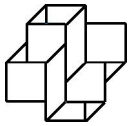

Laboratório Nacional de Computação Científica

Programa de Pós Graduação em Modelagem Computacional

Algoritmos Quânticos para o Problema do Isomorfismo de Grafos

Por

Edinelço Dalcumune

PETRÓPOLIS, RJ - BRASIL

MARÇO DE 2008

ALGORITMOS QUÂNTICOS PARA O PROBLEMA DO ISOMORFISMO DE GRAFOS

Edinelço Dalcumune

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM MODELAGEM COMPUTACIONAL

Aprovada por:

Prof. Renato Portugal, D.Sc

(Presidente)

Prof. Gilson Antônio Giraldi, D.Sc.

Prof. Celina Miraglia Herrera de Figueiredo, D.Sc.

PETRÓPOLIS, RJ - BRASIL
MARÇO DE 2008

Dalcumune, Edinelço

D138a Algoritmos quânticos para o problema do isomorfismo de grafos / Edinelço Dalcumune. Petrópolis, RJ. : Laboratório Nacional de Computação Científica, 2008.

xiii, 75 p. : il.; 29 cm

Orientador: Renato Portugal

Dissertação (M.Sc.) – Laboratório Nacional de Computação Científica, 2008.

1. Computação Quântica. 2. Isomorfismo de Grafos. 3. Interseção de Grupos. 4. Algoritmos Quânticos. I. Portugal, Renato. II. LNCC/MCT. III. Título.

CDD 004.1

"Sábio é aquele que conhece os limites da própria ignorância." (Sócrates)

Para meus pais e irmãos.

Agradecimentos

Agradeço ...

A meus pais e irmãos pelo apoio.

A todos os professores que ao longo da minha vida têm contribuído para minha formação, em especial o professor Renato Portugal pela orientação e amizade.

Aos colegas do grupo de computação quântica do LNCC.

Ao LNCC, a seus professores e funcionários.

A todos os colegas de graduação e às amizades que o mestrado tornou possíveis.

A FAPERJ pelo apoio financeiro.

Resumo da Dissertação apresentada ao LNCC/MCT como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMOS QUÂNTICOS PARA O PROBLEMA DO ISOMORFISMO DE GRAFOS

Edinelço Dalcumune

Março, 2008

Orientador: Renato Portugal, D.Sc

O problema do isomorfismo de grafos possui aplicações em diversas áreas da ciência. Tal problema não possui uma solução eficiente para o seu caso geral. No presente trabalho, apresentamos os conceitos básicos em teoria de grupos, teoria dos grafos e mecânica quântica. Apresentamos o problema do subgrupo oculto e uma conhecida redução polinomial do problema do isomorfismo de grafos no seu caso geral para o problema do subgrupo oculto sobre o grupo simétrico. Utilizamos um método que reduz o problema do isomorfismo de grafos para o problema de interseção de grupos. Este método utiliza resultados da computação quântica e da teoria dos grupos solúveis, nos permitindo obter uma solução eficiente através de um algoritmo quântico para o problema do isomorfismo de grafos para uma classe particular de grafos.

Abstract of Dissertation presented to LNCC/MCT as a partial fulfillment of the requirements for the degree of Master of Sciences (M.Sc.)

QUANTUM ALGORITHMS FOR THE GRAPH ISOMORPHISM PROBLEM

Edinelço Dalcumune

March, 2008.

Advisor: Renato Portugal, D.Sc

The graph isomorphism problem has applications in several areas of science. This problem has not an efficient solution to its general case. In this work, we present the basic concepts of group theory, graph theory and quantum mechanics. We introduce the hidden subgroup problem and a known polynomial reduction of the graph isomorphism problem in its general case to the hidden subgroup problem on the symmetric group. We use a method that reduces the graph isomorphism problem to the group intersection problem. This method combines results from quantum computing and solvable group theory providing a efficient solution through a quantum algorithm to the graph isomorphism problem for the particular class of graphs.

Sumário

Lista de Figuras

Figura

Lista de Siglas e Abreviaturas

- $\langle g_1, \ldots, g_k \rangle$: Conjunto gerado por $g_1, \ldots, g_k$
- $(G : H)$: índice de H em G
- G/H: Grupo quociente de G por H
- $[H : K]$: Comutador dos subgrupos H e K
- $\mathcal{P}(S)$: Conjunto das bijeções de S em S
- S_n: Grupo Simétrico de grau n
- $H^{\perp}$: Subgrupo ortogonal de G
- ω_t: Raiz complexa t-ésima da unidade, $\omega_t = e^{\frac{2\pi i}{t}}$
- log: Logaritmo na base 2
- $O(\cdot)$: Complexidade Computacional no pior caso; diz-se $f(x) = O(g(x))$ se $\exists\ C$, x_0 tal que $|f(x)| < Cg(x)$, $\forall\ x > x_0$
- $\mathrm{Iso}(\Gamma_1, \Gamma_2)$: Conjunto de isomorfismos entre um grafo Γ_1 e um grafo Γ_2
- $\mathrm{Aut}(\Gamma)$: Grupo de automorfismos de um grafo Γ
- $P_1 \propto_p P_2$: O problema P_1 é redutível polinomialmente a um problema P_2
- PIG: Problema do Isomorfismo de Grafos
- PAG: Problema do Automorfismos de Grafos
- $\lfloor \cdot \rfloor$: Maior número inteiro que seja menor ou igual a $\cdot$
- $\lceil \cdot \rceil$: Menor número inteiro que seja maior ou igual a $\cdot$
- $(\cdot)^*$: Complexo conjugado
- $(\cdot)^T$: Transposto
- $(\cdot)^{\dagger}$: Transposto conjugado
- $|\psi\rangle$: Vetor (em notação de Dirac), também chamado *ket*

- $\langle\psi|$: Vetor dual (em notação de Dirac), também chamado *bra*
- $\langle\varphi|\psi\rangle$: Produto escalar entre $|\varphi\rangle$ e $|\psi\rangle$
- $|\varphi\rangle \otimes |\psi\rangle$: Produto tensorial entre $|\varphi\rangle$ e $|\psi\rangle$
- $|\varphi\rangle\langle\psi|$: Produto tensorial entre $|\varphi\rangle$ e $|\psi\rangle$
- $V \otimes W$: Produto tensorial entre V e W
- $|\psi\rangle^{\otimes k}$: Produto tensorial de $|\psi\rangle$ por ele mesmo k vezes
- PSO: Problema do Subgrupo Oculto

Capítulo 1

Introdução

Estruturas químicas podem ser descritas em termos da teoria de grafos; átomos são interpretados como vértices e as ligações são interpretadas como arestas. O problema de isomorfismo de grafos tem atraído muitos pesquisadores devido a inúmeras aplicações práticas em várias áreas da ciência, como por exemplo, química [Liu e Klein (1991), Raymond e Willett (2002), Valiente (2002)], biologia computacional [Valiente (2002), Gan et al. (2003)], entre outras [Köbler et al. (1993), Fortin (1996)]. Além, é claro, da computação teórica, sendo assim um grande desafio matemático. Este problema pertence à classe dos problemas **NP**, mas não se sabe se está nas classes **P** ou **NP-Completa** [Papadimitriou (1994), Kaye et al. (2007)]. Existem fortes evidências de que não pertença a classe **NP-Completa**.

O Problema de isomorfismo de grafos pode ser visto da seguinte forma: Sejam Γ_1 e Γ_2 dois grafos não-orientados sobre os vértices $V = \{v_1, \ldots, v_n\}$. São Γ_1 e Γ_2 isomorfos? Ou seja, existe uma função bijetora $\varphi : V \to V$ tal que a aresta (v_i, v_j) estará contida em Γ_1 se e somente se $(\varphi(v_i), \varphi(v_j))$ estiver contida em Γ_2?

Não é conhecido na literatura um algoritmo clássico eficiente para o problema do isomorfismo de grafos para o caso geral. O melhor algoritmo conhecido para tal problema "roda"em tempo $e^{O(\sqrt{n \log n})}$ [Babai (1980), Babai e Luks (1983), Zemlyachenko et al. (1985)]. Dizemos que tal algoritmo é subexponencial no número de vértices dos grafos de entrada. Existem algoritmos clássicos eficientes para algumas classes de grafos, ou seja, algoritmos polinomiais no número de vértices dos

grafos de entrada. Contudo, o problema geral continua desafiando a comunidade científica.

A Computação Quântica [Nielsen e Chuang (2000), Kaye et al. (2007)] é uma área de pesquisa recente que utiliza elementos de três áreas importantes: Matemática, Física e Computação. O objetivo da Computação Quântica é estudar métodos para processar, transmitir e armazenar informações contidas em estados quânticos. Os seguintes questionamentos nos motivam a estudar Computação Quântica: Existem problemas que os computadores quânticos podem resolver mais rapidamente do que os clássicos? O que faz os computadores quânticos serem mais eficientes do que os clássicos?

A Computação Quântica nasce no início da década de 1980 quando Feynman apontou a existência de grandes dificuldades para simular sistemas quânticos em computadores clássicos, sugerindo que a construção de computadores baseados nos princípios da Mecânica Quântica evitaria tais dificuldades [Feynman (1982)]. Os argumentos de Feynman estimularam David Deutsch a generalizar o modelo mais fundamental da Computação Clássica, a saber, a máquina de Turing, para o seu equivalente quântico num trabalho histórico [Deutsch (1985)]. Posteriormente, generalizou também o modelo de circuitos baseados em portas lógicas. Operadores unitários tomaram o lugar das usuais portas lógicas AND, OR e NOT. Em 1994, [Shor (1994)] construiu com base nos trabalhos de [Deutsch (1985)] e [Simon (1994)] um algoritmo quântico que pode fatorar inteiros grandes exponencialmente mais rápido do que qualquer método clássico conhecido. Esse algoritmo permite a quebra dos principais códigos de criptografia usados atualmente, como RSA, Diffie-Hellman e ElGamal [Koblitz (1998)], caso um computador quântico de tamanho razoável esteja disponível.

A maioria dos algoritmos quânticos com ganho exponencial em relação aos seus equivalentes clássicos, tais como, o algoritmo de Shor, pode ser considerada como um caso particular do chamado Problema do Subgrupo Oculto (PSO). O PSO consiste em achar os geradores de um subgrupo H de um determinado grupo

finito G com uma função oráculo f definida de G para um conjunto finito X tal que $f(a) = f(b)$ se, e somente se, $aH = bH$ para todo a,$b \in G$.

No presente trabalho, veremos que o Problema do Isomorfismo de Grafos pode ser visto como um Problema do Subgrupo Oculto não abeliano sobre o grupo simétrico S_n, como foi mostrado por [Beals (1997), Ettinger e Høyer (1999), Jozsa (2000), Ahn (2002), Lomont (2004)], onde devemos encontrar um conjunto de geradores para o grupo de automorfismos do grafo, que é um subgrupo do grupo simétrico S_n. Utilizaremos o Problema de Interseção de Grupos de permutações [Hoffman (1979), Fenner e Zhang (2005)], juntamente com resultados da Computação Quântica para determinar, dado o estágio atual de desenvolvimento da Computação Quântica a existência de algoritmos quânticos eficientes para o Problema do Isomorfismo de Grafos em muitas classes de grafos ou não. Mostraremos que para a classe de grafos A_n esse método empregado produz um algoritmo quântico eficiente, com algumas restrições. Mostrando assim, aplicações da Computação Quântica ao Problema do Isomorfismo de Grafos.

No capítulo 2, apresentaremos uma revisão de tópicos de teoria de grupos necessários ao entendimento do restante da dissertação. Inclui-se nessa revisão conceitos sobre o grupo simétrico e grupos solúveis. No capítulo 3, daremos as noções básicas sobre teoria dos grafos. Ainda no capítulo 3, enunciaremos os problemas do isomorfismo e automorfismo de grafos, bem como, relações entre os dois problemas. No capítulo 4, apresentaremos uma descrição matemática do problema do isomorfismo de grafos. Definiremos o problema de interseção de grupos e sua relação com o problema do isomorfismo de grafos. Daremos exemplos de grafos, incluindo classes onde o problema do isomorfismo de grafos é resolvido eficientemente num computador clássico. Ainda no capítulo 4, mostraremos a complexidade de tal problema. No capítulo 5, faremos uma revisão sobre mecânica quântica. Apresentaremos também o problema do subgrupo oculto e o algoritmo quântico para o caso abeliano. Finalmente, no capítulo 6, apresentaremos aplicações da computação quântica para o problema do isomorfismo de grafos, incluindo uma visão

do problema do isomorfismo de grafos como um problema do subgrupo oculto. Ainda neste capítulo, daremos um algoritmo quântico eficiente usando o problema de interseção de grupos para resolver o problema do isomorfismo de grafos para uma classe particular de grafos, como foi dito no parágrafo anterior.

Capítulo 2

Teoria de Grupos

Neste capítulo, apresentamos uma revisão de tópicos de Teoria de Grupos necessários ao entendimento do restante da dissertação. Seguimos a linguagem clássica de Álgebra [Herstein (1986), Garcia e Lequain (2001)].

2.1 Teoria Básica de Grupos

Nesta seção veremos algumas definições básicas e exemplos da Teoria de Grupos que serão necessários para o entendimento do restante do trabalho.

Definição 1 Um conjunto não-vazio G é dito um **grupo** se em G existe uma operação $(*)$ definida tal que:

(i) $a, b \in G$ implica que $a * b \in G$.

(ii) Dados $a, b, c \in G$, então $a * (b * c) = (a * b) * c$.

(iii) $\exists\ e \in G; a * e = e * a = a, \forall\ a \in G$.

(iv) $\forall\ a \in G, \exists\ b \in G; a * b = b * a = e\ (b = a^{-1})$.

Por simplicidade denotaremos $a * b$ por ab para todo $a, b \in G$.

Definição 2 Se G é um grupo tal que $ab = ba, \forall\ a, b \in G$, nós dizemos que G é um grupo **abeliano**.

Definição 3 Seja G um grupo. Um subconjunto não-vazio H de G é um **subgrupo** de G (denotamos $H \leq G$) quando, com a operação de G, o conjunto H é um grupo.

Proposição 1 Seja H um subconjunto não-vazio do grupo G. Então H é um subgrupo de G se e somente se as duas condições seguintes são satisfeitas:

1) $h_1 h_2 \in H$, $\forall\ h_1,\ h_2 \in H$.

2) $h^{-1} \in H$, $\forall\ h \in H$.

Lema 1 Seja G um grupo e H um subconjunto finito não-vazio de G fechado sobre a operação em G. Então H é um subgrupo de G.

Demonstração: Vide [Herstein (1986)].

■

Corolário 1 Se G é um grupo finito e H é um subconjunto não-vazio de G fechado sobre a operação em G, então H é um subgrupo de G.

Definição 4 Seja G um grupo.

(i) A ordem de G (denotamos $|G|$) é a cardinalidade do conjunto G.

(ii) Se $a \in G$, então a ordem do elemento a (denotamos $|a|$ ou $o(a)$) é o menor inteiro positivo n tal que $a^n = e$ (Se n não existir, dizemos que $o(a) = \infty$).

(iii) O expoente de G, $exp(G)$ é o menor inteiro positivo m tal que $a^m = e$ para todo $a \in G$ (Se m não existir, dizemos que $exp(G) = \infty$).

Dizemos que um conjunto de elementos $g_1, ..., g_k$ **gera** um grupo G se cada elemento de G puder ser escrito como um produto de elementos (e seus inversos) da lista $g_1, ..., g_k$ e denota-se $G = \langle g_1, ..., g_k \rangle$.

O seguinte teorema é importante pois tem implicações em algoritmos quânticos.

Teorema 1 Todo grupo G de ordem finita $|G| > 1$ admite um conjunto gerador de tamanho no máximo $\lfloor \log_2 |G| \rfloor$.

Demonstração: Vide [Hoffman (1979)].

■

Definição 5 Um grupo G é dito **cíclico** se existe algum elemento $g \in G$ tal que $\langle g \rangle = G$. Neste caso, g é dito gerador de G.

O grupo $\mathbb{Z}$ com a operação de adição é cíclico. Os grupos $\mathbb{Z}_n$ com a operação de adição módulo n são cíclicos. Note que se G é cíclico, então G é abeliano.

Sendo $(G_1, *_1)$ e $(G_2, *_2)$ dois grupos. No conjunto $G_1 \times G_2$ defina a operação

$$(g_1, g_2) \cdot (g_1^{'}, g_2^{'}) = (g_1 *_1 g_1^{'}, g_2 *_2 g_2^{'})$$

Logo, $(G_1 \times G_2, \cdot)$ é um grupo chamado **produto direto** de G_1 com G_2. Mais geralmente, dados grupos $(G_1, *_1), \ldots, (G_n, *_n)$, defina a noção de **produto direto** $G_1 \times G_2 \times \ldots \times G_n$.

Outro importante teorema que tem implicações em algoritmos quânticos é o seguinte.

Teorema 2 (Teorema Fundamental sobre Grupos Abelianos Finitos) Um grupo abeliano finito é o produto direto de grupos cíclicos.

Demonstração: Vide [Herstein (1986)].

■

Seja G um grupo e H um subgrupo de G. Sendo $x \in G$, definimos uma **classe lateral à esquerda** de H em G como sendo o conjunto $xH = \{xh | h \in H\}$. Em particular, H é uma classe lateral do elemento e à esquerda. Analogamente, definimos uma **classe lateral à direita** de H em G como sendo o conjunto $Hx = \{hx | h \in H\}$.

Duas classes laterais à esquerda (direita) de H em G são disjuntas ou são iguais. Portanto, podemos escrever G como uma união disjunta de classes laterais, ou seja, $G = H \cup x_1 H \cup x_2 H \cup ... \cup x_r H$.

Definição 6 A cardinalidade do conjunto das classes laterais à esquerda (direita) é o **índice** de H em G; ele será denotado por $(G : H)$.

Tomando um conjunto de representantes das classes laterais à esquerda de H em G (um elemento de cada classe lateral) vemos que a cardinalidade deste conjunto é igual a $(G : H)$.

Proposição 2 Todas as classes laterais de H em G têm a mesma cardinalidade, igual à cardinalidade de H.

Teorema 3 (Teorema de Lagrange) Sejam G um grupo finito e H um subgrupo de G. Então $|G| = |H|(G : H)$; em particular, a ordem e o índice de H dividem a ordem de G.

Suponha que G é um grupo. Dois elementos a e b de G são chamados **conjugados** se existe um elemento $g \in G$ com $gag^{-1} = b$. Além disso, dado qualquer subconjunto S de G (S não necessariamente um subgrupo), definimos um subconjunto T de G como conjugado a S se e somente se existe algum $g \in G$ tal que $T = gSg^{-1}$.

Seja G um grupo e seja H um subgrupo de G. A operação de G induz de maneira natural uma operação sobre o conjunto de classes laterais à esquerda de H em G, isto é, a operação

$$(xH, yH) \to xyH$$

é bem definida, no sentido de não depender da escolha dos representantes x e y.

