UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

Testabilidade de Propriedades de Estruturas Discretas

por

Rafael Calegari

Dissertação submetida como requisito parcial
para a obtenção do grau de

Mestre em Matemática Aplicada

Prof. Dr. Carlos Hoppen
Orientador

Porto Alegre, agosto de 2022

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Calegari, Rafael

Testabilidade de Propriedades de Estruturas Discretas / Rafael Calegari.—Porto Alegre: PPGMAp da UFRGS, 2022.

121 p.: il.

Dissertação (mestrado)— Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, 2022.
Orientador: Hoppen, Carlos

Dissertação: Matemática Aplicada: Matemática Discreta, Grafos, Estruturas discretas, Testabilidade, Propriedades

Testabilidade de Propriedades de Estruturas Discretas

por

Rafael Calegari

Dissertação submetida ao Programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

Mestre em Matemática Aplicada

Linha de Pesquisa: Matemática Discreta

Orientador: Prof. Dr. Carlos Hoppen

Banca examinadora:

Prof. Dr. Rudini Menezes Sampaio
Universidade Federal do Ceará

Profa. Dra. Juliane Golubinski Capaverde
Universidade Federal do Rio Grande do Sul

Prof. Dr. Luiz Emilio Allem
Universidade Federal do Rio Grande do Sul

Dissertação apresentada e aprovada em julho de 2022.

Prof. Dr. Lucas Silva Oliveira
Coordenador

AGRADECIMENTOS

Agradeço a duas pessoas nesta etapa da minha jornada: ao meu orientador, hábil, dedicado e atento; e à minha querida esposa Lisi, que tanto me tem apoiado por todos esses anos.

SUMÁRIO

LISTA DE FIGURAS

RESUMO

Diante de instâncias intratáveis de problemas de decisão, assim entendidas como entradas para os respectivos algoritmos que sejam tão longas, que não permitem que se assegure o processamento de tal algoritmo sobre a dada cadeia num tempo aceitável por um computador real, uma das estratégias comumente empregadas consiste em aceitarmos uma probabilidade de erro numa margem tolerável em troca de garantia de execução assintoticamente mais rápida ou menos dependente de armazenamento de informação, seja no pior caso, seja no caso médio. Nesta linha, encontram-se as técnicas de testabilidade de propriedades de estruturas discretas. Neste trabalho, busca-se apresentar modelos de testabilidade estudados na literatura e contribuir com o aprofundamento e sistematização de determinada abordagem do estudo da testabilidade de algumas destas estruturas discretas sob a ótica de uma teoria de isomorfismos.

ABSTRACT

In face of intractable instances of decision problems, so understood as entries for the respective algorithms that are so long that do not enable to ensure the processing of such algorithm on the given chain in an acceptable time by a real computer, one of the commonly employed strategies consists in accepting a probability of error in a tolerable margin in exchange of an asymptotically faster performance guarantee, or less dependent on the storage of information, whether in the worst case or in the average case. In this line, are found the techniques of testability of properties of discrete structures. In this work, it is sought to present models of testability studied in the literature and contribute with the deepening and systematization of certain approach of the study of testability of some of these discrete structures under the perspective of a theory of isomorphisms.

1 INTRODUÇÃO

Tradicionalmente o tempo de execução de um algoritmo, normalmente contabilizado em número de operações que o algoritmo leva para processar a entrada, seja no pior caso, seja no melhor caso, seja no caso médio, é uma das mais importantes preocupações do estudo dos algoritmos na Teoria da Computação. Muitas vezes conseguimos melhorar o tempo de execução de um algoritmo, quando abrimos mão da certeza do resultado dado pelo mesmo, o que abre espaço para os algoritmos probabilísticos. Esses algoritmos tomam a informação de entrada, processam-na mediante um processo aleatório e dão uma saída com determinado percentual (alto) de certeza. Dada a sua própria natureza, assume-se que o algoritmo pode falhar em certas execuções, mas o alto grau de certeza e o ganho de tempo de execução em relação aos seus rivais (algoritmos determinísticos para o mesmo problema), em certa medida, compensam o risco de falha. Dentre os algoritmos probabilísticos, há os chamados algoritmos de Las Vegas, para os quais a probabilidade de acerto do resultado é igual a 1, mas o número de operações realizadas é uma variável aleatória, e há os algoritmos de Monte Carlo, para os quais se admite que haja erro no resultado encontrado, mas a probabilidade de acerto deve ser maior do que $\frac{1}{2}$. Quando mencionarmos, ao longo deste trabalho, algoritmos probabilísticos, estaremos nos referindo a algoritmos de Monte Carlo.

Um *algoritmo sublinear* é um algoritmo capaz de produzir uma saída sem acessar toda a entrada, mas utilizando somente a uma parte dela. Tipicamente esse tipo de algoritmo é concebido para problemas em que lidamos com grandes quantidades de informação. Em outras palavras, convém cogitar projetarmos algoritmos sublineares em situações em que suas entradas sejam representadas por expressões gigantescas. Trata-se de algoritmos que não leem toda a entrada, processam uma informação correspondente a um pequeno fragmento da entrada, e dão a sua saída, que, em regra, poderá ser correta com maior ou menor probabilidade, dependendo da magnitude relativa do fragmento lido, da natureza do problema e do meio de acesso à entrada. Se um algoritmo selecionar uma pequena amostra da entrada sempre pelo mesmo procedimento, sujeitará a computação a diversos vieses do ponto de vista estatístico. Nesse sentido, imagine-se um algoritmo que, ao processar cadeias de símbolos sobre um alfabeto (ou *strings*), lê os primeiros k símbolos de qualquer entrada mais longa do que k, para algum k constante, e depois faz suas escolhas usando apenas estas informações preliminares, antes de chegar à informação de saída. Esse algoritmo poderá ignorar aspectos relevantes da entrada, em geral, mas isso também depende da natureza do problema. Por esse motivo, é natural esperar que um algoritmo sublinear, para que possa garantir alta probabilidade de correção, deve ser, também, um algoritmo probabilístico. Ao menos a seleção da amostra da entrada deve ser um processo aleatório, evitando, assim, vieses de seleção da amostra.

Nesse contexto de algoritmos probabilísticos sublineares, encontra-se a testabilidade de propriedades, em que se procura verificar por algoritmos de decisão, com alta probabilidade de certeza, se um objeto de entrada possui certa propriedade ou se está longe de satisfazê-la (sob alguma função distância pré-estabelecida). Tais algoritmos são os chamados *testadores*. Os testadores são algoritmos sublineares, pois não têm acesso à entrada inteira. Seu acesso à entrada se dá por meio do seu oráculo, e isso ocorre por um processo aleatório.

Antes de tratarmos do cerne deste trabalho, ou seja, suas delimitações e motivações, convém fixarmos uma terminologia, de maneira informal, porém guardando o sentido usual dos termos pertinentes no ramo. Com isso, ficará mais clara a descrição do que se quer e do que se propõe aqui, pois os termos definidos nos próximos parágrafos, por si só, permitem-nos compreender o tipo de algoritmo que se tem abordado em matéria de testabilidade de propriedades.

As definições rigorosas serão dadas oportunamente, especialmente nos Capítulos 2, 3 e 4.

A seguir, quando falarmos em grafos sem especificarmos o tipo de grafo a que nos referimos, entenda-se que se trata de grafos não-direcionados, o que será feito sempre que o contexto afastar o risco de más interpretações. Na Seção 2.1 definimos o conceito de grafo e introduzimos alguns dos principais conceitos introdutórios da teoria de grafos, para fixarmos uma terminologia a ser empregada ao longo de todo este trabalho.

Uma propriedade de um grafo é um conjunto de grafos fechado sob isomorfismo, ou seja, qualquer conjunto de grafos que contenha os grafos isomorfos a todos os seus elementos. Costuma-se dizer que um grafo G satisfaz a propriedade P, quando G pertence a P.

Diremos neste capítulo que um grafo G, com n vértices, está a uma distância pelo menos ε de uma propriedade P ($0 \leq \varepsilon \leq 1$), quando é necessário modificar pelo menos εn^2 adjacências em G (acrescentando ou excluindo arestas) para transformá-lo num grafo que satisfaça P. A notação utilizada é $d(G,P) \geq \varepsilon$. Veremos, ao longo da dissertação, que há outras métricas que podem ser adotadas.

O algoritmo A é um ε-testador para a propriedade P, se for um algoritmo probabilístico que, ao tomar a descrição de um grafo G como entrada e selecionar apenas parte desta, correspondente a um subgrafo de tamanho limitado superiormente por uma função constante em relação ao tamanho do grafo (a "complexidade de consultas", do termo em inglês *query complexity*, conforme se definirá), aceita a cadeia de entrada com probabilidade maior ou igual a $2/3$, se o grafo representado na entrada satisfizer P (ou seja: quando $d(G,P) = 0$), e rejeita a entrada com probabilidade maior ou igual a $2/3$, caso $d(G,P) \geq \varepsilon$. É importante salientar que a definição deste tipo de algoritmo não impõe condição alguma no caso de $0 < d(G,P) < \varepsilon$. A escolha da constante $2/3$ é convencional, no sentido de que a literatura existente em testabilidade de propriedades a habituou-se a escolhê-la como limitante inferior inferior das probabilidades de aceitação e de rejeição dos aceitadores nos casos citados acima, mas, rigorosamente, pode-se fixar qualquer outra constante maior do que $1/2$ e menor do que 1 para que os ε-testadores sejam algoritmos de Monte Carlo.

Uma propriedade P é testável, quando, para todo ε positivo menor do que 1, existir um ε-testador para P. Percebe-se que, de certa forma, não é pouco o que se exige para que uma propriedade de grafos seja testável, o que nos leva a nos perguntarmos se tais propriedades existem ou quão raras elas são. Investigando questões como estas, alguns autores já identificaram diversas propriedades testáveis de grafos, como as propriedades monótonas e hereditárias, das quais trataremos no Capítulo 3. A título de exemplos, a propriedade de ter um homomorfismo para um grafo fixo K é uma propriedade monótona [3] e, portanto, testável (como se verá), e a propriedade de ser um grafo perfeito é uma propriedade hereditária [6].

O que dissemos até aqui sobre grafos pode ser estendido para um universo muito mais amplo. O estudo de testabilidade de propriedades de grafos, tal como definido neste capítulo, tem sido estendido a outros tipos de estruturas combinatórias, como permutações, grafos direcionados e torneios. Como já deve ter ficado claro, a grande vantagem do estudo destes objetos e relações consiste no fato de que tais algoritmos possibilitam, em geral, grande economia em tempo de execução e espaço de execução, se compararmos os testadores de propriedades com os algoritmos não aproximativos e não probabilísticos para o problema de decidir sobre a satisfação de P por um grafo em particular (ou outra estrutura discreta com propriedades "testáveis"). Os sacrifícios em precisão e em certeza podem ser compensados de diversas maneiras, se forem adequadamente combinadas técnicas inerentes à testabilidade com algoritmos exatos. Em [1], [3], [4],

[6] e [7], entre tantos outros artigos, discute-se testabilidade de propriedades de grafos.

A testabilidade de propriedades em estruturas discretas (não apenas grafos) tem sido amplamente estudada desde a década de 1990 (vide [19] e [26]). Segundo Goldreich [20], o estudo de testabilidade de propriedades surgiu implicitamente no trabalho de Blum, Luby e Rubinfeld [9], o qual foi seguido por outros autores, culminando no trabalho de Rubinfeld e Sudan [26]. Inicialmente esse estudo tratou de propriedades de homomorfismos de grupos e, com o tempo diversificou-se. A literatura já existente acerca da testabilidade de propriedades de grafos não-direcionados, originada em [19], tem muito a nos ensinar sobre a testabilidade de propriedades de dígrafos, e vice-versa, embora se trate de objetos matemáticos distintos. As publicações acerca destas duas classes de estruturas combinatórias, no que tange à testabilidade de propriedades, também nos fornecem pistas para deduzirmos teoremas acerca da testabilidade de propriedades de outras classes de estruturas discretas, tais como permutações, torneios, dentre tantas. Esperamos que tais "pistas", uma vez classificadas e compreendidas, venham a formar um conhecimento novo sobre testabilidade mais abstrato e, possivelmente (assim se espera), não menos relevante para a Matemática Aplicada que o conhecimento que hoje se tem sobre testabilidade de propriedades de cada uma destas classes de objetos. O objetivo da pesquisa implementada no presente trabalho consistiu em identificar um rol de ferramentas metodológicas capazes de importar para o âmbito da testabilidade de propriedades de certas estruturas combinatórias teoremas que envolvam testabilidade de propriedades de estruturas combinatórias de outras classes, tal como se exemplificou logo acima, partindo do que já se sabia no meio científico e tendo como ponto de partida publicações anteriores a respeito.

Apesar de a testabilidade de propriedades ser um caminho para algoritmos rápidos, Fox e Wei [16] observam que muitos resultados gerais fixam enormes limitantes superiores da complexidade dos testadores. Por outro lado, tem havido progresso nas pesquisas de melhores limitantes em testabilidade de propriedades, como se vê em [2], [5], [11], [12] e [15].

Este trabalho está estruturado em sete capítulos. Começamos por um resumo da literatura pré-existente acerca de testabilidade de propriedades, o que fazemos nos Capítulos 2, 3 e 4, respectivamente sobre teste de propriedades de grafos, de torneios e de outras estruturas discretas. No Capítulo 5, fixamos uma terminologia adequada para modelarmos a testabilidade de propriedades de estruturas discretas num contexto mais geral e teorizarmos sobre o foco deste trabalho, que consiste em responder à seguinte pergunta: Como podemos derivar conhecimento sobre testabilidade de uma determinada classe de objetos matemáticos de um universo enumerável a partir de conhecimento precedente sobre certa classe de objetos (não necessariamente a mesma)? Neste capítulo, após definirmos precisamente os conceitos de estrutura discreta e de modelo de testabilidade empregados neste trabalho, definimos os tipos de bijeções que nos permitem representar objetos conceituais de uma teoria de testabilidade de propriedades (de grafos, por exemplo) no universo de outra teoria de testabilidade (de permutações, por exemplo, ou de torneios, ou, ainda, de grupos finitos). Estas bijeções serão chamadas de transformações entre estruturas discretas ou transformações entre modelos de testabilidade. A mais importante classe de transformações entre estruturas ou modelos, para os propósitos deste trabalho, é a das transformações regulares, o que será apresentado ainda no Capítulo 5.

Uma vez fixadas formalmente as ideias de estrutura discreta, modelo de estrutura discreta e transformações entre estruturas discretas e modelos de testabilidade, o que se dá no Capítulo 5, já mencionado, o Capítulo 6 inicia-se com uma série de definições que nos habilita a enunciar o Lema da Equivalência, o qual apresentamos e demonstramos logo a seguir, quando, também, divagamos sobre métodos de dedução de teoremas dentro do objeto desta dissertação, que é o de trazer conhecimento novo numa teoria de testabilidade a partir de conhecimento preliminar noutra teoria de testabilidade, conforme já explicamos resumidamente nas linhas acima. Finalmente, tal capítulo culmina no capítulo seguinte, em que formulamos nossas considerações finais e analisamos a viabilidade de desenvolvimentos de vertentes da matéria desta dissertação em trabalhos futuros.

2 DISPOSIÇÕES PRELIMINARES

Este capítulo é subdividido em três partes: na primeira, apresentamos a terminologia padrão sobre as teorias dos grafos, dígrafos e torneios, objetos centrais desta dissertação; na segunda, definimos três classes de propriedades de grafos de grande importância em matéria de testabiliade de propriedades de grafos (propriedades monótonas, hereditárias e semi-hereditárias); e na terceira, expomos alguns teoremas muito explorados nos estudos de testabilidade de propriedades de grafos e de torneios. O leitor familiarizado com esta matéria poderá pular este capítulo, passando diretamente para o Capítulo 3, sem prejuízo.

Os conceitos preliminares fixados na primeira seção deste Capítulo e seus desdobramentos de que tratam as duas seções posteriores fundam-se na literatura do âmbito da testabilidade de propriedades de grafos e torneios, em particular em [3], [4], [6] e [27].

2.1 Conceitos Preliminares

No que segue, sempre que C for um conjunto e k for um inteiro positivo, denotaremos por $[C]^k$ a coleção de todos os subconjuntos de C com cardinalidade k.

Um *grafo simples* é um par de conjuntos finitos (V,A), em que $A \subseteq [V]^2$, ou seja, A é uma coleção de subconjuntos binários de V. Nestas condições, se $G = (V,A)$ diremos que os elementos de V são chamados os *vértices* do grafo G e os elementos de A são as *arestas* de G. Para representar o conjunto de vértices de um grafo G e o seu conjunto de arestas, usaremos a notação $V(G)$ e $A(G)$, respectivamente.

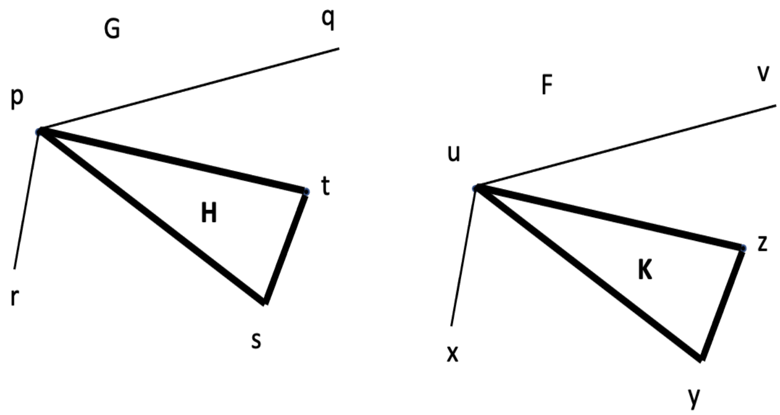

Figura 2.1: Exemplos de isomorfismos: os grafos G e F são isomorfos, enquanto K é isomorfo a H

Muitas vezes, permite-se que haja mais de uma aresta entre os mesmos vértices (o que chamaremos de *arestas múltiplas*) ou que haja arestas em que ambos os vértices são iguais (o que chamaremos de *laços*). Tais estruturas também são chamadas de grafos, mas não de grafos simples. Neste trabalho, sempre avisaremos, quando tratarmos de grafos com laços ou arestas múltiplas. Além disso, chamaremos simplesmente de grafos os grafos simples, sempre que não houver possibilidade de mal-entendidos.

É comum representarmos um grafo simples como um conjunto finito de pontos no plano e segmentos de reta com extremidades em pontos do referido conjunto, em que os pontos do conjunto finito representam os vértices do grafo e os segmentos de reta representam as arestas, como na Figura 2.1. Nesta figura, representamos os grafos G, H, F e K, em que $V(G) = \{p,q,r,s,t\}$, $A(G) = \{\{p,r\},\{p,q\},\{p,s\},\{p,t\},\{s,t\}\}$. O grafo H está representado de forma destacada, e $V(H) = \{p,t,s\}$. Suas arestas são $\{p,t\}$, $\{p,s\}$ e $\{s,t\}$. De maneira análoga, descrevemos os conjuntos de vértices e de arestas de F e de K.

O número de vértices de um grafo G é a *ordem* de G, e o número de arestas de G é o *tamanho* de G. Para simplificar a notação, ao invés de usarmos $|V(G)|$, usaremos $|G|$, e, ao invés de $|A(G)|$, preferiremos empregar $\|G\|$.

Um *homomorfismo* de G para G' é uma função $\varphi : V(G) \longrightarrow V(G')$ que preserva as relações de adjacência, ou seja, uma função tal que $\{u,v\} \in A(G)$ implica que $\{\varphi(u),\varphi(v)\} \in A(G')$. Se um homomorfismo for inversível e a sua função inversa também for um homomorfismo, diz-se que tal função é um *isomorfismo*. Assim, um isomorfismo de G para G' é uma bijeção $\varphi : V(G) \longrightarrow V(G')$ tal que $\{u,v\} \in A(G)$ se, e somente se, $\{\varphi(u),\varphi(v)\} \in A(G')$. Se tal função existir, então diremos que G e G' são grafos *isomorfos* e escrevemos $G \simeq G'$. Tomando novamente o exemplo da Figura 2.1, definamos uma função $\varphi : V(G) \longrightarrow V(F)$ da seguinte forma: $\varphi(p) = u$, $\varphi(q) = v$, $\varphi(r) = x$, $\varphi(s) = y$, $\varphi(t) = z$. Tomando qualquer par de pontos de G (por exemplo, p e q), temos que a relação de adjacência é preservada por φ e pela sua inversa (no caso, como p e q é uma aresta de G, $\{\varphi(p),\varphi(q)\} = \{u,v\}$, que é uma aresta de F. Os vértices q e r não são adjacentes em G, e suas imagens sobre a função φ (os vértices v e x) também não são adjacentes em F. Para qualquer par de pontos de G, a função φ leva vértices adjacentes a vértices adjacentes e vértices não adjacentes a vértices não adjacentes, preservando a relação de adjacência em ambos os sentidos, de maneira que φ é um isomorfismo e os grafos G e F são isomorfos.

Se $V(G') \subseteq V(G)$ e $A(G') \subseteq A(G)$, então diremos que G' é um *subgrafo* de G. Escreve-se $G' \subseteq G$. Se G' contém todas as arestas de G incidentes em dois vértices de G', então G' é um *subgrafo induzido* de G. Neste caso, escreve-se $G' \lhd G$. Se $G' \lhd G$ e $V(G') = S$, G' será denotado por $G[S]$. Uma cópia de um grafo G em um grafo G' é um subgrafo de G' isomorfo a G. Retornando à Figura 2.1, H é um subgrafo induzido de G, e K (grafo com três vértices e três arestas que está destacado na figura) é um subgrafo induzido de F. Se removermos uma aresta de H ou de K e mantivermos os três vértices, teremos como resultado um subgrafo não induzido de G ou de F (conforme o caso). E uma cópia induzida de um grafo G em um grafo G' é um subgrafo induzido de G' isomorfo a G.

Uma sequência de arestas $(\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{k-1}, u_k\})$ de um grafo G, tal que $u_1, \ldots u_k$ são distintos, é um *caminho* de comprimento $k-1$ em G que contém os vértices $u_1, u_2, \ldots, u_k$.

Se uma sequência de arestas $(\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{k-1}, u_k\}, \{u_k, u_1\})$ de G for tal que $u_1, \ldots u_{k-1}$ forem distintos, então ela é um ciclo.

Uma *propriedade* de grafos é um conjunto de grafos fechado sob isomorfismos. Dizer que um grafo satisfaz uma propriedade equivale a dizer que ele pertence a tal propriedade. Se um grafo G pertence à propriedade P e G' é isomorfo a G, então G' também pertence à propriedade P.

