponte de Königsberg.

Euler publicou um artigo em 1736, no qual ele demonstrava que tal passeio era impossível. O argumento é um tanto simples. Suponha que exista tal passeio. Considere quaisquer das quatro partes da cidade, digamos a ilha de Kneiphoff, e suponha que nosso passeio não começa aqui. Então, em algum ponto no tempo entramos na ilha atravessando uma ponte; mais tarde a deixamos por meio de uma outra ponte (pelas regras do passeio). Então entramos nela novamente através de uma quinta, mas para sair, teríamos que ter uma sexta ponte, mas não há. Temos que terminar nosso passeio na ilha.

Portanto temos que começar ou terminar nosso passeio na ilha. O problema é que podemos tirar a mesma conclusão para qualquer dos outros três distritos da cidade. A única diferença é que ao invés de cinco pontes, esses distritos são conectados ao restante da cidade por três pontes apenas; logo, se não começamos aí, ficamos presos lá na segunda visita, não na terceira.

Mas agora estamos com problemas: nosso passeio começa em um dos distritos termina em um outro - para quaisquer dos dois distritos remanescentes, o argumento acima leva a uma contradição. Isso prova que tal passeio não existe, nenhum passeio pode atravessar toda ponte exatamente uma vez.

O resultado de Euler é geralmente considerado como o primeiro teorema da teoria dos grafos.

Definição 2.1.18 *Um passeio euleriano é um passeio que passa por toda aresta exatamente uma vez (o passeio pode ou não ser fechado).*

Teorema 2.1.5 Teorema de Euler (Euler - 1736). *Um grafo conexo (não necessariamente simples) G é euleriano se, e somente se, todos os seus vértices tem grau par.*

Demonstração

(\Rightarrow) Suponhamos que G tenha uma trilha fechada de comprimento m . Cada vez que a trilha passa por um vértice utiliza duas novas arestas, uma para entrar e outra para sair. Logo, o grau de cada vértice deve ser obrigatoriamente par.

(\Leftarrow) Usaremos indução sobre o número de arestas m do grafo. Por vacuidade, o teorema é válido quando $m = 0$. Suponhamos que o teorema seja válido para todos os grafos com menos do que m arestas. Sendo G conexo, todos os vértices têm grau maior do que 2, pois os graus são pares. Pelo lema anterior, G contém um ciclo (que é uma trilha fechada). Dentre todas as trilhas fechadas em G escolhemos uma trilha T com comprimento máximo. Se T tem comprimento m , o teorema está provado. Caso contrário, consideramos o grafo H resultante da retirada das arestas de T . Como retiramos um número par de arestas de cada vértice de T , e todos os vértices do grafo tem grau par (pela hipótese), pelo menos uma das componentes de H tem um vértice em comum com T e tem todos os vértices com grau par. Pela hipótese de indução, H tem uma trilha fechada que passa por todos os vértices de H , e podemos formar uma trilha fechada maior concatenando T com a trilha em H . Mas isto contraria a maximalidade na escolha de T .

□

Se o grafo não é euleriano, mas tem uma trilha aberta de comprimento m , ele é dito semi-euleriano. Um grafo conexo (não necessariamente simples) G é semi-euleriano se, e somente se, no máximo, dois vértices têm grau ímpar [JURKIEWICZ, 2009].

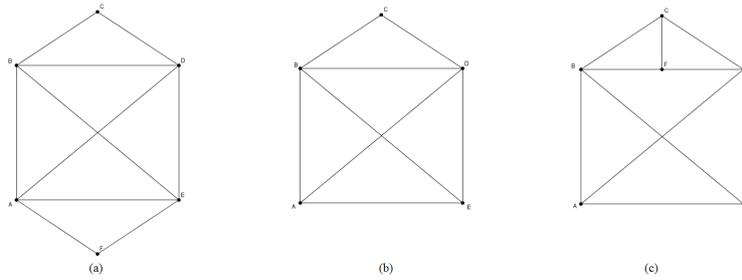


Figura 2.12: (a) É um grafo euleriano (trilha pode ser A-B-C-D-E-F-A-D-B-E-A), (b) é um grafo semieuleriano (a trilha pode ser A-E-B-D-C-B-A-D-E) e (c) não é euleriano, nem semieuleriano.

Um problema aparentemente similar ao dos grafos eulerianos é o de procurar em G uma trilha fechada que passe por todos os vértices uma e só uma vez. Este problema foi proposto em 1859 por William Rowan Hamilton como apenas um jogo matemático, a princípio sem aplicação prática. O jogo proposto por Hamilton ficou conhecido como o Problema do ciclo Hamiltoniano ("Hamilton Cycle Problem" - HCP, em inglês) e, mais de 100 anos depois, encontra várias aplicações práticas como, por exemplo, no projeto de circuitos impressos. O problema de saber se um grafo é hamiltoniano ou não é um dos mais estudados da teoria dos grafos por sua aplicabilidade em comunicação, transporte e planejamento.

O ciclo hamiltoniano é utilizado na definição formal do Problema do Caixeiro Viajante, onde dado um grafo completo ponderado (grafo que possui arestas valoradas) G , procuramos encontrar o ciclo hamiltoniano de menor valor. Entretanto, até hoje não existe tal fórmula para achá-lo.

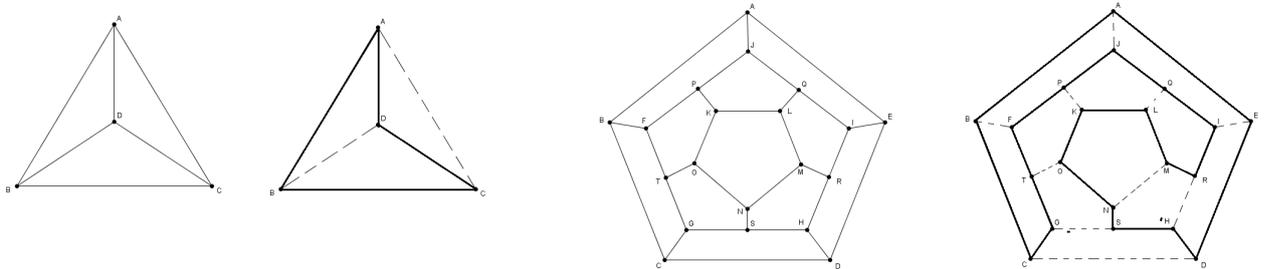


Figura 2.13: Exemplos de grafos e seus ciclos hamiltonianos.