Definição 7 Um subgrupo N de G é um subgrupo **normal** de G se $g^{-1}Ng = N$ para todo $g \in G$. Denotamos por $N \lhd G$. Neste caso, as classes laterais à esquerda de N são iguais às classes laterais à direita de N; vamos chamá-las de classes laterais de N.

Os grupos $\{e\}$, G são subgrupos normais de G. Dados dois grupos G e H, se $(G : H) = 2$, então $H \lhd G$.

Teorema 4 Sejam G um grupo e H um subgrupo normal de G. Então o conjunto das classes laterais, com a operação induzida de G, é um grupo.

Definição 8 Sejam G um grupo e H um subgrupo normal de G. O grupo de suas classes laterais com a operação induzida de G é o **grupo quociente** de G por H; ele será denotado por G/H ou por $\frac{G}{H}$.

Seja D_n o grupo não-abeliano chamado grupo **diedral** ou grupo de simetrias do polígono regular de n vértices, que é formado pelas rotações e reflexões do plano que preservam o polígono. Formalmente, temos

Definição 9 Se denotarmos por ρ uma rotação do ângulo $2\pi/n$ e por σ uma reflexão em relação a um eixo de simetria do polígono, então o grupo diedral é dado por

$$D_n = \langle \rho, \sigma \rangle = \{e, \rho, ..., \rho^{n-1}, \sigma, \sigma\rho, ..., \sigma\rho^{n-1}\}.$$

Este grupo possui uma apresentação com geradores ρ e σ que estão relacionados como a seguir

$$\rho^n = \sigma^2 = e, \rho\sigma = \sigma\rho^{-1}.$$

O conjunto formado pelas rotações formam um subgrupo abeliano de D_n que denotaremos por C_n.

2.2 O Grupo Simétrico

Definição 10 Seja S um conjunto qualquer. Considere o conjunto $\mathcal{P}(S) = \{f : S \to S | f \text{ é uma bijeção}\}$. O conjunto $\mathcal{P}(S)$ com a operação composição de funções é um grupo, não-abeliano em geral. Quando S tem um número finito de elementos, digamos n, então $\mathcal{P}(S)$ tem um nome especial. Ele será chamado o **grupo simétrico de grau** n ou **grupo das permutações** de n letras e será denotado por S_n. Temos ainda que a ordem de S_n é $n!$.

É muito importante estudar os grupos S_n e seus subgrupos devido ao teorema abaixo.

Teorema 5 (Teorema de Cayley) Seja G um grupo finito de ordem n. Então G é isomorfo a um subgrupo de S_n.

Se uma função $f : S \to S$ é uma bijeção, então f pode ser representada por uma permutação do conjunto S. Vamos usar a notação de produtos de ciclos para permutações, como definimos abaixo. Composições de permutações são lidas a partir da direita para a esquerda, isto é, se $\pi, \tau \in S_n$, então $\pi\tau$ é a permutação que primeiro aplica τ e então π.

Definição 11 Uma permutação $\sigma \in S_n$ é chamada de **r-ciclo** se existem elementos distintos $i_1, i_2, ..., i_r \in \{1, 2, ..., n\}$ tais que $\sigma(i_1) = i_2, \sigma(i_2) = i_3, ..., \sigma(i_{r-1}) = i_r, \sigma(i_r) = i_1$, e tais que $\sigma(i_j) = i_j$, $\forall\ j \in \{1, 2, ..., n\}\backslash\{i_1, i_2, ..., i_r\}$; tal r-ciclo será denotado por $(i_1...i_r)$; o número r é chamado o **comprimento** do ciclo. Os 2-ciclos são também chamados de **transposições**.

Sendo $\sigma \in S_n$ um r-ciclo e $\tau \in S_n$ um s-ciclo, dizemos que as permutações σ e τ são **disjuntas** se nenhum elemento de $\{1, 2, ..., n\}$ é movido simultaneamente por ambas, isto é, $\forall\ a \in \{1, 2, ..., n\}$, se $\sigma(a) \neq a$ então $\tau(a) = a$ e vice-versa, se $\tau(a) \neq a$ então $\sigma(a) = a$.

Lema 2 Toda permutação em S_n é o produto de ciclos disjuntos.

Não é difícil ver que, se σ, $\tau \in S_n$ são ciclos disjuntos, então $\sigma\tau = \tau\sigma$. Também pode-se mostrar que a ordem de um r-ciclo $\tau \in S_n$ é igual a r. Além disso, sendo $\sigma_1, \ldots, \sigma_k \in S_n$ ciclos disjuntos de comprimentos $r_1, \ldots, r_k$, respectivamente, então o produto $\sigma_1...\sigma_k$ tem ordem igual a $\text{mmc}(r_1, \ldots, r_k)$.

Temos a seguinte proposição,

Proposição 3

(i) Todo elemento de S_n é um produto de transposições, isto é, $S_n = \langle\{\text{transposições}\}\rangle$.

(ii) $S_n = \langle (12), (13), \ldots, (1n) \rangle$.

(iii) $S_n = \langle (12), (23), \ldots, (n-1n) \rangle$.

Demonstração: Vide [Garcia e Lequain (2001)].

■

Como consequência, temos que as permutações (12) e $(12 \ldots n)$ geram o grupo simétrico S_n, ou seja, $S_n = \langle (12), (12 \ldots n) \rangle$.

2.3 Grupos Solúveis

Informalmente, pode-se pensar nos grupos solúveis como "aproximadamente abelianos". Por exemplo, podemos considerar que um grupo G está "perto" de ser abeliano se ele contém um subgrupo normal H tal que tanto H quanto o quociente G/H são abelianos (um tal grupo diz-se **metabeliano**). Generalizaremos esta idéia.

Primeiramente, vamos dar algumas definições.

Definição 12 Seja G um grupo. Uma **série subnormal** de G é uma cadeia de subgrupos

$$G = G_0 \rhd G_1 \rhd G_2 \rhd \ldots \rhd G_n = \{e\} \tag{2.1}$$

onde G_{i+1} é um subgrupo normal de G_i, para $i = 0, 1, \ldots, n-1$.

Os grupos quocientes da série (2.1) são grupos G_i/G_{i+1}, para $i = 0, 1, \ldots, n-1$.

O **refinamento** da série subnormal (2.1) é uma série subnormal obtida a partir de (2.1) pela inserção de alguns (possivelmente nenhum) subgrupos. O refinamento é **próprio** se algum subgrupo distinto dos já existentes é inserido na série.

A série subnormal (2.1) é uma **série de composição** se ela não admite um refinamento próprio.

Definição 13 Seja G um grupo. O **comutador** de dois elementos $h, k \in G$ é definido como $[h, k] := hkh^{-1}k^{-1}$. O comutador de dois subgrupos $H, K \leq G$ é definido como o subgrupo de G gerado por todos os comutadores $[h, k]$ onde $h \in H$ e $k \in K$, isto é,

$$[H, K] := \langle hkh^{-1}k^{-1} | h \in H, k \in K \rangle.$$

Em particular, o grupo de comutadores $G^{'}$ é igual a $[G, G]$.

Definição 14 Seja G um grupo. A série de comutadores (série derivada)

$$G \geq G^{'} \geq G^{''} \geq G^{'''} \geq ... \geq G^{i} \geq ...$$

de G é definida indutivamente por

$$G^0 := G, \qquad G^{i+1} := [G^i, G^i].$$

Isto é, G^{i+1} é o subgrupo dos comutadores do grupo G^i, para cada $i = 0, 1, 2, \ldots$.

Proposição 4 Seja G um grupo. As seguintes condições são equivalentes:

(i) O grupo G possui uma série subnormal cujos grupos quocientes são abelianos.

(ii) Existe um inteiro n tal que $G^n = \{e\}$.

No caso de G ser finito, elas são também equivalentes a:

(iii) O grupo G possui uma série de composição cujos grupos quocientes são abelianos (e portanto, são cíclicos de ordem prima).

Definição 15 Um grupo G é dito **solúvel** se ele satisfaz as condições equivalentes da proposição 4.

Definição 16 Uma série de subgrupos de um grupo G

$$G = G_0 > G_1 > G_2 > \ldots > G_r = \{e\} \tag{2.2}$$

é dita uma **série normal** de G se temos $G_i \lhd G$, para cada $i = 0, 1, \ldots, r$. Em particular, uma série normal de G é uma série subnormal de G.

Pode-se provar que todo grupo abeliano, todo p-grupo finito e todo grupo finito de ordem ímpar são solúveis. No entanto, se $n \geq 5$, então o grupo simétrico S_n não é solúvel.

O grupo diedral (definição 9) é solúvel, de fato,

Teorema 6 Seja D_n o grupo diedral de ordem $2n$. Então D_n é solúvel.

Demonstração: Seja C_n o grupo das rotações do polígono. Então o grupo D_n possui a seguinte série subnormal com grupos quocientes abelianos

$$\{1\} \lhd C_n \lhd D_n \tag{2.3}$$

De fato, primeiramente vemos que $\{1\} \lhd C_n$ e como $|D_n|/|C_n| = 2$, ou seja, $(D_n : C_n) = 2$, temos que $C_n \lhd D_n$. Além disso, $C_n/\{1\}$ é abeliano e D_n/C_n tem ordem $p = 2$, portanto é abeliano.

■

Teorema 7 Seja G um grupo.
1) Seja H um subgrupo de G. Se G é solúvel, então H é solúvel.
2) Seja H um subgrupo normal de G. Então, o grupo G é solúvel se e somente se os grupos H e G/H são solúveis.

Demonstração: Vide [Garcia e Lequain (2001)].

■

Nós dizemos que uma família de grupos abelianos é **suavemente abeliano** se cada grupo na família pode ser expressado como produto direto de um subgrupo cujo expoente (veja Definição 4) é limitado por uma constante e um subgrupo de tamanho polilogarítmico na ordem do grupo.

Uma família de grupos solúveis é **suavemente solúvel** se o comprimento de cada série derivada é limitada por uma constante e a família de todos os grupos quocientes G^i/G^{i+1} é suavemente abeliano. O termo suavemente solúvel foi introduzido por [Friedl et al. (2003)].

Capítulo 3

Teoria dos Grafos

Serão descritos neste capítulo alguns conceitos básicos da Teoria dos Grafos, bem como a definição do Problema do Isomorfismo de Grafos e o Problema do Automorfismo de Grafos, além de algumas relações entre os dois problemas. A apresentação cobre o necessário para a compreensão dos problemas e algoritmos discutidos nos próximos capítulos. Também introduziremos aqui a terminologia e notação sobre grafos utilizados no restante da dissertação [Szwarcfiter (1984), Harary (1969)].

3.1 Noções Básicas de Grafos

Acredita-se que a Teoria dos Grafos teve início quando o famoso matemático suiço Leonhard Euler publicou um artigo sobre o problema das "Sete Pontes de Königsberg"em 1736. Na cidade de Königsberg (atualmente Kaliningrado, Rússia) havia um conjunto de sete pontes que cruzavam o rio Pregel (ver Figura 3.1). Elas conectavam duas ilhas entre si e as ilhas com as margens. Durante muito tempo os habitantes daquela cidade perguntavam-se se era possível cruzar as sete pontes numa caminhada contínua sem passar duas vezes por qualquer uma delas. Euler mostrou que a travessia proposta não era possível [Euler (1741), Alexanderson (2006)]. Mas, não se restringiu a resolver apenas este problema. Sua solução foi construída pensando o problema de forma mais abrangente, inaugurando assim a Teoria de Grafos. A Figura 3.2 mostra um grafo para o problema.

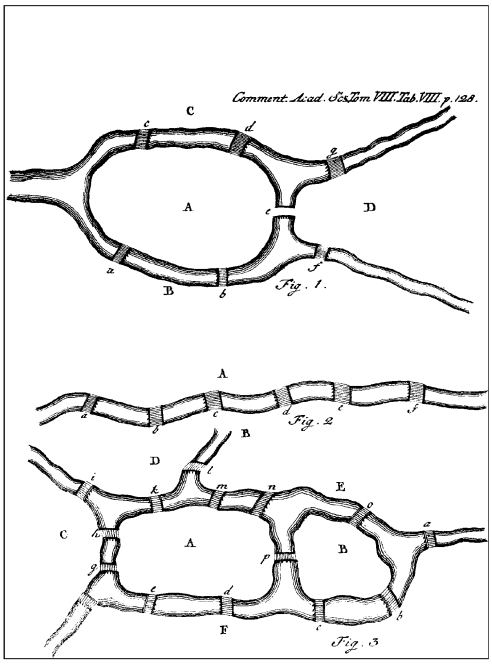

Figura 3.1: As gravuras ilustrando o documento de Euler de 1736 sobre as pontes de Königsberg. Fonte: Alexanderson (2006)

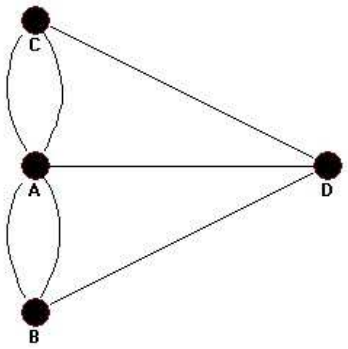

Figura 3.2: Grafo do Problema das Sete Pontes de Königsberg.

Um grafo (não-orientado) Γ é uma estrutura (V, E), onde V é um conjunto finito não-vazio e E é uma relação $E \subseteq V \times V$ do grafo. Os elementos de V são os **vértices** e os de E são as **arestas** de Γ, respectivamente. Cada aresta $e \in E$ será denotada pelo par de vértices $e = (v, w)$ que a forma. Neste caso, os vértices v, w são os extremos (ou extremidades) da aresta e, sendo denominados **adjacentes**. A aresta e é dita **incidente** a ambos v, w. O número de arestas incidentes no vértice v é chamado o **grau** de v. Duas arestas que possuem um extremo comum

são chamadas de **arestas adjacentes**.

Dizemos que um grafo $D = (V, A)$ é um **grafo orientado** (dígrafo) se V é um conjunto de vértices e A um conjunto de pares ordenados de vértices, comumente chamados de **arcos** ou **arestas direcionadas**.

Dizemos que $\Gamma' = (V', E')$ é um **subgrafo** do grafo $\Gamma = (V, E)$ se Γ' é um grafo e $V' \subseteq V$, $E' \subseteq E$.

Uma sequência de vértices $v_1, \ldots, v_k$ tal que $(v_j, v_{j+1}) \in E$, $1 \leq j < k-1$, é denominada um **caminho**. Um caminho de k vértices é formado por $k-1$ arestas que definem o comprimento do caminho. Se todos os vértices em um caminho são distintos, o caminho recebe o nome de **simples** ou **elementar**.

Um **ciclo** é um caminho $v_1, \ldots, v_k, v_{k+1}$, onde $v_1 = v_{k+1}$, $k \geq 3$. Se o caminho $v_1, \ldots, v_k$ for simples o ciclo também é dito simples. O grafo que não possui ciclos é dito **acíclico**.

Um grafo é dito **conexo** quando existe um caminho ligando cada par de vértices distintos, caso contrário o grafo é dito **desconexo**.

Seja S um conjunto e $S' \subseteq S$. Diz-se que S' é maximal em relação a uma certa propriedade P, quando S' satisfaz a propriedade P e não existe subconjunto $S'' \supset S'$, que também satisfaz P. Ou seja, S' não está propriamente contido em nenhum subconjunto de S que satisfaça P.

Denominam-se **componentes conexas** de um grafo Γ aos subgrafos maximais de Γ que sejam conexos. A propriedade P, neste caso, é equivalente a ser conexo.

Assumiremos que o conjunto de vértices seja dado por $V = \{1, \ldots, n\}$. Permutações dos vértices são então dadas por permutações no grupo simétrico S_n. Seja $\pi \in S_n$ uma permutação, escreveremos $\pi\Gamma$ para o grafo que é obtido pela permutação dos vértices de Γ com π, isto é, $\pi\Gamma = \pi(V, E) = (V, \pi E) = (V, \{(\pi(i), \pi(j)) | (i, j) \in E\})$.

Um grafo pode ser visualizado através de uma **representação geométrica**, na qual seus vértices correspondem a pontos distintos do plano em posições arbi-

trárias, enquanto que a cada aresta (v, w) é associada uma linha arbitrária unindo os pontos correspondentes a v, w. Para maior facilidade de exposição, é usual identificar um grafo com a sua representação geométrica. Isto é, no decorrer do texto será utilizado o termo **grafo**, significando também a sua representação geométrica.

3.2 Problema do Isomorfismo de Grafos

A partir do que já foi dito é possível formular o seguinte problema. Dadas duas representações geométricas, correspondem elas a um mesmo grafo? Em outras palavras, é possível fazer coincidir, respectivamente, os pontos de duas representações geométricas, de modo a preservar adjacência (ou seja, de modo a fazer também coincidir as arestas)? Responderemos estas perguntas logo abaixo. Mas primeiro, definiremos o que é isomorfismo de grafos.

Definição 17 (Isomorfismo de Grafos) Dois grafos (orientados ou não-orientados) $\Gamma_1 = (V_1, E_1)$, $\Gamma_2 = (V_2, E_2)$ são chamados isomorfos se existe uma permutação τ tal que $\tau\Gamma_1 = \Gamma_2$, ou seja, tal que $(\tau(u), \tau(v)) \in E_2 \Leftrightarrow (u, v) \in E_1$. Neste caso, escrevemos $\Gamma_1 \simeq \Gamma_2$ e τ é chamado um isomorfismo de grafos entre Γ_1 e Γ_2. O conjunto de isomorfismos entre um grafo Γ_1 e um grafo Γ_2 é denotado por $\mathrm{Iso}(\Gamma_1, \Gamma_2)$.

Exemplo 1 Os grafos $\Gamma_1 = (V_1, E_1)$ e $\Gamma_2 = (V_2, E_2)$, onde $E_1 = \{(1,2), (1,4), (2,5), (2,6), (3,5), (4,5)\}$ e $E_2 = \{(1,5), (2,3), (2,4), (2,5), (3,6), (5,6)\}$ são isomorfos via a permutação $\tau = (1,3)(4,6)$, ou seja, temos que $\tau E_1 = E_2$.

Exemplo 2 Podemos obter mais do que um isomorfismo para um par de grafos isomorfos, como podemos ver na Figura 3.3. Neste caso, nós temos $|\mathrm{Iso}(\Gamma_1, \Gamma_2)| = 4$, onde os isomorfismos são dados pelas permutações $\tau_1 = (2,5)$, $\tau_2 = (1,6)(3,4)$, $\tau_3 = (1,4)(3,6)$ e $\tau_4 = (1,3)(2,5)(4,6)$.

Formalmente, o problema pode ser enunciado da seguinte maneira.