Uma função que associa cada grafo a um número real é chamada um *parâmetro* de grafos. Se tal função mapeia para o mesmo valor grafos isomorfos, diremos que se trata de um *parâmetro invariante*, ou simplesmente, um *invariante*.

Muitos problemas clássicos de teoria dos grafos consistem em determinar se um grafo satisfaz ou não uma propriedade ou em determinar um parâmetro do grafo. Quando contamos o número de vértices ou de arestas do grafo, estamos determinando um parâmetro. O número de ciclos de comprimento ímpar é um parâmetro invariante de grafos, e a mera existência de um ou mais ciclos de comprimento impar é uma propriedade. Quando verificamos se o grafo G tem um grafo H como subgrafo, estamos verificando se G satisfaz uma propriedade: a de ter H como subgrafo.

Uma das maneiras de representar um grafo é por meio da sua matriz de adjacência. A matriz de adjacência de um grafo G com n vértices é uma matriz com n linhas e n colunas, em que o elemento disposto no cruzamento entre a i-

ésima linha com a j-ésima coluna é 1 se, e somente se, existe uma aresta contendo o i-ésimo vértice e o j-ésimo vértice, e o referido elemento é 0, caso contrário.

Um *dígrafo* é um par de conjuntos finitos $\left(V, \vec{A}\right)$, em que $\vec{A} \subseteq V \times V$. Nestas condições, se $D = \left(V, \vec{A}\right)$ diremos que os elementos de V são os *vértices* do dígrafo D e os elementos de $\vec{A}$ são os *arcos* de D. Tal como fixamos para o caso dos grafos simples, para representarmos o conjunto de vértices de um dígrafo D e o seu conjunto de arcos, usaremos a notação $V(D)$ e $\vec{A}(D)$, respectivamente.

Se (u, v) é arco de um dígrafo D, dizemos que a orientação deste arco é de u para v. Temos um 2-ciclo em D, quando $(u, v) \in \vec{A}(D)$ e $(v, u) \in \vec{A}(D)$, pois, nesse caso, u e v formam um ciclo dirigido de comprimento 2.

Um *laço*, num dígrafo, é um arco (u, u), para algum vértice u. E um *arco múltiplo* é uma coleção de dois ou mais arcos idênticos de um dígrafo, numa contagem em que estejamos contando cada ocorrência do referido arco. Salvo se dissermos expressamente o contrário, quando falarmos sobre dígrafos, falaremos sobre dígrafos sem laços e sem arcos múltiplos, apesar de que, em alguns momentos, precisaremos lidar com dígrafos com laços ou com arcos múltiplos.

Todos esses conceitos permitem-nos apresentar a seguinte definição, importante no que tange ao foco deste trabalho:

Definição 2.1. *Um torneio é um dígrafo $D = \left(V, \vec{A}\right)$ sem laços e tal que, para cada $u \in V$ e cada $v \in V$, sendo $u \neq v$, $(u, v) \in \vec{A}(D) \Leftrightarrow (v, u) \notin \vec{A}(D)$.*

Torneio é um caso particular de dígrafo em que existe um arco para cada subconjunto binário do conjunto de vértices. Dado um par de vértices de um torneio, o que cabe é definir qual é a orientação do arco que os une, pois sabemos que existe um arco que os contém como componentes. Assim como no caso geral, quando (u, v) é arco de um torneio T, dizemos que a orientação deste arco é de u para v e, por se tratar de um torneio, dizemos também que u domina v. Se

u não dominasse v, v dominaria u, mas, por definição, alguma destas hipóteses deve se concretizar. O número de vértices de um torneio T é o *tamanho* de T. Para simplificar a notação, ao invés de usarmos $|V(T)|$, usaremos $|T|$, valendo a notação para dígrafos também.

Um *isomorfismo* de T para T' é uma função $\varphi : V(T) \longrightarrow V(T')$ que preserva as relações de dominância, ou seja, uma bijeção tal que $(u,v) \in \vec{A}(T)$ implica que $(\varphi(u), \varphi(v)) \in \vec{A}(T')$. Se existir uma função com tal atributo, então diremos que T e T' são torneios *isomorfos*.

Suponhamos que $V(T') \subseteq V(T)$. Neste caso, se $\vec{A}(T') = \{(u,v) : u \in V(T') \wedge v \in V(T') \wedge (u,v) \in \vec{A}(T)\}$, então diremos que T' é um *subtorneio induzido* de T ou simplesmente um *subtorneio* de T. Escreve-se $T' \subseteq T$. Vimos neste capítulo que, se G', um sugrafo G, contém todas as arestas de G incidentes em dois vértices de G', então G' é um *subgrafo induzido* de G. No caso de torneios, porém, sempre que um torneio T' for subtorneio de T, T' será, também, um subtorneio induzido de T', pois, como já observamos, para cada par de vértices, existe um, e apenas um, arco que os contém como componentes.

As noções de propriedade e de parâmetro de torneios é similar à ideia análoga para grafos. Uma *propriedade* de torneios é um conjunto de torneios fechado sob isomorfismos. Dizer que um torneio satisfaz uma propriedade equivale a dizer que ele pertence a tal propriedade. Se um torneio T pertence à propriedade P e T' é isomorfo a T, então T' também pertence à propriedade P. Uma função que associa cada torneio a um número real é chamada um *parâmetro* de torneios. Se tal função mapeia para o mesmo valor torneios isomorfos, diremos que se trata de um *parâmetro invariante*, ou simplesmente, um *invariante*. De modo similar, define-se matriz de adjacência, tanto de torneios, quanto de dígrafos em geral.

Uma sequência de arcos $((u_1,u_2),(u_2,u_3),(u_3,u_4)\dots(u_{k-1},u_k))$ de um torneio T, tal que $u_1,\dots u_k$ são distintos, é um *caminho* de comprimento $k-1$ em T, que contem os vértices $u_1,u_2,\dots,u_k$.

Se a sequência de arcos $((u_1,u_2),(u_2,u_3),(u_3,u_4)\dots(u_{k-1},u_k),(u_k,u_1))$ de um torneio T é tal que $u_1,\dots u_{k-1}$ são vértices distintos, então ela é um ciclo dirigido. Um *torneio transitivo* é um torneio que não contém ciclos dirigidos.

2.2 Propriedades hereditárias, monótonas e semi-hereditárias

Dizemos que uma propriedade de grafos P é *hereditária*, sempre que P é fechada sob remoção de vértices. Desta forma, sempre que G pertence a P e G' decorre da remoção de um vértice de G, teremos que G' pertence a P. Alternativamente, pode-se definir a classe das propriedades hereditárias como a classe das propriedades fechadas sob a opeeração de extração de subgrafos induzidos, pois ambas as definições são equivalentes, isto é, a fixação de uma delas como verdadeira é condição necessária e suficiente para a dedução da outra.

Dizemos que uma propriedade de grafos P é *monótona*, sempre que P é fechada sob remoção de vértices e de arestas. Desta forma, sempre que G pertence a P e G' decorre da remoção de um vértice ou de uma aresta de G, teremos que G' pertence a P. A propriedade de ter um homomorfismo para um grafo dado H é um exemplo de propriedade monótona. Uma outra maneira de definir o que é uma propriedade monótona é fixar uma tal propriedade como aquela que é fechada sob a opeeração de extração de subgrafos (não necessariamente induzidos). Tal como no caso das propriedades hereditárias, a equivalência lógica também se observa aqui.

Uma propriedade hereditária pode não ser monótona, pois há propriedades fechadas sob a remoção de vértices que não são fechadas sob a remoção de arestas. Assim, estas são propriedades hereditárias que não são monótonas. Um exemplo de uma propriedade hereditária que não é monótona é a propriedade de não conter cópia induzida de um grafo dado H. Se um grafo G não contém uma cópia de H como subgrafo induzido, então qualquer subgrafo induzido de G também satisfaz a propriedade de não conter uma cópia de H como subgrafo induzido. Entretanto esta propriedade pode não ser monótona, pois pode haver um subgrafo não induzido de G que contém uma cópia induzida de H e, consequentemente, não satisfaz a esta propriedade.

Também em matéria de torneios, as propriedades fechadas sob tomada de subestruturas (os seus subtorneios) requerem uma atenção especial. A diferença é que, no caso dos grafos, tais subestruturas podem ser os subgrafos e os subgrafos induzidos. Já no caso dos torneios, não há esta distinção: todo subtorneio de um torneio T é um subtorneio induzido de T. Uma propriedade de torneios P é *hereditária*, sempre que P é fechada sob a extração de subtorneios. Desta forma, sempre que T pertence a P e T' é subtorneio de T, teremos que T' pertence a P.

Em tudo o que segue, diremos que um grafo G está ε-longe de uma propriedade P quando a menor distância relativa entre G e qualquer grafo pertencente à propriedade P é menor do que ε, sob algua métrica de grafos. No Capítulo 3 apresentaremos algumas métricas de grafos. Uma propriedade S de grafos é *semi-hereditária* se, e somente se, existe uma propriedade hereditária H tal que:

1- S está contida em H; e

2- Para todo ε positivo, existe M para o qual todo grafo G de ordem maior do que M é tal que, se G está ε-longe de S, então $T \notin H$, sendo que, por definição, um grafo G está ε-longe da propriedade P, se é necessário adicionar ou remover εn^2 arestas de G para formar um grafo pertencente a P.

Similarmente, uma propriedade S de torneios é *semi-hereditária* se, e somente se, existe uma propriedade hereditária de torneios H tal que:

1- S está contida em H; e

2- Para todo ε positivo, existe M para o qual todo torneio T de tamanho maior do que M é tal que, se T está ε-longe de S, então $T \notin H$, sendo que, por definição, um torneio T está ε-longe da propriedade P, se é necessário inverter a orientação de pelo menos εn^2 arcos de T para formar um torneio pertencente a P.

Destas definições, podemos perceber que, tanto no caso de grafos, como no caso de torneios:

I- Toda propriedade hereditária é semi-hereditária (basta tomar $H = S$, que ambas as condições acima se satisfazem).

II- Toda propriedade semi-hereditária está contida em uma propriedade hereditária (item 1 da definição).

III- Nem toda propriedade semi-hereditária é hereditária.

2.3 Teoremas Primordiais

As próximas quatro definições habilitam-nos a enunciar um dos mais importantes teoremas com aplicações em testabilidade de propriedades de grafos: o Lema da Regularidade para grafos, o que as sucederá no texto. Adotamos a expressão "para grafos" no nome que atribuímos a este teorema para evitar que haja confusões, tendo em mente que há também o Lema de Regularidade para torneios (o Lema 2.3).

Definição 2.2. *: Um par de conjuntos* (X', Y') refina *o par de conjuntos disjuntos* (X, Y)*, quando* $X' \subseteq X$ *e* $Y' \subseteq Y$*.*

Definição 2.3. *Dados um grafo* $G = (V, A)$ *e os conjuntos não vazios* X *e* Y *tais que* $X \subseteq V$, $Y \subseteq V$ *e* $X \cap Y = \emptyset$*, definimos a* densidade *do par* (X, Y) *no grafo* G *como*

$$dens_G(X, Y) = \frac{e_G(X, Y)}{|X||Y|}$$

onde $e_G(X, Y) = \left|\{\{x, y\} \subseteq A : x \in X \wedge y \in Y\}\right|$.

Definição 2.4. *Sejam* $G = (V, A)$ *um grafo,* $\gamma > 0$*,* X *e* Y *dois subconjuntos disjuntos de* V*. Dizemos que o par* (X, Y) *é* γ-regular *em* G*, se, para quaisquer conjuntos* X' *e* Y' *tais que* (X', Y') *refina* (X, Y)*,* $|X'| \geq \gamma |X|$ *e* $|Y'| \geq \gamma |Y|$*, tem-se que*

$$\left|dens_G(X, Y) - dens_G(X', Y')\right| < \gamma$$

(Quando não houver dúvida no contexto, diremos apenas que certo par é regular, sem especificar o parâmetro γ*.)*

Definição 2.5. *Dado um grafo* $G = (V, A)$*, uma* equipartição γ-regular de G *de cardinalidade* k *é uma coleção de conjuntos disjuntos* C_1*,* C_2*, ...,* C_k *tal que vale o seguinte:*

$$\bigcup_{j=1}^{k} C_j = V$$

$$\forall i \forall j \left(\left| |C_i| - \left|C_j\right| \right| \leq 1 \right)$$

e o número de pares $\left(C_i, C_j\right)$ *que não são* γ*-regulares é no máximo* $\gamma\binom{k}{2}$*.*

Lema 2.1 (Lema da Regularidade para grafos). *Para todo $\gamma \in (0,1)$ e todo inteiro positivo m, existe um inteiro positivo M tal que todo grafo G com ordem maior do que M admite uma equipartição γ-regular de cardinalidade maior ou igual a m e menor ou igual a M.*

Este lema, empregado na demonstração de diversos teoremas de testabilidade de grafos, informa-nos que sempre poderemos encontrar num grafo de ordem grande o suficiente uma equipartição com parâmetros de regularidade e cardinalidade nas especificações dadas (uma equipartição γ-regular com cardinalidade em certo intervalo, com γ dado). Em diversas situações, é útil identificar equipartições γ-regulares de um grafo. Nesses casos, voltamos ao Lema da Regularidade para gerar a equipartição desejada, observando as condições necessárias dadas pelas variáveis quantificadas do lema. Este lema de Szemerédi, desde a sua apresentação (vide [28]), tem sido uma importante ferramenta em teoria extremal dos grafos. Em [3], [4] e [6], pode-se encontrar mais detalhes sobre este resultado, do qual derivam o lema seguinte e outros lemas importantes que veremos adiante.

Sejam $G = (V, A)$ um grafo e $U = \{U_1, U_2, \ldots, U_k\}$ uma equipartição de V. Para todo $\eta \in [0,1]$ e todo $\gamma \in [0,1]$, chamaremos de *grafo reduzido* ou *grafo regularidade* $R = R_\eta^\gamma(G, U) := (U, A(R))$ em que $A(R)$ satisfaz ao seguinte:

$\left(U_i, U_j\right) \in A(R)$ se, e somente se, $\left(U_i, U_j\right)$ é γ-regular e $dens_G\left(U_i, U_j\right) \geq \eta$.

A Figura 2.2 mostra uma equipartição de um grafo com sete vértices (os pontos p, q, r, s, t, u e v). A equipartição tem três partes (os conjuntos disjuntos A, B e C). Calculemos as densidades de cada parte.

$$dens_G(A, B) = \frac{e_G(A, B)}{|A||B|} = \frac{0}{3 \cdot 2} = 0$$

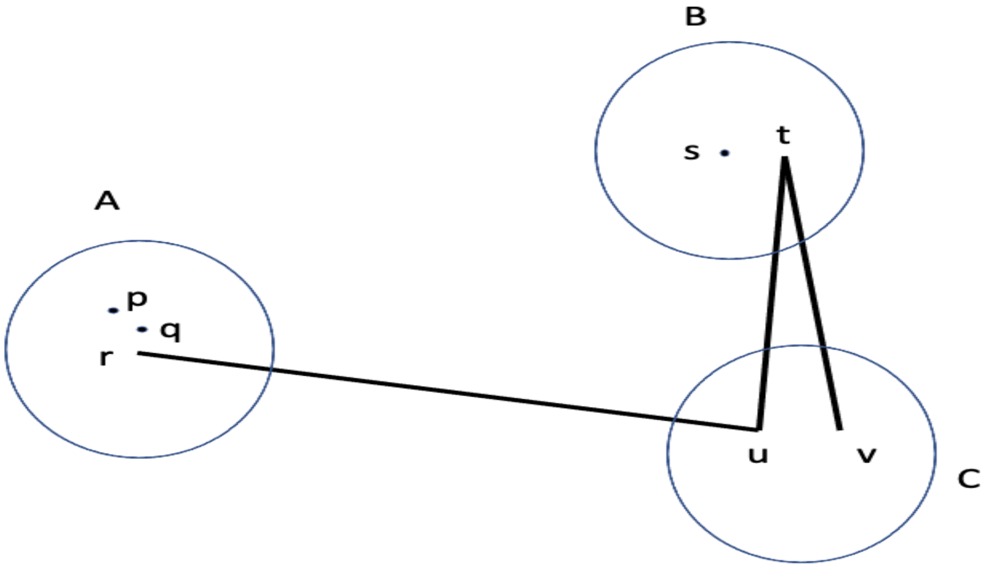

Figura 2.2: Representação de uma equipartição de cardinalidade 3 de um grafo com vértices p, q, r, s, t, u e v. As partes desta equipartição são os conjuntos A, B e C.

$$dens_G(A,C) = \frac{e_G(A,C)}{|A||C|} = \frac{1}{3 \cdot 2} = \frac{1}{6}$$

$$dens_G(B,C) = \frac{e_G(B,C)}{|B||C|} = \frac{2}{2 \cdot 2} = \frac{1}{2}$$

Sejam A', B' e C', respectivamente, subconjuntos de A, B e C com cardinalidade maior ou igual (respectivamente) a $\frac{2}{3}|A|$, $\frac{2}{3}|B|$ e $\frac{2}{3}|C|$. Como a cardinalidade de um conjunto finito é um número natural, para que $|B'| \geq \frac{2}{3}|B|$, $|B'| = 2$, de maneira que $B' = B$, e, da mesma forma, para que $|C'| \geq \frac{2}{3}|C|$, $|C'| = 2$, o que implica que $C' = C$. Já o conjunto A', para que $|A'| \geq \frac{2}{3}|A|$, é suficiente e necessário que $|A'| = 2$, o que significa que $A' = \{p,q,r\}$ (situação já analisada), $A' = \{p,q\}$, $A' = \{p,r\}$ ou $A' = \{q,r\}$. Como $B' = B$ e $C' = C$, $dens_G(B,C) = dens_G(B',C') = \frac{1}{2}$. Calculemos as densidades de (A',B') e (A',C'). Como não há arestas conectando A com B, em qualquer caso teremos que $e_G(A',B') = 0$. Consequentemente, $dens_G(A',B') = 0$. Quanto à densidade do outro par, temos o seguinte:

$$A' = \{p,q\} \Rightarrow dens_G(A',C') = \frac{0}{4} = 0$$
$$A' = \{p,r\} \Rightarrow dens_G(A',C') = \frac{1}{4}$$
$$A' = \{q,r\} \Rightarrow dens_G(A',C') = \frac{1}{4}$$

Em qualquer caso, todos os pares de partes da partição do grafo representado pela Figura 2.2 são $\frac{2}{3}$-regulares, de maneira que tal equipartição é $\frac{2}{3}$-regular, conforme as Definições 2.2 a 2.5. Quando (A,B) é γ-regular, (A,B) é α-regular para qualquer $\alpha > \gamma$. Em outras palavras, quando substituímos γ por um número real maior do que γ, enfraquecemos a assertiva de que (A,B) é γ-regular. Relembrando a Definição 2.5, em que a caracterização de uma equipartição regular admite uma quantidade máxima de pares não regulares de partes de uma equipartição, uma questão que nos vem é se podemos encontrar $\gamma < \frac{2}{3}$ tal que a equipartição representada pela Figura 2.2 é γ-regular. Conforme a definição mencionada logo acima, o número de pares de (A,B,C) que não são γ-regulares é no máximo $\gamma\binom{3}{2}$ para que esta equipartição seja γ-regular, ou seja, o número de pares que não são γ-regulares é menor ou igual a 3γ. Pode-se verificar que, se $\gamma > \frac{1}{2}$, chegaremos à mesma conclusão de antes de que $|A'| = 2$, $|B'| = 2 = |B|$ e $|C'| = 2 = |C|$. Disso, como vimos, decorre que $\{A,B,C\}$ é uma equipartição γ-regular.

Outro resultado útil para testabilidade de grafos é o seguinte:

Lema 2.2 (Lema da Imersão para grafos)**.** *Para todo $\varepsilon \in (0,1)$, todo inteiro k e todo inteiro $f \geq 1$, existem γ, δ e J com a seguinte propriedade. Seja F um grafo de ordem f e sejam $U_1, U_2, \ldots, U_k$ subconjuntos disjuntos de vértices de um grafo G tais que $|U_1| = |U_2| = \cdots = |U_k| = c \geq J$. Suponha que existe uma função $\varphi : V(F) \longrightarrow \{1,2,\ldots,k\}$ que respeite a seguinte condição: Se $(i,j) \in A(F)$, então $\left(U_{\varphi(i)}, U_{\varphi(j)}\right)$ é γ-regular e $dens_G\left(U_{\varphi(i)}, U_{\varphi(j)}\right) \geq \varepsilon$. Então os conjuntos $U_1, U_2, \ldots, U_k$ geram pelo menos δc^f cópias de F em G.*

Como no caso anterior, também existe um Lema da Imersão para torneios, de maneira que convém especificar de qual deles estamos falando. Este lema nos mostra que, para encontrarmos um limitante inferior do número de cópias de um grafo F em um grafo G, é suficiente verificarmos a existência de uma função φ que satisfaça as condições descritas acima. Tudo se passa como se estivéssemos lidando com 3 grafos: F, G e K. O grafo G é formado pela substituição de cada vértice de K por um conjunto de vértices (de G) de tal forma que cada aresta de K corresponda a um par de conjuntos disjuntos de vértices de G com densidade e regularidade (o parâmetro γ) mínimas. Sendo assim, o G é o grafo regularidade $R_{\varepsilon}^{\gamma}(F,U)$, em que U é a equipartição formada por U_1, U_2, ..., U_k. Este lema está demonstrado em diversos artigos a partir do Lema da Regularidade, como nos trabalhos de Alon e Shapira [4] e [6].

Definição 2.6. *Seja G um grafo e $\mathcal{F}$ uma família de grafos. Dizemos que G é* F-livre, *ou que G é* livre de $\mathcal{F}$ *se, e somente se, G não contém cópia de algum elemento de $\mathcal{F}$.*

Definição 2.7. *Seja G um grafo e $\mathcal{F}$ uma família de grafos. Dizemos que G é* $\mathcal{F}$-livre **induzido**, *ou que G é* livre de $\mathcal{F}$ **induzido** *se, e somente se, G não contém cópia* ***induzida*** *de algum elemento de $\mathcal{F}$.*

Em alguns momentos, por simplicidade, diremos que G é livre de um grafo L, quando deveríamos dizer que G é livre do conjunto unitário composto pelo grafo L. Faremos este singelo abuso de linguagem para não carregarmos demais a linguagem, o que dificultaria, sem necessidade, a exposição do assunto. As Definições 2.6 e 2.7 são úteis para generalizarmos a noção de que não ser supergrafo de alguma cópia de um certo grafo constante F_1 é uma propriedade de grafos. Se estendermos essa propriedade para os grafos constantes F_1, F_2, etc..., teremos a Definição 2.6. A Definição 2.7 é equivalente à anterior, mas para cópias induzidas. Ambas as definições permitem-nos enunciar os Teoremas 2.1 e 2.2, os quais fixam condições necessárias e suficientes para que uma propriedade seja, respectivamente, hereditária ou monótonas. Vejamos.