2.2 Árvores

Um grafo G é chamado uma árvore se ele é conexo e não contém qualquer ciclo como subgrafo. A árvore mais simples tem um nó e nenhuma aresta. A segunda árvore mais simples consiste de dois nós conectados por uma aresta. O seguinte teorema mostra que árvores podem ser caracterizados como grafos "minimamente conexos", bem como grafos "maximamente livres de ciclo".

Teorema 2.2.1 (a) um grafo G é uma árvore se e somente se ele é conexo, mas a remoção de qualquer de suas arestas resulta em um grafo desconexo.

(b) Um grafo G é uma árvore se e somente se ele não contém nenhum ciclo, mas a adição de qualquer nova aresta cria um ciclo.

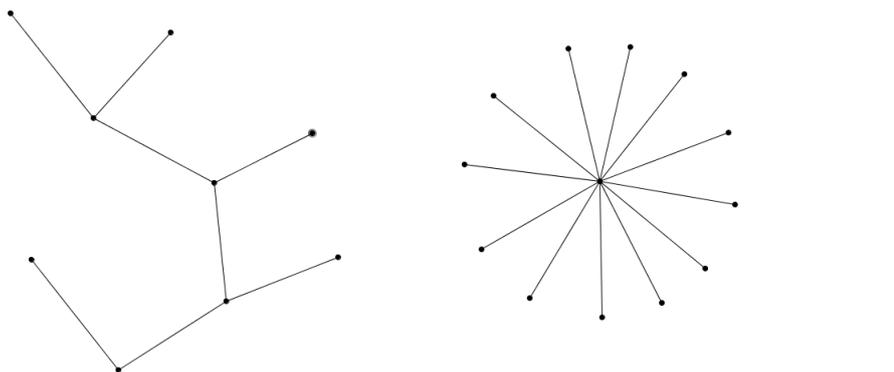


Figura 2.14: Três exemplos de árvores.

Prova. (a) Primeiro, temos que provar que se G é uma árvore então ele satisfaz a condição dada no teorema. Está claro que G é conexo (pela definição de árvore). Queremos provar que após remover qualquer aresta, ele não permanece conexo. A prova é indireta: assuma que ao remover a aresta uv de uma árvore G , o grafo resultante G' seja conexo. Então G' contém um caminho P conectando u e v . Mas então se colocarmos a aresta uv de volta, o caminho P e a aresta uv formará um ciclo em G , o que contradiz a definição de árvores.

Segundo, temos que provar que se G satisfaz a condição dada no teorema, então ele é uma árvore. Está claro que G é conexo, portanto temos apenas que argumentar que G não contém um ciclo. Novamente por um argumento indireto, assuma que G contém um ciclo C . Então ao remover qualquer aresta de C , obtemos um grafo conexo. Mas isso contradiz a condição no teorema.

(b) Suponha que G é uma árvore. Considere dois vértices u e v . Como G é conexo, então existe um caminho P entre u e v . Ao adicionar uma nova aresta ligando u e v então o grafo G terá um ciclo formado por P e por uv . Como u e v foram tomados arbitrariamente, a adição de qualquer aresta forma um ciclo.

Agora suponha que G é um grafo sem ciclos, mas que a adição de qualquer aresta cria um ciclo. Sejam u e v dois vértices quaisquer. Se for adicionada uma aresta ligando u e v , por hipótese G terá um ciclo. Assim sendo, sabe-se que um caminho é um ciclo sem uma das arestas. Então, ao retirar a aresta uv tem-se que existe um caminho entre u e v , logo G é conexo, pois u e v foram tomados arbitrariamente. Como G não contém ciclos e é conexo, G é uma árvore.

□

Definição 2.2.1 *Seja um grafo G . Um subgrafo de G com o mesmo o mesmo conjunto de nós n que é também uma árvore é chamado de árvore geradora de G e possui $n - 1$ arestas.*

Para se encontrar uma árvore geradora, devemos saber o que são pontes ou aresta-de-cortes. Considere um grafo G com n vértices, e uma aresta e . Se removermos e do grafo, o grafo resultante pode ou não permanecer conexo. Se ele ficar desconexo, então chamamos a aresta e de *aresta-de-corte* ou de *ponte*. Essa definição implica que todas as aresta de uma árvore são pontes.

Se encontrarmos uma aresta que não seja uma ponte, remova-a. Prossiga este procedimento até obter um grafo que seja conexo, mas que a remoção de qualquer aresta desse grafo o deixe desconexo.

Frequentemente usamos árvores que têm um vértice especial, que se chama *raiz*. Podemos tomar qualquer árvore, escolher qualquer um de seus vértices, e chamá-lo de raiz. uma árvore

com uma raiz determinada é chamada de *árvore enraizada* .

O que se segue é uma das mais importantes propriedades de árvores.

Teorema 2.2.2 *Toda árvore com pelo menos dois nós tem pelo menos dois nós de grau 1.*

Demonstração: Vamos começar de qualquer nó v_o da árvore e fazer um passeio sobre a árvore. Digamos que não queremos nunca voltar de um nó sobre a aresta através da qual entramos nesse nó; isso é possível a menos que tomemos um nó de grau 1, e nesse caso paramos e a prova está terminada.

Portanto vamos argumentar que isso tem que acontecer mais cedo ou mais tarde. Se não acontecer, em algum momento adiante temos que retornar a um nó que já visitamos; mas então os nós e arestas que percorremos entre as duas visitas formam um ciclo. Isso contradiz nossa suposição de que G é uma árvore e portanto não contém qualquer ciclo.

□

Uma árvore de verdade cresce desenvolvendo um novo ramo, e mais outro, e assim por diante. Mostramos que árvores-grafos podem ser levadas a crescer da mesma maneira. Para ser preciso, considere o seguinte procedimento, que chamamos de *Procedimento de Crescimento-de-Árvore*:

– Comece com um simples nó.

– Repita o que se segue um número qualquer de vezes: se você tem um grafo qualquer G , crie um novo nó e conecte-o por uma nova aresta a qualquer nó de G .