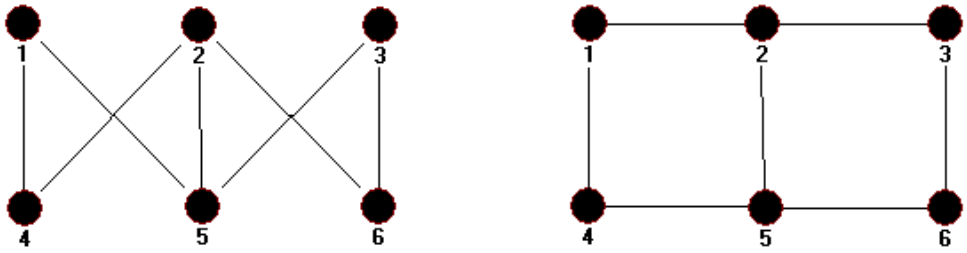

Figura 3.3: Par de Grafos com 4 isomorfismos.

Problema 1 (Problema do Isomorfismo de Grafos) Dados dois grafos Γ_1 e Γ_2 com n vértices cada, determinar se eles são isomorfos.

Não existe atualmente um algoritmo eficiente para resolver este problema no caso geral. Poderíamos tentar todas as permutações possíveis, mas isso resultaria em um algoritmo de complexidade $O(n!)$. Para que dois grafos sejam isomorfos, no mínimo as seguintes condições têm que ser respeitadas (condições necessárias):

1- Ter o mesmo número de vértices.

2- Ter o mesmo número de arestas.

3- Ter 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 isso não é suficiente para que sejam isomorfos. Por exemplo, os grafos da Figura 3.4 respeitam essas condições e não são isomorfos.

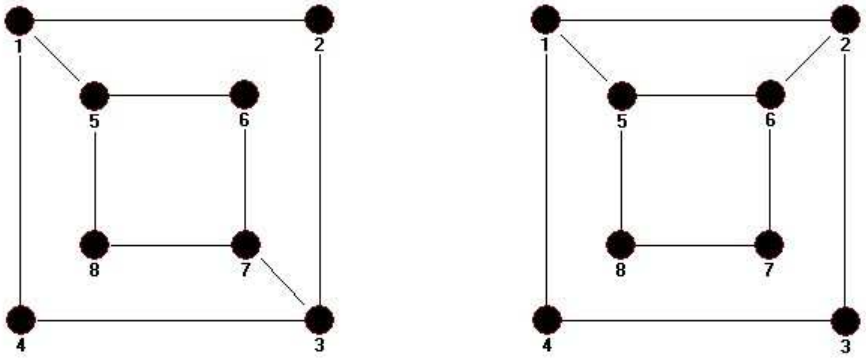

Figura 3.4: Par de grafos não isomorfos.

Mais detalhes sobre o Problema do Isomorfismo de Grafos pode ser encontrado em [Köbler et al. (1993)].

3.3 Problema do Automorfismo de Grafos

Um caso especial importante de isomorfismos de grafos são os automorfismos de grafos.

Definição 18 (Automorfismo de Grafos) Seja $\Gamma = (V, E)$ um grafo, onde $V = \{1, \ldots, n\}$. Uma permutação $\pi \in S_n$ é um automorfismo de Γ se $(\pi(u), \pi(v)) \in E$ sempre que $(u, v) \in E$. O conjunto de automorfismos de um grafo Γ é denotado por $\mathrm{Aut}(\Gamma)$.

Em outras palavras, um automorfismo é uma permutação de vértices que preserva a relação de adjacência bem como a de não-adjacência. Claramente, cada grafo tem a permutação trivial como um automorfismo. Além disso, temos:

Proposição 5 O conjunto $\mathrm{Aut}(\Gamma)$ é um subgrupo do grupo simétrico $S_n : \mathrm{Aut}(\Gamma) < S_n$.

Demonstração: Claramente $\mathrm{Aut}(\Gamma) \subset S_n$, uma vez que $\mathrm{Aut}(\Gamma)$ consiste de bijeções de um conjunto de n elementos sobre si mesmo. Como S_n é finito, basta mostrarmos que $\mathrm{Aut}(\Gamma)$ é fechado com respeito a operação de composição em S_n. De fato, tomemos $\phi, \psi \in \mathrm{Aut}(\Gamma)$ e sejam $u, v \in V$ (conjunto de vértices), então ψ leva a aresta (u, v) para a aresta $(\psi(u), \psi(v))$, da mesma forma ϕ leva a aresta $(\psi(u), \psi(v))$ para a aresta $(\phi(\psi(u)), \phi(\psi(v))) = (\phi\psi(u), \phi\psi(v))$. Portanto, $\phi\psi$ leva aresta em aresta, ou seja, $\phi\psi \in \mathrm{Aut}(\Gamma)$.

■

Este é chamado o grupo de automorfismos do grafo. Um grafo é chamado **rígido** se a permutação trivial é seu único automorfismo, isto é, se seu grupo de automorfismos é o grupo trivial.

Temos então o seguinte problema,

Problema 2 (Problema do Automorfismo de Grafos) Dado um grafo Γ, o Problema do Automorfismo de Grafos consiste em encontrar um conjunto de geradores para $\mathrm{Aut}(\Gamma)$.

Dizemos que um problema P_1 é redutível polinomialmente a um problema P_2, $P_1 \propto_p P_2$, se a existência de um algoritmo polinomial para P_2 implica na existência de um algoritmo polinomial para P_1. Dois problemas são equivalentes polinomialmente se cada um é redutível polinomialmente para o outro.

3.4 Relações entre Isomorfismos e Automorfismos de Grafos

A partir deste ponto denotaremos o Problema do Isomorfismo de Grafos e o Problema do Automorfismo de Grafos por PIG e PAG, respectivamente. Seguem algumas propriedades básicas de isomorfismos e automorfismos de grafos e relações importantes entre os dois conceitos.

Primeiramente, temos por definição que, dado um grafo Γ então $\mathrm{Aut}(\Gamma) = \mathrm{Iso}(\Gamma, \Gamma)$. Segundo, temos o seguinte resultado.

Teorema 8 Sejam Γ_1 e Γ_2 dois grafos isomorfos com o mesmo conjunto de vértices $V = \{1, \ldots, n\}$. Então, $\mathrm{Iso}(\Gamma_1, \Gamma_2) = \tau\mathrm{Aut}(\Gamma_1) = \mathrm{Aut}(\Gamma_2)\tau$ para cada $\tau \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$.

Demonstração: Sejam $\Gamma_1 = (V, E_1)$ e $\Gamma_2 = (V, E_2)$ dois grafos e sejam ϕ e ψ isomorfismos de Γ_1 para Γ_2, note que eles são permutações em S_n. Da definição, temos

$$(v, w) \in E_1 \Leftrightarrow (\phi(v), \phi(w)), (\psi(v), \psi(w)) \in E_2.$$

Por isso, $\psi^{-1}\phi$ é um automorfismo de Γ_1. Assim, ϕ e ψ estão na mesma classe lateral à esquerda de $\mathrm{Aut}(\Gamma_1)$. De fato,

$$\psi \in \psi\mathrm{Aut}(\Gamma_1) \text{ e } \phi = \psi\psi^{-1}\phi \in \psi\psi^{-1}\phi\mathrm{Aut}(\Gamma_1) = \psi\mathrm{Aut}(\Gamma_1).$$

Reciprocamente, sejam ϕ um isomorfismo de Γ_1 para Γ_2 e ρ um automorfismo de Γ_1. Então $\phi\rho$ é novamente um isomorfismo de Γ_1 para Γ_2. Assim, a classe lateral à esquerda $\phi\mathrm{Aut}(\Gamma_1)$ é o conjunto de todos os isomorfismos de Γ_1 para Γ_2.

Analogamente, mostra-se que $\mathrm{Iso}(\Gamma_1, \Gamma_2) = \mathrm{Aut}(\Gamma_2)\phi$.

■

Corolário 2 Sejam Γ_1 e Γ_2 grafos isomorfos. Então o número de isomorfismos de Γ_1 para Γ_2 é igual a ordem de $\mathrm{Aut}(\Gamma_1)$ e igual a ordem de $\mathrm{Aut}(\Gamma_2)$.

Corolário 3 Sejam Γ_1 e Γ_2 grafos isomorfos. Então $\mathrm{Aut}(\Gamma_1)$ e $\mathrm{Aut}(\Gamma_2)$ são conjugados em $\mathcal{P}(V)$, onde V é o conjunto de vértices de Γ_1 e Γ_2.

Demonstração: Seja $\phi \in \mathrm{Iso}(\Gamma_1, \Gamma_2)$, $\rho \in \mathrm{Aut}(\Gamma_1)$, $\tau \in \mathrm{Aut}(\Gamma_2)$. Então $\phi\rho\phi^{-1} \in \mathrm{Aut}(\Gamma_2)$ e $\phi^{-1}\tau\phi \in \mathrm{Aut}(\Gamma_1)$.

■

Notemos que a volta do Corolário 3 não é verdade. Antes de dar um exemplo, considere a seguinte definição.

Definição 19 Seja $\Gamma = (V, E)$ um grafo. O grafo complementar $\overline{\Gamma}$ de Γ é o grafo $(V, \overline{E})$, onde $(v, w) \in \overline{E}$ se $(v, w) \notin E$. Podemos também escrever $\overline{\Gamma} = (V, V \times V - E)$.

Exemplo 3 O grafo completo K_n é um grafo com vértices $V = \{1, \ldots, n\}$ tais que todos dois vértices distintos são adjacentes. O grafo complementar $\overline{K_n}$ de K_n é um grafo que tem $V = \{1, \ldots, n\}$ como conjunto de vértices e nenhuma aresta. Claramente, $S_n = \mathrm{Aut}(K_n) = \mathrm{Aut}(\overline{K_n})$, mas K_n e $\overline{K_n}$ não são isomorfos para $n > 1$.

Vamos ilustrar o Teorema 8 e seus corolários a seguir:

Exemplo 4 Sejam $\Gamma_1 = (V, E_1)$ e $\Gamma_2 = (V, E_2)$, onde $V = \{1, \ldots, 5\}$, $E_1 = \{(1,2), (1,4), (2,3), (3,4), (3,5)\}$ e $E_2 = \{(1,4), (1,5), (2,3), (3,4), (3,5)\}$. Os grafos Γ_1 e Γ_2 são isomorfos (ver Figura 3.5). Existem dois isomorfismos de Γ_1 para Γ_2, a saber, $\pi_1 = (2,4,5)$ e $\pi_2 = (2,5)$. Portanto, $\mathrm{Iso}(\Gamma_1, \Gamma_2) = \{(2,4,5), (2,5)\}$. Além disso, verifica-se que $\mathrm{Aut}(\Gamma_1) = \{(\quad), (2,4)\}$ e $\mathrm{Aut}(\Gamma_2) = \{(\quad), (4,5)\}$.

Tomando $\pi_1 = (2,4,5)$, temos

$$\pi_1 \mathrm{Aut}(\Gamma_1) = \{(2,4,5),(2,4,5)(2,4)\} = \{(2,4,5),(2,5)\}$$
$$\mathrm{Aut}(\Gamma_2)\pi_1 = \{(2,4,5),(4,5)(2,4,5)\} = \{(2,4,5),(2,5)\}$$

Analogamente, tomando $\pi_2 = (2,5)$, temos

$$\pi_2 \mathrm{Aut}(\Gamma_1) = \{(2,5),(2,5)(2,4)\} = \{(2,5),(2,4,5)\}$$
$$\mathrm{Aut}(\Gamma_2)\pi_2 = \{(2,5),(4,5)(2,5)\} = \{(2,5),(2,4,5)\}$$

Portanto, $\mathrm{Iso}(\Gamma_1,\Gamma_2) = \pi\mathrm{Aut}(\Gamma_1) = \mathrm{Aut}(\Gamma_2)\pi$ para cada $\pi \in \mathrm{Iso}(\Gamma_1,\Gamma_2)$.

Note ainda que, $(4,5) = (2,5)^{-1}(2,4)(2,5)$, ou seja, $(4,5)$ e $(2,4)$ pertencem a mesma classe de conjugação. Portanto, $\mathrm{Aut}(\Gamma_2) = (2,5)\mathrm{Aut}(\Gamma_1)$.

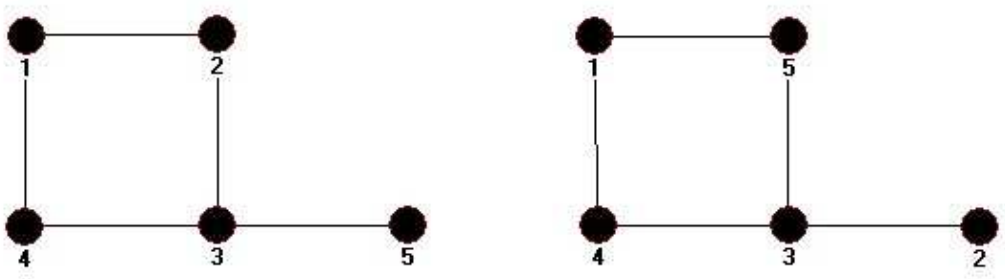

Figura 3.5: Grafos isomorfos para ilustrar o Teorema 8 e seus corolários.

Agora, daremos uma relação ainda mais importante entre os dois problemas. Um algoritmo eficiente para resolver o PIG implica num algoritmo eficiente para resolver o PAG.

Teorema 9 O Problema do Automorfismos de Grafos é redutível polinomialmente para o Problema do Isomorfismos de Grafos, PAG $\propto_p$ PIG.

Demonstração: Vide [Mathon (1979)].

■

Para cada permutação $\pi \in S_n$ temos que $\pi\overline{\Gamma} = \overline{\pi\Gamma}$. De fato, $\pi\overline{\Gamma} = \pi(V, V \times V - E) = (V, V \times V - \pi E)$. Por outro lado, temos que $\overline{\pi\Gamma} = \overline{(V, \pi E)} = (V, V \times V - \pi E)$. Uma consequência imediata disto é que se π é um isomorfismo entre Γ_1 e Γ_2, então π é também um isomorfismo entre $\overline{\Gamma_1}$ e $\overline{\Gamma_2}$: $\pi\overline{\Gamma_1} = \overline{\pi\Gamma_1} = \overline{\Gamma_2}$. Em outras palavras,

$$Iso(\Gamma_1, \Gamma_2) = Iso(\overline{\Gamma_1}, \overline{\Gamma_2}) \tag{3.1}$$

Se o grafo Γ não é conexo, então seu grafo complementar $\overline{\Gamma}$ o será. Portanto, sempre que considerarmos isomorfismos entre grafos, podemos assumir que no mínimo um dos grafos é conexo. Além disso, se um dos grafos é conexo e o outro não é, então sabemos que eles não são isomorfos. Logo, podemos sempre assumir que ambos os grafos são conexos.

Outra importante "transformação de grafos" é a **união disjunta de grafos** dada pela seguinte definição,

Definição 20 Dados dois grafos $\Gamma_1 = (V_1, E_1)$ e $\Gamma_2 = (V_2, E_2)$ com conjuntos disjuntos de vértices V_1 e V_2 (e portanto, conjuntos disjuntos de arestas), sua união disjunta é dada por $\Gamma_1 \cup \Gamma_2 = (V_1 \cup V_2, E_1 \cup E_2)$, ou seja, é o grafo obtido pela união dos conjuntos de vértices e união dos conjuntos de arestas dos grafos.

Exemplo 5 O grupo de automorfismos do ciclo de comprimento n é o grupo Diedral D_n de ordem $2n$. Além disso, o grupo de automorfismos do ciclo orientado de comprimento n é o grupo cíclico $\mathbb{Z}_n$ de ordem n.

No entanto, um caminho de comprimento ≥ 1 tem apenas 2 automorfismos.

Capítulo 4

Descrição Matemática do Problema do Isomorfismo de Grafos

4.1 Isomorfismos de Grafos e Interseção de Grupos

Neste capítulo, mostraremos que o Problema do Isomorfismo de Grafos pode ser reduzido para o problema de encontrar o conjunto de geradores para a interseção de dois grupos de permutações, ou seja, um algoritmo de tempo polinomial para o problema de interseção de grupos de permutações implica na existência de um algoritmo de tempo polinomial para o problema do isomorfismo de grafos [Hoffman (1979)]. Para isso, mostraremos que o grupo de automorfismos de todo grafo é isomorfo a interseção de dois grupos de permutações para os quais os conjuntos geradores são conhecidos.

Definimos então o Problema de Interseção de Grupos para grupos de permutações:

Problema 3 (Problema de Interseção de Grupos) Dados conjuntos de geradores para dois grupos de permutações $A < S_n$ e $B < S_n$ para algum n, determinar um conjunto gerador para $C = A \cap B$.

Não existe algoritmo clássico eficiente para tal problema.

Para todo grupo G, existe um grafo cujo grupo de automorfismos é isomorfo a G [Frucht (1939)]. O grupo de automorfismos de um grafo caracteriza suas simetrias, e além disso, é muito usado na determinação de certas propriedades.

Teorema 10 Todo grupo é um grupo de automorfismos de algum grafo. Além disso, se o grupo é finito, então o grafo pode ser tomado como finito.

Demonstração: Vide [Frucht (1939)].

■

Subsequentemente, [Frucht (1949)] mostrou que todo grupo é o grupo de automorfismo de um grafo de grau 3, e isto tem inspirado um grande número de resultados similares.

Seja $\Gamma = (V, E)$ um grafo com conjunto de vértices $V = \{1, \ldots, n\}$. Construiremos uma nova representação para S_n fazendo o grupo atuar no conjunto de todos os pares não-ordenados $(i, j), 1 \leq i, j \leq n, i \neq j$, ou seja, o conjunto de arestas e não-arestas, $E \cup \overline{E}$. Repare que $|E \cup \overline{E}| = \frac{n(n-1)}{2}$. A nova representação é obtida através da associação de π em S_n com uma permutação $\psi \in S_{n(n-1)/2}$ agindo sobre os pares não-ordenados (i, j) da seguinte maneira:

$$\psi(i, j) = (\pi(i), \pi(j)). \tag{4.1}$$

Vamos denotar por $S_n^{'}$ essa nova representação para S_n. Então, temos que S_n atua no conjunto $\{1, \ldots, n\}$ enquanto $S_n^{'}$ atua no conjunto $E \cup \overline{E} = \{(i, j), 1 \leq i, j \leq n, i \neq j\}$. Podemos dizer ainda que $\{1, \ldots, n\}$ são os pontos de S_n e que os elementos de $E \cup \overline{E}$ são os pontos de $S_n^{'}$. Temos então o seguinte resultado,

Proposição 6 Seja $S_n^{'}$ definido como acima e seja a função

$$\begin{aligned} \phi : S_n &\to S_n^{'} \\ \pi &\mapsto \psi_\pi \end{aligned}$$

tal que $\psi_\pi(i, j) = (\pi(i), \pi(j))$, onde $1 \leq i, j \leq n, i \neq j$. Então a função ϕ é um isomorfismo.