Teorema 2.1. *Uma propriedade P é hereditária se, e somente se, todo grafo G obedece o seguinte: Existe uma família de grafos $\mathcal{F}$ tal que G satisfaz P se, e somente se, G é livre de $\mathcal{F}$* ***induzido****.*

Teorema 2.2. *Uma propriedade P é monótona, se e somente se, todo grafo G obedece ao seguinte: Existe uma família de grafos $\mathcal{F}$ tal que G satisfaz P, se e somente se G é livre de $\mathcal{F}$.*

Chamaremos de $Livre\ \mathcal{F}$ a propriedade de ser livre da família $\mathcal{F}$ e de $LivreInd\ \mathcal{F}$ a propriedade de ser livre induzido da família $\mathcal{F}$. O que os dois teoremas acima nos contam é que uma propriedade P é hereditária se, e somente se, $P = LivreInd\ \mathcal{F}$ e uma propriedade é monótona se, e somente se, $P = Livre\ \mathcal{F}$, para alguma família de grafos F.

Para provar o Teorema 2.1, dividiremos a demonstração em duas partes. Na primeira demonstraremos um sentido da bi-implicação ($\Rightarrow$) e na outra demonstraremos o outro sentido ($\Leftarrow$). Inicialmente, suponhamos que $C = LivreInd\ \mathcal{F}$. Tomemos algum grafo G tal que $G \in C$. Fixemos um grafo H tal que $H \triangleleft G$. Se $K \in F$ e $K \triangleleft H$, chegamos ao absurdo de que $K \triangleleft G$. Esta proposição não pode ser verdadeira, porque, como $G \in C = LivreInd\ \mathcal{F}$, G não pode ter um subgrafo induzido pertencente a $\mathcal{F}$. Assim, temos o seguinte.

$$\forall K\,(K \triangleleft H \Rightarrow K \notin \mathcal{F})$$

$$\therefore H \in LivreInd\ \mathcal{F}$$

Em síntese, se $G \in LivreInd\ \mathcal{F}$ e $H \triangleleft G$, então necessariamente $H \in LivreInd\ \mathcal{F}$, o que implica que a propriedade $LivreInd\ \mathcal{F}$ é fechada sob tomada de subgrafos induzidos, o que é o mesmo que dizer que ela é fechada sob remoção de vértices e, portanto, hereditária.

Para verificarmos o sentido contrário da bi-implicação afirmada no Teorema 2.1, suponhamos que C seja uma propriedade hereditária que contém o grafo G e tomemos arbitrariamente um grafo H que seja subgrafo induzido de G.

Seja $\mathcal{F}$ o conjunto complementar de C. Sabemos que C é hereditária. Segue, pois, que

$$H \in C$$

$$\therefore H \in \mathcal{F}^c$$

E, finalmente,

$$H \notin \mathcal{F}$$

Lembrando que H é tomado arbitrariamente dentre os subgrafos induzidos de G, encontramos que

$$\forall H\,(H \lhd G \Rightarrow H \notin \mathcal{F})$$

$$\therefore G \in LivreInd\ \mathcal{F}$$

onde $\mathcal{F}$ é determinado conjunto de grafos (o conjunto complementar de C).

Qualquer que seja o G fixado entre os grafos pertencentes a C, concluiremos sempre que $G \in LivreInd\ \mathcal{F}$. Sendo assim, $C \subseteq LivreInd\ \mathcal{F}$. Por outro lado,

$$G \in \mathcal{F} \Rightarrow G \notin LivreInd\ \mathcal{F},$$

o que nos habilita a concluir que $C = LivreInd\ \mathcal{F}$, como queríamos provar. O Teorema 2.2 admite demonstração análoga, substituindo os subgrafos induzidos por subgrafos não necessariamente induzidos, $LivreInd\ \mathcal{F}$ por $Livre\ \mathcal{F}$, propriedades hereditárias por propriedades monótonas e substituindo, também, remoção de vértices por remoção de vértices ou arestas.

O Teorema 2.1 admite um análogo para torneios, em que a Definição 2.8, que segue abaixo, faz o papel que a Definição 2.7 faz no contexto dos grafos e o Teorema 2.3, logo a seguir, é o análogo do Teorema 2.1 no âmbito dos torneios.

Definição 2.8. *Seja T um torneio e $\mathcal{F}$ uma família de torneios. Dizemos que T é* F-livre, *ou que T é* livre de F *se, e somente se, G não contém cópia de algum elemento de $\mathcal{F}$. Escreveremos, nesse caso, $T \in Livre(F)$, em que $Livre(\mathcal{F})$ é a propriedade de ser livre de $\mathcal{F}$.*

Teorema 2.3. *Uma propriedade de torneios P é hereditária se, e somente se, $P = Livre(\mathcal{F})$, para alguma família de torneios $\mathcal{F}$.*

Os Lemas da Regularidade e da Imersão também têm análogos no âmbito dos torneios. A seguir enunciamos as Definições 2.9 a 2.11, as quais são equiparáveis às Definições 2.3 a 2.5, entretanto enquanto estas últimas se referem ao universo dos grafos, tornando-nos aptos a enunciar o Lema da Regularidade de grafos (Lema 2.1), as primeiras referem-se ao universo dos torneios e o Lema da Regularidade para torneios (Lema 2.3).

Definição 2.9. *Dados um torneio $T = (V, \vec{A})$ e os conjuntos não vazios X e Y tais que $X \subseteq V$, $Y \subseteq V$ e $X \cap Y = \emptyset$, definimos a* densidade *do par (X, Y) no torneio T como*

$$dens_T(X, Y) = \frac{e_T(X, Y)}{|X||Y|}$$

onde $e_T(X, Y) = \left|\left\{(x, y) \subseteq \vec{A} : x \in X \wedge y \in Y\right\}\right|$.

Definição 2.10 (Par Regular)**.** *Sejam $T = (V, A)$ um torneio, $\gamma > 0$, X e Y dois subconjuntos disjuntos de V. Dizemos que o par (X, Y) é γ-*regular *em T, se, para quaisquer conjuntos X' e Y' tais que (X', Y') refina (X,Y), $|X'| \geq \gamma |X|$ e $|Y'| \geq \gamma |Y|$, tem-se que $|dens_T(X, Y) - dens_T(X', Y')| < \gamma$.*

Definição 2.11. *Dado o torneio $T = (V, \vec{A})$, uma* equipartição γ-regular de T *de cardinalidade k é uma coleção de conjuntos disjuntos C_1, C_2, ..., C_k tal que vale o seguinte:*

$$\bigcup_{j=1}^{k} C_j = V$$

$$\forall j \forall m \left(\left| \left| C_j \right| - |C_m| \right| \leq 1 \right)$$

e o número de pares $\left(C_j, C_m\right)$ que não são γ-regulares é no máximo γk^2.

Lema 2.3 (Lema da Regularidade para torneios)**.** *Para todo $\gamma \in (0,1)$ e todo inteiro positivo m, existe um inteiro positivo M tal que todo torneio T com tamanho maior do que M admite uma equipartição γ-regular de cardinalidade maior ou igual a m e menor ou igual a M.*

Como se passa no caso do Lema da Regularidade para Grafos, este lema é empregado na demonstração de diversos teoremas de testabilidade de torneios e informa-nos que sempre poderemos encontrar um torneio de tamanho grande o suficiente que admita uma equipartição com parâmetros de regularidade e cardinalidade nas especificações dadas (uma equipartição γ-regular com cardinalidade em certo intervalo, com γ dado). Em diversas situações, é útil identificar equipartições γ-regulares de um torneios. Nesses casos, voltamos ao Lema da Regularidade para torneios para gerar a equipartição desejada, observando as condições necessárias dadas pelas variáveis quantificadas do lema.

Lema 2.4 (Lema da Imersão para torneios)**.** *Para todo $\eta \in (0,1)$ e $h \in \mathbb{N}^*$, existem números reais positivos γ e δ satisfazendo o seguinte. Sejam $T = (V, A)$ um torneio e U uma coleção de h subconjuntos disjuntos de V dois a dois γ-regulares. Seja H um torneio com tamanho h admitindo uma função bijetora $\phi : V(H) \to U$ tal que*

$(u,v) \in A(H) \Rightarrow dens_T(\phi(u), \phi(v)) \geq \eta$. *Então existem pelo menos* δm^h *cópias de* H *em* T, *onde* $m = \min\{|C| : C \in U\}$.

O que este lema nos diz é que, se quisermos encontrar muitas cópias de um torneio $H = \left(\{u_1, u_2, \dots, u_h\}, \vec{A}(H)\right)$ em um torneio $T = \left(V, \vec{A}\right)$, é suficiente buscarmos uma coleção de conjuntos disjuntos $U_1, U_2, \dots, U_h$ de elementos de V, de cardinalidade considerável e dois a dois regulares, e que respeitem as mesmas relações de dominância de H, ou seja, que sejam tais que, sempre que $\left(u_i, u_j\right)$ for arco de T $\left(U_i, U_j\right)$ tem alta densidade, para todo $1 \leq i < j \leq h$.

O conceito de grafo reduzido tem um análogo para torneios. O Lema 2.4 admite formulação mais amigável, com o auxílio desta definição que segue abaixo. Vejamos.

Definição 2.12. *Sejam* $T = \left(V, \vec{A}\right)$ *um torneio e* U *uma coleção de* k *subconjuntos disjuntos de* V. *Para todo* $\eta \in [0,1]$ *e todo* $\gamma \in [0,1]$, *chamaremos de* dígrafo reduzido $R = R_\eta^\gamma(T,U) := \left(U, \vec{A}(R)\right)$ *em que* $\vec{A}(R)$ *satisfaz ao seguinte:* $\left(U_i, U_j\right) \in \vec{A}(R)$ *se, e somente se,* $\left(U_i, U_j\right)$ *é* γ*-regular e* $dens_T\left(U_i, U_j\right) \geq \eta$.

Lema 2.5. *Para qualquer* $\eta \in (0,1)$, $h \in \mathbb{N}^*$ *e* $M \in \mathbb{N}^*$, *existem números reais positivos* γ *e* δ *satisfazendo o seguinte. Sejam* $T = \left(V, \vec{A}\right)$ *um torneio e* U *uma equipartição de* T *de cardinalidade menor ou igual a* M. *Então para qualquer torneio* H *de tamanho* h *que seja subgrafo de* $R_\eta^\gamma(T,U)$ *existem pelo menos* δn^h *cópias de* H *em* T.

Uma demonstração dos Lemas 2.4 e 2.5 pode ser encontrada em Stagni [27].

Dos lemas apresentados, verificamos que suas estruturas basicamente começam com o quantificador universal ligado a uma ou mais variável e logo em seguida há o quantificador existencial ligado a uma ou mais variáveis. Em casos assim, podemos assumir que as variáveis quantificadas por $\exists$ são funções das variáveis quantificadas por $\forall$. Nesse sentido, podemos assumir que o parâmetro M

do Lema 2.1 é uma função de m e de γ. Isso nos permite enunciar a Definição 2.13, apresentada a seguir.

Definição 2.13. *Se β:$\mathbb{N} \longrightarrow (0,1)$ é uma função decrescente e m é um inteiro positivo, então definiremos como $W_{\beta,m}$:$\mathbb{N} \longrightarrow \mathbb{N}$, a seguinte função:*

$$W_{\beta,m}(i) = M\left(R, \frac{\beta(R)}{R^2}\right)$$

onde $R = W_{\beta,m}(i-1)$ e sendo que $W_{\beta,m}(1) = M(m, \beta(0))$

Lema 2.6. *Se S for definida como a função tal que $S = S(m,\beta) = W_{\beta,m}\left(\frac{100}{\beta(0)^4}\right)$ para dada função decrescente β:$\mathbb{N} \longrightarrow (0,1)$, onde $W_{\beta,m}$ é a função fixada na Definição 2.13 então vale o seguinte: Para todo grafo G tal que $|G| \geq S(m,\beta)$, existem uma equipartição $A = \{V_i : 1 \leq i \leq k\}$ e um subgrafo induzido U de G com uma equipartição $B = \{U_i : 1 \leq i \leq k\}$ tais que:*

1- $m \leq k \leq S$.

2- $U_i \subseteq V_i$, $\forall i \geq 1$, e $U_i \geq \frac{n}{S}$.

3- Todos os pares da equipartição B são $\beta(k)$-regulares.

4- Todos a menos de $\beta(0)\binom{k}{2}$ dos pares (i,j) tais que $1 \leq i < j \leq k$ obedecem a $\left|dens_G\left(V_i, V_j\right) - dens_G\left(U_i, U_j\right)\right| < \beta(0)$.

Observe-se que, conforme o item 3 acima, neste lema conseguimos uma equipartição B em que todos os pares são regulares. Por outro lado, B é equipartição de um subgrafo induzido de G, mas pode não ser uma equipartição de G. Ainda assim, conforme o item 4, conseguimos uma equipartição regular de G. Vê-se claramente que esta é uma versão mais forte do Lema 2.1. Este lema será explorado na Seção 3.7, o qual revelará seu potencial no processo dedutivo para gerar inferências.

Os Lemas 2.1, 2.2 e 2.6 empregam parâmetros dependentes de parâmetros dados. Os primeiros podem ser entendidos como funções. Uma discussão mais detalhada sobre as funções que figuram em algumas variantes destes lemas pode ser encontrada em [11] e [25].

3 TESTABILIDADE DE PROPRIEDADES DE GRAFOS

3.1 Testadores de propriedades de grafos

Começaremos nosso estudo com a testabilidade de propriedades de grafos, sobre o que dedicaremos o presente capítulo. Entre as fontes de testabilidade de propriedades de grafos, estão os artigos [3], [4], [5], [6], [7] e [17], entre outros que, ao longo deste capítulo, serão contextualizados.

Os algoritmos de teste de propriedades são os chamados testadores, e estes estão ligados a determinado dispositivo chamado oráculo. O *oráculo* é um dispositivo associado a um grafo que faz consultas a esse grafo, verificando parte da estrutura do grafo a cada consulta. A maneira como isso ocorre será descrita formalmente na Seção 3.3, em que trataremos dos três principais modelos de testabilidade de propriedades de grafos, sendo que, em cada qual, as consultas dos oráculos seguem diferentes regras.

Um *testador* para uma propriedade de grafos é um algoritmo probabilístico associado a um parâmetro de tamanho n, a um parâmetro de distância ε e a um oráculo com acesso à representação de um grafo. Ao oráculo é dada a faculdade de fazer consultas à representação do grafo de entrada, de maneira que o algoritmo possa acionar o oráculo em certos momentos e fazer escolhas a partir das respostas às consultas já realizadas pelo oráculo ou de escolhas feitas a partir dos dados resultantes de consultas anteriores ao oráculo. No final da computação de um testador sobre um grafo, o testador aceita ou rejeita o grafo de entrada, e vale o seguinte:

1- Se o grafo G satisfaz a propriedade P, então a probabilidade de o testador T aceitar G é maior do que $\frac{2}{3}$.

2- Se o grafo G está a uma distância maior ou igual a ε da propriedade P, então a probabilidade de o testador T rejeitar G é maior do que $\frac{2}{3}$, para alguma métrica sobre o conjunto dos grafos.

Se existir um testador tal que a probabilidade citada no item 1 acima for igual a 1, dizemos que a propriedade em questão é testável *com erro unilateral*. Caso contrário, dizemos que ela é testável *com erro bilateral*. Em qualquer caso, o número de consultas é, necessariamente, constante (alternativamente, sublinear) em relação a n, mas pode variar em função de ε. Para sermos mais precisos, há alguma divergência na literatura a esse respeito: a maioria dos autores define testadores constantes em relação ao parâmetro de distância, mas alguns autores, em alguns dos seus artigos, definem testadores dependentes deste parâmetro, desde que a dependência seja sublinear. Testadores constantes, que são os testadores adotados pela maior parte da literatura, são sublineares. Também são sublineares testadores em que o número de consultas é delimitado superiormente por uma função logarítmica, por exemplo. Como regra, um algoritmo é sublinear se, e somente se, sua complexidade computacional (a qual, no caso dos testadores, é o número de consultas) pode ser delimitada superiormente por uma função f tal que $\frac{f(n)}{n}$ tende a 0, quando n tende a ∞. Dizer que tais algoritmos são *sublineares* significa, em termos simplificados, dizer que sua complexidade computacional é "abaixo da linear". De uma forma ou de outra, em relação a isso cabe, aqui, a fixação de mais alguns termos técnicos.

Cabe aqui fazermos uma breve síntese da questão da eficiência dos testadores. Como de praxe, chamamos de complexidade certo parâmetro que nos indica a eficiência de um algoritmo. Quando trabalhamos com algoritmos determinísticos, é comum chamarmos de *complexidade de tempo* o parâmetro de complexidade que mede o número de operações que o algoritmo executará sobre dada entrada, o qual esperamos que seja sempre o menor possível; da mesma forma, quando o parâmetro de complexidade busca mensurar a quantidade de informação a ser armazenada numa mesma etapa da computação do algoritmo sobre dada entrada, falamos em *complexidade de espaço*. Mais eficiente é o algoritmo que consome menos tempo de execução ou menos espaço de execução no contexto determinístico. É claro que, como um algoritmo admite uma extensa, ou mesmo infinita, quantidade de entradas possíveis, a complexidade de tempo ou de espaço varia conforme o objeto tomado como entrada para processamento pelo algoritmo. Essa é a sua complexidade exata de tempo ou espaço, uma medida de menor interesse teórico, pois os parâmetros de eficiência que mais importam são quase sempre a *complexidade no pior caso*, a *complexidade no caso médio* e, em alguns casos, a *complexidade no melhor caso*, que são, respectivamente, uma limitante superior das complexidades exatas do algoritmo, uma média das complexidades exatas e uma limitante inferior da complexidade exata.

No caso da testabilidade de propriedades, nosso foco, em termos de eficiência, é o número de consultas que o oráculo precisará fazer ao grafo de entrada ao longo de todo o processamento do testador. O testador será tão mais eficiente, quanto menor for o número de consultas que ele fizer ao grafo de entrada por meio do seu oráculo. Chamaremos de *complexidade de consultas* de um testador sobre uma entrada o número de consultas que o seu oráculo precisa fazer ao longo do processamento da representação do grafo de entrada, acessível mediante o oráculo. Falaremos de *complexidade de consultas no pior caso* para nos referirmos a uma limitante superior do número de consultas de um testador. Não nos preocuparemos, neste trabalho, com complexidades médias, nem com complexidades no melhor caso.

Como existem diferentes maneiras de representar um grafo, existem diferentes modelos de testabilidade de grafos. O principal modelo de testabilidade de grafos adotado neste trabalho é o *modelo denso*, no qual o oráculo tem acesso à matriz de adjacência do grafo de entrada. O parâmetro de tamanho representa a ordem do grafo de entrada. Neste modelo, o testador pode, em determinados momentos da sua computação, acionar o oráculo para que este consulte um componente da matriz de adjacência do grafo de entrada. Obviamente, a resposta a esta consulta informa ao testador sobre a existência ou inexistência de uma aresta contendo u e v, em que u e v são vértices do grafo de entrada. Uma consulta do oráculo de um testador, no modelo denso, pode ser entendida como a pergunta de se determinado par de vértices corresponde a uma aresta. A resposta é sim ou não (1 ou 0).

3.2 Principais Tipos de Testadores

Qualquer algoritmo que se enquadre na definição dada no início da Seção 2.1 é um testador de uma propriedade de grafos. Não é difícil perceber o quanto isso é amplo e quão difícil é organizar o estudo destes algoritmos sem delimitá-los. Se pudermos definir algumas classes mais restritas de testadores que correspondam a todas as propriedades testáveis ou, pelo menos, a classes extremamente extensas de propriedades testáveis, teremos simplificado uma relevante parte de todo esse estudo. E, de fato, isso já foi feito. Como veremos, todas as propriedades testáveis admitem um *testador não-adaptativo*, e uma relevante e extensa classe de propriedades testáveis admite *testadores alienados* e *testadores canônicos*, tanto um, quanto o outro, definidos adiante. Na Seção 2.6, em que abordaremos aspectos mais gerais da teoria da testabilidade dos grafos, ficará mais clara a relevância das definições fixadas abaixo destes três tipos de testadores.

Testador não-adaptativo

Definição 3.1. *Um testador não-adaptativo é um testador T associado a uma função $q : [0,1] \times \mathbb{N} \longrightarrow \mathbb{N}$ tal que T procede da seguinte maneira sobre um grafo de entrada G de ordem n e sobre o parâmetro ε:*

1- *Inicialmente T escolhe aleatoriamente e com distribuição uniforme um subconjunto S de cardinalidade $q(\varepsilon, n)$ de V(G).*

2- *Posteriormente, T faz $\binom{q}{2}$ consultas ao oráculo para obter o subgrafo induzido por S.*

3- *Finalmente, T aceita ou rejeita G tendo acesso apenas a $G[S]$ e a n, ou seja, sem fazer mais consultas ao oráculo.*

Testador alienado (do termo em inglês *oblivious tester*)

Definição 3.2. *Um testador T é um* testador alienado, *se ele é associado a uma função $q : [0,1] \longrightarrow \mathbb{N}$ tal que T procede da seguinte maneira sobre a entrada G e sobre o parâmetro de distância ε, sendo G um grafo de ordem n:*

1- *Primeiramente, T computa $q(\varepsilon)$ e sorteia randomicamente e com distribuição uniforme um subconjunto $S \subseteq V(A)$ com $q(\varepsilon)$ vértices, se $q(\varepsilon) \leq n$, ou com n vértices, caso contrário.*

2- *Posteriormente, T faz as consultas ao oráculo para obter o subgrafo de G induzido por S.*

3- *Finalmente, T aceita ou rejeita G tendo acesso apenas a $G[S]$, <u>mas não a n</u>.*

Comparando-se as duas definições acima, pode-se verificar que a principal diferença entre um testador não-adaptativo e um testador alienado é que, neste último, sua complexidade de consultas é independente de n, e suas escolhas, a cada passo da computação, não dependem de n.