Teorema 2.2.3 *Todo grafo obtido pelo Procedimento de Crescimento-de-Árvore é uma árvore, e toda árvore pode ser obtida dessa maneira.*

Demonstração: Primeiro, considere qualquer grafo que pode ser obtido por esse procedimento. O grafo inicial é certamente uma árvore, portanto basta argumentar que nunca criamos uma "não-árvore"; em outras palavras se G é uma árvore, e G' for obtido de G criando-se um novo nó v e conectando-o a um nó u de G , então G' é uma árvore. Isso é imediato: G' é conexo, pois quaisquer dois "velhos" nós podem ser conectados por um caminho em G , enquanto que v pode ser conectado a qualquer outro w primeiro indo a u e então conectando u a w . Além do mais, G não contém ciclo: v tem grau 1 e portanto nenhum ciclo pode passar por v , mas um ciclo que não passa por v seria um ciclo no grafo anterior, o qual supostamente é uma árvore.

Segundo, vamos argumentar que toda árvore pode ser construída dessa maneira. Provamos isso por indução sobre o número de nós. Se o número de nós é 1, então a árvore surge pela construção, pois essa é a maneira pela qual começamos. Assuma que G é uma árvore com pelo menos 2 nós. Então pelo teorema 2.2.3, G tem um nó de grau 1 (pelo menos dois nós, na verdade). Seja v um nó com grau 1. Remova v de G , juntamente com aresta com extremidade v para obter um grafo G' .

Afirmamos que o G' é uma árvore. De fato, G' é conexo: quaisquer dois nós de G' podem ser conectados por um caminho em G , e esse caminho não pode passar por v pois v tem grau 1. Portanto esse caminho também é um caminho em G' . Além do mais, G' não contém um ciclo pois G não o contém.

Pela hipótese da indução, toda árvore com menos nós que G surge pela construção; em particular, G' assim o faz. Mas então G surge de G' por uma iteração a mais do segundo passo. Isso completa a prova do teorema 2.2.3.

□

A figura 2.15 mostra como árvores com até 4 nós surgem a partir dessa construção. Note que há uma "árvore de árvores" aqui. O fato de que a estrutura lógica dessa construção é uma árvore não tem nada a ver com o fato de que estamos construindo árvores: qualquer construção iterativa com escolhas livres a cada passo resulta em uma "árvore de descendência" semelhante.

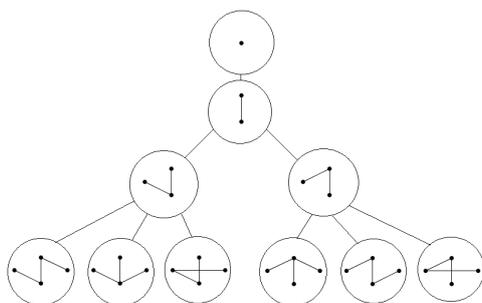


Figura 2.15: A árvore de descendência de árvores.

Quantas arestas tem uma árvore? É claro, isso depende do número de nós; mas surpreendentemente, depende *somente* do número de nós:

Teorema 2.2.4 *Toda árvore sobre n nós tem $n - 1$ arestas.*

Demonstração: De fato, começamos com um nó a mais que arestas, e a cada passo, um novo nó e uma nova aresta são adicionados, e essa diferença de 1 é mantida.

□

2.2.1 Armazenamento de árvores

Suponha que você deseja armazenar uma árvore rotulada em um computador. Para fazer isso depende de qual a necessidade em armazenar uma árvore, que informação sobre ela você deseja recuperar e quão frequentemente. No entanto estamos preocupados com a quantidade de memória que precisamos. Desejamos armazenar a árvore de tal modo que ela ocupe a mínima quantidade de memória. Algumas maneiras serão apresentadas a seguir.

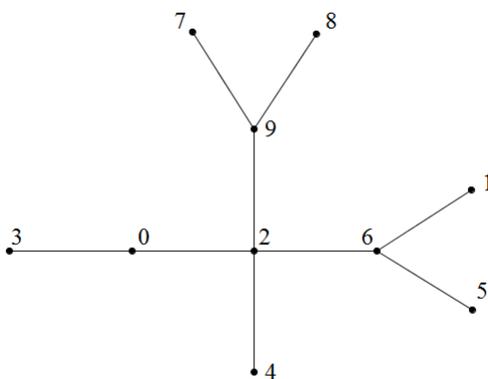


Figura 2.16: Árvore rotulada.

(a) Suponha que temos uma árvore G com n nós. Uma coisa que vem à cabeça é fazer uma grande tabela, com n linhas e n colunas, e colocar o número 1 na j -ésima posição da i -ésima linha se os nós i e j estão conectados por uma aresta, e o número 0, se eles não estão. Esse método de armazenar a árvore chamado de *matriz de adjacência do grafo* pode ser usada para qualquer grafo. Apesar deste método ser eficiente, gasta muita memória do computador. Precisamos de um bit para armazenar cada entrada dessa tabela, portanto isso toma n^2 bits.

(b) Fazemos melhor se especificarmos cada árvore listando todas as suas arestas. Podemos especificar cada aresta por duas extremidades. Será conveniente arranjar essa lista num vetor

cujas colunas correspondam às arestas. Por exemplo, a árvore na Figura 2.16 pode ser codificada por

7	8	9	6	3	0	2	6	6
9	9	2	2	0	2	4	1	5

Ao invés de uma tabela com n linhas, obtemos uma tabela com duas linhas. Pagamos um pouco por isso, ao invés de somente 0 e 1, a tabela contém números entre 0 e $n - 1$. Mas isso certamente vale a pena: mesmo se contarmos bits, escreve o rótulo de um nó toma $\log_2 n$ bits, portanto a tabela inteira ocupa apenas $2n \log_2 n$ bits, que é muito menos que n^2 se n grande.