Demonstração: Sejam $\pi, \tau \in S_n$. Então,

$$\begin{aligned}
\phi(\pi\tau)(i,j) &= \psi_{\pi\tau}(i,j) = (\pi\tau(i), \pi\tau(j)) \\
&= (\pi(\tau(i)), \pi(\tau(j))) = \psi_\pi(\tau(i), \tau(j)) \\
&= \psi_\pi(\psi_\tau(i,j)) = \psi_\pi\psi_\tau(i,j) \\
&= \phi(\pi)\phi(\tau)(i,j)
\end{aligned}$$

Portanto ϕ é um homomorfismo. Além disso, por definição, temos que ϕ é uma bijeção.

■

Note que S_n pode ser gerado por duas permutações, por exemplo por $\pi_1 = (1,2)$ e $\pi_2 = (1,\ldots,n)$ (consequência da Proposição 3). Portanto, $S_n^{'}$ também pode ser gerado por duas permutações, a saber, ψ_{π_1} e ψ_{π_2}. Seja $\overline{E}$ o conjunto de não-arestas de Γ. Seja $T = \mathcal{P}(E) \times \mathcal{P}(\overline{E})$, o produto direto de $\mathcal{P}(E)$ e $\mathcal{P}(\overline{E})$. Pelo que acabamos de ver, sabemos que T é gerado por quatro permutações que nós conhecemos. Então, afirmamos o seguinte:

Teorema 11 Seja $\Gamma = (V, E)$ um grafo com conjunto de vértices $V = \{1,\ldots,n\}$. Então, $\mathrm{Aut}(\Gamma) \simeq S_n^{'} \cap T$.

Demonstração: Seja $A = \mathrm{Aut}(\Gamma) < S_n$, o grupo de automorfismos de Γ. Seja $A^{'}$ o conjunto de elementos em $S_n^{'}$ correspondentes aos elementos de A, obtidos através do isomorfismo $\phi : S_n \to S_n^{'}$ (ver proposição 6), de modo que tenhamos $A \simeq A^{'}$. Mostraremos então que $A^{'} = S_n^{'} \cap T$.

Seja $\pi \in A$ um automorfismo de Γ e seja ψ_π o elemento correspondente em $A^{'}$. Então, ψ_π é um elemento de $S_n^{'}$. Além disso, como π leva arestas em arestas e não-arestas em não-arestas, ψ_π está também em T pela definição de tal conjunto.

Reciprocamente, seja ψ_π em $S_n^{'} \cap T$. Uma vez que $\psi_\pi \in S_n^{'}$, existe uma permutação de vértices associada π em S_n, além disso, como $\psi_\pi \in T$, a permutação de vértices π correspondente leva arestas em arestas e não-arestas para não-arestas, logo π é um automorfismo, ou seja, $\pi \in A = \mathrm{Aut}(\Gamma)$, daí $\psi_\pi \in A^{'}$.

Portanto, $A' = S'_n \cap T$. Mas, como $A \simeq A'$, temos que $A \simeq S'_n \cap T$, conforme queríamos demonstrar.

■

Notemos que a interseção acima faz sentido porque T é isomorfo a um subgrupo de $S_{n(n-1)/2}$.

Corolário 4 Se o problema de interseção de grupos está em P (ver seção 4.3), então segue que o problema de isomorfismo de grafos também está em P.

É uma questão em aberto se a volta do Corolário 4 é verdadeira.

Vamos ilustrar o Teorema 11 com um exemplo.

Exemplo 6 Seja $\Gamma = (V, E)$ um grafo tal que $V = \{1, 2, 3, 4\}$ e $E = \{(1, 2), (1, 3), (2, 3), (3, 4)\}$. Na Figura 4.1 temos uma representação para tal grafo.

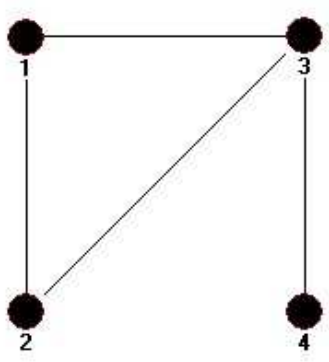

Figura 4.1: grafo $\Gamma = (V, E)$

Primeiramente, vemos que $\overline{E} = \{(1, 4), (2, 4)\}$, logo, temos que $E \cup \overline{E} = \{(1, 2), (1, 3), (2, 3), (3, 4), (1, 4), (2, 4)\}$. É fácil ver, neste caso, que $\mathrm{Aut}(\Gamma) = \{(\ \ ), (1, 2)\}$, porém vamos achar esse resultado através do Teorema 11.

Vamos reescrever o conjunto E de tal forma que 1 represente a aresta $(1, 2)$, 2 represente a aresta $(1, 3)$, 3 represente a aresta $(2, 3)$ e 4 represente a aresta $(3, 4)$. Da mesma forma vamos representar a não-aresta $(1, 4)$ por 5 e a não-aresta $(2, 4)$ por 6, portanto o conjunto $E \cup \overline{E}$ será dado nessa nova representação por $E \cup \overline{E} = \{1, 2, 3, 4, 5, 6\}$.

Logo, usando essa nova representação temos também os seguintes conjuntos

$$\mathcal{P}(\overline{E}) = \{(\quad), (5,6)\}$$

$$\mathcal{P}(E) = \{(\quad), (1,2), (1,3), (1,4), \ldots\} = S_4$$

Definimos agora o conjunto $T = \mathcal{P}(E) \times \mathcal{P}(\overline{E}) = \{(\pi, \tau); \pi \in \mathcal{P}(E), \tau \in \mathcal{P}(\overline{E})\}$, produto direto de $\mathcal{P}(E)$ e $\mathcal{P}(\overline{E})$.

Obtemos também o grupo $S_4^{'}$ a partir de S_4, pelo isomorfismo (conforme Proposição 6) definido por $\psi_\pi(i,j) = (\pi(i), \pi(j))$, onde $\pi \in S_4$, $\psi_\pi \in S_4^{'}$, $1 \leq i, j \leq 4$, $i \neq j$. Por exemplo, se $\pi = (1,2)$ então $\psi_\pi = (2,3)(5,6)$. Note que $\psi_\pi \in T$. Por outro lado, se $\pi = (3,4)$ então $\psi_\pi = (2,5)(3,6)$. Note que nesse caso $\psi_\pi \notin T$.

Repare que $S_4^{'} < S_6$ e T é isomorfo a um subgrupo de S_6. Portanto podemos fazer a interseção de tais grupos, que neste caso será igual a:

$$A^{'} = S_4^{'} \cap T = \{(\quad), (2,3)(5,6)\}.$$

Portanto, pelo Teorema 11, temos que $A^{'} \simeq A = \mathrm{Aut}(\Gamma)$. Usando o isomorfismo citado acima e sabendo que $A^{'} = \{(\quad), (2,3)(5,6)\}$, temos que a permutação $(2,3)(5,6)$ em $A^{'} = S_4^{'} \cap T$ equivale a permutação $(1,2)$ em $A = \mathrm{Aut}(\Gamma)$, portanto $A = \mathrm{Aut}(\Gamma) = \{(\quad), (1,2)\}$, confirmando o que já sabíamos.

4.2 Grafos com grupo de automorfismos solúvel

Dado um grafo Γ com n vértices, vimos pela Proposição 5 que seu grupo de automorfismos $\mathrm{Aut}(\Gamma)$ é subgrupo do grupo simétrico S_n. Além disso, pelo Teorema 11, vimos que $\mathrm{Aut}(\Gamma) \simeq S_n^{'} \cap T$, onde $S_n^{'}$ e T foram definidos anteriormente.

Seja a classe de grafos $A_n = (V_{A_n}, E_{A_n})$, onde $V_{A_n} = \{a_i, b_i, c_i : 0 \leq i < n\}$ e $E_{A_n} = \{(a_i, a_{i+1}), (a_i, b_i), (a_i, c_i), (b_i, c_i), (c_i, a_{i+1}) : 0 \leq i < n\}$, onde todos os índices são lidos módulo n, isto é, A_n é composto de um n-ciclo $(a_0, \ldots, a_{n-1})$ com um retângulo desenhado sobre cada lado mais uma diagonal em cada retângulo. Veja a Figura 4.2.

É fácil ver que $|V_{A_n}| = 3n$. Podemos também verificar que, $\mathrm{Aut}(A_n) = C_n$ [Harary (1969)], onde C_n é o grupo cíclico de ordem n. Rotações por $2\pi/n$ geram um subgrupo isomorfo a C_n, o grupo cíclico de ordem n. Portanto, temos que $\mathrm{Aut}(A_n) < S_{3n}$.

Vejamos um exemplo,

Exemplo 7 Seja o grafo $A_3 = (V_{A_3}, E_{A_3})$, onde $V_{A_3} = \{a_i, b_i, c_i : 0 \leq i < 3\}$ e $E_{A_3} = \{(a_i, a_{i+1}), (a_i, b_i), (a_i, c_i), (b_i, c_i), (c_i, a_{i+1}) : 0 \leq i < 3\}$, descrito na Figura 4.2. Este é o menor grafo cujo grupo de automorfismos é o C_3 [Harary e Palmer (1966/1972)].

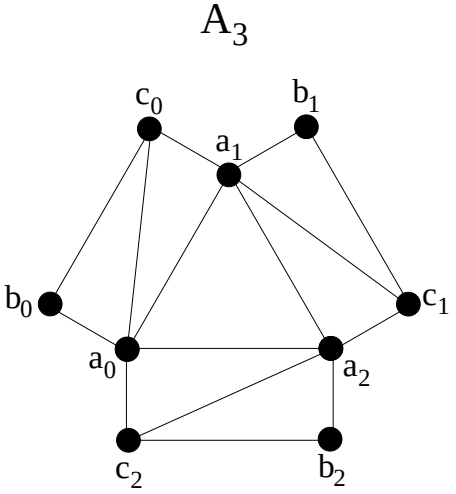

Figura 4.2: Exemplo de um grafo da classe A_n, com $n = 3$.

Renomeando os elementos do conjunto V_{A_3}, chamando a_0 de 1, a_1 de 2 até c_2 de 9, então teremos $V_{A_3} = \{1, 2, \ldots, 9\}$ e portanto, $E_{A_3} = \{(1,2), (2,3), (1,3), (1,4), (2,5), (3,6), (1,7), (2,8), (3,9), (4,7), (5,8), (6,9), (2,7), (3,8), (1,9)\}$. Temos então que, $\overline{E}_{A_3} = E(K_9) - E_{A_3}$ (onde K_9 é o grafo completo, definido no exemplo 3).

Definiremos agora o conjunto T como sendo o produto direto de $\mathcal{P}(E_{A_3})$ com $\mathcal{P}(\overline{E}_{A_3})$, ou seja, $T = \mathcal{P}(E_{A_3}) \times \mathcal{P}(\overline{E}_{A_3})$. Como $|E(K_9)| = 36$, temos que $|\overline{E}_{A_3}| = 21$, a saber, $\overline{E}_{A_3} = \{(1,5), (1,6), (1,8), (2,4), (2,6), (2,9), (3,4), (3,5), (3,7), (4,5), (4,6), (4,8), (4,9), (5,6), (5,7), (5,9), (6,7), (6,8), (7,8), (7,9), (8,9)\}$.

Por outro lado, construiremos também o conjunto S_9' a partir de S_9, usando

o seguinte homomorfismo, conforme Proposição 6.

$$\psi_\pi(i,j) = (\pi(i), \pi(j)), \pi \in S_9, 1 \leq i, j \leq 9, i \neq j.$$

Logo,

$$S_9^{'} = \{\psi_\pi; \psi_\pi(i,j) = (\pi(i), \pi(j)), \pi \in S_9, 1 \leq i, j \leq 9, i \neq j\} < S_{36}$$

Assim, pelo teorema 11 temos que $\mathrm{Aut}(A_3) \simeq S_9^{'} \cap T$. Por outro lado, temos que $\mathrm{Aut}(A_3) = C_3$ (o grupo cíclico C_3). Mas, $C_3 \lhd D_3$.

Em geral, temos que o grupo de automorfismos $\mathrm{Aut}(A_n) = C_n \lhd D_n$. Além disso, vimos no Teorema 6 que o grupo D_n é solúvel.

4.3 Problemas de Reconhecimento de Linguagens e Classes de Complexidade

Nesta seção vamos fazer uma breve revisão das classes de complexidade clássica e quântica.

A fim de computar, precisamos de uma maneira razoável para representar informação. Codificação unária (ou seja, que representam o número j por uma seqüência de $1s$ de comprimento j) é exponencialmente menos eficaz do que usar seqüências de símbolos a partir de qualquer alfabeto fixado de tamanho, pelo menos, 2. Passar de um alfabeto de tamanho 2 para um alfabeto maior de tamanho fixo só muda o tamanho da representação do problema por um fator constante. Então, vamos simplesmente utilizar o alfabeto $\Sigma = \{0, 1\}$. O conjunto Σ^* denota todas as seqüências (strings) de comprimento finito sobre este alfabeto. Uma linguagem L é um subconjunto de Σ^*. Em particular, usualmente L é um conjunto de seqüências com alguma propriedade de interesse.

Um algoritmo 'resolve o problema de reconhecimento de linguagem para L' se ele aceita toda seqüência $x \in L$ e rejeita toda seqüência $x \notin L$.

Por exemplo, o problema de decidir se um inteiro n (representado como uma

seqüência de bits) é primo é refeito como o problema de reconhecer se a seqüência representando n está na linguagem PRIME $= \{10, 11, 010, 011, 101, 111, \ldots\}$ (que consiste no conjunto de todas as seqüências que representam números primos, de acordo com alguma codificação razoável, o que neste caso é a codificação binária 'padrão').

Agora que mostramos como expressar problemas de decisão em termos de reconhecer os elementos de uma linguagem, podemos definir diferentes classes de linguagens. Por exemplo, definimos formalmente **P** ('tempo polinomial') como sendo a classe de linguagens L para as quais existe um algoritmo clássico determinístico A executando no pior caso em tempo polinomial, isto é, existe um polinômio $p(n)$ tal que A executa por tempo no máximo $p(n)$ instruções elementares sobre entradas de comprimento n de tal forma que, para uma eventual entrada $x \in \Sigma^*$ o algoritmo A sobre a entrada x, produz 'aceitar' se e somente se $x \in L$. Note que nesta classe não é possível capturar as vantagens da utilização da aleatoriedade para resolver problemas.

A classe **BPP** ('tempo polinomial probabilístico com erro limitado') é composta por todas as linguagens L para as quais existem um algoritmo clássico randômico A executando com pior caso tempo polinomial tal que, para qualquer entrada $x \in \Sigma^*$ temos

- Se $x \in L$ então a probabilidade de A aceitar x é no mínimo $\frac{2}{3}$
- Se $x \notin L$ então a probabilidade de A aceitar x é no máximo $\frac{1}{3}$.

É importante notar que quando nos referimos ao 'a probabilidade de A aceitar', estamos nos referindo à probabilidade sobre escolhas aleatórias de caminhos da computação sobre a entrada fixada $x \in L$. É também importante notar que não há nada de especial a respeito da constante $\frac{2}{3}$. Qualquer constante $\frac{1}{2} + \delta$ funcionará [Papadimitriou (1994), Kaye et al. (2007)].

A classe **BQP** ('tempo polinomial quântico com erro limitado') é composta por todas as linguagens L para as quais existe um algoritmo quântico A executando com pior caso tempo polinomial tal que, para qualquer entrada $x \in \Sigma^*$ temos

- Se $x \in L$ então a probabilidade de A aceitar x é no mínimo $\frac{2}{3}$
- Se $x \notin L$ então a probabilidade de A aceitar x é no máximo $\frac{1}{3}$.

Tratamos algoritmos com complexidade polinomial como 'eficientes' e problemas que podem ser resolvidos com complexidade polinomial como 'tratáveis', e problemas sem solução polinomial como 'intratáveis'.

A classe **NP** ('tempo polinomial não-determinístico') consiste de todas as linguagens L para as quais existe um algoritmo clássico de tempo polinomial $A(a, b)$ tal que para qualquer entrada $x \in \Sigma^*$ temos

- Se $x \in L$ então existe uma entrada y tal que $A(x, y)$ produz 'aceitar'
- Se $x \notin L$ então $A(x, y)$ produz 'rejeitar' para todo y

e o comprimento de y é limitado por um polinômio no comprimento de x.

O problema de decisão da fatoração de inteiros é um exemplo de problema da classe **NP**. Tal problema consiste no seguinte, dado um número inteiro composto m, e $l < m$, teria m algum fator não-trivial menor do que l? O que caracteriza os problemas **NP** é o fato de que os casos "sim" (aceitar) de um problema podem facilmente ser verificados uma vez que se tenha uma evidência apropriada. Existe uma assimetria interessante na definição de **NP**. Embora devamos ser capazes de testar rapidamente se uma possível evidência para $x \in L$ é de fato uma evidência, tal necessidade não existe para $x \notin L$. Por exemplo, no caso do problema da fatoração existe uma maneira simples de demonstrarmos se um dado número possui um fator menor do que m, mas apresentar uma prova para mostrar que um número não possui fatores menores do que m é mais difícil.

Os problemas da classe **NP** considerados mais difíceis formam uma subclasse chamada '**NP-Completa**'. Qualquer problema dessa classe é tão difícil quanto qualquer outro problema da classe **NP**, em outras palavras, um algoritmo que resolva um problema da classe **NP-Completa** pode ser adaptado para resolver qualquer outro problema da classe **NP**, com pequeno custo [Papadimitriou (1994)].

Os problemas de fatoração de inteiros e isomorfismo de grafos estão na classe **NP**, mas não acredita-se que estejam na classe **NP-Completa**.

A classe **PESPAÇO** consiste em todas as linguagens L para as quais existe um algoritmo clássico A usando no pior caso espaço polinomial tal que para qualquer entrada $x \in \Sigma^*$ o algoritmo A aceita x se e somente se $x \in L$. Ou seja, os problemas em **PESPAÇO** podem ser resolvidos em uma máquina de Turing com número polinomial de bits, sem limite de tempo.

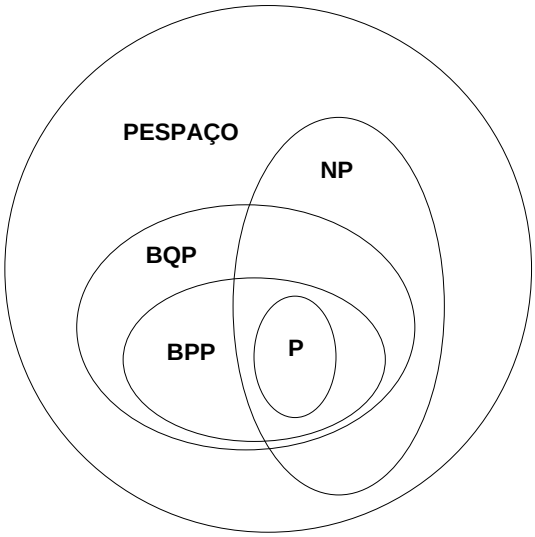

Figura 4.3: Este diagrama ilustra as conhecidas relações entre algumas das mais importantes classes de complexidades. Atualmente, nenhuma das inclusões são sabidas serem estritas. Por exemplo, atualmente não existe qualquer prova de que **P** $\neq$ **PESPAÇO**.