Testador canônico

Definição 3.3. *Um testador canônico para a propriedade P é um testador não-adaptativo que, após sortear um subconjunto S do conjunto de vértices do grafo de entrada G*

e formar o subgrafo induzido por S, decide (determinística ou randomicamente) se tal subgrafo induzido satisfaz ou não certa propriedade P' (em que P' não é necessariamente igual a P).

Vê-se que todo testador canônico é não-adaptativo, mas a recíproca não é verdadeira. Mais adiante, nas Seções 3.5 e 3.6, veremos um exemplo de um testador canônico e de um testador que não é canônico.

3.3 Modelos de Testabilidade de Grafos

Existem pelo menos três modelos de testabilidade de grafos na literatura: o modelo denso, o modelo de grau limitado e o modelo geral. No modelo denso, a métrica depende da *função de adjacência* $g_G : [V]^2 \longrightarrow \{0,1\}$, onde V é o conjunto de vértices de um grafo G. Essa função associa a cada conjunto binário de vértices o elemento 1, se houver uma mesma aresta incidente em ambos, e o elemento 0, caso contrário. Em outras palavras, essa função fornece o componente da matriz de adjacência do grafo na linha correspondente a um vértice e na coluna correspondente a outro vértice.

Sejam G e G' dois grafos com o mesmo conjunto de vértices. A *distância absoluta* entre estes dois grafos, no modelo denso, conhecida como *distância de edição*, é o número de pontos do domínio da função de adjacência para os quais a imagem é diferente. Simbolicamente, escrevemos:

$$dist_{Abs}(G,G') = |\{\{u,v\} \subseteq V : g_G(\{u,v\}) \neq g_{G'}(\{u,v\})\}|$$

onde V é o conjunto de vértices de G e de G'. A *distância relativa* entre G e G' é a distância absoluta entre eles normalizada, isto é, dividida por n^2, sendo que n é um limitante superior da ordem de G e de G'. Nesse caso, escrevemos assim:

$$dist_{Rel}(G,G') = \frac{|\{\{u,v\} \subseteq V : g_G(\{u,v\}) \neq g_{G'}(\{u,v\})\}|}{n^2}$$

É importante observar que estas definições de distância requerem que os grafos tenham o mesmo conjunto de vértices, o que é uma condição mais forte do que os grafos terem a mesma ordem.

Também é importante o conceito de distância entre um grafo e uma propriedade, definido da seguinte forma:

$$dist(G,P) = \min_{G' \in P}[dist_{Rel}(G,G')]$$

Uma forma equivalente de definir a distância entre grafos e propriedades no modelo denso, a qual também é usual, é por meio do conceito de grafo *ε-longe* de certa propriedade P. Um grafo G está *ε-longe* da propriedade P, quando $dist(G,P) \geq \varepsilon$.

No modelo denso, o oráculo consulta conjuntos binários de vértices do grafo de entrada e fornece a imagem da função de adjacência sobre esses conjuntos binários. Como vimos, esta função indica-nos se um subconjunto binário do conjunto de vértices de um grafo corresponde a uma aresta ou não, e é somente esse o tipo de informação que o oráculo de um testador consulta no modelo denso.

No modelo de grau limitado, a métrica depende da *função de incidência* $g_G : V \times [d] \longrightarrow V \cup \{0\}$, onde V é o conjunto de vértices de um grafo G e d é um limitante superior dos graus dos vértices de G. Essa função associa a cada par composto por um vértice u e um número natural não nulo i, limitado por d, o elemento v, se v for o i-ésimo vizinho de u, caso i seja menor ou igual ao grau de u. Caso contrário, o par (u,i) é associado ao elemento 0. Sendo assim, essa função pressupõe uma ordem entre os vizinhos de cada vértice. Além disso, há redundância na representação das arestas, que são contabilizadas duas vezes, uma para cada vértice sobre o qual a aresta incide.

Assim como no modelo denso, no modelo de grau limitado, a métrica também é dada por uma função g_G que capacita um testador a fazer consultas para o seu oráculo acerca do grafo de entrada. A diferença é que, neste último modelo, g_G é a função de incidência, que informa o testador qual é o j-ésimo vizinho do k-ésimo vértice, se houver; já no modelo denso, g_G é a função de adjacência, que informa o testador sobre as relações de adjacência para os pares de vértices consultados.

Sejam G e G' dois grafos com o mesmo conjunto de vértices. A *distância absoluta* entre estes dois grafos no modelo de grau limitado, é o número de pontos do domínio da função de incidência para os quais a imagem é diferente. Simbolicamente, escrevemos: $dist_{Abs}(G, G') = |\{(u,i) \in V \times [d] : g_G(u,i) \neq g_{G'}(u,i)\}|$. A *distância relativa* entre G e G', nesse mesmo modelo, é a distância absoluta entre eles normalizada, isto é, dividida por nd, um limitante superior do número de componentes unitários das matrizes de incidência de G e de G'. Nesse caso, escrevemos

$$dist_{Rel}(G, G') = \frac{|\{(u,i) \in V \times [d] : g_G(u,i) \neq g_{G'}(u,i)\}|}{nd}$$

É importante observar que, tal como ocorre no modelo denso, todas estas definições de distância requerem que os grafos tenham o mesmo conjunto de vértices, o que, como antes, é uma condição mais forte do que os grafos terem a mesma ordem.

No *modelo geral*, os grafos são representados tanto pela função de adjacência, quanto pela função de incidência, redundantemente. O oráculo, nas suas consultas ao grafo de entrada no decorrer da execução de um testador, pode acionar tanto a função de adjacência, quanto a função de incidência, obtendo como respostas as informações de que tratamos acima.

Além dos tipos de consultas, outro aspecto que define o modelo geral de testabilidade de grafos são as definições de distância absoluta e de distância relativa no modelo, ambas dadas logo a seguir:

$$dist_{Abs}(G, G') = \left| A(G) \Delta A(G') \right|$$

$$dist_{Rel}(G, G') = \frac{\|A(G) \Delta A(G')\|}{\max(|A(G)|, |A(G')|)}$$

onde o operador Δ representa a diferença simétrica entre dois conjuntos, ou seja, a diferença entre sua união e sua intersecção.

$$A \Delta B = (A \cup B) \setminus (A \cap B)$$

Aqui, como nos outros dois modelos definidos anteriormente, a distância relativa é a distância absoluta normalizada (neste caso, pelo máximo do número de arestas).

Dos três modelos de testabilidade de grafos tratados nesta seção, o modelo denso é o mais estudado, no sentido de que há mais publicações sobre ele do que sobre os outros dois, em parte em razão das limitações à aplicabilidade do modelo de grau limitado e em parte em razão da alta dificuldade técnica do estudo do modelo geral.

3.4 Algumas Importantes Classes de Propriedades de Grafos

Dizemos que uma propriedade é "natural", quando admite um testador que não depende do parâmetro de tamanho (n). As propriedades não naturais incluem, por exemplo, a propriedade de um grafo possuir ciclo hamiltoniano de ordem par ou a propriedade de um grafo ter ordem prima e não ter triângulos como subgrafos. Pode parecer paradoxal assumir que um algoritmo não depende do tamanho da representação da entrada, mas tal aparência não passa de um engano. Os testadores são algoritmos de tempo sublinear, pois não têm acesso à entrada inteira. Seu acesso à entrada se dá por meio do seu oráculo, e isso ocorre por um processo aleatório. Por esse motivo, o mero acesso de um testador à sua entrada não implica que ele possa computar o tamanho da entrada, já que tal acesso não é integral. Em outras palavras, se não nos é dado conhecer uma representação completa de um objeto, não poderemos dizer qual é o tamanho da sua representação na forma de uma *string* (uma sequência de símbolos de uma coleção finita). Desta forma, um testador alienado, que não depende do parâmetro de tamanho, é um testador e, como tal, recebe um parâmetro n e um parâmetro ε, mas despreza o parâmetro de tamanho (n) e computa mediante as consultas feitas pelo oráculo até aceitar ou rejeitar o grafo de entrada.

Podemos verificar, da definição acima, que uma propriedade que admita um testador alienado é, necessariamente, uma propriedade natural, porque esse testador, por definição, não depende do parâmetro de tamanho.

Em alguns artigos (como em [7]) aparece a expressão "fortemente testável" para denotar aquelas propriedades testáveis por testadores de erro unilateral com complexidade de consultas constante em relação à ordem do grafo de entrada no pior caso. Em outros artigos, por outro lado, considera-se que os testadores têm, necessariamente e por definição, complexidade constante em relação à ordem dos grafos de entrada no pior caso. Nesses casos, ao invés de dizer-se que uma propriedade é *fortemente testável*, diz-se simplesmente que a propriedade é *testável*, termo que adotamos neste trabalho.

Uma propriedade *facilmente testável* é uma propriedade para a qual existe um testador com erro unilateral e complexidade de consultas polinomial em função de ε^{-1} no pior caso. Esta expressão aparece, também, em [7].

Além das propriedades mencionadas acima, são de relevo para o estudo da testabilidade de propriedades de grafos as propriedades monótonas, hereditárias e semi-hereditárias, definidas no capítulo 2. Todas elas são propriedades testáveis, de acordo com Alon e Shapira [6].

3.5 Alguns Teoremas e suas Aplicações

Nesta seção, abordaremos alguns dos principais teoremas sobre testabilidade de propriedades de grafos no modelo denso e debruçar-nos-emos mais pormenorizadamente em alguns métodos dedutivos em que tais teoremas são explorados. O teorema a seguir nos dá uma ideia de como podemos restringir o âmbito dos testadores que realmente precisamos estudar para compreendermos bem a testabilidade de propriedades de grafos. Dos três teoremas seguintes, os dois primeiros são demonstrados no trabalho de Goldreich e Trevisan [17], e o terceiro é demonstrado no trabalho de Alon e Shapira [6].

Teorema 3.1. *Toda propriedade testável admite um testador não-adaptativo.*

Com isso, sempre que houver um testador para uma propriedade P de grafos, haverá, também, um testador não-adaptativo para P. Assim, para qualquer efeito, nós sempre poderemos restringir nossa busca aos testadores não-adaptativos, quando queremos saber se certas propriedades admitem ou não testadores. Em outras palavras, problemas sobre existência de testadores para dadas propriedades convertem-se, sem perda ou ganho de generalidade, em problemas sobre existência de testadores não-adaptativos para as referidas propriedades. O próximo teorema nos permite estender tal entendimento para a testabilidade de propriedades de grafos com erro unilateral.

Teorema 3.2. *Toda propriedade testável com erro unilateral admite um testador não-adaptativo com erro unilateral.*

O teorema seguinte permite-nos fazer algumas interessantes observações envolvendo propriedades hereditárias e semi-hereditárias, testadores alienados e testadabilidade com erro unilateral.

Teorema 3.3. *Uma propriedade admite um testador alienado de erro unilateral se, e somente se, ela é semi-hereditária.*

Isso nos garante que sempre que lidarmos com propriedades monótonas ou hereditárias poderemos estar certos da existência de testadores alienados para tais propriedades. Mais do que isso: poderemos encontrar testadores alienados de erro unilateral para essas propriedades. O teorema acima é mais forte do que esta conclusão, pois ela decorre do teorema, mas a recíproca não é verdadeira. Para verificar esta relação de decorrência unidirecional, basta atestar que toda propriedade monótona é hereditária, toda propriedade hereditária é semi-hereditária e que o teorema em questão é um bicondicional que pode ser expresso como a conjunção dos dois condicionais expostos logo a seguir, dos quais o segundo completa a prova da conclusão:

1- Se uma propriedade admite um testador alienado de erro unilateral, então ela é semi-hereditária.

2- Se uma propriedade é semi-hereditária, então ela admite um testador alienado de erro unilateral.

O Teorema 3.3, assim, reúne todas essas informações de maneira compacta.

Nesse mesmo sentido há o teorema transcrito a seguir, o qual pode ser encontrado com sua demonstração em Goldreich e Trevisan [17].

Teorema 3.4. *Seja P uma propriedade de grafos. Se existe um testador com complexidade de consultas $q(n,\varepsilon)$ para P, então existe um testador canônico para P com complexidade de consultas $O(q(n,\varepsilon))$. Além disso, se o testador original tem erro unilateral, o* *testador canônico também tem erro unilateral e sua complexidade de consultas no pior caso (a do testador canônico) é de* $2q(n,\varepsilon)$

O lema a seguir é crucial para o estudo das propriedades hereditárias, conforme os métodos descritos posteriormente.

Lema 3.1 (Lema da Remoção para grafos)**.** *Para todo* $\varepsilon > 0$ *e toda família de grafos* $\mathcal{F}$*, existem* $\delta > 0$ *e inteiro positivo* k *tais que todo grafo* G *de ordem* n *que seja* ε*-longe de ser livre de* $\mathcal{F}$ ***induzido*** *contém pelo menos* δn^h *cópias* ***induzidas*** *de um grafo* $H \in \mathcal{F}$ *tal que* $h = |H| \leq k$*.*

Este lema tem uma variante para a propriedade de ser F livre e para cópias não necessariamente induzidas, transcrita abaixo. Ambos podem ser encontrados em Alon e Shapira [4] e [6].

Lema 3.2. *Para todo* $\varepsilon > 0$ *e toda família de grafos* $\mathcal{F}$*, existem* $\delta > 0$ *e inteiro positivo* k *tais que todo grafo* G *de ordem* n *que seja* ε*-longe de ser livre de* $\mathcal{F}$ *contém pelo menos* δn^h *cópias de um grafo* $H \in \mathcal{F}$ *tal que* $h = |H| \leq k$*.*

Destes dois últimos teoremas, nossa atenção se concentra no primeiro deles, o Lema 3.1, pois podemos projetar testadores canônicos para propriedades hereditárias aplicando-o. Vejamos como podemos fazer isso.

Consideremos o seguinte algoritmo T sobre um grafo de entrada $G = (V,A)$, definido em função de dado grafo H, de ordem h:

1- Amostramos uniformemente e ao acaso $\left\lceil 2\delta^{-1} \right\rceil h$ vértices de V, onde δ depende de ε e de H em conformidade com o Lema 3.1. (Rigorosamente, devemos fixar o mínimo entre $\left\lceil 2\delta^{-1} \right\rceil h$ e n, mas este último caso é trivial e não será, adiante, considerado.)

2- Consideramos S formado pelos vértices amostrados no item 1.

3- Aceitamos G se, e somente se, $G[S]$ é livre de H.

Para provar que T é um testador para a propriedade "ser livre de H induzido", basta provar que:

I- Se G é livre de H induzido, então, $G[S]$ é livre de H induzido.

II-

$$P(T \ rejeita \ G | dist_{Rel}(G, LivreInd \ H) > \varepsilon) > \frac{2}{3}.$$

No item II acima, e no que segue, usamos a notação $P(A)$ para denotar a probabilidade associada ao evento A para dado espaço amostral.

A seguir, provaremos as assertivas I e II. A afirmação I é trivialmente verdadeira, já que não é possível que G seja livre de H induzido e um subgrafo induzido de G contenha H como subgrafo induzido (isso seria um absurdo). Mais do que isso: a probabilidade de que o testador aceite G, caso G seja livre de H induzido, é igual a 1, de modo que esse testador tem erro unilateral.

Suponhamos, agora, que G esteja ε-longe de ser livre de H induzido. A afirmação II significa que, neste caso, $P(T \ rejeita \ G) > \frac{2}{3}$. Queremos provar essa desigualdade.

Primeiramente definamos os eventos C_1 e C_2 de um experimento aleatório E_1, que consiste em sortear os conjuntos A e B de vértices do grafo G, com distribuição uniforme, sobre o universo W, assim definido:

$$W = \left\{(A, B) : A \subseteq V \wedge B \subseteq V \wedge |A| = h \wedge |B| = \left\lceil 2\delta^{-1} \right\rceil h\right\}$$

Chamemos $\left\lceil 2\delta^{-1} \right\rceil h$ de g para simplificar a escrita das expressões posteriores. Sejam

$$C_1 = \{(A, B) \in W : A \subseteq B\}$$

$$C_2 = \{(A,B) \in W : G[A] \simeq H\}$$

Podemos contar a cardinalidade de C_1, o que será levado em conta mais adiante. Ao concluirmos a contagem, concluímos o seguinte:

$$|C_1| = \binom{n}{h}\binom{n-h}{g-h}$$

Também podemos contar um limitante inferior da cardinalidade de $C_1 \cap C_2$, levando em conta o teor do Lema 3.1. Aplicando-se este lema, concluímos que o número de conjuntos A que podem ser sorteados no experimento E_1 é maior ou igual a δn^h, e, para cada A selecionado pelo experimento E_1 há $\binom{n-h}{g-h}$ pares (A,B). Consequentemente,

$$|C_1 \cap C_2| \geq \delta n^h \binom{n-h}{g-h}$$

Sabemos que

$$|C_1 \cap C_2| + \left|C_1 \cap C_2^c\right| = |C_1|$$

Portanto

$$\left|C_1 \cap C_2^c\right| = |C_1| - |C_1 \cap C_2|$$
$$\left|C_1 \cap C_2^c\right| \leq \binom{n}{h}\binom{n-h}{g-h} - \delta n^h \binom{n-h}{g-h}$$

O que queremos calcular é o seguinte. Suponhamos que sorteamos uma amostra de tamanho g do conjunto de vértices de G com distribuição uniforme. Consideremos, também que, desta amostra, selecionamos, também com distribuição uniforme, h vértices. Qual é a probabilidade de o grafo induzido por este conjunto com cardinalidade h não ser cópia de H? Esta propriedade é dada por

$$P\left((A,B) \in C_2^c | (A,B) \in C_1\right) = \frac{P\left((A,B) \in C_2^c \cap C_1\right)}{P\left((A,B) \in C_1\right)}$$

$$
\begin{aligned}
&= \frac{\left|C_1 \cap C_2^c\right|}{|C_1|} \\
&\leq \frac{\binom{n}{h}\binom{n-h}{g-h} - \delta n^h \binom{n-h}{g-h}}{\binom{n}{h}\binom{n-h}{g-h}} \leq 1-\delta
\end{aligned}
$$

Consequentemente, a probabilidade de uma seleção aleatória com distribuição uniforme de h vértices extraídos de uma amostra de tamanho g do grafo de entrada, também com distribuição uniforme, não formar uma cópia induzida de H é menor ou igual a $1-\delta$.

Consideremos, agora, o experimento E_2, que consiste em selecionar com probabilidade uniforme uma amostra L de V com cardinalidade g e, posteriormente, selecionar uma partição $\left\{U_1, U_2, \ldots, U_{\frac{g}{h}}\right\}$ desta amostra em uma coleção de $\frac{g}{h}$ subconjuntos de cardinalidade h. Chamemos de D_1 o evento de que todos os subconjuntos da amostra L com cardinalidade h não formem cópias induzidas de H em G. E chamemos de D_2 o evento em que U_j, para todo $j \in \left[\frac{g}{h}\right]$, não forma cópia induzida de H. Antes verificamos que

$$
\begin{aligned}
P\left(\sim G[U_k] \simeq H | \bigwedge_{j=1}^{k-1} \sim G\left[U_j\right] \simeq H\right) &\leq P\left(\sim G[U_k] \simeq H\right) \\
&= P\left((A,B) \in C_2^c | (A,B) \in C_1\right) \leq 1-\delta
\end{aligned}
$$

Com isso, verifica-se que

$$
\begin{aligned}
& P\left(\bigwedge_{S \in V', |S|=h} \sim G[S] \simeq H\right) \leq P\left(\bigwedge_{j=1}^{g} \sim G\left[U_j\right] \simeq H\right) \\
\leq\ & P\left(\sim G[U_1] \simeq H\right) \prod_{k=2}^{\frac{g}{h}} P\left(\sim G[U_k] \simeq H | \bigwedge_{j=1}^{k-1} \sim G\left[U_j\right] \simeq H\right) \\
\leq\ & (1-\delta) \prod_{k=2}^{\frac{g}{h}} (1-\delta) = (1-\delta)^{\lceil 2\delta^{-1} \rceil}
\end{aligned}
$$

O evento $\bigwedge_{\substack{S \in V' \\ |S|=h}} \sim G[S] \simeq H$, de cuja probabilidade acabamos de calcular um limitante superior em função de δ, é o evento "T aceita G". Para concluir:

$$P(T\ aceita\ G) = P\left(\bigwedge_{\substack{S \in V' \\ |S|=h}} \sim G[S] \simeq H\right) \leq (1-\delta)^{\lceil 2\delta^{-1} \rceil} \leq \left(1 - \frac{1}{\delta^{-1}}\right)^{2\delta^{-1}} \leq e^{-2} \leq \frac{1}{3}$$

Isso prova que o algoritmo exposto é um testador canônico com erro unilateral para a propriedade de ser livre de um grafo H induzido. Este método pode ser generalizado para a propriedade de ser livre de F em que F seja uma família finita de grafos não necessariamente unitária, o que contribuirá para caracterizarmos a testabilidade de propriedades hereditárias. O testador canônico para a propriedade de ser livre da família de grafos $\mathcal{F}$ induzido opera da seguinte forma:

1- Sobre o grafo de entrada $G = (V, A)$, amostra-se uniformemente e ao acaso $\lceil 2\delta^{-1} \rceil h$ vértices de V, onde h é a ordem do maior grafo de $\mathcal{F}$ e δ depende de ε e de F em conformidade com o Lema 3.1.

2- Considera-se S formado pelos vértices amostrados no item 1.

3- Aceita-se G se, e somente se, $G[S]$ é livre de F.

A prova de validade deste testador é análoga à do testador para famílias unitárias, sobre o qual tratamos logo acima. Como no caso do testador canônico apresentado acima para a propriedade de ser livre de H induzido, é trivial que, se G é livre de uma família $\mathcal{F}$ induzido, então o subgrafo induzido pela amostra S também é livre de $\mathcal{F}$ induzido. E, como no caso do mencionado testador canônico, por demonstração análoga prova-se que a probabilidade de T rejeitar G condicionada ao evento de G ser ε-longe de $LivreInd\ G[F]$ é maior do que $\frac{2}{3}$.

O Teorema 2.1 caracteriza as propriedades hereditárias como as propriedades de um grafo ser livre de $\mathcal{F}$ induzido para alguma família $\mathcal{F}$. Assim, um testador para a propriedade de ser livre de $\mathcal{F}$ induzido, para alguma família $\mathcal{F}$, é um testador de uma propriedade hereditária.

3.6 Um testador que não é canônico

Neste capítulo, apresentaremos um exemplo de um testador que não é canônico. Desta forma, definiremos o algoritmo e demonstraremos que ele satisfaz os requisitos necessários para não ser classificado como canônico nos termos da Definição 3.3.