(c) **O código de pai.** De agora em diante, o nó com rótulo 0 terá um papel especial; consideraremos esse nó o nó "raiz" árvore. Então podemos listar os dois nós finais de uma aresta listando a extremidade mais distante da raiz primeiro, e depois a extremidade mais próxima da raiz a seguir.. Portanto para toda aresta, o nó escrito abaixo é o pai do nó escrito acima. Para ordem na qual listamos as arestas, vamos tomar a ordem de seus primeiros nós. Para a árvore da Figura 2.16, obtemos a tabela

1	2	3	4	5	6	7	8	9
6	0	0	2	6	2	9	9	2

A primeira linha consiste dos números 1, 2, 3, 4, 5, 6, 7, 8, 9 nessa ordem. Ordenamos as arestas pela ordem crescente de suas primeiras extremidades. A raiz 0 não ocorre, pois ela não é o filho de qualquer outro nó. Mas por que obtemos todo outro número exatamente uma vez? Após um pouco de reflexão, isso deve também ficar claro: se um nó ocorre na primeira linha, então seu pai ocorre abaixo dele. Como um nó tem apenas um pai, ele pode ocorrer apenas uma vez. Como todo nó diferente da raiz tem um pai, todo nó diferente da raiz ocorre na primeira linha.

Por conseguinte, sabemos que se tivermos uma árvore sobre n nós, e escrevermos o vetor usando esse método, então a primeira linha consistirá de 1, 2, 3, ..., $n - 1$. Com isso, podemos suprimir a primeira linha sem perder nenhuma informação, bastando armazenar a segunda. Portanto especificamos uma árvore de $n - 1$ números, cada um entre 0 e $n - 1$. isso toma $(n - 1)[\log_2 n]$ bits.

(d) Agora descrevemos o chamado *código de Prüfer*, que atribuirá a qualquer árvore rotulada de n -pontos uma sequência de comprimento $n - 2$, e não $n - 1$, consistindo dos números 0, ..., $n - 1$. O ganho é pequeno, mas importante: mostraremos que toda sequência dessa corresponde a uma árvore. Por conseguinte estabeleceremos uma *bijecção*, uma correspondência um-para-um entre árvores rotuladas sobre n nós e sequências de comprimento $n - 2$, consistindo de números 0, 1, ..., $n - 1$. Como o número de tais sequências é n^{n-2} .

Ainda consideramos 0 como raiz, e ordenamos as duas extremidades de uma aresta tal que filho vem primeiro, mas ordenamos as arestas (as colunas do vetor) não pela magnitude de sua primeira extremidade porém um pouco diferente, mais diretamente relacionada à própria árvore.

Portanto, construímos uma tabela com duas linhas cujas colunas correspondem às arestas, e cada aresta é listada de modo que o nó mais distante de 0 esteja no topo e seu pai embaixo. A ordem na qual listamos as arestas é diferente do código do pai. No código de *Prüfer* procuramos por um nó de grau 1, diferente de 0, com o menor rótulo, e escrevemos a aresta formada por 1 e 6. Então removemos esse nó e aresta da árvore, e repetimos: procuramos pela extremidade com o menor rótulo, diferente de 0, e escrevemos a aresta incidente a ela. Em nosso caso, isso significa a aresta com extremidades 3 e 0. Então removemos esse nó e aresta etc. Prosseguimos até que todas as arestas sejam listadas. O vetor que obtemos é chamado de *código estendido de Prüfer* da árvore (chamamos de estendido porque, como veremos, precisamos apenas de uma parte dele como o código de Prüfer "real"). O código estendido de Prüfer na Figura 2.16 é:

1	3	4	5	6	7	8	9	2
6	0	2	6	2	9	9	2	0

Uma pequena observação é que a última entrada na segunda linha é sempre 0, pois ela vem da última aresta e como nunca tocamos o nó 0, essa última aresta deve ser incidente a ela. Agora não está claro que a primeira linha é supérflua; ela ainda consiste dos números $1, 2, \dots, n - 1$, mas agora elas não estão mais na ordem crescente.

A proposição chave é que a primeira linha é determinada pela segunda:

Proposição 2.2.1 *A segunda linha de um código de Prüfer determina a primeira.*

Demonstração: De fato, pois cada entrada na primeira linha do código estendido de Prüfer é o menor inteiro que não ocorre na primeira linha antes dele, nem na segunda linha abaixo ou depois dele, pois quando essa entrada (por exemplo a k -ésima entrada da primeira linha) foi registrada, então os vértices antes dela na primeira linha foram removidos (juntamente com as arestas correspondentes às $k - 1$ colunas). As entradas remanescentes na segunda coluna são exatamente aqueles vértices que são pais ao mesmo tempo, o que significa que eles não são folhas. Isto descrever como a primeira linha pode ser reconstruída a partir da segunda.

□

Portanto, não é preciso armazenar toda a matriz do código estendido de Prüfer para armazenar a árvore, basta armazenar a segunda linha. Na verdade, como se sabe que a última entrada é 0, não é necessário o armazenamento desta entrada. A lista consistindo das primeiras $n - 2$ entradas da segunda linha é conhecido como *código de Prüfer* da árvore. Por conseguinte, este código é uma lista de comprimento $n - 2$ onde cada entrada é um número inteiro entre 0 e $n - 1$, o que é bem semelhante ao código de pai, sendo apenas um dígito mais curto. Mas a principal característica do código de Prüfer é que ele é ótimo, como diz a Proposição 2.2.3

Proposição 2.2.2 *Toda lista de números entre 0 e $n - 1$, de comprimento $n - 2$, é um código de Prüfer de alguma árvore sobre n vértices.*

Demonstração: É bem simples, considere uma de $n - 2$ entradas com cada entrada assumindo valores inteiros entre 0 e $n - 1$. Então acrescenta-se 0 após a última entrada aumentando o tamanho da lista. Após isso, escreve-se uma linha acima desta lista de mesmo comprimento, de maneira que acima de cada entrada na primeira linha seja o menor inteiro que não ocorre na primeira linha antes dele e nem na segunda abaixo ou depois dele (o que é sempre possível pois esta condição exclui no máximo $n - 1$ valores de n possíveis). O resultado é um código de Prüfer estendido que é uma árvore que é única. De fato, pois se duas listas distintas representassem a mesma árvore, então as matrizes teriam que ser iguais, o que implicaria que cada entrada teria que ser igual a sua correspondente na outra lista, sendo então as duas listas iguais.