A figura 4.3 (adaptada de [Kaye et al. (2007)], página 185) ilustra as relações conhecidas entre as classes de complexidade que acabamos de definir. Por exemplo, claramente **P** $\subseteq$ **BPP** $\subseteq$ **BQP** $\subseteq$ **PESPAÇO** e **P** $\subseteq$ **NP** $\subseteq$ **PESPAÇO**. Infelizmente, embora se acredite que cada uma dessas inclusões seja estrita, nenhuma jamais foi demonstrada. Mas acredita-se que **P**$\neq$**NP** e que **NP**$\neq$**PESPAÇO**. Esperamos também que **BPP**$\neq$**BQP**.

Sabemos que **P** é subconjunto de **NP**, pois a capacidade de se resolver um problema implica a verificação de suas soluções, no entanto, não se sabe se existem problemas de **NP** que não pertencem a **P**. O problema em aberto mais famoso em ciência da computação é determinar se existem ou não problemas em **NP** que não

pertençam a **P**, abreviado como "problema **P**≠**NP**". A maioria dos cientistas da computação acredita que **P**≠**NP**, no entanto, após décadas de trabalho ninguém demonstrou tal fato, e a possibilidade **P**=**NP** permanece.

O problema do isomorfismo de grafos ocupa uma importante posição no mundo da análise de complexidade. Ele é um dos problemas que está na classe **NP** mas não se sabe se está nas classes **P** ou **NP-Completa**. Supondo **P**≠**NP**, é possível provar que existe uma classe não-vazia de problemas **NPI** (**NP** intermediária) que não são solucionáveis com recursos polinomiais, nem **NP-Completos**. É claro que não se conhece nenhum problema de **NPI**, pois, caso contrário, teríamos **P**≠**NP**, mas existem problemas considerados candidatos. Uma das hipóteses mais aceitas é que o problema do isomorfismo de grafos pertença a tal classe de problemas.

O maior desafio da área de algoritmos quânticos é encontrar problemas que estão em **BQP** mas não em **BPP**, isto é, encontrar problemas que sejam resolvidos de forma eficiente em um computador quântico, mas não em um computador clássico. O estudo dessas classes de complexidade e as relações entre elas pode ser útil para compreender a dificuldade destes problemas.

4.4 Algoritmos Clássicos

Não é conhecido na literatura um algoritmo clássico eficiente para o problema do isomorfismo de grafos para o caso geral. O melhor algoritmo conhecido para tal problema "roda" em tempo $e^{O(\sqrt{n \log n})}$ [Babai (1980), Babai e Luks (1983), Zemlyachenko et al. (1985)]. Dizemos que tal algoritmo é subexponencial no número de vértices dos grafos de entrada. No entanto, devido a grande dificuldade de se tratar o caso geral do problema e da necessidade de algoritmos que resolvam o problema para grafos em situações práticas, pesquisadores têm adotado outros caminhos para resolver o problema.

Uma das estratégias adotadas foi desenvolver algoritmos heurísticos que resolvam o problema de forma satisfatória. Este é o caso do algoritmo desenvolvido

por Mckay, que encontra geradores para o grupo de automorfismos do grafo [McKay (1981), McKay (1990)]. Tal algoritmo utiliza informações à respeito da vizinhança imediata de cada vértice e realiza assim um refinamento, classificando seus vértices, que juntamente com técnicas da teoria de grupos reduz o número de possíveis soluções a serem analisadas. Este algoritmo é considerado mais poderoso do que qualquer outro algoritmo publicado para a solução prática do caso geral do problema de isomorfismo de grafos. No entanto, possui pior caso exponencial.

Uma outra estratégia usada consiste em tentar desenvolver algoritmos polinomiais para classes restritas de grafos.

4.4.1 Algoritmos Clássicos para certas classes de grafos

Existem algoritmos clássicos polinomiais para determinadas classes de grafos [Fortin (1996)]. Temos então uma versão restrita do Problema de Isomorfismo de Grafos, onde os grafos de entrada pertencem a uma classe $\mathcal{C} \subseteq \mathcal{G}$, onde $\mathcal{G}$ denota todos os grafos conexos e não-orientados.

O primeiro grande resultado neste campo foi um artigo de Luks para a classe de grafos que tem grau limitado por uma constante (isto é, grau máximo $\leq$ alguma constante k) [Luks (1982)]. Ele mostrou que para tais grafos existe um algoritmo em tempo polinomial para checar isomorfismos. O algoritmo apresentado no artigo tem complexidade $O(n^{ck \log k})$, onde $c > 1$ é uma constante e k é o grau máximo. No entanto, mesmo o grau máximo sendo apenas 10, esta técnica não produz um algoritmo prático.

Foi mostrado por [Hopcroft e Wong (1974)] que grafos planares poderiam ser checados por isomorfismos em tempo linear, embora o seu algoritmo possua uma grande constante, o que também o torna impróprio na prática. Outra grande classe de grafos sobre os quais isomorfismo de grafos pode ser eficientemente resolvido são os grafos arco-circulares. [Hsu (1995)] mostrou que existe um algoritmo $O(mn)$ para testar isomorfismos destes grafos, onde m é o número de arestas, e n é o número de vértices. Este resultado é interessante na medida em que não exige

alguns parâmetros explícitos para ser constante, e parece aplicar-se a um grande grupo de grafos práticos.

Existem diversas outras classes de grafos restritas para as quais isomorfismos de grafos podem ser resolvidos em tempo polinomial. Por exemplo, [Cogis e Guinaldo (1995)] mostraram que isomorfismo de grafos conceituais pode ser resolvido em tempo polinomial. Isto é de particular interesse para a comunidade de inteligência artificial. Outras classes de grafos como as árvores [Aho et al. (1974)], grafos de permutação [Colbourn (1981)], grafos intervalo [Lueker e Booth (1979)], grafos de intervalo próprio [de Figueiredo et al. (1995)], partial k-trees [Bodlaender (1990)] têm problemas de isomorfismo que estão em **P**.

Capítulo 5

Computação Quântica

A Computação Quântica (CQ) é uma área de pesquisa recente que utiliza elementos de três áreas importantes: Matemática, Física e Computação. O objetivo da CQ é estudar métodos para processar, transmitir e armazenar informações contidas em estados quânticos. Os seguintes questionamentos nos motivam a estudar CQ: Existem problemas que os computadores quânticos podem resolver mais rapidamente do que os clássicos? O que faz os computadores quânticos serem mais eficientes do que os clássicos?

5.1 Mecânica Quântica

Nesta seção apresentaremos uma breve revisão de Mecânica Quântica utilizando a notação adotada para o estudo da Computação Quântica [Nielsen e Chuang (2000), Kaye et al. (2007), Lomont (2004)].

Definição 21 (Q-bit) Um q-bit (bit quântico) é um vetor unitário em $\mathbb{C}^2$.

Definição 22 (Estado) O estado de um sistema quântico é um vetor (coluna) em algum espaço vetorial, escrevemos $|\psi\rangle$.

Estados quânticos com $n > 1$ q-bits são freqüentemente chamados de registradores quânticos.

Na computação clássica temos o bit como conceito fundamental. Analogamente, na computação quântica temos como conceito fundamental o bit quântico

ou q-bit, abreviadamente. Um bit pode assumir somente os valores 0 ou 1. Na computação quântica, os valores 0 e 1 são substituídos pelos vetores $|0\rangle$ e $|1\rangle$. Esta notação para vetores é chamada notação de Dirac e é padrão na Mecânica Quântica. A diferença entre bits e q-bits é que um q-bit $|\psi\rangle$ pode também estar em uma combinação linear dos vetores $|0\rangle$ e $|1\rangle$,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{5.1}$$

onde α e β são números complexos. Dizemos que $|\psi\rangle$ é uma superposição dos estados $|0\rangle$ e $|1\rangle$ com amplitudes α e β, que satisfazem a $|\alpha|^2 + |\beta|^2 = 1$. A interpretação física de $|\psi\rangle$ é a co-existência do q-bit nestes dois estados. Assim, $|\psi\rangle$ é um vetor em um espaço vetorial complexo de dimensão 2, onde $\{|0\rangle\,, |1\rangle\}$ forma uma base ortonormal, chamada base computacional. O estado $|0\rangle$ não é o vetor zero, mas simplesmente o primeiro vetor da base. As matrizes representando os vetores $|0\rangle$ e $|1\rangle$ são dadas por

$$|0\rangle = \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}.$$

O estado $|\psi\rangle$ pode guardar uma enorme quantidade de informação em seus coeficientes α e β, mas essa informação mora num nível quântico. Para trazer essa informação quântica para o nível clássico devemos **medir** (ver Apêndice A) o q-bit. A Mecânica Quântica nos diz que o processo de medida causa um distúrbio no estado quântico, projetando o estado $|\psi\rangle$ nos subespaços gerados por $|0\rangle$ e $|1\rangle$, produzindo o estado $|0\rangle$ com probabilidade $|\alpha|^2$ e o estado $|1\rangle$ com probabilidade $|\beta|^2$.

As leis da Mecânica Quântica determinam que se o computador estiver isolado, a direção de $|\psi\rangle$ pode mudar mas não a sua norma. Na Álgebra Linear isso é descrito pela ação de um operador unitário U, que é uma matriz complexa 2×2 satisfazendo

$$UU^{\dagger} = I, \tag{5.2}$$

onde $U^\dagger = (U^*)^T$ ($*$ indica complexo conjugado e T indica a operação transposta) e I é a matriz identidade 2×2.

O **produto tensorial** é uma forma de se juntar espaços vetoriais para formar espaços vetoriais maiores. É necessário introduzir tal conceito para considerarmos casos de múltiplos q-bits.

Suponha que V e W sejam espaços vetoriais de dimensões m e n, respectivamente. Portanto, $V \otimes W$ é um espaço vetorial com dimensão mn. Os elementos de $V \otimes W$ são combinações lineares de produtos tensoriais $|v\rangle \otimes |w\rangle$ dos elementos $|v\rangle$ de V e $|w\rangle$ de W. Em particular, se $|i\rangle$ e $|j\rangle$ são bases ortonormais de V e W, então $|i\rangle \otimes |j\rangle$ é uma base de $V \otimes W$. Freqüentemente se usa a notação abreviada $|v\rangle\,|w\rangle$, $|v, w\rangle$ ou ainda $|vw\rangle$ para o produto tensorial $|v\rangle \otimes |w\rangle$.

Por definição, o produto tensorial satisfaz as seguintes propriedades:

1. Para um escalar z arbitrário, e elementos $|v\rangle$ de V e $|w\rangle$ de W,

$$z(|v\rangle \otimes |w\rangle) = (z\,|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z\,|w\rangle) \tag{5.3}$$

2. Para $|v_1\rangle$ e $|v_2\rangle$ arbitrários em V e $|w\rangle$ em W,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle \tag{5.4}$$

3. Para $|v\rangle$ arbitrário em V e $|w_1\rangle$ e $|w_2\rangle$ em W,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle \tag{5.5}$$

Mencionaremos a notação $|\psi\rangle^{\otimes k}$ para denotar o produto tensorial de $|\psi\rangle$ por ele mesmo k vezes.

Como exemplo, se nós temos um computador quântico com 2 q-bits e o primeiro q-bit está no estado $|0\rangle$ e o segundo está no estado $|1\rangle$, então o computador

quântico está no estado $|0\rangle \otimes |1\rangle$, dado por

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{5.6}$$

O vetor resultante está no espaço vetorial 4-dimensional. O estado geral $|\psi\rangle$ de um computador quântico com 2 q-bits é uma superposição dos estados $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$ (costuma-se representar tais estados na notação decimal como $|0\rangle$, $|1\rangle$, $|2\rangle$ e $|3\rangle$ respectivamente),

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle\,, \tag{5.7}$$

satisfazendo a $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$.

Em geral, o estado $|\psi\rangle$ de um computador quântico com n q-bits está numa superposição dos 2^n estados $|0\rangle$, $|1\rangle$,..., $|2^n - 1\rangle$,

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\,, \tag{5.8}$$

com as amplitudes α_i satisfazendo a

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{5.9}$$

Note que a base ortonormal $\{|0\rangle\,, \ldots, |2^n - 1\rangle\}$ é a base computacional na notação decimal. O estado de um computador quântico com n q-bits é um vetor no espaço vetorial complexo 2^n-dimensional.

Um espaço vetorial complexo V é um espaço de Hilbert se existe um produto interno, escrevemos na forma $\langle\varphi|\psi\rangle$, definido pelas seguintes regras ($a, b \in \mathbb{C}$ e $|\varphi\rangle$, $|\psi\rangle$, $|u\rangle$, $|v\rangle \in V$):

1. $\langle\psi|\varphi\rangle = \langle\varphi|\psi\rangle^*$,

2. $\langle\varphi|(a\,|u\rangle + b\,|v\rangle)\rangle = a\langle\varphi|u\rangle + b\langle\varphi|v\rangle$,

3. $\langle\varphi|\varphi\rangle > 0$ se $|\varphi\rangle \neq 0$.

onde $\langle\varphi|$ é o vetor dual (transposto conjugado) de $|\varphi\rangle$.

5.2 Operadores Unitários

Operações sobre q-bits devem preservar a norma, e portanto são descritas por matrizes unitárias. Algumas das mais importantes (considerando um q-bit) são as matrizes de Pauli, reproduzidas abaixo:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \qquad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; \qquad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{5.10}$$

A matriz X é a representação da porta quântica NOT.

Outros operadores quânticos importantes são descritos a seguir, a porta de Hadamard (denotada por H), a porta de fase (denotada por S) e a porta $\pi/8$ (denotada por T):

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix} \tag{5.11}$$

A porta Hadamard é uma das portas quânticas mais úteis. Esta porta é algumas vezes denominada de "raiz quadrada de NOT", e transforma $|0\rangle$ em $(|0\rangle + |1\rangle)/\sqrt{2}$ (primeira coluna de H), "meio caminho" entre $|0\rangle$ e $|1\rangle$, e transforma $|1\rangle$ em $(|0\rangle - |1\rangle)/\sqrt{2}$ (segunda coluna de H), também "meio caminho" entre $|0\rangle$ e $|1\rangle$. Note, contudo, que H^2 não é uma porta NOT; é fácil mostrar que $H^2 = I$, e portanto, aplicando H duas vezes, não altera o estado.

Mais detalhes sobre Computação Quântica pode ser encontrado em [Nielsen e Chuang (2000), Kaye et al. (2007), de Abreu (2004), Marquezino (2006), Lavor et al. (2003), Lomont (2004)].

5.3 Problema do Subgrupo Oculto

Uma motivação para estudar o Problema do Subgrupo Oculto (abreviaremos por PSO) é que a maioria dos algoritmos quânticos encontrados, que são exponencialmente mais rápidos do que seus equivalentes clássicos são casos particulares de algoritmos quânticos para a solução do Problema do Subgrupo Oculto. Como exemplo, citamos o Algoritmo de Simon [Simon (1994), Simon (1997)] e o Algoritmo de Shor [Shor (1994), Shor (1997), Lavor et al. (2003)] para fatoração de inteiros grandes. Os pesquisadores dessa área estão interessados em estender a família de grupos para os quais o PSO pode ser resolvido eficientemente, e assim, desenvolver algoritmos quânticos eficientes para problemas onde não existam algoritmos clássicos eficientes, tais como determinar isomorfismos de grafos [Beals (1997), Ettinger e Høyer (1999), Jozsa (2000), Ahn (2002), Lomont (2004)] ou encontrar o menor vetor em um reticulado [Regev (2002), Regev (2004)].

Em 1994, [Shor (1994)] construiu com base nos trabalhos de [Deutsch (1985)] e [Simon (1994)] um algoritmo quântico que pode fatorar inteiros grandes exponencialmente mais rápido do que qualquer método clássico conhecido, e assim abriu as portas para a pesquisa sobre computação quântica. Esse algoritmo permite a quebra dos principais códigos de criptografia usados atualmente, como RSA, Diffie-Hellman e ElGamal [Koblitz (1998)] caso um computador quântico de tamanho razoável esteja disponível. Shor também deu um algoritmo que resolve o problema de logaritmo discreto, que é usado em diversos outros criptosistemas. Kitaev [Kitaev (1996)] notou que esses algoritmos, assim como outros, enquadram-se num mesmo conjunto de problemas que consiste em encontrar geradores de um subgrupo a partir do grupo usando uma função que "oculta" o subgrupo, e assim o PSO estava nascendo.

Definição 23 Dada uma função eficientemente computável $f : G \to X$, de um grupo finito G para um conjunto X, que é constante nas classes laterais (à esquerda) de algum subgrupo H de G e que toma valores distintos nas distintas classes laterais de H em G, o problema do subgrupo oculto é achar um conjunto gerador para H.

Dizemos que o subgrupo H é "oculto" por f e a função f é chamada função separadora de classes laterais.

Não é conhecido nenhum algoritmo clássico eficiente para a solução do PSO. No entanto, através de algoritmos quânticos, tem se conseguido resolver, eficientemente, o problema para várias classes de grupos.

Para maiores detalhes sobre o Problema do Subgrupo Oculto, consultar as seguintes referências, [Lomont (2004), Mosca e Ekert (1999), Gonçalves (2005)].

5.3.1 Problema do Subgrupo Oculto Abeliano

Seja G um grupo abeliano finito. Sabemos do Teorema Fundamental para grupos abelianos finitos (Teorema 2) que G é isomorfo ao produto direto de grupos cíclicos, de ordens $t_1, t_2, ..., t_n$. Ou seja,

$$G \simeq \mathbb{Z}_{t_1} \times \mathbb{Z}_{t_2} \times ... \times \mathbb{Z}_{t_n}. \tag{5.12}$$

Por simplicidade assumiremos que G é igual a este produto direto. Seja $\mathcal{H}$ o espaço de Hilbert com base ortonormal $\{|g\rangle : g \in G\}$ indexada pelos elementos de G. Suponhamos um computador quântico atuando em $\mathcal{H}$ (sabemos que se o número de q-bits aumenta linearmente então $\dim \mathcal{H}$ aumenta exponencialmente). Portanto, para implementarmos isto fisicamente precisamos de $n = \lceil \log_2 |G| \rceil$ q-bits. O estado de um computador quântico com n q-bits é um vetor num espaço vetorial complexo de dimensão 2^n.

Definição 24 Um **caráter** de um grupo G é um homomorfismo de grupo de G para o grupo multiplicativo de números complexos diferentes de zero $\mathbb{C}^*$. Ou seja, é uma função de conjuntos $\chi : G \to \mathbb{C}^*$ tal que $\chi(g_1 + g_2) = \chi(g_1)\chi(g_2)$.

Supondo $G = \mathbb{Z}_t$, temos para cada elemento $g \in \mathbb{Z}_t$,

$$\chi_g(h) = e^{\frac{2\pi i}{t} gh} = \omega_t^{gh} \tag{5.13}$$

onde $\omega_t = e^{\frac{2\pi i}{t}}$ é a raiz complexa t-ésima da unidade.