Sejam $\mathcal{F}_1$ e $\mathcal{F}_2$ duas famílias finitas de grafos. Sejam T_1 um testador canônico para a propriedade $LivreInd\ \mathcal{F}_1$ e T_2 um testador canônico para a propriedade $LivreInd\ \mathcal{F}_2$, ambos definidos tal como foi feito no final da Seção 3.5, quando definimos (e depois generalizamos) um testador que amostra uniformemente e ao acaso $\lceil 2\delta^{-1} \rceil h$ vértices de V, onde H é um grafo dado de ordem h e δ depende de ε e de H em conformidade com o Lema 3.1, e aceita G se, e somente se, $G[S]$ é livre de H, onde S é formado pelos vértices amostrados. Vamos definir um testador T_3 como o testador que procede da seguinte maneira:

Passo nº 1- Primeiramente, T_3 executa T_1.

Passo nº 2- Depois, T_3 executa T_2.

Passo nº 3 - Por fim, T_3 aceita, se T_1 e T_2 aceitam; ou rejeita, caso contrário.

Afirmamos que T_3 é um testador para a propriedade $LivreInd(\mathcal{F}_1 \cup \mathcal{F}_2)$. Se isso for verdade, T_3 é um exemplo de um testador que não é canônico. Para provar esta afirmação, precisamos demonstrar que:

Requisito nº 1- Se G, o grafo de entrada, pertence a $LivreInd(\mathcal{F}_1 \cup \mathcal{F}_2)$, T_3 aceita G com probabilidade maior do que $\frac{2}{3}$.

Requisito nº 2- Se G está ε-longe de $LivreInd(\mathcal{F}_1 \cup \mathcal{F}_2)$, T_3 rejeita o grafo de entrada (G) com probabilidade maior do que $\frac{2}{3}$.

A seguir, definimos dezesseis proposições matemáticas que nomearemos como $p_1, p_2, ..., p_{16}$. Na notação que empregaremos, escreveremos

$$p_k : \alpha$$

quando estivermos atribuindo o nome p_k para a proposição α. O método que adotaremos para demonstrar o Requisito nº 1 e o Requisito nº 2, expostos acima, visa a facilitar a conferência de cada passo.

No que segue daremos interpretação às expressões de p_1, p_2, etc..., estabeleceremos algumas relações entre as proposições recém-definidas a partir do conteúdo dos capítulos anteriores e, finalmente, exporemos uma sequência de argumentos expostos de forma padronizada que demonstrem que o Requisito nº 1 e o Requisito nº 2, fixados acima, são verdadeiros, concluindo a demonstração.

Nas proposições $p_1, p_2, ..., p_{16}$, as expressões F_1, F_2 e G terão o mesmo significado que o adotado até aqui. No Lema 3.1, a variável δ, que depende de ε e de F, pode ser entendida como uma função de ε para dada família F. No rol abaixo, a função $\delta(\varepsilon)$ será tal função assumindo que $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$. Da mesma forma, $\delta_1(\varepsilon)$ representará a função de tal variante do Lema 3.1 com $\mathcal{F} = \mathcal{F}_1$ e $\delta_2(\varepsilon)$ será a função correspondente à variável δ do Lema 3.1 no caso em que $\mathcal{F} = \mathcal{F}_2$.

Mais adiante estabeleceremos algumas relações entre $\delta(\varepsilon)$, $\delta_1(\varepsilon)$ e $\delta_2(\varepsilon)$.

$$p_1 : G \in LivreInd\ \mathcal{F}_1$$

$$p_2 : G \in LivreInd\ \mathcal{F}_2$$

$$p_3 : G \in LivreInd\,(\mathcal{F}_1 \cup \mathcal{F}_2)$$

$$p_4 : T_1\ aceita\ G$$

$$p_5 : T_2\ aceita\ G$$

$$p_6 : T_3\ aceita\ G$$

$$p_7 : dist_{Rel}\,(G, LivreInd\,(\mathcal{F}_1 \cup \mathcal{F}_2)) > \varepsilon$$

$$p_8 : \exists \delta > 0 \exists H \in (\mathcal{F}_1 \cup \mathcal{F}_2)\left(|\{K : K \simeq H \wedge K \subseteq G\}| \geq \delta n^h\right)$$

$$p_9 : \exists \delta \left(\left(\exists H \in \mathcal{F}_1 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta n^h\right)\right) \vee \left(\exists H \in F_2 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta n^h\right)\right)\right)$$

$$p_{10} : \exists H \in \mathcal{F}_1 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta(\varepsilon) n^h\right)$$

$$p_{11} : \exists H \in \mathcal{F}_2 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta(\varepsilon) n^h\right)$$

$$p_{12} : \exists H \in \mathcal{F}_1 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta_1(\varepsilon)\, n^h\right)$$

$$p_{13} : Prob\,(T_1\ rejeita\ G) > \frac{2}{3}$$

$$p_{14} : \exists H \in \mathcal{F}_2 \left(|\{K : K \simeq H \wedge K \lhd G\}| \geq \delta_2(\varepsilon)\, n^h\right)$$

$$p_{15} : Prob\,(T_2\ rejeita\ G) > \frac{2}{3}$$

$$p_{16} : Prob\,(T_3\ rejeita\ G) > \frac{2}{3}$$

Não afirmamos nem negamos as dezesseis proposições acima. Apenas as nomeamos, para facilitar a manipulação simbólica no posterior processo dedutivo.

Consideremos a seguinte tese.

Tese 3.1:

$$p_3 \Rightarrow (p_1 \wedge p_2)$$

Esta é uma forma abreviada de escrever que

$$G \in LivreInd\,(\mathcal{F}_1 \cup \mathcal{F}_2) \Rightarrow (G \in LivreInd\,(\mathcal{F}_1) \wedge G \in LivreInd(\mathcal{F}_2))$$

A veracidade desta afirmação, a Tese 3.1, decorre da definição de união de famílias. Assim, tal tese será assumida como verdadeira na demonstração que faremos.

A seguir, expomos uma sequência de doze outras teses que também serão assumidas como verdadeiras na demonstração imediatamente posterior e justificá-las-emos sucintamente, sendo todas redigidas de forma abreviada em função de p_1, p_2, p_3, etc..., tal como a Tese 3.1 foi apresentada em função de p_1, p_2 e p_3.

Tese 3.2:

$$p_1 \Rightarrow p_4$$

Esta afirmação decorre da definição de T_1 em função de $\mathcal{F}_1$.

Tese 3.3:

$$p_2 \Rightarrow p_5$$

Esta afirmação decorre da definição de T_2 em função de $\mathcal{F}_2$.

Tese 3.4:

$$(p_4 \wedge p_5) \Rightarrow p_6$$

Tal assertiva decorre do passo nº 3 da descrição do testador T_3, segundo o qual T_3 aceita, se T_1 e T_2 aceitam G.

Tese 3.5:

$$p_7 \Rightarrow p_8$$

Este é o enunciado do Lema 3.1 quando a família $\mathcal{F}$ é $\mathcal{F}_1 \cup \mathcal{F}_2$.

Tese 3.6:

$$p_8 \Leftrightarrow p_9$$

Esta afirmação decorre da definição de união de conjuntos.

Tese 3.7:

$$p_8 \Rightarrow (p_{10} \vee p_{11})$$

Esta tese decorre da Tese 3.5 e do fato de que, nesta última, δ pode ser entendido como função de ε.

Tese 3.8:

$$p_{12} \Rightarrow p_{13}$$

Esta afirmação decorre da definição do testador T_1.

Tese 3.9:

$$p_{14} \Rightarrow p_{15}$$

Esta afirmação decorre da definição do testador T_1.

Tese 3.10:

$$p_{10} \Rightarrow p_{12}$$

Em síntese, conforme a interpretação que estamos dando a $\delta(\varepsilon)$ e $\delta_1(\varepsilon)$, podemos assumir que $\delta(\varepsilon) \geq \delta_1(\varepsilon)$ para todo ε (vide parágrafo anterior às definições das proposições p_1, p_2, ..., p_{16}), pois, quando estendemos uma família F de grafos, mantemos ou aumentamos o número de cópias induzidas de membros da família resultante num grafo dado.

Para sermos mais rigorosos, segundo o Lema 3.1, para todo $\varepsilon > 0$ e toda família de grafos $\mathcal{F}$, existem $\delta > 0$ e inteiro positivo k tais que todo grafo G de ordem n que seja ε-longe de ser livre de F induzido contém pelo menos δn^h cópias induzidas de um grafo $H \in \mathcal{F}$ tal que $h = |H| \leq k$. Sendo assim, temos que:

1- Para todo ε, se o grafo G de ordem n é ε-longe de ser livre de $\mathcal{F}_1$ induzido, então G contém pelo menos $\delta_1(\varepsilon)n^h$ cópias induzidas de um grafo $H \in \mathcal{F}_1$ tal que $h = |H|$.

2- Para todo ε, se o grafo G de ordem n é ε-longe de ser livre de $\mathcal{F}_1 \cup \mathcal{F}_2$ induzido, então G contém pelo menos $\delta(\varepsilon)n^h$ cópias induzidas de um grafo $H \in \mathcal{F}_1 \cup \mathcal{F}_2$ tal que $h = |H|$.

Analisemos as duas sentenças acima. Se G é ε-longe de ser livre de $\mathcal{F}_1$ induzido, então G também é ε-longe de ser livre de $\mathcal{F}_1 \cup \mathcal{F}_2$ induzido e, conforme a segunda sentença acima, G contém pelo menos $\delta(\varepsilon)n^h$ cópias induzidas de um grafo $H \in \mathcal{F}_1 \cup \mathcal{F}_2$ tal que $h = |H|$. Porém, conforme a primeira sentença das duas acima, nesta mesma situação G contém pelo menos $\delta_1(\varepsilon)n^h$ cópias induzidas de um grafo $H \in \mathcal{F}_1$ tal que $h = |H|$. Naturalmente, pode haver mais cópias de algum grafo $H \in \mathcal{F}_1 \cup \mathcal{F}_2$ em G do que cópias de $H \in \mathcal{F}_1$ em G, pois $\mathcal{F}_1 \subseteq \mathcal{F}_1 \cup \mathcal{F}_2$. Assim, podemos assumir que $\delta(\varepsilon) \geq \delta_1(\varepsilon)$.

Tese 3.11:

$$p_{11} \Rightarrow p_{14}$$

Conforme a interpretação que estamos dando a $\delta(\varepsilon)$ e $\delta_2(\varepsilon)$, podemos assumir que $\delta(\varepsilon) \geq \delta_2(\varepsilon)$ para todo ε, pelo mesmo motivo exposto na justificação da Tese 3.10.

Tese 3.12:

$$p_{13} \Rightarrow p_{16}$$

O evento "T_1 rejeita G" está contido no evento "T_3 rejeita G", porque T_3 rejeita G se, e somente se, T_1 rejeita G ou T_2 rejeita G. Convém lembrar que a probabilidade de uma disjunção é maior ou igual à probabilidade de um de seus disjuntivos, pois um disjuntivo está contido no evento correspondente à disjunção.

Tese 3.13:

$$p_{15} \Rightarrow p_{16}$$

Como no caso da Tese 3.12, o evento "T_2 rejeita G" está contido no evento "T_3 rejeita G", porque T_3 rejeita G se, e somente se, T_1 rejeita G ou T_2 rejeita G.

No que segue, expomos uma sequência de sete argumentos em que tomamos como pressupostos as 13 teses enunciadas e justificadas acima.

Argumento 1:

$$p_1 \Rightarrow p_4 \text{ (Tese 3.2)}$$

$$p_2 \Rightarrow p_5 \text{ (Tese 3.3)}$$

$$\therefore (p_1 \wedge p_2) \Rightarrow (p_4 \wedge p_5)$$

Argumento 2:

$$(p_1 \wedge p_2) \Rightarrow (p_4 \wedge p_5) \text{ (Argumento 1)}$$

$$(p_4 \wedge p_5) \Rightarrow p_6 \text{ (Tese 3.4)}$$

$$\therefore (p_1 \wedge p_2) \Rightarrow p_6$$

Argumento 3:

$$p_3 \Rightarrow (p_1 \wedge p_2) \text{ (Tese 3.1)}$$

$$(p_1 \wedge p_2) \Rightarrow p_6 \text{ (Argumento 2)}$$

$$\therefore p_3 \Rightarrow p_6$$

Argumento 4:

$$p_8 \Rightarrow (p_{10} \vee p_{11}) \text{ (Tese 3.7)}$$

$$p_{10} \Rightarrow p_{12} \text{ (Tese 3.10)}$$

$$p_{11} \Rightarrow p_{14} \text{ (Tese 3.11)}$$

$$\therefore p_8 \Rightarrow (p_{12} \vee p_{14})$$

Argumento 5:

$$p_8 \Rightarrow (p_{12} \vee p_{14}) \text{ (Argumento 4)}$$

$$p_{12} \Rightarrow p_{13} \text{ (Tese 3.8)}$$

$$p_{14} \Rightarrow p_{15} \text{ (Tese 3.9)}$$

$$\therefore p_8 \Rightarrow (p_{13} \vee p_{15})$$

Argumento 6:

$$p_8 \Rightarrow (p_{13} \vee p_{15}) \text{ (Argumento 5)}$$

$$p_{13} \Rightarrow p_{16} \text{ (Tese 3.12)}$$

$$p_{15} \Rightarrow p_{16} \text{ (Tese 3.13)}$$

$$\therefore p_8 \Rightarrow p_{16}$$

Argumento 7:

$$p_7 \Rightarrow p_8 \text{ (Tese 3.5)}$$

$$p_8 \Rightarrow p_{16} \text{ (Argumento 6)}$$

$$\therefore p_7 \Rightarrow p_{16}$$

Verifica-se que, nesta sequência de argumentos, cada argumento é conclusão de um dos argumentos anteriores (e, portanto, já demonstrado) uma das treze teses assumidas como verdadeiras. Consequentemente, a conclusão de qualquer um destes sete argumentos é, necessariamente, uma proposição verdadeira.

Na sequência de argumentos acima, mostramos que

$$G \in LivreInd(\mathcal{F}_1 \cup \mathcal{F}_2) \Rightarrow T_3 \ aceita \ G$$

pois essa é a conclusão do Argumento 3. Isso prova o Requisito nº 1 definido anteriormente para que T_3 seja um testador para a propriedade $LivreInd(F_1 \cup F_2)$.

Na mesma sequência de argumentos, acima, de acordo com a conclusão do Argumento 7, deduzimos que $p_7 \Rightarrow p_{16}$, o que é o mesmo que dizer que

$$dist_{Rel}(G, LivreInd(F_1 \cup F_2)) > \varepsilon \Rightarrow Prob(T_3 \ rejeita \ G) > \frac{2}{3}$$

Isso prova o Requisito nº 2 definido anteriormente para que T_3 seja um testador para a propriedade $LivreInd(F_1 \cup F_2)$. Mas T_3, tal como definido no início desta seção, não se enquadra na Definição 3.3, apesar de existir testador canônico equivalente a T_3. Portanto T_3 é um testador que não é canônico, como queríamos demonstrar.

3.7 A testabilidade das propriedades monótonas

Ser testável é um atributo forte de uma propriedade de grafos, no sentido de que, se escolhermos aleatoriamente uma propriedade de grafos, provavelmente ela não será testável. Apesar disso, há exemplos de propriedades que são sabidamente testáveis, conforme se demonstrou na literatura. Nosso foco nesta Seção 3.7 é o teorema abaixo.

Teorema 3.5. *Toda propriedade monótona é testável com erro unilateral.*

Uma função recursiva é uma função com domínio e contradomínio em conjuntos numéricos tal que a imagem de cada ponto do domínio pode ser computada a partir de tal ponto mediante a execução do mesmo algoritmo. Faremos aqui um resumo da prova do Teorema 3.5. Os detalhes podem ser encontrados em Alon e Shapira [3].

Partamos das definições seguintes.

Definição 3.4. *Para toda família de grafos $\mathcal{F}$ (possivelmente infinita) e todo inteiro r, $\mathcal{F}_r$ é o seguinte conjunto de grafos: Um grafo R (não necessariamente pertencente a $\mathcal{F}$) é membro de $\mathcal{F}_r$ se, e somente se, R tem no máximo r vértices e existe no mínimo um grafo H pertencente a $\mathcal{F}$ tal que existe um homomorfismo de H para R.*

Definição 3.5. *Para toda família de grafos $\mathcal{F}$ tal que $\mathcal{F}_r$ é não vazio e todo inteiro r, Ψ_F:$\mathbb{N} \longrightarrow \mathbb{N}$ é a função tal que*

$$\Psi_{\mathcal{F}}(r) = \max_{R \in \mathcal{F}_r} \min_{\{H \in \mathcal{F} : H \mapsto R\}} |V(H)|$$

Definição 3.6. *Entendendo o parâmetro γ do Lema 2.2 como uma função dos parâmetros (ε, k, f), α':$\mathbb{N} \longrightarrow (0,1)$ será definida como a função tal que $\alpha'(0) = \frac{\varepsilon}{8}$ e*

$$r \neq 0 \Rightarrow \alpha'(r) = \gamma\left(\frac{\varepsilon}{8}, r, \Psi_F(r)\right)$$

Definição 3.7. *α:$\mathbb{N} \longrightarrow (0,1)$ é a função tal que $\alpha(0) = \frac{\varepsilon}{8}$ e $r \neq 0$ implica que $\alpha(r) = min\left\{\alpha'(r), \frac{\alpha(0)}{4}, \frac{1}{4r^2}\right\}$*

A função S é definida a partir da função fixada mediante a Definição 2.13, da seguinte maneira

Definição 3.8. *S é a função que obedece a*

$$S(\varepsilon) = W_{\alpha, \frac{8}{\varepsilon}}\left(100\left(\frac{8}{\varepsilon}\right)^4\right)$$

sendo que S é a função com o maior domínio possível tal que a expressão acima é legítima.

A seguir, apresentamos as Definições 3.9 e 3.10. A primeira depende do parâmetro J do Lema 2.2, que pode ser entendido como função de (ε, k, f), e a segunda depende do parâmetro δ do mesmo lema, o qual também pode ser compreendido como função de (ε, k, f).

Definição 3.9. *N é a função tal que*

$$N(\varepsilon) = S(\varepsilon) J\left(\frac{\varepsilon}{8}, S(\varepsilon), \Psi_F(S(\varepsilon))\right)$$

com o maior domínio possível que não comprometa a legitimidade desta definição.

Definição 3.10. *Q é a função tal que*

$$Q(\varepsilon) = \frac{2\Psi_F(S(\varepsilon))}{\delta\left(\frac{\varepsilon}{8}, S(\varepsilon), \Psi_F(S(\varepsilon))\right)} S(\varepsilon)$$

com o maior domínio possível que não comprometa a legitimidade desta definição.

Alon e Shapira [3] mostraram o seguinte resultado:

Teorema 3.6. *Se Ψ_F é uma função recursiva, então δ, J, S, N e Q podem ser computadas.*

Seja $\mathcal{F}$ uma família de grafos (finita ou infinita). E seja G um grafo com $n \geq N(\varepsilon)$ vértices. Apliquemos o Lema 2.6 sobre G para gerar uma equipartição $\left\{V_j\right\}_{j=1,2,\ldots,k}$ e a coleção de conjuntos de vértices $\left\{U_j\right\}_{j=1,2,\ldots,k}$, em que, para cada i, $U_i \subseteq V_i$. Nesta aplicação do Lema 2.6, fixemos $m = \frac{8}{\varepsilon}$ e $\beta(\varepsilon) = \alpha(\varepsilon)$ (vide

Definição 3.7). U é o subgrafo de G induzido por $\bigcup_{j=1}^{k} U_j$ gerado pela partição $\left\{U_j\right\}_{j=1,2,\ldots,k}$.

O grafo G' decorre da remoção, no grafo G, das seguintes arestas, nesta ordem:

1- Primeiramente removemos as arestas $\{u,v\}$ tais que $u \in V_i$ e $v \in V_i$, para algum i. Como o número de vértices de cada V_i é no máximo $\frac{n}{k}+1$ e o número de partes V_i é no máximo igual a k, neste passo são removidas no máximo $k\left(\frac{n}{k}\right)^2$ arestas. Pelo fato de que $k \geq \frac{8}{\varepsilon}$, este limitante superior é menor do que $\frac{\varepsilon}{8}n^2$.

2- Para cada i e cada j tais que $\left|dens_G\left(V_i,V_j\right) - dens_G(U_i,U_j)\right| > \frac{\varepsilon}{8} = \zeta(0)$, removemos arestas $\{u,v\}$ tais que $u \in V_i$ e $v \in V_j$. De acordo com o item 4 do Lema 2.6, existem no máximo $\frac{\varepsilon}{8}k^2$ pares (i,j). Cada V_i tem no máximo $\frac{n}{k}+1$ vértices, de maneira que, neste segundo passo, no máximo $\frac{\varepsilon}{8}k^2\left(\frac{n}{k}+1\right)^2 \leq \frac{\varepsilon}{7}n^2$ vértices.

3- Para cada i e cada j tais que $dens_G\left(U_i,U_j\right) < \frac{\varepsilon}{8}$, removem-se as arestas $\{u,v\}$ tais que $u \in V_i$ e $v \in V_j$. Como no passo anterior removeram-se todas as arestas $\left\{u_i,u_j\right\}$, para cada i e cada j que obedecem a desigualdade $\left|dens_G\left(V_i,V_j\right) - dens_G(U_i,U_j)\right| > \frac{\varepsilon}{8}$, $u_i \in V_i$ e $u_j \in V_j$, temos que, neste terceiro passo, removemos, no máximo $k^2\frac{\varepsilon}{4}\left(\frac{n}{k}+1\right)^2 \leq \frac{\varepsilon}{3}n^2$ arestas, já que $dens_G\left(U_i,U_j\right) < \frac{\varepsilon}{8} \Rightarrow dens_G\left(V_i,V_j\right) < \frac{\varepsilon}{8} + \beta(0) = \frac{\varepsilon}{4}$.

Tendo em vista o número máximo de arestas que podem ser removidas em cada um dos três passos enumerados acima, podemos verificar que εn^2 é um limitante superior do número total de arestas removidas neste procedimento, isto é, $|G'| < \varepsilon n^2$.

Definamos R como o grafo regularidade de U, com parâmetro de densidade $\frac{\varepsilon}{8}$ e parâmetro de regularidade $\frac{\varepsilon}{8}$ em relação à partição $\left\{U_j\right\}_{j=1,2,\dots,k}$. Seus vértices serão chamados de r_1, r_2, ..., r_k.