□

2.2.2 Árvores rotuladas e não-rotuladas

Como costumamos contar todo tipo de coisas, e já que temos conhecimento acerca de árvores, é natural perguntar: *quantas árvores existem sobre n vértices?*

Para respondermos isso, temos que definir cuidadosamente quando consideramos duas árvores iguais. A seguir estão duas possibilidades:

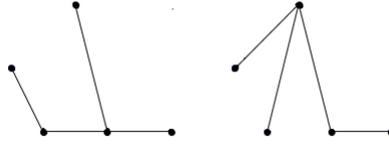


Figura 2.17: Essas árvores são iguais?

- Fixamos o conjunto de vértices, e consideramos duas árvores iguais se os mesmos pares de nós são conectados em cada um. Nesse caso é aconselhável dar nomes aos nós, de modo que possamos distinguí-los. É conveniente usar os números $0, 1, 2, \dots, n - 1$ como nomes, onde n é o número de vértices da árvore. Expressamos isso dizendo que os vértices da árvore são rotulados por $0, 1, 2, \dots, n - 1$.
- Não damos nomes aos nós, e consideramos duas árvores iguais se podemos rearranjar os nós de um modo que obtemos uma outra árvore. Mais exatamente, consideramos árvores iguais se existe uma correspondência um-para-um entre os nós da primeira árvore e os nós da segunda árvore tal que dois nós na primeira árvore que estão conectados por uma aresta, e vice-versa (isomorfismo). Se falamos sobre *árvores não-rotuladas*, queremos dizer que não distinguimos árvores isomorfas das outras.

Portanto podemos fazer duas perguntas: quantas árvores rotuladas existem sobre n nós? e quantas árvores não-rotuladas existem sobre n nós. Essas duas questões são bastante diferentes, e por isso temos que considerá-las separadamente.

Para o caso de árvores rotuladas, existe uma solução muito bonita.

Teorema 2.2.5 Teorema de Cayley *O número de árvores rotuladas sobre n nós é n^{n-2} .*

Demonstração: De fato, toda lista de $n - 2$ elementos com cada entrada entre 0 e $n - 1$ corresponde a uma árvore sobre n vértices, assim sendo o número de sequências é n^{n-2} .

O número de árvores não-rotuladas sobre n nós, usualmente representada por T_n , é ainda mais difícil de se manusear. Nenhuma fórmula simples como o teorema de Cayley é conhecida para se determinar esse número. Nosso objetivo é obter uma ideia aproximada sobre quão grande é esse número.

Fica claro que o número de árvores não-rotuladas é menor que o número de árvores rotuladas.: toda árvore pode ser rotulada de muitas maneiras. Se desenharmos uma árvore não rotulada, podemos rotular de $n!$ maneiras. As árvores rotuladas que se obtêm dessa maneira não são necessariamente todas diferentes. Por exemplo, se a árvore for um estrela, então independentemente de como se permutam os rótulos das folhas, obtêm-se a mesma árvore rotulada.

□

Proposição 2.2.3 *O número de T_n de árvores não rotuladas sobre n vértices satisfaz*

$$\frac{n^{n-2}}{n!} \leq T_n$$

Demonstração: Se sabe que cada árvore rotulada pode ser rotulada de no máximo $n!$ maneiras. Como o número de árvores rotuladas é n^{n-2} , segue que:

$$\frac{n^{n-2}}{n!} \leq T_n$$

□

Perceba que a Proposição 2.2.4 representa apenas um limitante inferior para T_n . Como calcular então um limitante superior? Informalmente, como deve-se descrever uma árvore, se deseja-se apenas o "formato" dela e não se está preocupado com qual vértice possui qual rótulo?

Tome uma árvore G sobre n vértices, e especifique uma de suas folhas como sendo a raiz. Em seguida desenhe G no plano sem cruzar as arestas, isso pode sempre ser feito e quase sempre se desenham árvores dessa maneira. Agora imagine que as arestas da árvore são paredes perpendiculares ao plano. Começando na raiz, caminhe ao redor desse sistema de paredes, mantendo sempre a parede à sua direita. Chama-se o passeio sobre uma aresta de um "passo". Como existem $n - 1$ arestas e cada arestas tem dois lados, será dado $2(n - 1)$ passos antes de se retornar a raiz.

A cada vez que se dá um passo para longe da raiz (isto é, um passo de um pai para um de seus filhos), escreve-se 1, e cada vez que se dá um passo em direção a raiz, escreve-se 0. Dessa maneira se determina uma lista binária com $2(n - 1)$ elementos. Esse código é conhecido como *código planar* da árvore não rotulada. O código planar tem a seguinte propriedade: toda árvore não rotulada é univocamente determinada pelo seu código planar (REFERENCIA).

Proposição 2.2.4 *O número T_n de árvores não rotuladas satisfaz*

$$T_n \leq 4^{n-1}$$

Demonstração: Como o número de possíveis códigos planares é no máximo $2^{2(n-1)}$, obtem-se que o número de árvores não rotuladas é no máximo 4^{n-1}

□

Teorema 2.2.6 *O número T_n de árvores não rotuladas sobre n vértices satisfaz:*

$$\frac{n^{n-2}}{n!} \leq T_n \leq 4^{n-1}$$

Demonstração: Pela Proposição 2.2.4 se obtem a primeira parte da desigualdade e pela Proposição 2.2.5 se obtem a segunda parte.

□

2.2.3 Encontrando a melhor árvore

Por exemplo, se queremos realizar a ligação de computadores em rede a custo mínimo, que ligações deveremos fazer?

A resposta será uma árvore geradora, é claro. Mas qual?

Para resolver o problema, usaremos o **algoritmo de Kruskal**. Este algoritmo consiste em tomar a aresta de menor valor; se ela não forma ciclo, a acrescentamos à nossa árvore. Caso contrário, nós a desprezamos. Quando tivermos conseguido $n - 1$ arestas, nossa árvore estará pronta. Este método também é chamado de *Algoritmo Guloso*.

2.2.4 Caixeiro viajante

O *Problema do Caixeiro Viajante*, PCV, é um dos problemas mais estudados no campo da pesquisa operacional, mas até hoje não foi encontrado um algoritmo computacionalmente eficiente para resolvê-lo. Sua formulação é simples: dado um grafo completo valorado G , desejamos determinar o valor do menor ciclo hamiltoniano de G .