Veja que g e h podem ser pensados como inteiros entre 0 e $t-1$. Para um caso mais geral, $G = \mathbb{Z}_{t_1} \times \mathbb{Z}_{t_2} \times ... \times \mathbb{Z}_{t_n}$, podemos pensar nos elementos g, h como $g = (g_1, g_2, ..., g_n)$ e $h = (h_1, h_2, ..., h_n)$. Neste caso, definimos

$$\chi_g(h) = \prod_{i=1}^{n} \omega_{t_i}^{g_i h_i} \tag{5.14}$$

Algumas propriedades dos caráteres.

Lema 3 Para qualquer g, $h \in G$, $\chi_g(h) = \chi_h(g)$.

Demonstração: Trivial, G é abeliano.

■

Lema 4 Considere $|G|$ vetores de valor complexo

$$|v_g\rangle = \frac{1}{\sqrt{|G|}} \begin{pmatrix} \chi_g(h_1) \\ \chi_g(h_2) \\ \vdots \\ \chi_g(h_{|G|}) \end{pmatrix} \tag{5.15}$$

Um vetor para cada $g \in G$, onde $h_1, ..., h_{|G|}$ é uma lista completa de elementos de G. Esses vetores são unitários e são ortogonais dois a dois. Em particular, nós temos para algum $g \neq 0$ que $\sum_i \chi_g(h_i) = 0$.

Demonstração: Ver [Damgard (2004)].

■

Construiremos uma matriz $|G| \times |G|$, cujas colunas são os vetores $|v_g\rangle$ definidos acima (Equação 5.15).

$$F_G = \frac{1}{\sqrt{|G|}} \begin{bmatrix} \chi_{g_1}(h_1) & \chi_{g_2}(h_1) & \dots & \chi_{g_{|G|}}(h_1) \\ \chi_{g_1}(h_2) & \chi_{g_2}(h_2) & \dots & \chi_{g_{|G|}}(h_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{g_1}(h_{|G|}) & \chi_{g_2}(h_{|G|}) & \dots & \chi_{g_{|G|}}(h_{|G|}) \end{bmatrix} \tag{5.16}$$

Segue das relações de ortogonalidade de caráteres [Serre (1977)] que esta matriz é unitária. Logo o operador definido por esta matriz, chamado F_G é unitário. Dizemos então que F_G é a **Transformada de Fourier** em grupos abelianos. Podemos também definí-la por sua atuação nos vetores da base como:

$$F_G \left|g\right\rangle = \frac{1}{\sqrt{|G|}} \sum_{h \in G} \chi_g(h) \left|h\right\rangle \tag{5.17}$$

A saída de F_G é o estado quântico correspondente ao vetor $|v_g\rangle$, que tem a forma acima quando escrito na notação usual para um estado. Já que o operador é unitário ele pode ser implementado num computador quântico.

Se $G = \mathbb{Z}_2 \times \mathbb{Z}_2 \times ... \times \mathbb{Z}_2$ é o grupo com a operação XOR (adição módulo 2), temos que $\omega_t = \omega_2 = -1$. Então nós podemos pensar nos elementos $g, h \in G$ como $n-$uplas $g = (g_1, ..., g_n)$, $h = (h_1, ..., h_n)$ onde g_i, h_i assumem os valores 0 ou 1. Consequentemente nós temos $\chi_g(h) = \prod_{i=1}^{n} (-1)^{g_i h_i}$. Usando isso na definição de F_G acima (Equação 5.17), nós temos:

$$\begin{aligned} F_G \left|g\right\rangle &= \frac{1}{\sqrt{|G|}} \sum_{h \in G} \chi_g(h) \left|h\right\rangle \\ &= \frac{1}{\sqrt{|G|}} \sum_{h \in G} (\prod_{i=1}^{n} (-1)^{g_i h_i}) \left|h\right\rangle \\ &= \frac{1}{\sqrt{|G|}} \sum_{h \in G} (-1)^{g \cdot h} \left|h\right\rangle , \end{aligned}$$

onde $g \cdot h$ é o produto interno usual. Em outras palavras, a transformada de Hadamard, $F_G = H^{\otimes n}$.

Definição 25 (O Subgrupo Ortogonal) Para qualquer subgrupo H de um grupo finito G, definimos o subgrupo ortogonal $H^{\perp}$, como

$$H^{\perp} = \{g \in G \mid \chi_g(h) = 1, \forall h \in H\} \tag{5.18}$$

Como G é finito e $\chi_{g'+g} = \chi_{g'} \chi_g$, ou seja, $H^{\perp}$ é fechado, temos que $H^{\perp}$ é realmente um subgrupo de G [Damgard (2004)].

O teorema abaixo mostra uma importante relação entre $H^{\perp}$ e H.

Teorema 12 Temos $H^{\perp} \simeq G/H$, em particular $|H^{\perp}| = |G|/|H|$.

Demonstração: Ver [Lomont (2004)].

■

Mostraremos agora que para um grupo abeliano genérico (finito) G, F_G produz uma superposição sobre os elementos em $H^{\perp}$, qualquer que seja $H < G$.

Lema 5 Para qualquer classe lateral H_i de H em G, temos

$$F_G\big(\frac{1}{\sqrt{|H|}}\sum_{g\in H_i}|g\rangle\big) = \frac{1}{\sqrt{|H^{\perp}|}}\sum_{h\in H^{\perp}}\chi_h(g_i)\,|h\rangle \tag{5.19}$$

onde g_i é um elemento fixo representante da classe lateral H_i.

Demonstração: Ver [Damgard (2004), Gonçalves (2005)].

■

Apresentaremos agora um algoritmo eficiente (método padrão de solução para grupos cíclicos) para o PSO, isto é, um algoritmo que rode em tempo polinomial em $\log_2|G|$ [Damgard (2004), Lomont (2004)].

Assumiremos que temos dois registradores: um registrador com $n = \lceil \log_2|G| \rceil$ q-bits e o segundo registrador com m q-bits, onde $n \geq m$. O algoritmo usa a seguinte subrotina:

Passo 1. Inicialize o computador quântico no estado $|0_G\rangle\,|0^m\rangle$, onde $|0_G\rangle$ é o estado da base correspondente ao elemento neutro de G. Depois aplique F_G no primeiro registrador para obter

$$\frac{1}{\sqrt{|G|}}\sum_{g\in G}|g\rangle\,|0^m\rangle\,.$$

Passo 2. Aplicando U_f no estado anterior, obtemos

$$\frac{1}{\sqrt{|G|}}\sum_{g\in G}|g\rangle\,|f(g)\rangle\,.$$

Repare, que como U_f é linear, ele atua em todos os $|g\rangle\,|0^m\rangle$ para $|G|$ valores de g, então isto gera todos os $f(g)$ simultaneamente. Este é o chamado **paralelismo quântico**.

Passo 3. Meça o segundo registrador do estado anterior. A medida, segundo um postulado da mecânica quântica, provoca um distúrbio no estado original. Este estado por sua vez é levado em

$$\frac{1}{\sqrt{|H|}} \sum_{g_0 \in H_i} |g_0\rangle\,|f(g_0)\rangle\,.$$

onde H_i é alguma classe lateral de H. A constante foi renormalizada já que sobraram apenas $|H|$ elementos na soma. Estes elementos são aqueles cuja imagem é $f(g_0)$.

Passo 4. Aplique F_G no primeiro registrador e obtenha então uma superposição sobre $H^\perp$. Pelo Lema 5, temos

$$\frac{1}{\sqrt{|H^\perp|}} \sum_{h \in H^\perp} \chi_h(g_i)\,|h\rangle\,|f(g_0)\rangle\,.$$

onde g_i é um elemento fixo representante da classe lateral H_i.

Passo 5. Meça agora o primeiro registrador. A medida produz um elemento randômico em $H^\perp$.

Passo 6. Segue do Teorema 1 que se este procedimento for repetido um número de vezes logaritmico em $|G|$ (logo polinomial em n) obtemos um conjunto de geradores para $H^\perp$. Usando as relações entre H e $H^\perp$ podemos calcular um conjunto de geradores para H com probabilidade próxima de 1.

5.3.2 Problema do Subgrupo Oculto Não Abeliano

Porque nós queremos encontrar os subgrupos ocultos de grupos não-abelianos? Já vimos que um algoritmo eficiente para o PSO abeliano produz um algoritmo

de fatoração de inteiros que é exponencialmente mais rápido do que qualquer algoritmo clássico conhecido. Similarmente, encontrar algoritmos eficientes para o PSO sobre certos grupos não-abelianos poderia produzir algoritmos mais rápidos do que qualquer algoritmo clássico para diversos problemas importantes.

Uma das principais razões que levam muitos pesquisadores a estudar o PSO para grupos não-abelianos é o desejo em encontrar um algoritmo eficiente para o problema de isomorfismos de grafos. Uma redução (ver Seção 6.1) mostra que se o PSO puder ser resolvido eficientemente para o grupo simétrico S_n, então nós teríamos um algoritmo em tempo polinomial para o problema de isomorfismo de grafos [Beals (1997), Ettinger e Høyer (1999), Jozsa (2000), Ahn (2002), Lomont (2004)].

Outra razão é que um algoritmo quântico eficiente para resolver o PSO para o grupo diedral D_n produz um algoritmo rápido para encontrar o menor vetor num reticulado, primeiro mostrado por [Regev (2002)]. Isso poderia produzir outro algoritmo cuja a contrapartida clássica é menos eficiente do que a versão quântica. Encontrar o menor vetor num reticulado tem muitas aplicações, incluindo aplicações para a criptografia. Foi desenvolvido por [Kuperberg (2005)] um algoritmo quântico subexponencial no tamanho da entrada. Posteriormente, [Regev (2004)] apresentou uma versão melhorada do algoritmo de Kuperberg, onde o tempo de processamento do algoritmo ainda é subexponencial, mas a quantidade de memória utilizada é polinomial no tamanho da entrada.

Embora grande esforço esteja sendo empregado, estes dois casos do PSO não abeliano, continuam em aberto e desafiando a comunidade científica. Mas, alguns sucessos em grupos não abelianos foram conseguidos. Por exemplo,[Hallgren et al. (2000)] demonstraram que existe um algoritmo quântico eficiente para a solução do PSO em um grupo qualquer desde que a Transformada de Fourier Quântica seja eficientemente implementável no grupo e que o subgrupo oculto seja normal.

Foi mostrado [Ivanyos et al. (2003)] ser possível resolver eficientemente o PSO em grupos não abelianos onde a ordem do subgrupo de comutadores seja pequena

em relação à ordem do grupo, no sentido de que $|G'|$ seja $O(\log|G|)$.

Como o grupo diedral pode ser escrito como o produto semidireto $\mathbb{Z}_n \rtimes \mathbb{Z}_2$, um caminho natural foi estudar novos algoritmos quânticos para o PSO para grupos não abelianos que se escrevem como o produto semidireto de grupos cíclicos. Em [Inui e Le Gall (2005)], os autores apresentaram uma solução para o PSO no grupo $\mathbb{Z}_{p^r} \rtimes \mathbb{Z}_p$, p inteiro primo e r um inteiro positivo. Neste trabalho os autores utilizaram a Transformada de Fourier Quântica no grupo abeliano. Seguindo esta linha, [Cosme e Portugal (2007)] apresentaram um algoritmo quântico eficiente para o PSO, no produto semidireto de grupos cíclicos $\mathbb{Z}_{p^r} \rtimes_\phi \mathbb{Z}_{p^2}$, onde p é qualquer número primo e r é um inteiro qualquer tal que $r > 4$. Também foi abordado o PSO no grupo $\mathbb{Z}_N \rtimes_\phi \mathbb{Z}_{p^2}$, onde N é um número inteiro com uma fatoração prima especial. Estes algoritmos quânticos são exponencialmente mais rápido do que qualquer algoritmo clássico para o mesmo fim. Também nesta direção, [Cosme (2008)] apresentou um algoritmo quântico eficiente para o PSO no produto semidireto $\mathbb{Z}_{p^r} \rtimes_\phi \mathbb{Z}_{p^s}$, onde p é qualquer número primo ímpar, r e s são inteiros positivos e o homomorfismo ϕ está em função de p, r e s. Como consequência, pode-se resolver eficientemente o PSO também no grupo $\mathbb{Z}_N \rtimes_\phi \mathbb{Z}_{p^s}$, onde o inteiro $\mathbb{N}$ possui uma fatoração prima especial.

Foi dado por [Ivanyos et al. (2007a)] um algoritmo quântico eficiente para a classe dos grupos extra-especiais. Num trabalho posterior, os mesmos autores resolveram o problema do PSO para a classe de grupos nilpotentes de classe 2 [Ivanyos et al. (2007b)]. Para uma leitura mais detalhada sobre este último problema podemos também consultar [Fernandes (2008)].

Para mais detalhes sobre o PSO não abeliano, podemos consultar [Ivanyos et al. (2003), Inui e Le Gall (2005), Ivanyos et al. (2007a), Cosme (2008)].

Capítulo 6

Aplicações da Computação Quântica ao Problema do Isomorfismo de Grafos

Nesse capítulo, vamos mostrar que uma solução eficiente do PSO no grupo simétrico S_n implicará num algoritmo de tempo polinomial para decidir se dois grafos são isomorfos ou não [Ahn (2002)]. Utilizaremos o Problema de Interseção de Grupos de permutações [Hoffman (1979), Fenner e Zhang (2005)] para encontrar um algoritmo quântico para o Problema do Isomorfismo de Grafos para a classe de grafos A_n (definida na Seção 4.2), mostrando assim o poder da Computação Quântica.

6.1 Redução do Problema do Isomorfismo de Grafos para o Problema do Subgrupo Oculto

Existem algumas maneiras de reduzir o PIG para o PSO [Beals (1997), Ettinger e Høyer (1999), Jozsa (2000), Ahn (2002), Lomont (2004)]. Nesta seção vamos apresentar o modelo descrito por [Ahn (2002)]. A redução possui dois passos: No primeiro passo, o PAG é reduzido para o PSO no grupo simétrico S_n e então, no segundo passo, o PIG é reduzido para o PAG.

Podemos pensar em S_n agindo num grafo Γ de ordem n como a seguir: Cada elemento $\sigma \in S_n$ é uma permutação (ou reordenação) dos números $\{1, 2, \ldots, n\}$; podemos pensar em $\sigma\Gamma$ como uma reclassificação (nova rotulação) dos n vértices de Γ. Por exemplo, se σ é a permutação que leva $\{1, 2, 3, 4\}$ para $\{4, 3, 2, 1\}$ e Γ é

como na Figura 6.1, então $\sigma\Gamma$ é como na Figura 6.2.

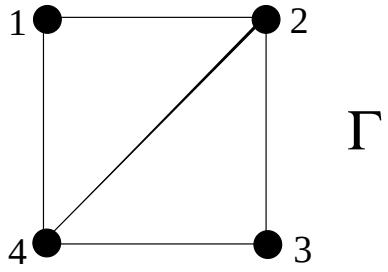

Figura 6.1: Grafo Γ

Uma boa maneira de visualizar um automorfismo de Γ é pensar a respeito da ação de S_n em Γ. Elementos de S_n renomeiam os vértices de Γ: Começamos com um grafo Γ (como na Figura 6.1) e então, sem mover nenhuma aresta, nós simplesmente renomeamos os vértices; o resultado deste processo é mostrado no lado esquerdo da Figura 6.2 para a permutação $\sigma = (1,4)(2,3)$. Após isso, podemos mover os vértices para suas posições originais; o resultado desse movimento é mostrado no lado direito da Figura 6.2. Um automorfismo de um grafo é um elemento de S_n para o qual os grafos inicial e final deste processo coincidem perfeitamente. Portanto $\sigma = (1,4)(2,3)$ não é um automorfismo. Denotamos por fix($\sigma\Gamma$) a representação do grafo que resulta deste processo (assim, em nosso exemplo, a representação do lado direito da Figura 6.2 é fix($\sigma\Gamma$). Com esta notação, σ é um automorfismo se e só se fix($\sigma\Gamma$) =fix($e\Gamma$), onde e é a permutação identidade de S_n.

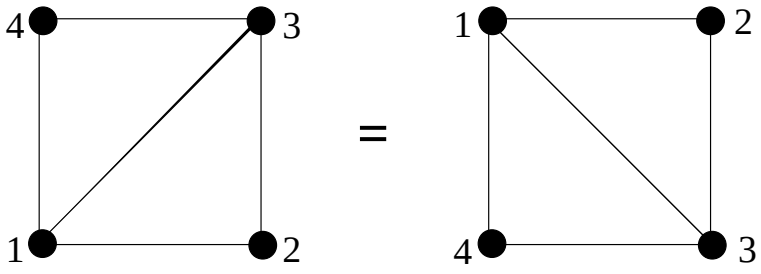

Figura 6.2: Grafo $\sigma\Gamma$

Vamos denotar por $\mathcal{R}$ o conjunto de todas as representações dos grafos fix($\sigma\Gamma$) para todo $\sigma \in S_n$.

Mostraremos agora que o PAG pode ser visto como um PSO, PAG $\propto_p$ PSO (PAG é redutível polinomialmente para o PSO, ou seja, resolver eficientemente o PSO implica resolver eficientemente o PAG). De fato, seja Γ um grafo com n vértices, consideremos $V = \{1, \ldots, n\}$. Seja f_Γ uma função de S_n que leva cada elemento (permutação) σ para fix($\sigma\Gamma$). A saber,

$$f_\Gamma : S_n \to \mathcal{R}$$
$$\sigma \mapsto \text{fix}(\sigma\Gamma)$$

Sejam $\sigma_1\text{Aut}(\Gamma), \sigma_2\text{Aut}(\Gamma), \ldots, \sigma_k\text{Aut}(\Gamma)$ classes laterais de Aut(Γ). Sabemos que um grupo finito pode ser escrito como a união disjunta de suas classes laterais (ver Seção 2.1), ou seja, $S_n = \sigma_1\text{Aut}(\Gamma) \cup \sigma_2\text{Aut}(\Gamma) \cup \ldots \cup \sigma_k\text{Aut}(\Gamma)$, $\sigma_i \in S_n$, $\sigma_i \neq \sigma_j$. Segue então que f_Γ é constante nas classes laterais de Aut(Γ) e distinta em cada classe lateral. Portanto, PAG é um PSO, onde $G = S_n$, $X = \mathcal{R}$, H =Aut(Γ) e f_Γ é a função que esconde o subgrupo Aut(Γ), de acordo com a Definição 23.

Agora, mostraremos o último passo, ou seja, um algoritmo eficiente para resolver o PAG implica num algoritmo eficiente para resolver o PIG [Mathon (1979)].

Teorema 13 O Problema do Isomorfismos de Grafos é redutível polinomialmente para o Problema do Automorfismos de Grafos, PIG $\propto_p$ PAG.