O grafo G' é formado por remoção de arestas de G e, por isso, todo subgrafo de G' é também subgrafo de G. Tendo em mente a definição de grafo ε-longe de uma propriedade, como nós removemos menos do que εn^2 arestas de G para formar G', existe um membro de F que tem cópia em G' e, portanto, em G, o qual chamaremos de F'.

Para cada $i \in [k]$, definiremos $R_i = V(F') \cap V_i$. Sabemos que, se $\{v_i, v_j\}$ é aresta de G', $v_i \in V_i$ e $v_j \in V_j$, então (U_i, U_j) é um par $\alpha(k)$-regular com densidade maior ou igual a $\frac{\varepsilon}{8}$ em G. Portanto existe um homomorfismo $\varphi : V(F') \longrightarrow V(R)$ que mapeia todo $R_i \subseteq V(F')$ a r_i.

Como $|V(R)| = k$ e existe o homomorfismo mencionado acima, conclui-se, conforme a Definição 3.4, que $R \in F_k$. Consequentemente existe um grafo $K \in F$ de tamanho menor ou igual a $\Psi_F(k)$ tal que existe um homomorfismo $\varphi : V(K) \longrightarrow V(R)$, conforme a definição de Ψ_F.

O homomorfismo φ atesta que se (i, j) é aresta de F', $(\varphi(i), \varphi(j))$ é uma aresta de R. Da mesma forma, sempre que $(\varphi(i), \varphi(j))$ é uma aresta de r, $\left(U_{\varphi(i)}, U_{\varphi(j)})\right)$ é um par $\alpha(k)$-regular com densidade maior ou igual a $\frac{\varepsilon}{8}$ em G. Alon e Shapira [3] deduziram um limitante superior para a ordem de F', que é o valor $\Psi_{\mathrm{F}}(k)$.

Portanto, se $\left\{u_i, u_j\right\} \in A(F')$, então $\left(U_i, U_j\right)$ é um par $\alpha(k)$-regular com densidade maior ou igual a $\frac{\varepsilon}{8}$ em G.

Pode-se aplicar o Lema 2.2 sobre a partição $\left\{U_j\right\}_{j=1,\dots,k}$ de U, levando-nos a concluir que há no mínimo $\delta \prod_{i=1}^{k} |U_i|$ cópias de F em U e, consequentemente, em G (pois U é subgrafo de G), onde δ é uma das variáveis do enunciado do Lema 2.2, a qual pode ser assumida como uma função de ε, sem prejuízo. Escolhendo uma função δ que satisfaça isso, podemos sustentar a veracidade da seguinte afirmação:

Afirmação 3.1. *Existem no mínimo $\delta(\varepsilon)\prod_{i=1}^{k}|U_i|$ cópias de F em G.*

Naturalmente, a função δ depende da família $\mathcal{F}$. É fácil notar que é vantajoso encontrarmos uma definição precisa para δ. Nesse sentido, partindo da Afirmação 3.1 e fixando as funções referidas nas Definições 3.4, 3.5, 3.6, 3.7, 3.8, 3.9 e 3.10, Alon e Shapira [3] mostraram que é válido o seguinte teorema:

Teorema 3.7. *Para toda família de grafos $\mathcal{F}$ (possivelmente infinita) e todo grafo G tal que $|G| > N_{\mathcal{F}}(\varepsilon)$ e G é ε-longe de ser livre de $\mathcal{F}$, então o subgrafo induzido randômico de $Q_{\mathcal{F}}(\varepsilon)$ vértices de G gera um membro de $\mathcal{F}$ com probabilidade maior ou igual a $\frac{2}{3}$.*

O conceito que será definido adiante corresponde a uma caracterização da propriedade de ser livre de determinada família de grafos (finita ou infinita).

Definição 3.11. *Para toda propriedade monótona P, chamaremos de F_P o conjunto de grafos tais que todo membro de F_P não satisfaze P e todo grafo obtido de um membro de F_P por remoção de uma aresta ou vértice satisfaz P.*

Primeiramente, observemos que satisfazer P é equivalente a ser livre de F_P. Definamos o seguinte testador T. Tomemos os valores das funções N e Q para dado ε. Como N e Q são funções cujas imagens podem ser computadas, T pode, sempre que possível, levar em conta os valores de N e Q. Seja G um grafo. Se a ordem de G for menor do que N, T consulta todas as arestas possíveis de G, testa se G é livre de F, aceitando em caso afirmativo (nesse caso, T não erra). Se a ordem de G é maior do que N, T sorteia Q vértices, encontra o subgrafo induzido por este conjunto de vértices, o qual chamaremos de H, e aceita se, e somente se, H é livre de F. Com esta descrição, percebe-se que, se G é livre de F, então T aceita com probabilidade igual a 1. Desta descrição e do Teorema 3.7, também se verifica que, se G estiver ε-longe de satisfazer P (ou de ser livre de F_P), o grafo amostrado de ordem Q conterá um subgrafo pertencente a F_P com probabilidade maior ou igual a $\frac{2}{3}$. Nesse caso, T rejeitará G com probabilidade maior ou igual a $\frac{2}{3}$. De uma forma ou de outra, a complexidade máxima de consultas está delimitada superiormente por $\max(N(\varepsilon), Q(\varepsilon))$, uma função constante em relação à ordem do grafo de entrada. Isso prova o Teorema 3.5.

4 TESTABILIDADE DE PROPRIEDADES EM GERAL

Neste capítulo, abordaremos testabilidade de outros tipos de objetos matemáticos, não estudados no capítulo anterior. Focar-nos-emos aqui prinipalmente em testabilidade de propriedades de torneios, dígrafos e permutações, mas salienta-se que isso não esgota o estudo de testabilidade de propriedades. Stagni [27] fez uma síntese dos conceitos e resultados elementares em matéria de testabilidade de propriedades de torneios. Hoppen, Kohayakawa, Moreira e Sampaio [23] estudaram testabilidade de propriedades de permutações, assim como Fox e Wei [16], sendo este último trabalho mais específico sobre as diferentes métricas sobre espaços de permutações e suas relações. Bender e Ron [8] apresentam alguns conceitos elementares sobre testabilidade de propriedades de dígrafos e, no mesmo trabalho, estudaram algumas propriedades de dígrafos. Em [20] tem-se uma introdução de testabilidade de propriedades de vários tipos de objetos matemáticos.

4.1 Testabilidade de propriedades de torneios

Vimos no Capítulo 2 que existem diferentes modelos de testabilidade de grafos, entre os quais o modelo denso, o modelo de grau limitado e o modelo geral. No caso dos torneios, para nossos propósitos, adotaremos um modelo de testabilidade análogo ao modelo denso para grafos. Nesse modelo, a métrica depende da *função de adjacência* $g_T : V(T) \times V(T) \longrightarrow \{0,1\}$, onde V é o conjunto de vértices de um torneio T. Essa função associa a cada par ordenado de vértices o elemento 1, se o referido par for arco do torneio, e o elemento 0, caso contrário. Assim, essa função fornece o componente da matriz de adjacência do torneio na linha correspondente a um vértice e na coluna correspondente a outro vértice, definida de modo análogo à matriz de adjacência de um grafo. Pela maneira como a função de adjacência de um torneio T, a função g_T, está definida, podemos dizer que dados os vértices u e v do torneio T, vale que:

$$g_T(u,v) = 1 \Leftrightarrow (u,v) \in \vec{A}(T)$$

$$g_T(u,v) = 0 \Leftrightarrow (v,u) \in \vec{A}(T)$$

Sejam T e T' dois torneios com o mesmo conjunto de vértices. A *distância absoluta* entre estes torneios é o número de pontos do domínio da função de adjacência para os quais a imagem é diferente. Simbolicamente, escrevemos: $dist_{Abs}(T,T') = |\{(u,v) \subseteq V \times V : g_T(u,v) \neq g_{T'}(u,v)\}|$. A *distância relativa* entre T e T' é a distância absoluta entre eles normalizada, isto é, dividida por n^2, sendo n o tamanho T e de T'. Nesse caso, escreve-se

$$dist_{Rel}(T,T') = |\{(u,v) \subseteq V \times V : g_T(u,v) \neq g_{T'}(u,v)\}|/n^2$$

Tal como no caso do modelo denso para grafos, aqui também cabe observar que estas definições de distância requerem que os torneios tenham o mesmo conjunto de vértices, o que é uma condição mais forte do que os referidos torneios terem o mesmo tamanho.

Aqui também é importante o conceito de distância entre um torneio e uma propriedade, definido da seguinte forma:

$$dist(T,P) = \min_{T' \in P} [dist_{Rel}(T,T')]$$

Uma forma de definir a distância entre torneios e propriedades no modelo aqui adotado de testabilidade de torneios, equivalente ao modelo denso de testabilidade de propriedades de grafos, é por meio do conceito de torneio ε-*longe* de certa propriedade P. Um torneio T está ε-*longe* da propriedade P, quando é necessário inverter εn^2 ou mais arcos de G para transformá-lo num torneio que satisfaz P. A distância entre T e P pode ser entendida como o menor ε tal que T está ε-longe de P.

Assim como há testadores de propriedades de grafos, há, também, testadores de propriedades de torneios. Um *testador* para uma propriedade de torneios é um algoritmo randômico sublinear associado a um parâmetro de tamanho (n), a um parâmetro de distância (ε) e a um oráculo com acesso à representação de um torneio. Ao oráculo é dada a faculdade de fazer consultas à representação do torneio de entrada, de maneira que o algoritmo possa acionar o oráculo em certos momentos e fazer escolhas a partir das respostas às consultas já realizadas pelo oráculo ou de escolhas feitas a partir dos dados resultantes de consultas anteriores ao oráculo. No final da computação de um testador sobre um torneio, o testador aceita ou rejeita o torneio de entrada, e vale o seguinte:

1- Se o torneio T satisfaz a propriedade P, então a probabilidade de o testador A aceitar T é maior do que $\frac{2}{3}$.

2- Se o torneio T está a uma distância maior ou igual a ε da propriedade P, então a probabilidade de o testador A rejeitar G é maior do que $\frac{2}{3}$, para a métrica sobre o conjunto dos torneios.

Se existir um testador tal que a probabilidade citada no item 1 acima for igual a 1, dizemos que a propriedade em questão é testável *com erro unilateral*. Caso contrário, dizemos que ela é testável *com erro bilateral*. Em ambos os casos, dizemos que a propriedade de torneios em questão é *testável*.

Passemos a uma breve análise da computação de um testador. O oráculo tem acesso à matriz de adjacência do torneio de entrada. O parâmetro de tamanho representa o tamanho do torneio de entrada. Neste modelo, o testador pode, em determinados momentos da sua computação, acionar o oráculo para que este consulte um componente da matriz de adjacência do torneio de entrada. Obviamente, a resposta a esta consulta informa ao testador sobre a relação de dominância entre os vértices u e v, em que u e v são vértices do torneio de entrada. Uma consulta do oráculo de um testador, pode ser entendida como a pergunta de se determinado par de vértices distintos (u,v) corresponde a um arco do torneio ou se (v,u) é um arco do mesmo torneio, dado que um destas respostas, e apenas uma destas, deverá ser sim. Essa resposta à consulta do oráculo é representada por 0 ou 1.

O oráculo consulta pares ordenados de vértices do torneio de entrada e fornece a imagem da função de adjacência sobre pares. Como vimos, esta função indica-nos se um par ordenado sobre o conjunto de vértices de um torneio corresponde a um arco ou não, e é somente esse o tipo de informação sobre o que o oráculo de um testador consulta.

Com o propósito de simplificar o estudo dos testadores de propriedades de grafos, na Seção 3.2, definimos *testador não-adaptativo*, *testador alienado* e *testador canônico* de propriedades de grafos, e aqui, no presente, fazemos o mesmo para torneios. O leitor não terá dificuldade em fazer as cabíveis analogias entre estes tipos de testadores de propriedades de torneios e os correspondentes tipos de testadores de propriedades de grafos.

Definição 4.1 (Testador não-adaptativo de torneios)**.** *Um* testador não-adaptativo *é um testador A associado a uma função $q : [0,1] \times \mathbb{N} \longrightarrow \mathbb{N}$ tal que A procede da seguinte maneira sobre um torneio de entrada T de tamanho n e sobre o parâmetro ε:*

1- *Inicialmente A escolhe randomicamente e com distribuição uniforme um subconjunto S de cardinalidade $q(\varepsilon, n)$ de $V(T)$.*

2- *Posteriormente, A faz $\binom{q}{2}$ consultas ao oráculo para obter o subtorneio induzido por S.*

3- *Finalmente, A aceita ou rejeita T tendo acesso apenas a $T[S]$ e a n, ou seja, sem fazer mais consultas ao oráculo.*

De maneira análoga, definem-se também os conceitos de *testador alienado de torneios* e *testador canônico de torneios*.

A mesma discussão que fizemos sobre propriedades naturais e propriedades fortemente testáveis cabe na discussão sobre testabilidade de propriedades de torneios. No caso dos torneios, as propriedades naturais são as propriedades que não dependem do parâmetro de tamanho, e falamos de propriedades fortemente testáveis num contexto em que ampliamos o conceito de testador para abranger também algoritmos que não sejam necessariamente limitados superiormente por uma função constante em relação ao tamanho do torneio, mas por uma função necessariamente sublinear em relação a este mesmo parâmetro. Stagni [27] observa que todas as propriedades testáveis estudadas na literatura são naturais, ou seja, admitem testadores que não dependem de n.

Podemos verificar, da definição acima, que uma propriedade que admita um testador alienado é, necessariamente, uma propriedade natural, porque esse testador, por definição, não depende do parâmetro de tamanho.

Uma propriedade de torneios *facilmente testável* é uma propriedade para a qual existe um testador com erro unilateral de complexidade polinomial em função de ε^{-1} no pior caso. A analogia com o caso correspondente para grafos é evidente.

Abordaremos aqui alguns dos principais teoremas sobre testabilidade de propriedades de torneios e debruçar-nos-emos mais pormenorizadamente em alguns métodos dedutivos em que tais teoremas são explorados. Uma demonstração destes teoremas pode ser encontrada no trabalho de Henrique Stagni [27].O primeiro teorema a seguir nos dá uma ideia de como podemos restringir o âmbito dos testadores que realmente precisamos estudar para compreendermos bem a testabilidade de propriedades de testabilidade.

Teorema 4.1. *Toda propriedade testável admite um testador não-adaptativo.*

Com isso, sempre que houver um testador para uma propriedade P de torneios, haverá, também, um testador não-adaptativo para P. Isso também acontecia no caso dos grafos, conforme vimos no Capítulo 3. Assim, para qualquer efeito, problemas de existência de testadores para certa propriedade convertem-se em problemas de existência de testadores não-adaptativos para as mesmas. O próximo teorema aprofunda esse entendimento.

Teorema 4.2. *Toda propriedade testável com erro unilateral admite um testador não-adaptativo com erro unilateral.*

O teorema seguinte permite-nos relacionar propriedades hereditárias, propriedades semi-hereditárias de torneios e testabilidade de propriedades de torneios com erro unilateral.

Teorema 4.3. *Uma propriedade de torneios é testável com erro unilateral se, e somente se, ela é semi-hereditária.*

Corolário 4.1. *Toda propriedade de torneios hereditária é testável com erro unilateral.*

A demonstração do Corolário 4.1 é imediata a partir do Teorema 4.3, tendo em vista que toda propriedade hereditária é semi-hereditária.

Na Seção 3.5 abordamos o Lema da Remoção para grafos e, posteriormente, na mesma seção, expusemos testadores canônicos para propriedades hereditárias em função da família $\mathcal{F}$ tal que a propriedade hereditária em questão é igual à propriedade de ser livre de $\mathcal{F}$ induzido (vide Teorema 2.1, segundo o qual toda propriedade hereditária pode ser definida em função uma família de grafos da qual é "livre induzido"). Abaixo apresentamos o Lema da Remoção para torneios.

Lema 4.1 (Lema da Remoção para torneios)**.** *Para todo $\varepsilon > 0$ e toda família de torneios $\mathcal{F}$, existem $\delta > 0$ e inteiro positivo k tais que, todo torneio T de tamanho n que seja ε-longe de ser livre de F contém pelo menos δn^h cópias de um torneio $H \in \mathcal{F}$ tal que $h = |H| \leq k$.*

Vejamos a seguir um testador para a propriedade de ser livre de um torneio H (a demonstração apoia-se no Lema 4.1). Consideremos o seguinte algoritmo a partir de dado torneio (V, A):

1- Amostramos uniformemente e ao acaso $\lceil 2\delta^{-1} \rceil h$ vértices de V, em que h o tamanho de H.

2- Consideramos S formado pelos vértices amostrados no item 1.

3- Aceitamos T se, e somente se, $T[S]$ é livre de H.

Pode-se provar que:

1- Se T é livre de H, então, o torneio induzido por S é livre de H.

2- $P < \frac{1}{3}$, onde P é a probabilidade de T ser aceito, se T estiver ε-longe de $Livre(H)$.

Provando estas duas afirmações acima, poderemos dizer que o algoritmo exposto é um testador canônico para a propriedade "ser livre de H". A demonstração segue basicamente os mesmos passos da demonstração para o testador análogo para propriedades de grafos, exposto na Seção 3.5 para a propriedade de ser livre de um grafo H induzido. Como a relação de ser subtorneio é transitiva, a afirmação 1 é verdadeira, de maneira que o erro do testador é unilateral.

Suponhamos, agora, que T esteja ε-longe de ser livre de H. A afirmação 2 nos diz que, neste caso, a probabilidade de T ser aceito é menor do que $\frac{1}{3}$. Conforme o Lema 4.1 (Lema da Remoção para torneios), a probabilidade de uma amostra de tamanho h do torneio de entrada, de tamanho n, formar uma cópia de H é maior do que δ. A amostra de T selecionada pelo testador contém $\lceil 2\delta^{-1} \rceil$ uplas distintas. Com isso,

$$P < (1-\delta)^{2\delta^{-1}} = \left(1 - \frac{1}{\delta^{-1}}\right)^{2\delta^{-1}} \leq e^{-2} < \frac{1}{3}$$

onde P é a probabilidade de T ser aceito.

Assim, o algoritmo exposto é um testador canônico com erro unilateral para a propriedade de ser livre de um torneio H. Como no caso dos grafos tratado na Seção 3.5, este método pode ser generalizado para a propriedade de ser livre de $\mathcal{F}$ em que $\mathcal{F}$ seja uma família finita de torneios.

4.2 Testabilidade de outros tipos de objetos matemáticos

No Capítulo 2.1, definimos dígrafos como um par de conjuntos finitos $(V, \vec{A})$, em que $\vec{A} \subseteq V \times V$. Definimos, também, seus vértices como os elementos de V e seus arcos como os elementos de A. No caso dos dígrafos, como seus arcos, ao contrário das arestas dos grafos simples, são direcionados, convém nomear um conceito semelhante ao caminho de um grafo simples, mas que leva em conta a direção dos arcos. Nesse sentido, chamamos de *caminho direcionado* de um dígrafo D uma sequência $((u_1, u_2), (u_2, u_3), (u_3, u_4) \dots (u_{k-1}, v_k))$ tal que D contenha os vértices distintos $u_1, u_2, \dots, u_k$. Se incluirmos em tal sequência de pares de vértices o par (v_k, v_1), gerando a sequência $((u_1, u_2), (u_2, u_3), (u_3, u_4) \dots (u_{k-1}, v_k), (v_k, v_1))$ temos um *ciclo direcionado*.

Um dígrafo D é *fortemente conexo*, quando existe um caminho direcionado no dígrafo entre quaisquer dois vértices distintos, ou seja, $\forall u \forall v$, tem-se que $(u_i, u_{i+1}) \in A(D)$ para $i = 1, 2, \ldots, k$, onde $u_1 = u$ e $u_{k+1} = v$. E um dígrafo D é *acíclico*, quando não contém ciclos direcionados, isto é, quando não contém vértices distintos $u_1, u_2, \ldots, u_k$ tais que (u_i, u_{i+1}) e (u_k, u_1), com $i = 1, 2, \ldots, k-1$. A noção de isomorfismo de dígrafos é definida de maneira análoga a isomorfismo de grafos ou de torneios, o que se discutiu no Capítulo 2. Uma propriedade de dígafos é um conjunto de dígrafos fechado sob isomorfismo. As propriedades de aciclicidade (conjunto de dígrafos que não contém ciclos direcionados) e conectividade forte (conjunto de dígrafos que têm pelo menos um caminho direcionado entre cada par de vértices) são úteis para fazermos algumas considerações sobre a testabilidade de dígrafos, motivo pelo qual definimo-las acima.

Um testador de dígrafos é um algoritmo T que verifica se um dígrafo satisfaz uma propriedade P ou se está ε-longe de satisfazê-la para qualquer $\varepsilon \in (0,1)$ de tal maneira que, se o dígrafo de entrada satisfizer P, T o aceita com probabilidade maior ou igual a $\frac{2}{3}$, e, se o dígrafo de entrada estiver ε-longe de satisfazer P, T rejeita este dígrafo com probabilidade maior ou igual a $\frac{2}{3}$.

Segundo Bender e Ron [8], há dois modelos de testabilidade de dígrafos: o modelo da matriz de adjacência e o modelo da matriz de incidência.

No modelo da matriz de adjacência, cada dígrafo é representado pela sua matriz de adjacência e cada testador pode fazer consultas das entradas desta matriz. A métrica é similar à do modelo denso de testabilidade de grafos, visto nas Seções 3.1 e 3.3, com a diferença de que, ali, verificam-se subconjuntos binários de vértices nas matrizes de adjacência, que são matrizes simétricas, enquanto aqui, verificam-se pares ordenados de vértices das matrizes de adjacência de dígrafos, as quais podem não ser simétricas. Como no caso do modelo denso de testabilidade de grafos, a distância absoluta entre dois dígrafos corresponde ao número de posições da matriz de adjacência em que as suas matrizes de adjacência diferem, e a distância relativa é a distância absoluta normalizada pelo tamanho de ambos os dígrafos, o qual deve ser o mesmo para que a métrica esteja definida. Sobre este modelo, podem se encontrar mais detalhes em Goldreich, Goldwasser e Ron [19].

Assim como o modelo da matriz de adjacência de testabilidade de dígrafos assemelha-se ao modelo denso para grafos, o modelo da matriz de incidência para dígrafos assemelha-se ao modelo de grau limitado para grafos. Neste último modelo, os arcos de cada dígrafo são associados aos números naturais não nulos menores ou iguais a d, sendo d um limitante superior dos graus dos seus vértices, e o dígrafo é representado pela matriz de incidência, e não pela matriz de adjacência. Sobre o tema, podem ser citados Goldreich e Ron [18]. Bender e Ron [8] observam que o modelo da matriz de incidência tem duas variantes diferentes. Na primeira, a matriz de incidência lista os vértices de saída associados a cada par (v, i), em que i é o i-ésimo arco do vértice v. Suponhamos que, ao consultar o par (v, i), o testador obtenha, como resposta, o vértice u. Neste caso, (v, u) é o i-ésimo arco de saída de v. Mas há também uma variante do modelo da matriz de incidência em que os testadores podem consultar não apenas os arcos que saem, como também os arcos que entram em cada vértice do dígrafo de entrada.