Por exemplo, uma máquina tem que perfurar um número de buracos em uma placa de circuitos impressos, e então retornar para o ponto inicial. Nesse caso, a quantidade importante é o tempo em que a máquina leva para mover a ponta perfurante de um buraco para o próximo, pois o tempo total que uma dada placa tem que gastar na máquina determina o número de placas que podem ser processadas em um dia. Portanto, se for tomado o tempo necessário para mover a ponta de um buraco para outro como o "custo" dessa aresta, precisa-se encontrar um ciclo com os buracos como vértices e com o custo mínimo.

O Problema do Caixeiro Viajante é muito mais difícil que o problema de se encontrar a árvore mais barata, e não existe algoritmo para resolvê-lo que seja tão simples, elegante e eficiente.

2.3 Emparelhamento de grafos

Emparelhamento de grafos está intimamente ligado a grafos bipartidos (já vistos anteriormente na seção de grafos). Porém, um novo conceito será introduzido, *emparelhamento perfeito*. O *emparelhamento perfeito* em um grafo bipartido onde seus dois subconjuntos independentes de nós tem o mesmo número n de nós, é um conjunto de arestas tal que todo nó é incidente a exatamente a uma delas.

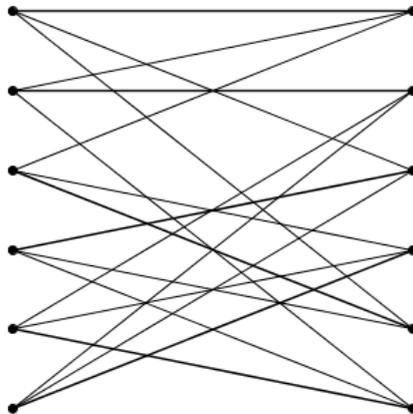


Figura 2.18: Um grafo bipartido com emparelhamento perfeito.

Se todo nó de um grafo bipartido tem o mesmo grau $d \geq 1$, então ele contém um emparelhamento perfeito (REFERÊNCIA).

Teorema 2.3.1 *Seja G um grafo bipartido, onde A e B são seus conjuntos independentes, isto é, ambos constituem a partição do conjunto de vértices. Então G tem um emparelhamento perfeito se e somente se $|A| = |B|$ e para qualquer subconjunto de k vértices de A existem pelo menos k vértices em B que estão conectados a pelo menos um deles.*

Demonstração: Será definido como um grafo bipartido "bom" se ele tem o mesmo número de nós esquerdos e direitos, e quaisquer k nós "esquerdos" estão conectados a pelo menos k nós "direitos".

Para um grafo com apenas dois nós, ser bom significa que esses dois nós estão conectados. Por conseguinte, para um grafo ter um emparelhamento perfeito significa que ele pode ser particionado em dois grafos "bons" com dois nós.

Nosso plano é particionar nosso grafo em duas partes "boas" e então particionar cada uma dessas em duas partes "boas" etc., até que obtenhamos partes "boas" com dois nós. Então as arestas que permanecem formam um emparelhamento perfeito. Selecione um nó $a \in A$ e $b \in B$ que estejam conectados por uma aresta; tome-os como a primeira parte, e os nós remanescentes como a outra parte. Não há problema com a primeira parte: ela é "boa". Mas a segunda parte pode não ser boa: ela pode ter algum conjunto S de k nós na esquerda conectados a menos de k nós na direita. No grafo original, esses k nós estavam conectados a pelo menos k nós em B ; isso somente pode acontecer se o k -ésimo dos tais nós for o nó b . Suponha que T denote o conjunto de vizinhos de S no grafo original. O que é importante lembrar que $|S| = |T|$.

Agora tentamos uma outra maneira de particionar o grafo: tomamos $S \cup T$ (juntamente com as arestas entre eles) como uma parte e o resto dos nós, como a outra. (Este resto não é vazio: o nó a pertence a ele, por exemplo.)

Vamos argumentar que ambas partes são "boas". Tome o primeiro grafo inicialmente, tome qualquer subconjunto de, digamos, j nós em S (o lado esquerdo do primeiro grafo). Como o grafo original era bom, eles estão conectados a pelo menos j nós, que estão todos em T pela definição de T .

Para o segundo grafo, segue igualmente que ele é "bom", se intercambiarmos "esquerda" e "direita". Isso completa a prova.

□

2.4 Coloração

Antes de começarmos a falar sobre a propriedade de coloração dos vértices, é importante abordarmos sobre conjunto independente dos vértices.

Um conjunto independente de vértices em um grafo G é o conjunto dos vértices tal que existem dois vértices adjacentes. Quando não podemos mais adicionar nenhum vértice sem que ele perca sua independência, dizemos que esse conjunto de vértices é maximal.

O número de independência $\alpha(G)$ é a cardinalidade do subconjunto independente máximo dos vértices do grafo.

Aplicações do conceito de conjunto independente surgem quando, por exemplo desejamos evitar duplicação de esforços.

Definição 2.4.1 *A propriedade da coloração dos vértices de um grafo é a atribuição de uma cor (geralmente representada por um número ou qualquer outro símbolo) para cada vértice do grafo de um modo que nenhuma cor ocorra em vértices adjacentes. Outro jeito de declarar essa condição é que cada aresta tenha extremidades de cores diferentes. Ainda pode-se dizer que o conjunto de vértices de dada cor é um conjunto independente de vértices.*

Se podemos colorir um grafo com k cores, dizemos que o grafo é k -colorvel. A respeito da coloração dos vértices em um grafo G , estamos interessados em encontrar o menor número possível de cores diferentes, ou seja o menor k . Este número é chamado de **número cromático** de G e é denotado por $\chi(G)$.

Teorema 2.4.1 *Para todo grafo G , tem-se que $\chi(G) \leq \Delta + 1$.*

Demonstração: Colorimos vértice por vértice. Cada vértice pode ser adjacente a, no máximo, Δ vértices. Podemos sempre encontrar uma cor com a qual colorir o vértice da vez.

□

Apresentaremos, sem demonstrar, um teorema clássico que reduz um pouco o limite do teorema acima.