Demonstração: Sejam $\Gamma_1 = (V_1, E_1)$ e $\Gamma_2 = (V_2, E_2)$ dois grafos conexos com n vértices cada e seja $\Gamma_3 = \Gamma_1 \cup \Gamma_2$ a união disjunta de Γ_1 e Γ_2 (ver Definição 20). Suponhamos que exista um algoritmo eficiente para resolver o PAG. Aplicamos então tal algoritmo em $\Gamma_3 = \Gamma_1 \cup \Gamma_2$. Se $\exists\ \sigma \in \text{Aut}(\Gamma_3)$ tal que $\sigma(v) = u$, para algum $v \in V_1$, $u \in V_2$ então para qualquer $v' \in V_1$ teremos $\sigma(v') = u'$, $u' \in V_2$, pois os conjuntos V_1 e V_2 são disjuntos e σ leva aresta em aresta. Então teremos $\sigma(V_1) = V_2$, portanto $\Gamma_1 \simeq \Gamma_2$. Caso contrário, ou seja, se $\nexists\ \sigma \in \text{Aut}(\Gamma_3)$ com tal propriedade então $\Gamma_1 \not\simeq \Gamma_2$.

■

Já tínhamos do Teorema 9 que resolver eficientemente o PIG implica em resolver eficientemente o PAG. Portanto os dois problemas são equivalentes polinomialmente. Na verdade, temos um conjunto de problemas equivalentes polinomialmente [Mathon (1979)]. Os Problemas 1 e 2 são também equivalentes aos seguintes problemas: Construir explicitamente um isomorfismo entre grafos; determinar o número de isomorfismos entre grafos e determinar a ordem do grupo de automorfismos do grafo.

6.2 Grupos Oráculo

O modelo de grupos Oráculo foi introduzido por [Babai e Szemerédi (1984)] como uma estrutura geral para estudos de problemas algorítmicos para grupos finitos e desde então têm sido estudados extensivamente. Temos a seguinte definição:

Definição 26 Seja um grupo G tal que:

(i) Cada elemento de G pode ser codificado como uma palavra binária de mesmo comprimento, isto é, número de bits. Este número é chamado **comprimento** da codificação;

(ii) Há um **oráculo (Caixa Preta)** que realiza a operação do grupo nesta codificação com custo unitário;

(iii) Caso a codificação não seja única, isto é, um elemento em G possuir mais de uma palavra o representando, então um segundo oráculo é requerido para testar identidade na codificação de G.

Se o grupo G é dado por um conjunto de palavras representando geradores para o mesmo, então, ele é chamado um **grupo oráculo**.

Assim, um grupo oráculo com comprimento de codificação n tem ordem limitada por 2^n, o número de palavras binárias de comprimento n, portanto todo grupo oráculo é finito. Note que nem toda palavra binária de comprimento n necessariamente corresponde a um elemento do grupo. Sendo assim, podemos

imaginar que nosso grupo oráculo tenha algum comportamento arbitrário dadas codificações inválidas. Logo, não o acessaremos com um elemento inválido do grupo. Se dissermos que um grupo particular ou subgrupo oráculo é dado (para algum algoritmo), queremos dizer com isso que um conjunto de palavras que geram o grupo ou subgrupo é dado.

Admitiremos que os grupos oráculo aqui tratados têm codificação única, fato este que evita a exigência do segundo oráculo (Caixa Preta) da Definição 26. Dado um grupo oráculo G de comprimento de codificação n, teremos associado uma porta quântica (transformação unitária) U_G agindo sobre $2n$ q-bits como segue:

$$U_G \left|g\right\rangle \left|h\right\rangle = \left|g\right\rangle \left|gh\right\rangle . \tag{6.1}$$

Aqui, como já foi dito, assumimos que g e h são dados como codificações válidas do grupo. A inversa de U_G é

$$U_G^{-1} \left|g\right\rangle \left|h\right\rangle = \left|g\right\rangle \left|g^{-1}h\right\rangle . \tag{6.2}$$

As portas U_G e U_G^{-1} são os oráculos do grupo. São elas que efetuam a operação do grupo na codificação especificada.

Para saber como implementar grupos oráculo na forma de circuitos quânticos, ver [Watrous (2001)].

6.3 Problemas relacionados com o Problema do Subgrupo Oculto

Em [Friedl et al. (2003)] foram introduzidos diversos problemas que estão intimamente relacionados com o Problema do Subgrupo Oculto. Em particular, eles introduziram o **Problema Estabilizador** que generaliza o Problema do Subgrupo Oculto. De fato, a única diferença entre Problema Estabilizador e o Problema do Subgrupo Oculto é que na definição do Problema Estabilizador a função f pode ser uma função quântica que leva elementos do grupo para estados quânticos mutuamente ortogonais com norma unitária.

Seja G um grupo finito. Seja Δ um conjunto de estados quânticos mutuamente ortogonais. Seja $\alpha : G \times \Delta \to \Delta$ uma ação de grupo de G sobre Δ, isto é, para todo $x \in G$ a função $\alpha_x : |\phi\rangle \to |\alpha(x, |\phi\rangle)\rangle$ é uma permutação sobre Δ e a função h de G para o grupo simétrico sobre Δ definida por $h(x) = \alpha_x$ é um isomorfismo.

Usaremos a notação $|x \cdot \phi\rangle$ para o estado $|\alpha(x, |\phi\rangle)\rangle$, quando α é claro no contexto. Seja $G(|\phi\rangle)$ denotando o conjunto $\{|x \cdot \phi\rangle = |\phi\rangle\}$, e seja $G_{|\phi\rangle}$ denotando o subgrupo Estabilizador de $|\phi\rangle$ em G, isto é, $\{x \in G; |x \cdot \phi\rangle = |\phi\rangle\}$. Dado um inteiro positivo t, seja α^t a ação do grupo G sobre $\Delta^t = \{|\phi\rangle^{\otimes t}; |\phi\rangle \in \Delta\}$ definido por $\alpha^t(x, |\phi\rangle^{\otimes t}) = |x \cdot \phi\rangle^{\otimes t}$. Nós precisamos de α^t porque as superposições de entrada não podem ser clonadas em geral (Teorema da Não-Clonagem).

Agora, definiremos o chamado Problema Estabilizador, que consiste em encontrar $O(\log |G|)$ geradores para o subgrupo $G_{|\phi\rangle}$.

Problema 4 Seja G um grupo finito e Δ um conjunto de estados quânticos mutuamente ortogonais. Fixamos a ação de grupo $\alpha : G \times \Delta \to \Delta$. Então, dados geradores para G e estados quânticos $|\phi\rangle \in \Delta$, o Problema Estabilizador consiste em encontrar um conjunto gerador para o subgrupo $G_{|\phi\rangle} = \{x \in G; |x \cdot \phi\rangle = |\phi\rangle\}$.

Para grupos abelianos finitos, o Problema Estabilizador é resolvido eficientemente por um algoritmo quântico com erro ϵ [Kitaev (1996)]. Além disso, os problemas de fatoração e do logaritmo discreto podem ser reduzidos para o Problema Estabilizador Abeliano.

Se o grupo G for solúvel tal problema pode ser resolvido em tempo polinomial quântico sobre certos critérios de solubilidade fortes. De fato, temos o seguinte teorema devido a [Friedl et al. (2003)].

Teorema 14 Seja G um grupo solúvel finito que tem um subgrupo comutador suavemente solúvel e seja α uma ação de grupo de G. Quando $t = (\log^{\Omega(1)} |G|) \times \log(1/\epsilon)$, o Problema Estabilizador pode ser resolvido em G para α^t em tempo $poli(\log |G|)\log(1/\epsilon)$ quântico com erro ϵ.

Foram descritas por [Fenner e Zhang (2005)] reduções quânticas para vários problemas. Algoritmos quânticos para esses problemas freqüentemente requerem diversas cópias idênticas de um estado quântico ou porta unitária para trabalhar com uma precisão desejada. Consequentemente, vamos assumir implicitamente que tais reduções podem ser repetidas t vezes, onde t é algum parâmetro polinomial apropriado no tamanho da entrada e logarítmico no tamanho do erro desejado.

6.4 Algoritmo Quântico

Relataremos progressos conseguidos por [Fenner e Zhang (2005)] em encontrar algoritmos quânticos para Interseção de Grupos. Mostraremos que se um dos grupos em questão for solúvel, existe um algoritmo quântico eficiente para o Problema de Interseção de Grupos se tal grupo possui um subgrupo comutador suavemente solúvel. Além disso, segue como corolário que se ambos os grupos forem solúveis existe um algoritmo quântico eficiente para o Problema de Interseção de Grupos se um dos grupos em questão possui um subgrupo comutador suavemente solúvel.

Antes disso, um algoritmo quântico eficiente para computar a ordem de um grupo solúvel foi dado por [Watrous (2001)], além disso, ele obteve um método para construir (aproximadamente) superposições quânticas uniformes sobre elementos deste grupo solúvel. Conforme vemos no teorema abaixo.

Teorema 15 (Watrous (2001)) No modelo de grupos Oráculo com codificação única, existe um algoritmo quântico operando da seguinte forma (relativo a um grupo oráculo arbitrário). Dados geradores $g_1, \ldots, g_m$ tais que $G = \langle g_1, \ldots, g_m \rangle$ é solúvel, a saída do algoritmo é a ordem de G com probabilidade de erro limitado por ϵ em tempo polinomial em $mn + \log(1/\epsilon)$(onde n é o comprimento das palavras que representam os geradores). Além disso, o algoritmo produz um estado quântico ρ que aproxima o estado $|G\rangle = |G|^{-1/2} \Sigma_{g \in G} |g\rangle$ com precisão ϵ(no traço da norma métrica).

Agora, podemos enunciar um teorema que relaciona o Problema de Interseção

de Grupos com o Problema Estabilizador. Na verdade, o primeiro problema reduz-se para o segundo, tendo como hipótese adicional um dos grupos em questão ser solúvel.

Teorema 16 O Problema de Interseção de Grupos reduz-se para o Problema Estabilizador em tempo polinomial quântico com erro limitado se um dos grupos em questão for solúvel.

Demonstração: Dada uma entrada (A, B, n) para o Problema de Interseção de Grupos, sem perda de generalidade, suponha que $G = \langle A \rangle$ é um grupo finito arbitrário e $H = \langle B \rangle$ é solúvel. Pelo Teorema 15 nós podemos construir (aproximadamente) uma superposição uniforme $|H\rangle = |H|^{-1/2} \Sigma_{h \in H} |h\rangle$. Para qualquer $g \in G$, seja $|gH\rangle$ denotando a superposição uniforme sobre classes laterais gH, isto é, $|gH\rangle = |H|^{-1/2} \Sigma_{h \in gH} |h\rangle$. Seja $\Delta = \{|gH\rangle \, ; g \in G\}$. Note que os estados quânticos em Δ são (aproximadamente) ortogonais dois a dois. Definimos a ação de grupo como $\alpha : G \times \Delta \to \Delta$ para cada $g \in G$ e cada $|\phi\rangle \in \Delta$, $\alpha(g, |\phi\rangle) = |g\phi\rangle$. Então a interseção de G e H é exatamente o subgrupo de G que estabiliza o estado quântico $|H\rangle$, a saber, $G_{|H\rangle} = \{g \in G; |g \cdot H\rangle = |H\rangle\}$.

■

Corolário 5 O Problema de Interseção de Grupos sobre grupos solúveis pode ser resolvido com erro ϵ por um algoritmo quântico que corre em tempo polinomial em $m + \log(1/\epsilon)$, onde m é o tamanho da entrada, dado que um dos grupos solúveis em questão tenha um subgrupo comutador suavemente solúvel.

Demonstração: Segue diretamente dos Teoremas 16 e 14.

■

6.5 Exemplo

A classe de grupos solúveis é a maior classe para a qual existe algoritmos quânticos eficientes até o presente momento, por exemplo o cálculo da ordem do grupo. No entanto, o método descrito no Teorema 11 requer algoritmos quânticos

eficientes para os grupos $S_n^{'}$ e T que não são solúveis. Se o grupo de automorfismos do grafo for solúvel, podemos usar uma estratégia de tentativa e erro com grupos solúveis que sejam subgrupos de $S_n^{'}$ e T. Se o grupo de automorfismos de um grafo estiver na interseção desses grupos solúveis, os geradores de Aut(Γ) podem ser encontrados eficientemente, desde que um dos grupos solúveis possua um subgrupo comutador **suavemente** solúvel. Essa restrição adicional é exigida no Corolário 5. A seguir vamos dar um exemplo onde essa estratégia funcione.

Exemplo 8 Considere a classe de grafos $A_n = (V_{A_n}, E_{A_n})$, onde $V_{A_n} = \{a_i, b_i, c_i : 0 \leq i < n\}$ e $E_{A_n} = \{(a_i, a_{i+1}), (a_i, b_i), (a_i, c_i), (b_i, c_i), (c_i, a_{i+1}) : 0 \leq i < n\}$, como descrito na Seção 4.2. Se $n = 3$ temos o grafo da Figura 4.2.

As não-arestas de A_n são dadas pelo seguinte conjunto

$$\overline{E}_{A_n} = \{(a_i, a_j) : i \neq j, j \neq i \pm 1; (a_i, b_j) : i \neq j; (a_i, c_j) : i \neq j, j \neq i - 1; \\ (b_i, b_j) : i \neq j; (b_i, c_j) : i \neq j; (c_i, c_j) : i \neq j; 0 \leq i, j < n\}.$$

Sabemos que $|E_{A_n}| = 5n$ e que $|E_{A_n}| + |\overline{E}_{A_n}| = |K_{3n}|$. Portanto,

$$|\overline{E}_{A_n}| = 3n(3n-1)/2 - 5n = (9n^2 - 13n)/2. \tag{6.3}$$

Agora, reescreveremos o conjunto E_{A_n} de tal forma que 1 represente (a_0, a_1), 2 represente (a_1, a_2), assim por diante, até n representando (a_{n-1}, a_0). Da mesma forma $n+1$ representa (a_0, b_0), $n+2$ representa (a_1, b_1), assim por diante, até $2n$ representando (a_{n-1}, b_{n-1}). Continuando, $2n+1$ representa (b_0, c_0), $2n+2$ representa (b_1, c_1), assim por diante, até $3n$ representando (b_{n-1}, c_{n-1}). Prosseguindo, $3n+1$ representa (a_0, c_0), $3n+2$ representa (a_1, c_1), assim por diante, até $4n$ representando (a_{n-1}, c_{n-1}). Finalmente $4n+1$ representando (a_1, c_0), $4n+2$ representa (a_2, c_1), assim por diante, até $5n$ representando (a_n, c_{n-1}). Da mesma forma, para o conjunto $\overline{E}_{A_n}$, vamos representar (b_0, a_1) por $5n+1$, (b_1, a_2) por $5n+2$, assim por diante, até (b_{n-1}, a_0) sendo representado por $6n$. Não identificaremos as outras não-arestas, pois elas não serão referidas posteriormente.

Definiremos agora o conjunto T como sendo o produto direto de $\mathcal{P}(E_{A_n})$ com $\mathcal{P}(\overline{E}_{A_n})$, ou seja, $T = \mathcal{P}(E_{A_n}) \times \mathcal{P}(\overline{E}_{A_n})$. Notemos que $|\mathcal{P}(E_{A_n})| = 5n!$ e que $|\mathcal{P}(\overline{E}_{A_n})| = (\frac{9n^2-13n}{2})!$. As permutações $(1,2)$,$(1,2,\ldots,5n)$,$(5n+1,5n+2)$ e $(5n+1,5n+2,\ldots,3n(3n-1)/2)$ formam um conjunto gerador para T.

Por outro lado, construiremos também o grupo S'_{3n} a partir de S_{3n}, usando o seguinte homomorfismo, conforme Proposição 6.

$$\psi_\pi(i,j) = (\pi(i),\pi(j)), \pi \in S_{3n}, 1 \leq i,j \leq 3n, i \neq j.$$

Logo,

$$S'_{3n} = \{\psi_\pi; \psi_\pi(i,j) = (\pi(i),\pi(j)), \pi \in S_{3n}, 1 \leq i,j \leq 3n, i \neq j\} < S_{3n(3n-1)/2}.$$

Como $\pi_1 = (1,2)$ e $\pi_2 = (1,2,\ldots,3n)$ são geradores para o grupo S_{3n}, então ψ_{π_1} e ψ_{π_2} são geradores para o grupo S'_{3n}.

Assim, pelo Teorema 11 temos que $\mathrm{Aut}(A_n) \simeq S'_{3n} \cap T$. Logo, achar $\mathrm{Aut}(A_n)$ reduz-se a achar $S'_{3n} \cap T$. Além disso, o Teorema 16 nos diz que o problema de interseção de grupos reduz-se ao problema estabilizador em tempo polinomial quântico com erro limitado se um dos grupos em questão for solúvel. Mais ainda, se ambos os grupos forem solúveis, o Corolário 5 nos diz que o problema de interseção de grupos pode ser resolvido eficientemente, desde que um dos grupos solúveis possua um subgrupo comutador suavemente solúvel. A restrição de subgrupos comutadores suavemente solúveis infelizmente reduz bastante a possibilidade de uso de algoritmos quânticos para a solução do PIG.

Um exemplo onde a estratégia adotada nessa seção funciona é tomar qualquer grupo abeliano subgrupo do S'_{3n} que contenha o grupo $\mathrm{Aut}(A_n)$ e o mesmo para T. Uma vez que o subgrupo comutador de grupos abelianos é suavemente solúvel, o Corolário 5 exibe um algoritmo quântico que acha os geradores de $\mathrm{Aut}(A_n)$ eficientemente.

Exemplo 9 Vamos dar um exemplo que mostra a limitação dos algoritmos quân-

ticos desenvolvidos até o presente momento para determinar eficientemente os geradores da interseção de dois grupos solúveis. Vamos tomar dois grupos diedrais, um subgrupo de S'_{3n} e outro subgrupo de T cuja interseção é o grupo $\mathrm{Aut}(A_n)$.

Vamos supor que n seja par, no entanto, se n for ímpar o método é o mesmo, só as permutações serão diferentes. Sejam então as seguintes permutações:

$$g_1 = \overbrace{(1,2,...,n)}^{\alpha_1}\overbrace{(n+1,...,2n)}^{\alpha_2}\overbrace{(2n+1,...,3n)}^{\alpha_3}\overbrace{(3n+1,...,4n)}^{\alpha_4}\overbrace{(4n+1,...,5n)}^{\alpha_5},$$

$$\begin{aligned} g_2 \;=\; & \overbrace{(1,n)(2,n-1)...(n/2,n/2)}^{\beta_1} \\ & \overbrace{(n+1,2n)(n+2,2n-1)...(n+n/2,2n-n/2)}^{\beta_2} \\ & \overbrace{(2n+1,3n)(2n+2,3n-1)...(2n+n/2,3n-n/2)}^{\beta_3} \\ & \overbrace{(3n+1,4n)(3n+2,4n-1)...(3n+n/2,4n-n/2)}^{\beta_4} \\ & \overbrace{(4n+1,5n)(4n+2,5n-1)...(4n+n/2,5n-n/2)}^{\beta_5}. \end{aligned}$$

Notemos que o grupo $\langle \alpha_1, \beta_1 \rangle$ é isomorfo ao grupo diedral. De fato, $(\alpha_1)^n = (\beta_1)^2 = e$. Além disso, verifica-se que $\alpha_1\beta_1 = \beta_1(\alpha_1)^{-1}$.