Dentre as duas variantes citadas acima do modelo da matriz de incidência, sempre devemos deixar claro a qual nos referimos a cada momento. Os testadores de ambas as variantes não são necessariamente os mesmos para a mesma propriedade de dígrafos, nem tampouco a complexidade de consultas máxima ou mínima associada a uma propriedade é preservada, quando mudamos de uma variante para a outra.

Bender e Ron [8] mostraram que a complexidade mínima de consultas da conectividade forte é de $O\left(\sqrt{n}\right)$ na primeira variante do modelo de matriz de incidência, e a complexidade máxima da conectividade forte é de $O\left(\frac{1}{\varepsilon}\right)$ na segunda variante. Isso significa que a complexidade de consultas desta propriedade maior ou igual a $c\sqrt{n}$ no primeiro caso e menor ou igual a $\frac{d}{\varepsilon}$ no segundo caso, onde c e d são duas constantes reais positivas. Note-se que ambos os limitantes de complexidade de consultas são sublineares. É fácil ver que a complexidade máxima da segunda variante não é válida para a primeira e que a complexidade mínima da primeira variante não é válida para a segunda, pois qualquer um destes dois casos culminaria num absurdo. Isso porque, sendo q a complexidade de consultas, quaisquer que sejam c e d, existem n e ε tais que é falso que $q \geq c\sqrt{n} \wedge q \leq \frac{d}{\varepsilon}$. Basta tomar n e ε tais que $c\sqrt{n} < \frac{d}{\varepsilon}$.

No caso da aciclicidade, os mesmos autores acima citados deduziram um limitante superior da complexidade de consultas para o modelo da matriz de adjacência e um limitante inferior para o modelo de matriz de incidência, a primeira em função apenas de ε, e a segunda somente em função de n (respeitada a sublinearidade, que sempre deve estar presente na testabilidade de propriedades). Por um caminho análogo ao do parágrafo anterior, pode-se mostrar que a complexidade de consultas deduzida para a aciclicidade no modelo da matriz de adjacência não vale para o modelo da matriz de incidência, e vice-versa.

Também podemos falar em testabilidade de propriedades de *permutações*. Uma permutação é uma bijeção de um conjunto finito em si mesmo. Não há grandes diferenças entre a definição de um testador de propriedade de permutações e os testadores vistos até agora. A diferença é que, aqui, a entrada é a representação de uma permutação, sendo que o tamanho da permutação é a cardinalidade do seu domínio. Um testador de permutações também está associado a um oráculo com acesso à permutação de entrada e também deve decidir com alta probabilidade de acerto se a permutação de entrada satisfaz uma propriedade ou se está ε-longe de satisfazê-la, sob alguma métrica. Entretanto há mais de um tipo de consultas que pode ser empregdo no modelo de testabilidade. Num dos casos, o oráculo consulta qual é a imagem da permutação de entrada sobre certo ponto do seu domínio. Percebe-se que, se a complexidade de consultas não fosse sublinear, o testador leria a representação completa da permutação de entrada, mas isso não se dá no âmbito da testabilidade de propriedades de estruturas discretas. Outro exemplo de tipo de consultas que um oráculo de um testador de propriedades de permutações pode fazer é verificar se, para dado i e dado j, sendo $i < j$, temos que $\pi(i) < \pi(j)$, onde π é a permutação de entrada.

Quando tratamos de grafos (no Capítulo 3), vimos que há mais de uma métrica para grafos em testabilidade. Quando estudamos dígrafos, pouco acima, enumeramos dois modelos, cada qual com uma representação diferente de dígrafos e, consequentemente, com uma métrica diferente. No caso das permutações, também há diversas métricas, adotadas para diferentes propósitos, como a distância retangular, a distância tau de Kendall, a distância de Spearman (em inglês: *Spearman's footrule distance*) e outras.

Sejam π_1 e π_2 duas permutações sobre $[n]$. A distância retangular entre π_1 e π_2, a seguir denotada ρ_Γ, é dada por

$$\rho_\Gamma(\pi_1, \pi_2) = \frac{1}{n} \max_{S,T} ||\pi_1(S) \cap T| - |\pi_2(S) \cap T||$$

Já a distância tau de Kendall, denotada ρ_{KT}, é definida como

$$\rho_{\mathrm{KT}}(\pi_1, \pi_2) = \frac{1}{\binom{n}{2}} |\{i \in [n] \wedge j \in [n] \wedge \pi_1(i) < \pi_1(j) \wedge \pi_2(i) > \pi_2(j)\}|$$

e a distância de Spearman, D, é dada por

$$D(\pi_1, \pi_2) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} |\pi_1(i) - \pi_2(i)|$$

onde, como antes, π_1 e π_2 são duas permutações sobre $[n]$.

Há algumas inequações lineares sobre diferentes métricas que já foram demonstradas. Dentre elas, Diaconis e Graham [14] provaram que:

$$\rho_{\mathrm{KT}}(\pi_1, \pi_2) \leq D(\pi_1, \pi_2) \leq 2\rho_{\mathrm{KT}}(\pi_1, \pi_2)$$

Em [16] e [23] podem ser encontrados maiores detalhes sobre testabilidade de propriedades de permutações. Em [21] discute-se testabilidade de propriedades de orientações de dígrafos, e em [20] estuda-se testabilidade de diversas classes de objetos matemáticos. No capítulo seguinte, iniciaremos a introdução de uma abstração do estudo de testabilidade de propriedades de estruturas discretas dentro dos propósitos enunciados no Capítulo 1 deste trabalho, e, no Capítulo 6, apresentaremos o Lema da Equivalência, com o qual pretendemos dar um passo rumo à unificação das diferentes teorias de testabilidade de diferentes classes de objetos matemáticos.

5 UMA FORMALIZAÇÃO DA TESTABILIDADE DE PROPRIEDADES

5.1 Uma Generalização dos Modelos de Testabilidade de Propriedades

O tipo de método que estudaremos requer uma definição precisa de estrutura discreta e de modelo de testabilidade. Para tal fim, adotaremos as definições apresentadas nas Seções 5.2 e 5.3 deste trabalho.

Deve-se fixar, para um modelo de testabilidade de uma estrutura discreta, uma métrica, um tipo de consulta para os testadores.

Uma mesma estrutura discreta pode ser usada na definição de mais de um modelo de testabilidade, como ocorre com o caso dos grafos. Vimos no Capítulo 3 que há pelo menos três modelos naturais de testabilidade de grafos: o modelo denso, o modelo de grau limitado e o modelo geral. Vimos também as diferentes métricas e maneiras de acesso dos testadores dos grafos de entrada, resultantes de representações distintas dos grafos. Este acesso é feito mediante os oráculos, que são capazes de fazer consultas ao grafo de entrada, captando informações sobre o grafo de entrada mesmo sem ler a entrada inteira. Assim, métrica e tipos de consulta de um oráculo associado a um testador são objetos matemáticos que caracterizam a testabilidade de propriedades de grafos.

Numa generalização destes três modelos de testabilidade para abranger modelos de testabilidade não apenas de grafos, mas também de outros tipos de estruturas discretas, consideraremos, também, mais um objeto: uma relação de equivalência sobre o universo da estrutura respectiva, para definir o que chamaremos de relação de *invariância* de modelo, conforme será descrito a seguir.

Qualquer relação de equivalência sobre as instâncias da estrutura pode ser escolhida como relação de invariância do modelo de testabilidade. Assim, um modelo de testabilidade ficará completamente definido, quando fixarmos uma estrutura discreta, uma métrica, um conjunto de tipos de consultas que serão permitidas pelos algoritmos de teste de propriedades (os testadores), uma relação de equivalência que definirá a invariância dos conjuntos de instâncias da estrutura discreta em questão e uma função tamanho, conforme a Seção 5.3 deste texto.

Dada uma classe de estruturas discretas, definimos como uma *propriedade* de instâncias de tal estrutura um conjunto de instâncias que seja fechado sob a relação de invariância. No caso dos grafos, esta relação de equivalência que chamamos de invariância é o *isomorfismo* de grafos; no caso de dígrafos, o *isomorfismo* de dígrafos, valendo esta relação, também, para os torneios, uma classe especial de dígrafos; no caso de outras estruturas discretas, poderemos escolher a relação de equivalência mais conveniente como relação de invariância.

5.2 Estruturas discretas

Antes de definirmos um modelo de testabilidade de uma estrutura discreta, convém fixar uma definição precisa de estrutura discreta, bem como uma notação padrão.

Uma estrutura discreta E é uma 5-upla $(W(E), C(E), R(E), F(E), \omega(E))$, onde $W(E)$ é o seu universo, $C(E)$ é o seu conjunto de constantes, $R(E)$ é a sua coleção de relações, $F(E)$ é o seu conjunto de operações e $\omega(E)$ é o seu conjunto de parâmetros, conforme segue. Dada uma estrutura discreta E, o seu *universo* é um conjunto enumerável $W(E)$, cujos elementos são as *instâncias* da estrutura discreta. Deste universo, selecionamos um subconjunto $C(E) \subset W(E)$ cujos elementos são as *constantes* da estrutura discreta.

Toda estrutura discreta E tem uma coleção de *relações* $R(E)$ a ela associadas. As relações de uma estrutura discreta são conjuntos contidos em B^n, para algum n natural não nulo, em que $B = W(E)$. Para $n = 1$, diremos que a relação é unária, para $n = 2$, diremos que ela é binária. Em geral, para $n = k$, chamaremos de k-ária a relação.

As *operações* de uma estrutura discreta são funções $f : B^n \supseteq A \longrightarrow B$, para algum n natural não nulo, em que B é o universo. Os *parâmetros* de uma estrutura discreta são funções $h : B^n \supseteq A \longrightarrow \mathbb{R}$, para algum n natural não nulo. Assim como as relações das estruturas discretas, as operações e parâmetros também podem ser unários(as), binários(as), ..., k-ários(as), etc...

Duas estruturas discretas são iguais, quando têm o mesmo universo, as mesmas constantes, relações, operações e os mesmos parâmetros.

Dado um universo B de uma estrutura discreta, constituído por um conjunto enumerável, seja ele finito ou infinito, fixa-se um conjunto de operações da estrutura, que são funções cujo domínio é um subconjunto de B^n e cujo contradomínio é B. Se o domínio for o conjunto B^n, diremos que tal operação é uma *operação total* da estrutura discreta, o que significa que esta operação está definida para todo elemento do produto cartesiano $B \times B \times \cdots \times B$ (n vezes). Por outro lado, se o domínio for um subconjunto próprio de B^n, diremos que se trata de uma *operação parcial* da estrutura, o que equivale a dizer que há n-uplas sobre o universo que não estão contidas no domínio desta operação. O mesmo vale para os parâmetros, os quais, sob o mesmo critério, podem ser *parâmetros totais* ou *parciais*. Vejamos alguns exemplos.

Suponhamos que o conjunto de instâncias da estrutura discreta em que estamos trabalhando seja o conjunto de grafos simples cujos vértices sejam números naturais. Podemos definir uma operação parcial desta estrutura tal que seu domínio seja o conjunto dos ciclos ímpares. Nesse sentido, a função

$f : A \longrightarrow B$, onde A é o conjunto dos ciclos ímpares, B é o conjunto dos grafos simples e f é a função que associa a cada elemento da entrada, que é um grafo, o seu grafo complementar, é uma função parcial. Um exemplo de parâmetro parcial é a métrica de um modelo de testabilidade de uma estrutura discreta, sobre o que trataremos com mais detalhe na Seção 5.3, em que o domínio é um subconjunto próprio de B^2, sendo B o universo, em que dois elementos serão comparáveis se estiverem definidos numa mesma parte de determinada partição do universo (como veremos). Numa estrutura discreta cujas instâncias são os números naturais, a paridade das instâncias (atributo de ser par ou ímpar) é um parâmetro total, pois todos os elementos do universo são valores que podem ser assumidos pela variável independente do parâmetro.

Um outro exemplo, bastante interessante, decorre da possibilidade de representação de permutações por uma determinada classe de dígrafos. Uma permutação é uma bijeção de um conjunto finito em si mesmo. Representemos cada elemento de certo conjunto finito C como um vértice de um dígrafo. Para definir o conjunto de arcos do dígrafo que representa a permutação $\sigma : C \longrightarrow C$, para cada u e cada v pertencentes a C tais que $\sigma(u) = v$, introduza-se o arco (u, v) no dígrafo que representa σ. No dígrafo assim composto, cada vértice terá grau de entrada igual a 1 e grau de saída igual a 1, e, para cada dígrafo desta classe (com graus de entrada e de saída iguais a 1), existe apenas uma permutação por ele representada. Isso permite que fixemos uma estrutura discreta de dígrafos que seja uma cópia de dada estrutura discreta de permutações.

Vejamos como definir uma estrutura discreta sobre grafos no conceito definido acima de estrutura discreta. Temos certa liberdade nesta tarefa, pois há infinitas formas de se fazer isso (por exemplo, basta incluirmos ou excluirmos uma relação, operação ou parâmetro, que passamos a tratar de um novo modelo). Definamos um exemplo de uma destas estruturas como segue abaixo:

Exemplo 5.1. Existe uma estrutura discreta E composta pelos seguintes objetos matemáticos:

Universo: $W = \{(V, A)\}$, em que $V \subset \mathbb{N}^*$, sendo que V é um conjunto finito, e sendo que $A \subseteq \{a | a \subseteq V \wedge |a| = 2\}$.

Constantes: As constantes desta estrutura discreta são as instâncias dos dois tipos a enumerados seguir.

Constante do tipo 1: $(\{1\},\emptyset)$, $(\{2\},\emptyset)$, $(\{3\},\emptyset)$, ...

Constante do tipo 2: $(\{1,2\},\{\{1,2\}\})$, $(\{1,3\},\{\{1,3\}\})$, $\ldots$ $(\{2,3\},\{\{2,3\}\})$, $\ldots$

Relações: São relações unárias de E os subconjuntos de W. São relações binárias de E as seguintes: $=$, $\neq$, $\simeq$, $\subseteq$, $\lhd$ (igualdade, diferença, isomorfismo, "...ser subgrafo de..." e "...ser subgrafo induzido de..."). São relações ternárias de E "...ser cópia de...em...", "...ser cópia induzida de ... em...". "...ser $\frac{1}{3}$-longe da propriedade de ser um grafo perfeito".

Operações: União de grafos, intersecção de grafos, junção de grafos, complementação de grafos, diferença de grafos, "inclusão de arestas", "remoção de vértices", "remoção de arestas", conforme definimos logo abaixo.

Parâmetros: abundância, distância relativa entre um grafo e a propriedade de ser um ciclo de comprimento ímpar, número cromático, número de clique, número de independência, grau médio, grau mínimo, tais como definidos logo abaixo.

A união de grafos e a intersecção de grafos, denotadas respectivamente por $\cup$ e $\cap$, são assim definidas: $G_1 \cup G_2 = (V(G_1) \cup V(G_2), A(G_1) \cup A(G_2))$, $G_1 \cap G_2 = (V(G_1) \cap V(G_2), A(G_1) \cap A(G_2))$. A complementação de G, denotada de G^c, é assim definida: $G^c = (V(G), K \backslash A(G))$, onde K é o grafo com o mesmo conjunto de vértices de G e com arestas para todos os pares de vértices. A diferença entre G_1 e G_2 consiste num grafo $G_3 = G_1 - G_2 = G_1 \cap G_2^c$. A junção de um grafo G_1 com o grafo G_2 é definida por $\left(G_1^c \cup G_2^c\right)^c$. Remoção de um vértice de G é, nesta estrutura, um abuso de linguagem para a operação de diferença entre G e uma constante do tipo 1. Outro abuso de linguagem no contexto desta estrutura é a expressão "inclusão de arestas" da definição da estrutura, como uma das suas operações. Trata-se, rigorosamente, da união de um grafo G com uma constante do tipo 2 cujos dois vértices estejam contidos em $V(G)$. A remoção de vértices define-se como de praxe. De todas estas definições, percebe-se que todos os grafos desta estrutura discreta podem ser construídos a partir das constantes e operações da estrutura. Uma das maneiras de se fazer isso é pela união de constantes do tipo 1 para todos os vértices do grafo seguida de uniões deste grafo sem arestas a constantes do tipo 2 correspondentes a todas as arestas do grafo. Dependendo do grafo, poderá haver outros caminhos de construí-lo a partir das constantes e operações da estrutura.

Definamos os parâmetros desta estrutura discreta. Se G_1 e G_2 são instâncias desta estrutura, diremos que a *abundância* de G_1 em G_2 é igual a δ, quando δ é o ínfimo dos números reais tais que existem pelo menos $\delta|G_2|^{|G_1|}$ cópias induzidas de G_1 em G_2. A importância deste parâmetro binário total desta estrutura discreta está em permitir a representação em linguagem L do Lema da Remoção para grafos, o Lema 3.1. A distância relativa entre um grafo e uma dada propriedade está definida na Seção 3.3, nos três modelos de testabilidade de grafos, sendo que, sempre que não especificarmos um deles, ficará implícito que nos referimos ao modelo denso. Os outros parâmetros definem-se como de praxe.

Quando passamos a analisar os torneios, dentre as diversas maneiras de definirmos uma estrutura discreta cuja instâncias sejam torneios que seja análoga ao modelo denso para grafos, temos a seguinte fixação dos seus objetos, conforme o Exemplo 5.2:

Exemplo 5.2. Existe uma estrutura discreta E' composta pelos seguintes objetos matemáticos:

> **Universo**: $W = \{(V, vecA)\}$, em que $V \subset \mathbb{N}^*$, sendo que V é um conjunto finito, e $\vec{A} \subseteq \{(j,k) | j \in V \wedge k \in V \wedge j \neq k\}$ tal que, para todo $j \in V$ e todo $k \in V \setminus \{j\}$, $(j,k) \in \vec{A}$ se, e somente se, $(k,j) \notin A$, $\forall j \forall k$ em V.
>
> **Constantes**: As constantes desta estrutura discreta são as instâncias dos três tipos a enumerados seguir.
>
> Constante do tipo 1 (torneios sem arcos): $(\{1\},\emptyset)$, $(\{2\},\emptyset)$, $(\{3\},\emptyset)$, ...
> Constante do tipo 2 (torneios com dois vértices e em que o arco é "crescente", ou seja, são pares-ordenados (u,v), em que $u < v$):
>
> $$(\{1,2\},\{(1,2)\}),(\{1,3\},\{(1,3)\}), \dots (\{2,3\},\{(2,3)\}),(\{2,4\},\{(2,4)\}), \dots$$
>
> Constante do tipo 3 (torneios com dois vértices e em que o arco é "decrescente", ou seja, são pares-ordenados (u,v), em que $u > v$)
>
> $$(\{1,2\},\{(2,1)\}),(\{1,3\},\{(3,1)\}), \dots (\{2,3\},\{(3,2)\}),(\{2,4\},\{(4,2)\}), \dots$$
>
> **Relações**: São relações unárias de E' os conjuntos de torneios. São relações binárias de E' as seguintes: $=$, $\neq$, $\simeq$, $\subseteq$. São relações ternárias de E' as seguintes: "ser subtorneio de...", "ser cópia de... em...".
>
> **Operações parciais e totais**: junção, inversão de todos os arcos, inversão de um arco específico, remoção de vértices.
>
> **Parâmetros**: abundância, inevitabilidade, distância relativa entre um torneio e a propriedade de ser acíclico (um torneio sem ciclos dirigidos), tais como definidos abaixo.

Como no caso da estrutura de grafos fixada um pouco antes, os torneios da nossa estrutura discreta são os definidos no Capítulo 2 com a restrição de que seus vértices pertencem a $\mathbb{N}^*$. Há três tipos de constantes: as constantes do tipo 1 são os torneios com um vértice; as constantes do tipo 2 são torneios com dois vértices cujo arco parte do vértice menor para o vértice maior, que são arcos que chamaremos de *arcos crescentes*; as constantes do tipo 3 também são torneios com dois vértices, mas, diferentemente das de tipo 2, seus arcos partem do vértice maior para o vértice menor, e este tipo de arco será chamado de *arco decrescente*. Conforme a definição, entre dois vértices distintos há um arco crescente ou um arco decrescente, necessariamente. A operação junção entre torneios, denotada por *, é definida da seguinte forma: se $V(T_1)$ e $V(T_2)$ são disjuntos, então $T_2 = T_1 * T_2$ tem os mesmos vértices e os mesmos arcos de T_1 e T_2 e mais os arcos orientados de cada vértice de T_1 para cada vértice de T_2. A inversão de um arco (u,v) é a sua substituição por (v,u). É fácil ver que, por meio das constantes e operações desta estrutura discreta, é possível construir todas as suas instâncias. Podemos, por exemplo, aplicar a operação junção a todos os vértices que queremos reunir num torneio e, posteriormente, inverter a orientação de uma parte dos arcos, conforme a conveniência. Assim, não há torneio que não possa ser construído por este caminho.

Uma *transversal* de um torneio T_2 em um torneio T_1 é um subconjunto $S \subseteq \vec{A}(T_1)$ tal que toda cópia de T_2 em T_1 tem arcos contidos em S. Dizemos que um torneio T_2 é *ε-inevitável* em T_1, quando toda transversal de T_2 em T_1 tem tamanho pelo menos εn^2, onde n é o tamanho de T_1. E um torneio T_2 é *δ-abundante* em um torneio T_1, se T_1 contém pelo menos δn^2 cópias de T_2, onde n é o tamanho de T_1. Definiremos os parâmetros da *inevitabilidade* de um torneio T_2 em um torneio T_1 e da *abundância* de um torneio T_2 em um torneio T_1, respectivamente, como o maior ε tal que T_2 é ε-inevitável em T_1 e o maior δ tal que T_2 é δ-abundante em T_1. Para dada propriedade, podemos fixar como parâmetro a distância relativa de um torneio a esta propriedade, tal como definimos a distância relativa no Capítulo 4.