Teorema 2.4.2 (Brooks - 1941). *Se G é um grafo conexo que não seja K_n e tal que $\Delta(G) \geq 3$, então $\chi(G) \leq \Delta(G)$.*

Teorema 2.4.3 *Um grafo G é bipartido se, e somente se, $\chi(G) = 2$.*

Demonstração: Basta fazer corresponder cada uma das partições independentes de G uma cor.

□

2.4.1 Teorema das quatro cores

O teorema nos diz que para colorir qualquer mapa, quatro cores sempre bastam.

Esse famoso teorema tem uma história de bem mais que um século. Foi levantado por Guthrie na Inglaterra em 1852. Por décadas ele foi jogado para cima por matemáticos como uma charada simples, porém enganadora, até que as dificuldades de se obter uma prova tornou-se aparente na década de 1870. uma prova errada foi publicada em 1879 por Kempe, e o problema foi considerado como resolvido por uma boa década antes que o erro fosse descoberto.

Após o colapso da prova de Kempe, por mais de um século muitos matemáticos, amadores e profissionais tentaram em vão resolver essa intrigante questão chamada *A Conjectura das Quatro Cores*. Várias outras provas erradas foram publicadas e depois refutadas.

Uma área nova toda da matemática, a Teoria dos Grafos, cresceu de tentativas de provar A Conjectura das Quatro cores. Finalmente, em 1976 os acontecimentos tomaram um rumo surpreendente: Appel e Haken dera uma prova da Conjectura das Quatro cores (que é por conseguinte o Teorema das Quatro cores agora), mas sua prova usou computadores muito intensamente para verificar um enorme número de casos. Ainda hoje, uso de computadores não poderia ser eliminado da prova (embora atualmente ele leve muito menos tempo que a primeira prova, em parte porque os computadores são mais rápidos, e em parte porque uma melhor arrumação da distinção de casos foi encontrada); ainda não temos uma prova matemática 'pura' desse teorema. Mas podemos usar resultados sobre grafos para prover o fato mais fraco de que cinco cores bastam para colorir todo mapa planar.

Teorema 2.4.4 *Todo grafo planar tem um ponto de grau no máximo 5.*

Demonstração: Portanto temos um grafo planar com n nós. Usamos indução sobre o número de nós, portanto assumimos que grafo planar com menos de n nós são 5-coloríveis. Também sabemos que nosso grafo tem um nó v com grau máximo 5.

Se v tem grau 4 ou menos, então o argumento é fácil: vamos remover v do grafo e colorir o grafo remanescente com cinco cores (o que é possível pois esse é um grafo planar com menos nós). O nó v tem no máximo quatro vizinhos, portanto podemos encontrar uma cor para v que seja diferente das cores de seus vizinhos, e estender a coloração para v .

Portanto o único caso difícil é quando o grau de v é exatamente 5. Sejam u e w dois vizinhos de v . Ao invés de simplesmente remover v , modificamos o grafo um pouco mais e usamos o lugar liberado, após a remoção de v , para juntar u e w em um único ponto, que chamamos de uw .

Agora esse grafo modificado é planar e tem n nós, portanto ele pode ser colorido com cinco cores, pela hipótese da indução. Se separarmos os dois pontos u e w novamente, obtemos uma

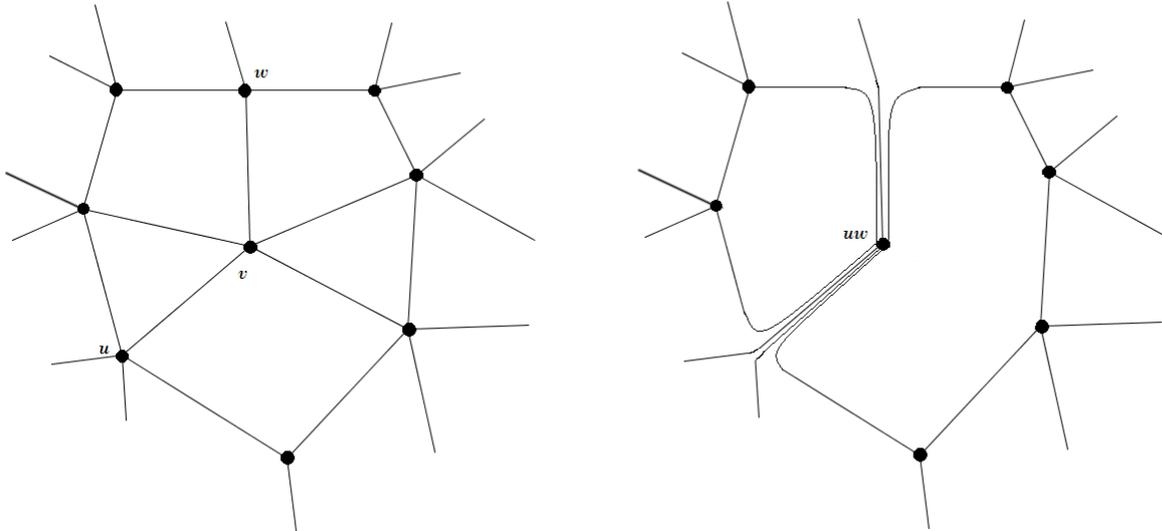


Figura 2.19: Prova do Teorema das 5-cores

coloração de todos os nós de G , exceto v com cinco cores. O que ganhamos com esse truque de juntar u e w é que nessa coloração eles têm a mesma cor! Portanto, muito embora v tenha cinco vizinhos, dois deles têm a mesma cor, daí uma das cinco cores não ocorrem de forma alguma entre os vizinhos. Podemos usar essa cor como a cor de v , completando a prova.

Advertência! Subestimamos uma dificuldade aqui. (Você pode ver quão fácil foi cometer erros nesses tipos de argumento!) Quando juntamos u e w , duas coisas ruins poderiam ter acontecido: (a) u e w estavam conectados por uma aresta, que após a junção tornou-se uma aresta conectando um nó a si próprio; (b) poderia ter havido um terceiro nó p conectado a ambos u e w , que após a junção tornou-se um nó conectado a uw por duas arestas. Não permitimos nenhuma dessas coisas!

O segundo problema (b) não é na verdade problema de forma alguma. Se obtemos duas arestas conectando o mesmo par de nós, poderíamos simplesmente ignorar um deles. O grafo permanece planar e na 5-coloração a cor de p seria diferente da cor comum de u e w , portanto quando os separamos, ambas as arestas conectando p a u e w conectariam nós com cor diferente.