Da mesma forma, os grupos $\langle \alpha_2, \beta_2 \rangle$, $\langle \alpha_3, \beta_3 \rangle$, $\langle \alpha_4, \beta_4 \rangle$ e $\langle \alpha_5, \beta_5 \rangle$ são isomorfos a grupos diedrais. Neste caso, podemos afirmar que o grupo $D = \langle g_1, g_2 \rangle$ é isomorfo ao grupo diedral.

Afirmamos que $D < T$, pois g_1 e g_2 levam arestas em arestas e não-arestas para não-arestas (não estamos movendo as não-arestas).

Note também que $g_2 \notin S'_{3n}$. Para estar em S'_{3n}, a permutação tem que ser feita nos vértices. Uma permutação de vértices inverte a diagonal gerando uma permutação diferente de g_2, pois envolve também não-arestas. A permutação g_2 foi construída fazendo permutações de arestas sem compromisso com as permutações de vértices.

Agora, renomearemos os vértices de A_n usando a mesma lógica empregada

para as arestas. Os vértices $a_0, a_1, \ldots, a_{n-1}$ serão representados por $1, 2, \ldots, n$, respectivamente. Prosseguindo, os vértices $b_0, b_1, \ldots, b_{n-1}$ serão representados por $n+1, n+2, \ldots, 2n$, respectivamente. Finalmente, os vértices $c_0, c_1, \ldots, c_{n-1}$ serão representados por $2n+1, 2n+2, \ldots, 3n$, respectivamente.

Nessa nova representação, definiremos as seguintes permutações de S_{3n},

$$h_1 = (1, 2, ..., n)(n+1, ..., 2n)(2n+1, ..., 3n)$$

$$h_2 = (2, n)(3, n-1)...(n/2, n/2)(n+1, 3n)(n+2, 3n-1)...(n+n/2, 3n-n/2)$$

Note que $\langle h_1, h_2 \rangle$ é isomorfo ao grupo diedral nessa nova representação. De fato, $(h_1)^n = (h_2)^2 = e$. Verifica-se também que

$$h_1 h_2 = h_2 h_1^{-1} \tag{6.4}$$

Podemos ver isso geometricamente, ou seja, no lado esquerdo da equação 6.4, temos uma reflexão seguida por uma rotação normal, enquanto no lado direito da equação temos uma rotação inversa seguida por uma reflexão.

No isomorfismo que leva S_{3n} em S'_{3n}, é fácil ver que h_1 é levado em g_1 (descrito no início do exemplo).

A permutação h_2 será levada pelo isomorfismo de S_{3n} em S'_{3n} para uma permutação $g_2^{'} \in S'_{3n}$. Portanto, o grupo $D^{'} = \langle g_1, g_2^{'} \rangle$ ($D^{'} < S'_{3n}$) será isomorfo ao grupo diedral nessa nova representação, ou seja, $D^{'}$ é um grupo solúvel. A permutação $g_2^{'}$ é dada por

$$\begin{aligned} g_2^{'} \;=\; & (1, n)(2, n-1)...(n/2, n/2) \\ & (n+1, 5n)(n+2, 5n-1)...(2n, 4n+1) \\ & (2n+1, 3n)(2n+2, 3n-1)...(2n+n/2, 3n-n/2) \\ & (3n+1, 6n)(3n+2, 6n-1)...(4n, 5n+1) \end{aligned}$$

Portanto, exibimos D e $D^{'}$ numa representação compatível. Esses grupos são

grupos solúveis e são subgrupos de T e S'_{3n}, respectivamente. No entanto, o grupo diedral não possui subgrupo comutador suavemente solúvel, como exige o Corolário 5. De fato, o subgrupo comutador do grupo diedral é $\langle \rho^2 \rangle$, onde ρ representa uma rotação, como descrito na Definição 9. Tal subgrupo possui série derivada limitada por uma constante, no entanto, o grupo quociente $\langle \rho^2 \rangle$ não é suavemente abeliano, pois não pode ser expresso como produto direto de um subgrupo cujo expoente é limitado por uma constante e um subgrupo de tamanho polilogarítmico na ordem do grupo. Logo, não podemos aplicar o método descrito pelo Corolário 5.

Capítulo 7

Conclusão

Este trabalho teve como objetivo principal a aplicação das ferramentas da Computação Quântica e da Teoria de Grupos ao Problema do Isomorfismo de Grafos.

Mostramos neste trabalho uma breve revisão sobre Teoria de Grupos. Em seguida, apresentamos alguns conceitos básicos sobre a Teoria de Grafos necessários para descrever o Problema do Isomorfismo de Grafos. Também foram apresentados conceitos da Mecânica Quântica. Apresentamos o Problema do Subgrupo Oculto (PSO) e mostramos (omitindo alguns detalhes) o algoritmo quântico padrão para a solução do PSO em grupos abelianos.

Mostramos que o Problema do Isomorfismo de Grafos é um PSO sobre o grupo simétrico S_n, ou seja, uma solução eficiente do PSO no grupo simétrico S_n implicará num algoritmo de tempo polinomial para decidir se dois grafos são isomorfos ou não.

Utilizamos uma redução do Problema do Isomorfismo de Grafos para o Problema de Interseção de Grupos, que juntamente com resultados da Computação Quântica nos permitiu obter um algoritmo quântico eficiente para uma classe particular de grafos, que infelizmente é bastante restrita. No entanto, um dos objetivos desta dissertação foi determinar, dado o estágio atual de desenvolvimento da Computação Quântica, se existem algoritmos quânticos eficientes para muitas classes de grafos ou não. Nossa conclusão é que a contribuição da Computação Quântica

é bastante restrita ainda, pois as duas estratégias adotadas: redução do Problema do Isomorfismo de Grafos para o PSO e uso do Problema de Interseção de Grupos, juntamente com resultados da Computação Quântica não produzem resultados significativos.

Como proposta para trabalhos futuros, procuraremos usar o conhecimento adquirido em Computação Quântica e Teoria de Grafos para dar mais exemplos onde essa segunda estratégia funcione, e também analisar possíveis generalizações de algoritmos quânticos para relaxar as restrições impostas a este método.

Referências Bibliográficas

L. Von Ahn. Survey: Quantum computation and the hidden subgroup problem. Relatório técnico, Dept. of Science Computer, Carnegie Mellon University, Pittsburgh, 2002.

A. V. Aho, J. E. Hopcroft, e J. D. Ullman. **The design and analysis of computer algorithms**. Addison-Wesley Series in Computer Science and Information Processing, Reading, MA: Addison-Wesley, 1974.

G. L. Alexanderson. About the cover: Euler and Königsberg's Bridges: A historical view. In: **Bull. Amer. Math. Soc. 43**, páginas 567–573, 2006.

L. Babai. On the complexity of canonical labeling of strongly regular graphs. **SIAM J. Comput.**, 9(1):212–216, 1980.

L. Babai e E. M. Luks. Canonical labeling of graphs. In: **STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing**, páginas 171–183, New York, NY, USA, 1983. ACM. ISBN 0-89791-099-0.

L. Babai e E. Szemerédi. On the complexity of matrix group problems i. **In Proc. of the 25th IEEE Symposium on Foudation of Computer Science**, páginas 229–240, 1984.

R. Beals. Quantum computation of Fourier transforms over symmetric groups. In: **Proc. 29th ACM Symp. on Theory of Computing**, páginas 48–53, New York, 1997. ACM.

H. L. Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. **J. Algorithms**, 11(4):631–643, 1990.

O. Cogis e O. Guinaldo. Linear descriptor for conceptual graphs and a class for polynomial isomorphism test. In: **Proceedings of the 3rd International Conference on Conceptual Structures, ICCS'95**, páginas 263–277, Santa Cruz, CA, USA, August 1995., 1995. Lecture Notes in AI 954, Springer-Verlag.

C. J. Colbourn. On testing isomorphism of permutation graphs. **Networks**, 11: 13–21, 1981.

C. M. M. Cosme. **Algoritmos Quânticos para o Problema do Subgrupo Oculto Não Abeliano**. Tese de Doutorado, Laboratório Nacional de Computação Científica - LNCC, 2008. A ser defendida.

C. M. M. Cosme e R. Portugal. Quantum algorithms for the hidden subgroup problem on a class of semidirect product groups. **ArXiv:quant-ph/0703223v2**, 2007.

I. Damgard. Qip note: On the quantum fourier transform and aplications. Relatório técnico, Computer Science Department of Aarhus University., 2004.

J. F. F. de Abreu. Computação quântica em sistemas abertos e uma aplicação ao modelo biológico de fröhlich. Dissertação de Mestrado, Laboratório Nacional de Computação Científica - LNCC, 2004.

C. M. H. de Figueiredo, J. Meidanis, e C. P. de Mello. A linear-time algorithm for proper interval graph recognition. **Inf. Process. Lett.**, 56(3):179–184, 1995. ISSN 0020-0190.

D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. **Proceedings of the Royal Society of London Ser. A**, A400: 97–117, 1985.

M. Ettinger e P. Høyer. A quantum observable for the graph isomorphism problem. **ArXiv:quant-ph/9901029**, 1999.

L. Euler. Solutio problematis ad geometriam situs pertinentis. In: **Commentarii Academiae Scientiarum Imperialis Petropolitanae 8, 128-140/Opera Omnia (1), Vol. 7, 1-10**, 1741.

S. A. Fenner e Y. Zhang. Quantum algorithms for a set of group theoretic problems. In: **ICTCS**, páginas 215–227, 2005.

T. D. Fernandes. Problema do subgrupo oculto em grupos nilpotentes de classe 2. Dissertação de Mestrado, Laboratório Nacional de Computação Científica - LNCC, 2008. A ser defendida.

R. Feynman. Simulating physics with computers. **International Journal of Theoretical Physics**, 21(6&7):467–488, 1982.

S. Fortin. The graph isomorphism problem. Relatório Técnico 96-20, University of Alberta, Edmonton, Alberta, Canada., 1996.

K. Friedl, G. Ivanyos, F. Magniez, M. Santha, e P. Sen. Hidden translation and orbit coset in quantum computing. In: **Proceedings of 35th ACM Symposium on Theory of Computing**, páginas 1–9, 2003.

R. Frucht. Herstellung von graphen mit vorgegebener abstrakter gruppe. **Compositio Mathematica**, 6:239–250, 1939.

R. Frucht. Graphs of degree three with a given abstract group. **Canad. J. Math.**, 1:365–378, 1949.

H. H. Gan, S. Pasquali, e T. Schlick. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. **Nucleic Acids Res**, 31(11):2926–2943, June 2003. ISSN 1362-4962.

A. Garcia e Y. Lequain. **Elementos de Álgebra**. IMPA, Rio de Janeiro, 2001.

D. N. Gonçalves. Transformada de fourier quântica no grupo diedral. Dissertação de Mestrado, Laboratório Nacional de Computação Científica - LNCC, 2005.

S. Hallgren, A. Russell, e A. Ta-Shma. Normal subgroup reconstruction and quantum computing using group representations. In: **Proc. 32nd ACM Symp. on Theory of Computing**, páginas 627–635. ACM, 2000.

F. Harary. **Graph Theory**. Addison-Wesley, Reading, Massachusetts, 1969.

F. Harary e E.M. Palmer. The smallest graph whose group is cyclic. **Czech. Math. J.**, 16/22:70–71/180, 1966/1972.

I.N. Herstein. **Abstract Algebra**. Macmillan Publishing Company, New York, 1986.

C. M. Hoffman. **Group-Theoretic Algorithms and Graph Isomorphism**, volume 136 of **Lecture Notes in Computer Science**. Springer-Verlag, Berlin, 1979.

J. E. Hopcroft e J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In: **STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing**, páginas 172–184, New York, NY, USA, 1974. ACM.

Wen-Lian Hsu. O($m.n$) algorithms for the recognition and isomorphism problems on circular-arc graphs. **SIAM J. Comput.**, 24(3):411–439, 1995. ISSN 0097-5397.

Y. Inui e F. Le Gall. An efficient quantum algorithm for the hidden subgroup problem over a class of semi-direct product groups. **Quantum Information and Computation (to apear) or ArXiv:quant-ph/0412033v2**, 2005.

G. Ivanyos, F. Magniez, e M. Santha. Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem. **International Journal of Foundations of Computer Science**, 14(5):723–739, 2003.

G. Ivanyos, L. Sanselme, e M. Santha. An efficient quantum algorithm for the

hidden subgroup problem in extraspecial groups. In: **Proc. of STACS'07**, 2007a.

G. Ivanyos, L. Sanselme, e M. Santha. An efficient quantum algorithm for the hidden subgroup problem in nil-2 groups. **arXiv:quant-ph/0707.1260v1**, 2007b.

R. Jozsa. Quantum factoring, discrete logarithms and the hidden subgroup problem. **ArXiv:quant-ph/0012084**, 2000.

P. Kaye, R. Laflamme, e M. Mosca. **An Introduction to Quantum Computing**. Oxford University Press, Inc., New York, NY, USA, 2007. ISBN 0198570007.

A. Kitaev. Quantum measurements and the abelian stabilizer problem. **Electronic Colloquium on Computational Complexity (ECCC)**, 3(3), 1996.

J. Köbler, U. Schöning, e J. Torán. **The graph isomorphism problem: its structural complexity**. Birkhäuser Verlag, Basel, Switzerland, 1993. ISBN 0-8176-3680-3.

N. Koblitz. **Algebraic aspects of cryptography**, volume 3 of **Algorithms And Computation In Mathematics**. Springer-Verlag, Berlin; New York, 1998. ISBN 3-540-63446-0.

G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. **SIAM Journal on Computing**, 30(1):170–188, 2005.

C. Lavor, L.R.U. Manssur, e R. Portugal. Shor's algorithm for factoring large integers. **ArXiv:quant-ph/0303175**, 2003.

X. Liu e D. J. Klein. The graph isomorphism problem. **Journal of Computational Chemistry**, 12(10):1243–1251, 1991.

C. Lomont. The hidden subgroup problem - review and open problems. **ArXiv:quant-ph/0411037**, 2004.

G. S. Lueker e K. S. Booth. A linear time algorithm for deciding interval graph isomorphism. **J. ACM**, 26(2):183–195, 1979. ISSN 0004-5411.

E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. **J. Comput. Syst. Sci.**, 25(1):42–65, 1982.

F. L. Marquezino. A Transformada de Fourier Quântica Aproximada e sua Simulação. Dissertação de Mestrado, Laboratório Nacional de Computação Científica - LNCC, 2006.

R. Mathon. A note on the graph isomorphism counting problem. **Inf. Process. Lett.**, 8(3):131–132, 1979.

B. D. McKay. Practical graph isomorphism. **Congressus Numerantium**, 30: 45–87, 1981.

B. D. McKay. **nauty** user's guide (version 1.5). Relatório Técnico TR-CS-90-02, Australian National University, Department of Computer Science, 1990.

M. Mosca e A. Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In: **Proc. of the 1st NASA International Conference on Quantum Computing and Quantum Communication**, number 1509, Palm Springs, 1999. Lecture Notes in Computer Science.

M. A. Nielsen e I. L. Chuang. **Quantum computation and quantum information**. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-63503-9.

C. M. Papadimitriou. **Computational complexity**. Addison-Wesley, Reading, Massachusetts, 1994. ISBN 0201530821.

J.W. Raymond e P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. **Journal of Computer-Aided Molecular Design**, 16(7):521–533, July 2002.

O. Regev. Quantum computation and lattice problems. In: **FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science**, páginas 520–529, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1822-2.

O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. **ArXiv:quant-ph/0406151v1**, 2004.

Jean-Pierre Serre. **Linear Representations of Finite Groups**. Number 42 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1977.

P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In: **IEEE Symposium on Foundations of Computer Science**, páginas 124–134, 1994.

P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Journal on Computing**, 26(5): 1484–1509, 1997.

D. R. Simon. On the power of quantum computation. In: **Proceedings of the 35th Annual Symposium on Foundations of Computer Science**, páginas 116–123, Los Alamitos, CA, 1994. Institute of Electrical and Electronic Engineers Computer Society Press.

D. R. Simon. On the power of quantum computation. **SIAM J. Comput.**, 26 (5):1474–1483, 1997. ISSN 0097-5397.

J. L. Szwarcfiter. **Grafos e Algoritmos Computacionais**. Editora Campus, Rio de Janeiro, 1984.

G. Valiente. **Algorithms on Trees and Graphs**. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002. ISBN 3540435506.

J. Watrous. Quantum algorithms for solvable groups. In: **STOC**, páginas 60–67, 2001.

V. N. Zemlyachenko, N. M. Kornienko, e R. I. Tyshkevich. Graph isomorphism problem. **Journal of Mathematical Sciences**, 29(4):1426–1481, 1985.

Apêndice A

Os Postulados da Mecânica Quântica

Aqui descrevemos os quatro postulados fundamentais da Mecânica Quântica [Nielsen e Chuang (2000)]. O primeiro postulado trata da descrição matemática de um sistema quântico isolado.

Postulado 1 A qualquer sistema físico isolado existe associado um espaço vetorial complexo com produto interno (ou seja, um espaço de Hilbert), conhecido como espaço de estados do sistema. O sistema é completamente descrito pelo seu vetor de estado, um vetor unitário no espaço de estados.

O segundo postulado trata da evolução dos sistemas físicos quânticos.

Postulado 2 A evolução de um sistema quântico fechado é descrita por uma transformação unitária. Ou seja, o estado $|\psi_0\rangle$ de um sistema em um tempo t_0 está relacionado ao estado $|\psi_f\rangle$ do sistema em t_f por um operador unitário U que depende somente de t_0 e t_f:

$$|\psi_f\rangle = U\,|\psi_0\rangle\,. \tag{A.1}$$

O terceiro postulado descreve a forma como pode-se extrair informações de um sistema quântico através de medições.

Postulado 3 As medidas quânticas são descritas por determinados operadores de medida $\{M_m\}$. Esses operadores atuam sobre o espaço de estados do sistema. O índice m se refere aos possíveis resultados da medida. Se o estado de um sistema for

$|\psi\rangle$, imediatamente antes da medida, a probabilidade de um resultado m ocorrer é dada por:

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle , \tag{A.2}$$

e o estado do sistema após a medida será:

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{A.3}$$

Os operadores de medida satisfazem a relação de completitude:

$$\sum_m M_m^\dagger M_m = I. \tag{A.4}$$

Por último, o quarto postulado descreve a forma como sistemas quânticos diferentes podem ser combinados.

Postulado 4 O espaço de estados de um sistema físico composto é o produto tensorial dos espaços de estados dos sistemas físicos individuais. Se os sistemas forem numerados de 1 até n, e o sistema i for preparado no estado $|\psi_i\rangle$, decorre que o estado do sistema composto será $|\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle$.