Mas o primeiro problema (a) é sério. Não poderíamos simplesmente ignorar a aresta conectando uw a si próprio; na verdade, não há como u e w possam receber a mesma cor na coloração final, pois eles estão conectados por uma aresta!

O que vem nos resgatar é o fato de que podemos escolher um outro par u, w de vizinhos de v . Poderia acontecer que tenhamos esse problema com todo par? não, porque então todo par de vizinhos estaria conectado, e isso significaria um grafo completo com cinco nós, que sabemos que não é planar. Portanto podemos encontrar pelo menos um par u e w para o qual o procedimento acima funciona. Isso completa a prova do Teorema das Cinco Cores.

□

2.4.2 Aplicação de coloração

As aplicações de coloração aparecem quando precisamos repartir o conjunto de vértices em conjuntos de vértices independentes disjuntos.

Um dos exemplos clássicos é o problema de exames. A tabela abaixo mostra a alocação de alunos nos exames finais que eles devem prestar:

Alunos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Matemática	X							X				X			X	
Português	X			X							X					X
Inglês						X	X			X					X	
Geografia				X	X		X		X							
História			X							X				X		X
Física			X		X							X	X			
Química		X						X	X					X		
Biologia		X				X					X		X			

Figura 2.20: Tabela de alocação de alunos nos exames finais

Duas disciplinas só podem ter exames realizados simultaneamente se não houver alunos comuns. Vamos construir um grafo com os vértices que representam as disciplinas $\{M, P, I, G, H, F, Q, B\}$; dois vértices estarão ligados se tiverem um aluno em comum.

A figura 2.19 mostra uma partição dos vértices em dois conjuntos independentes disjuntos. Os exames podem ser realizados em 2 horários, um para $\{B, G, H, M\}$ e outro para $\{F, I, P, Q\}$.

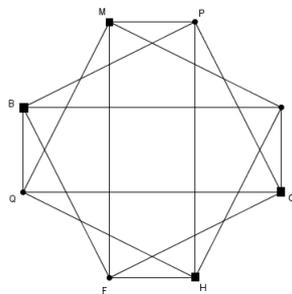


Figura 2.21: Dois conjuntos disjuntos independentes

Uma outra aplicação é a determinação de períodos de um sinal de trânsito. O desenho abaixo representa um cruzamento. As direções permitidas estão assinaladas por setas. Veja a figura 2.20.

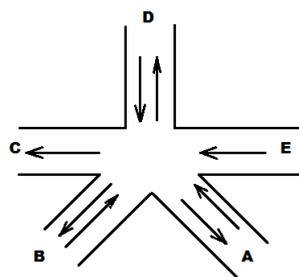


Figura 2.22: Cruzamento de vias de uma cidade

Como organizar o trânsito? Vamos formar um grafo de incompatibilidade. Os vértices serão as direções possíveis:

$$V = AB, AC, AD, BA, BC, BD, DA, DB, DC, EA, EB, EC, ED$$

Ligamos dois vértices sempre que as direções forem incompatíveis, pois poderia ocorrer um acidente, o que não queremos (por exemplo AD e EB). Veja a figura 2.21.

Observe que BA, DC e ED são compatíveis com todas as direções, sendo por isso vértices isolados. Uma coloração dos vértices corresponde a uma divisão em períodos. Poderíamos usar

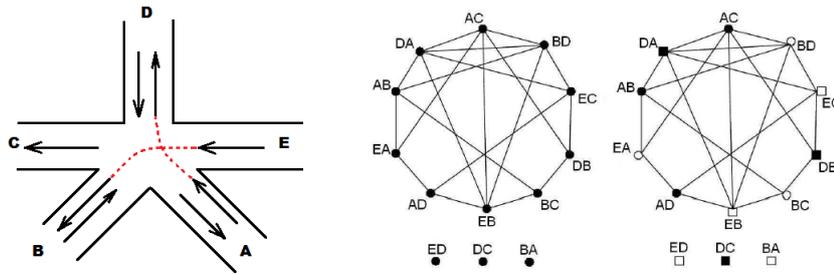


Figura 2.23: Cruzamento com direção incompatível à esquerda. Representação do grafo de direções incompatíveis.

13 cores, uma para cada direção, mas isso seria um desperdício de tempo. Como os vértices AC, BD, DA e EB formam um K_4 precisamos de pelo menos 4 cores. A partição em conjuntos independentes $\{AB, AC, AD\}$, $\{BC, BD, EA\}$, $\{BA, EB, EC, ED\}$, $\{DA, DB, DC\}$ mostra que de fato 4 cores (4 períodos) são suficientes, isto é, $\chi(G) = 4$.

Conclusão

Este relatório mostrou um panorama geral sobre os grafos, como os principais tipos de grafos existentes, definições e estrutura. A importância dos grafos também foram apresentadas, suas aplicações que envolvem o uso de Teoremas que facilitam a visão e resolução dos problemas aplicáveis na Teoria dos Grafos. Por exemplo, pudemos observar que o Problema do Caixeiro Viajante pode ser usado em um problema real. Uma placa de circuitos impressos deve ser perfurada no menor tempo possível com a finalidade de diminuir custos, para isso, a máquina deve encontrar o menor ciclo de menor custo, um ciclo hamiltoniano. Outras situações facilitadas pela Teoria dos Grafos também estão presentes na área de planejamento e comunicação. Entretanto, a fase reservada para simulações não pode ser realizada, pois o tempo destinado à capacitação teórica dos grafos foi além do planejado previamente no objetivo deste relatório.

Referências Bibliográficas

LOVÁSZ, László; PELIKÁN, József; VESZTERGOMBI, Katalin. Matemática Discreta. Tradução: Ruy J. G. B. de Queiroz. Rio de Janeiro: SBM, 2013.

MARCUS, Daniel. GRAPH THEORY, A Problem Oriented Approach. Washington: MAA Textbooks, 2008

JURKIEWICZ, Samuel. Grafos - Uma Introdução. Rio de Janeiro, 2009. (Apostila).

LUCCHESI, Cláudio Leonardo. Introdução à Teoria dos Grafos. Campinas, 1979. 152p.