5.3 Modelos de Testabilidade de uma Estrutura Discreta

As relações unárias e parâmetros unários de uma estrutura discreta podem ser invariantes ou não invariantes. As relações unárias invariantes serão chamadas de *propriedades* e correspondem às relações unárias da estrutura que são fechadas sob a relação de invariância (sobre a qual tratamos brevemente ainda na nesta Seção 5.2 e na Seção 5.3). Um parâmetro unário de uma estrutura será chamado de parâmetro invariante sempre que instâncias equivalentes sob a relação de invariância tiverem a mesma imagem sobre o parâmetro em questão. Por exemplo: dois grafos isomorfos têm o mesmo número cromático. O número cromático é um parâmetro invariante de grafos por esse motivo.

Conforme adiantamos na Seção 5.1, fixada uma estrutura discreta, uma métrica, um conjunto de tipos de consultas dos oráculos dos testadores, uma função tamanho e uma relação de invariância, definimos um modelo de testabilidade. Nos Capítulos 3 e 4 tratamos de alguns modelos de testabilidade que existem na literatura. Na Seção 5.2 formalizamos o conceito de estrutura discreta para que pudéssemos generalizar a noção já esboçada até então do que seja um modelo de testabilidade. Nossa pretensão nesta Seção 5.3 é ampliar aqueles conceitos para que possamos definir modelos de testabilidade associados a estruturas discretas dadas. E faremos isso de maneira tal que uma estrutura discreta admita mais de um modelo de testabilidade, como se dá com os grafos, os quais têm os três modelos descritos na Seção 3.3. Evidentemente, devemos definir o conceito de modelo de testabilidade de maneira tal que todos os modelos de testabilidade vistos nos Capítulos 3 e 4 se enquadrem.

Definiremos como um *modelo de testabilidade* uma 5-upla M tal que $M = (E, g, K, Z, t)$ em que:

1- E é uma estrutura discreta, composta pelo seu universo W e suas constantes, relações, operações e parâmetros.

2- g é uma métrica, que é uma função $g : \bigcup_{n=1}^{\infty} W_n \times W_n \longrightarrow \mathbb{R}_+$, tal que $\forall x \forall y \forall z$, $g(x,x) = 0$, $x \neq y \Rightarrow g(x,y) \neq 0$, $g(x,y) = g(y,x)$ e $g(x,z) \leq g(x,y) + g(y,z)$, onde W é o universo da estrutura discreta e $\{W_1, W_2, \dots\}$ é uma partição de W.

3- K é um conjunto de tipos de consulta do oráculo de um testador (funções que chamaremos de *funções-consulta* do modelo), isto é, $K = \{k_1, k_2, \dots\}$, onde cada k_j é uma função $k_j : W \longrightarrow \mathbb{N}$, tal que a sequência $(k_1(w), k_2(w), \dots)$ determina completamente o elemento w do universo W. Cada função k_j é dita uma *função-consulta* do modelo M.

4- Z é uma relação de *invariância*, que deverá ser uma relação de equivalência em W.

5- $t : W \rightarrow \mathbb{N}$ é uma função que associa um número natural a cada instância da estrutura discreta do modelo, o que definiremos como o seu *tamanho*.

Como se vê no item 2 acima, a métrica de um modelo de testabilidade não está necessariamente definida para todos os pares de instâncias. Ela é fixada após a definição de uma partição do universo da estrutura discreta, devendo estar definida apenas para pares de instâncias contidas na mesma parte da partição. No caso do modelo denso, do modelo de grau limitado e do modelo geral abordados na Seção 3.3, as partes da partição da métrica são os conjuntos de todos os grafos com o mesmo conjunto de vértices (lembremo-nos de que tais métricas somente estão definidas para grafos com o mesmo conjunto de vértices).

Um modelo de testabilidade também define as propriedades da respectiva estrutura discreta: um subconjunto do universo é uma propriedade, se for fechado sob a relação de invariância. No caso dos grafos, a relação de invariância é a relação de isomorfismo.

Uma vez definido um modelo de testabilidade, ficam determinados os *testadores* do modelo como sendo os algoritmos que respeitam o seguinte. Um *testador* para uma propriedade P de instâncias do modelo M é um algoritmo randômico associado a um parâmetro de tamanho da representação da instância de entrada n, a um parâmetro de distância ε e a um oráculo com acesso à representação de uma instância. Ao oráculo é dada a faculdade de fazer consultas à representação da instância de entrada, de maneira que o algoritmo possa acionar o oráculo em certos momentos e fazer escolhas a partir das respostas às consultas já realizadas pelo oráculo ou de escolhas feitas a partir dos dados resultantes de consultas anteriores ao oráculo. Os tipos de consultas que podem ser feitas pelo oráculo correspondem às funções-consulta do modelo de testabilidade em questão (item 3 da definição formal do modelo). No final da computação de um testador sobre uma instância da estrutura discreta, o testador aceita-a ou rejeita-a, e vale o seguinte:

1- Se a instância x satisfaz a propriedade P, então a probabilidade de o testador T aceitar x é maior do que $\frac{2}{3}$.

2- Se a instância x está a uma distância maior ou igual a ε da propriedade P segundo a métrica do modelo de testabilidade em questão (item 2 da definição formal do modelo), então a probabilidade de o testador T rejeitar x é maior do que $\frac{2}{3}$.

Se a probabilidade citada no item 1 acima for igual a 1, dizemos que a propriedade em questão é testável *com erro unilateral*. Caso contrário, dizemos que ela é testável *com erro bilateral*. Em qualquer caso, a *complexidade de consultas* (o número de vezes que o oráculo é acionado) é, necessariamente, sublinear em relação a n, mas pode variar em função de ε.

Formalmente, um testador associado à propriedade P é um algoritmo randômico T que executa uma sequência finita de passos $(P_1, P_2, ...P_k)$ em que, para cada j, P_j é a escolha de uma função-consulta a ser acionada pelo oráculo, o acionamento do oráculo ou a saída do algoritmo (a aceitação ou rejeição da instância de entrada respeitando o disposto nos dois itens do rol acima). Para uma função-consulta w e para a instância de entrada c, o oráculo fornece $w(c)$, sendo que, se w_j é a função-consulta escolhida no passo P_j, $\forall j < k$, $w_m(c)$ depende de $(w_1, w_1(c)), (w_2, w_2(c)), \ldots, (w_{m-1}, w_{m-1}(c))$.

O testador será tão mais eficiente, quanto menor for o número de consultas que ele fizer à instância de entrada por meio do seu oráculo. Chamaremos de *complexidade de consultas* de um testador sobre uma entrada o número de consultas que o seu oráculo faz ao longo do processamento da representação da instância de entrada. Quando falarmos na complexidade de consultas, não nos referimos à execução do testador sobre uma instância específica, e sim ao máximo do número de consultas realizados pelo testador a todas as possíveis instâncias de entrada com mesmo tamanho e para o mesmo parâmetro de distância.

Para tornar mais concreta a noção de modelo de testabilidade, vamos descrever uma possível definição de cada um dos três modelos descritos na Seção 3.3 deste trabalho restringindo os vértices ao conjunto dos números naturais, lembrando, porém, que há mais de uma maneira de se fazer isso.

Exemplo 5.3. Definição do modelo denso:

Estrutura discreta do modelo denso: Estrutura discreta de grafos simples descrita na Seção 5.2 ou outra com o mesmo universo.

Métrica do modelo denso: $dist_{Rel}(G,G') = \frac{|\{\{u,v\}\in V\times V(\{u,v\})\neq g_{G'}(\{u,v\})\}|}{n^2}$, onde $g_G : [V]^2 \longrightarrow \{0,1\}$ é a função de adjacência, a qual é tal que $g_G(\{u,v\}) = 1$, se e somente se $\{u,v\} \in A(G)$, sendo que, para que esta métrica esteja definida, é necessário que G e G' tenham o mesmo conjunto de vértices, o qual tem cardinalidade n.

Funções-consulta do modelo denso: $K = \{k_{1,1}, k_{2,1}, k_{1,2}, k_{2,2}, \dots\}$, onde cada $k_{u,v} : W \longrightarrow \mathbb{N}$, para vértices u e v dados, é uma função $g_G(\{u,v\})$ que associa G a um elemento do conjunto $\{0;1\}$, se $u \in V(G) \wedge v \in V(G)$.

Relação de invariância do modelo denso: Isomorfismo de grafos.

Função tamanho do modelo denso: função que associa a cada grafo o seu número de vértices.

Exemplo 5.4. Definição do modelo de grau limitado:

Estrutura discreta do modelo de grau limitado: Estrutura discreta de grafos simples descrita na Seção 5.2 ou outra com o mesmo universo.

Métrica do modelo de grau limitado:

$$dist_{Rel}(G,G') = \frac{|\{(u,i) \in V \times [d] : g_G(u,i) \neq g_{G'}(u,i)\}|}{nd}$$

onde $g_G : V \times [d] \longrightarrow V \cup \{\#\}$ é a função de incidência, a qual é tal que $g_G(u,i) = v$, se e somente se v é o i-ésimo vizinho de u e $g_G(u,i) = \#$, caso u tenha menos do que i vizinhos, sendo que, para que esta métrica esteja definida, é necessário que G e G' tenham o mesmo conjunto de vértices, o qual tem cardinalidade n.

Funções-consulta de do modelo de grau limitado: Conjunto K, que obedece ao seguinte: $K = \{k_{1,1}, k_{2,1}, k_{1,2}, k_{2,2}, \dots\}$, onde cada $k_{u,i} : W \longrightarrow \mathbb{N}$, para um vértice u e um número natural d dados, é uma função $g_G(\{u,i\})$ que associa G a um elemento do conjunto $\{0;1\}$, se $u \in V(G) \wedge i \in [d]$.

Relação de invariância do modelo de grau limitado: Isomorfismo de grafos.

Função tamanho do modelo de grau limitado: função que associa a cada grafo o seu número de vértices.

Também podemos definir formalmente o modelo geral de testabilidade de propriedades de grafos, a que nos referimos na Seção 5.2.

5.4 Sistematização do Uso de Bijeções para Relacionar Modelos de Testabilidade

De agora em diante, sempre que mencionarmos o termo *objeto* no estudo de certa estrutura discreta, estaremos nos referindo às instâncias, operações, relações e parâmetros da referida estrutura discreta. E chamaremos de objetos de um modelo de testabilidade os objetos da estrutura discreta subjacente e também a sua métrica, as suas funções-consulta, a relação de invariância e a função tamanho.

Tendo em vista o nosso objetivo de encontrar meios e ferramentas capazes de exportar o conhecimento que temos no estudo de testabilidade de determinado tipo de estrutura discreta para o estudo de testabilidade associada a outro tipo de estrutura discreta, dentre as diversas publicações existentes na literatura em matéria de testabilidade, buscaremos fazer um estudo de certas bijeções que nos permitem associar cada objeto de um dos modelos de testabilidade a um, e apenas um, objeto de outro modelo de testabilidade. As leis de correspondência entre os objetos de duas estruturas discretas que nos interessam satisfazem a certas condições que serão a seguir definidas, e serão, adiante, chamadas de "transformações regulares entre estruturas discretas" ou, conforme o caso, “transformações regulares entre modelos de testabilidade de estruturas discretas” e são o principal foco do estudo que faremos.

Suponhamos que desejemos definir uma regra de correspondência que associe univocamente cada objeto do modelo M_1, ou seja, as instâncias, relações, operações e parâmetros da estrutura que o integra, a um objeto do modelo M_2 de uma maneira que nos permita estudar M_2 a partir do que sabemos sobre os objetos de M_1. Em outras palavras, nossa intenção é escolher uma regra de correspondência entre ambos os modelos que, de alguma maneira, admita a existência de algoritmos que, por meio do processamento simbólico de teoremas situados no âmbito do estudo de M_1, forneçam-nos teoremas acerca do modelo de testabilidade M_2 guardando algum tipo de informação do modelo original (M_1).

Nem todas as transformações entre estruturas discretas ou modelos de testabilidade interessam ao nosso estudo. O objetivo que perseguimos requer que nos ocupemos de transformações que satisfaçam a certos requisitos, para que estas nos tenham utilidade prática na dedução de teoremas. Mais precisamente, nossa principal intenção é encontrar, em cada caso, alguma regra de associação dos objetos de duas estruturas discretas, E_1 e E_2, que respeite as seguintes restrições:

1- que associe cada instância de E_1 a uma instância de E_2.

2- que associe cada n-upla de instâncias de E_1 a uma n-upla de instâncias de E_2, para algum n natural não nulo;

3- que associe cada relação n-ária de E_1 a uma relação n-ária de E_2;

4- que associe cada operação n-ária de E_1 a uma operação n-ária de E_2;

5- que associe cada parâmetro n-ário de E_1 a uma parâmetro n-ário de E_2.

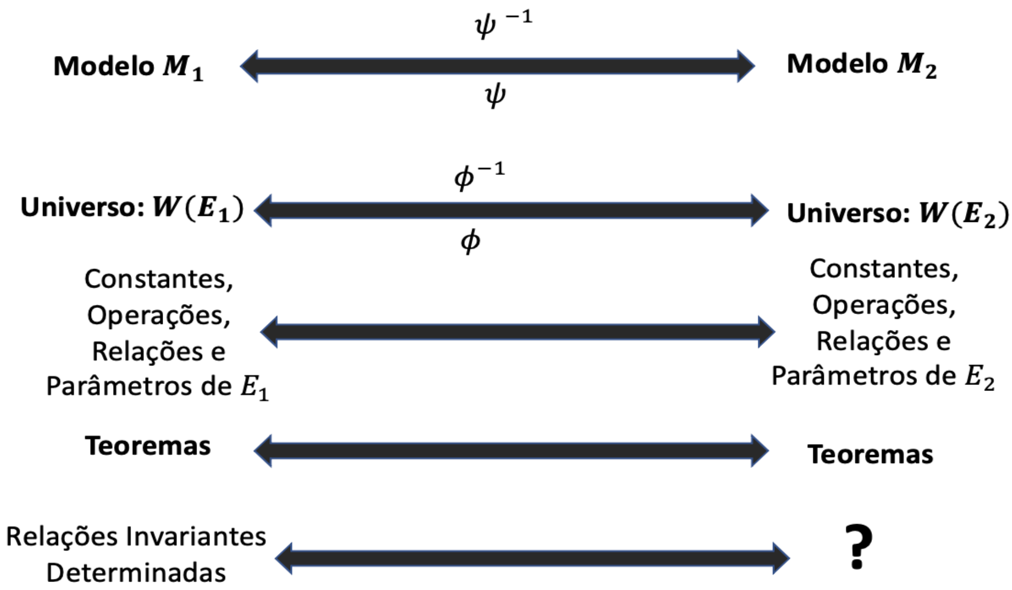

Figura 5.1: Representação de uma bijeção dos componentes de dois modelos de estruturas discretas com o fim de estabelecer uma correspondência entre teoremas de ambos os modelos.

Todo objeto matemático que satisfaz essas condições é uma transformação entre estruturas discretas. E, dentre estas, as transformações regulares são dotadas de propriedades muito úteis ao processo dedutivo, conforme disporemos no Capítulo 6.

Na Figura 5.1, esquematizamos o método que aqui se propõe a estudar. À esquerda, na Figura 5.1, temos menções a objetos do modelo M_1, como seu universo, suas constantes, suas operações, etc... Já à direita, são mencionados os objetos do modelo M_2 associados aos de M_1 pela função ψ. O universo da estrutura discreta de M_1 (a estrutura E_1) é associado ao universo da estrutura discreta de M_2 (a estrutura E_2), mediante a função $\phi_W : W(E_1) \longrightarrow W(E_2)$.

Formalmente, definiremos as transformações regulares entre estruturas discretas da seguinte forma, adotando a notação já fixada para os cinco componentes de uma estrutura discreta dada (vide a Seção 5.2).

Definição 5.1. *Chamaremos de* transformação regular entre as estruturas discretas *E_1 e E_2 uma 4-upla $(\psi_W, \psi_R, \psi_\otimes, \psi_\omega)$ em que $\psi_W : W(E_1) \longrightarrow W(E_2)$, $\psi_R : R(E_1) \longrightarrow R(E_2)$, $\psi_\otimes : F(E_1) \longrightarrow F(E_2)$, $\psi_\omega : \omega(E_1) \longrightarrow \omega(E_2)$ são funções bijetoras que satisfazem o seguinte. Sejam $S(E_1)$ o conjunto das sequências finitas de E_1, $S(E_2)$ o conjunto das sequências finitas de E_2 e $\psi_s : S(E_1) \longrightarrow S(E_2)$ a função tal que $\psi_s(x_1, x_2, \ldots, x_n) = (\psi_W(x_1), \psi_W(x_2), \ldots \psi_W(x_n))$. Então:*

1- $s_j \in R_k \Leftrightarrow \psi_s\left(s_j\right) \in \psi_R(R_k)$ para cada sequência s_j de E_1 e cada relação R_k de E_1.

2- $\otimes_i\left(s_j\right) = x_k \Leftrightarrow \psi_\otimes(\otimes_i)\left(\psi_s\left(s_j\right)\right) = \psi_W(x_k)$ para cada sequência s_j de E_1 e cada operação $\otimes_i$ de E_1

3- $\omega_i\left(s_j\right) = r \Leftrightarrow \psi_\omega(\omega_i)\left(\psi_s\left(s_j\right)\right) = r$, para cada sequência s_j de instâncias de E_1 e cada parâmetro ω_i de E_1.

Usaremos a seguinte notação. Se c é um elemento do domínio de uma das quatro funções que compõem uma transformação regular $\psi = (\psi_W, \psi_R, \psi_\otimes, \psi_\omega)$, a notação $\psi(c)$ representará a imagem correspondente. Da mesma forma, se c é uma sequência finita de elementos do domínio de ψ_W, $\psi(c) = \psi_s(c)$.

Por ser uma 4-upla de bijeções, a transformação ψ tem inversa, ψ^{-1}, a qual leva instâncias de E_2 a instâncias de E_1 e operações, relações e parâmetros de E_2 a operações, relações e parâmetros de E_1, respectivamente. Tudo isso está sinteticamente representado na Figura 5.1 pelas três primeiras setas. Queremos estender a função ψ de maneira que a função resultante se aplique a teoremas e leve teoremas da teoria subjacente ao modelo de testabilidade M_1 para teoremas da teoria correspondente a M_2, se possível, preservando a relação de invariância, já que é em função desta relação que as propriedades são caracterizadas. Com isso, uma vez definida a regra de associação entre objetos de M_1 e objetos de M_2, o que queremos saber é, dado um teorema sobre termos do modelo de testabilidade M_1 qual é o teorema correspondente que podemos encontrar no modelo de testabilidade M_2. O nosso primeiro desafio consiste em definir precisamente uma correspondência natural entre M_1 e M_2, em função de ψ, e isso será feito na Seção 6.1, ao enunciarmos um importante resultado no âmbito da matéria deste trabalho: o Lema da Equivalência.

Sempre que existir uma transformação regular entre duas estruturas discretas, existe uma bijeção entre os universos das estruturas discretas que determina uma transformação regular entre ambas as estruturas (os modelos podem ser considerados um caso especial de estrutura). Dada uma transformação regular ψ (cujo primeiro componente seja a função ψ_W), ficará definida a função ψ_s e, em função desta última, as bijeções ψ_R, $\psi_\otimes$ e ψ_ω ficam determinadas. Da mesma forma, toda transformação regular entre duas estruturas discretas é determinada por alguma bijeção cujo domínio seja um dos dois universos e o contradomínio seja o outro universo (o universo da outra estrutura discreta). Além disso, uma estrutura discreta e uma bijeção entre seu universo e dado conjunto determinam univocamente outra estrutura discreta tal que existe uma transformação regular entre ambas, o que se dá da seguinte forma: Sejam E_1 uma estrutura discreta, B um conjunto e $\phi : W(E_1) \to B$ uma bijeção. Então existe uma única estrutura discreta E_2 tal que existe uma transformação regular ψ entre E_1 e E_2. Neste caso, $W_2 = B$, $\psi_W = \phi$ e existem apenas um ψ_R, um $\psi_\otimes$ e um ψ_ω tais que $(\psi_W, \psi_R, \psi_\otimes, \psi_\omega)$ é uma transformação regular.

Em geral não é fácil definir transformações regulares. Na Seção 6.3 apresentaremos um exemplo de transformação regular (Exemplo 6.2).

5.5 Uma classe de linguagens artificiais para modelos de testabilidade

Existe uma abordagem para estudar propriedades de estruturas combinatórias por meio de limites de objetos combinatórios, à luz de certa definição de convergência de sequências de instâncias (vide [10]). Coregliano e Razborov [13] estudaram limites de objetos combinatórios por uma abordagem lógica, tendo definido linguagens de primeira ordem para representar objetos que, no presente trabalho, correspondem a estruturas discretas. Este tipo de abordagem é útil na generalização de uma coleção de teorias, quando definimos adequadamente uma classe de linguagens artificiais capaz de modelar cada teoria em questão. Por este motivo, aqui também formularemos uma classe de linguagens artificiais, conforme segue.

Definiremos uma linguagem para os propósitos do Capítulo 6, em que nos valeremos de procedimentos lógicos para a exposição genérica de dois métodos dedutivos. Trata-se da linguagem $L(M)$, a qual será usada para expressar sentenças sobre o modelo de testabilidade M, ou simplesmente linguagem L (quando o modelo M estiver subentendido no contexto). As expressões bem-formadas desta linguagem são sequências de símbolos de L que representem instâncias da estrutura discreta, relações, operações ou parâmetros da estrutura ou distâncias entre instâncias (conforme a métrica do modelo). Entre os símbolos de L, há variáveis, constantes, conectivos lógicos, quantificadores e parênteses. A seguir definiremos a linguagem, o que será feito de maneira similar à descrição de sistemas formais quantificados em manuais de Lógica Matemática, como [22] e [24], porém de maneira adaptada aos nossos propósitos.

Na linguagem L, as relações, operações e parâmetros serão representados com os seguintes símbolos. As relações da estrutura serão denotadas pelos símbolos $\{R_1, R_2 \ldots\}$, as operações serão denotadas pelos símbolos $\{\otimes_1, \otimes_2, \ldots\}$, e os parâmetros serão denotados pelos símbolos $\{\omega_1, \omega_2, \ldots\}$.

A linguagem L terá uma coleção de símbolos $\{C_1; C_2; C_3 \ldots\}$ que representarão conjuntos determinados. Cada C_j representa um subconjunto de $\mathbb{R}$ ou um subconjunto do universo da estrutura discreta E associada ao modelo de testabilidade M. Da mesma forma, a linguagem L terá um conjunto de símbolos $\{u_1; u_2; \ldots\}$ que representarão constantes reais determinadas, incluindo os elementos 0 e 1. Por fim, esta linguagem deverá ter um conjunto de símbolos $\{s_1; s_2; \ldots\}$ para denotar as constantes da estrutura discreta em questão.

Os símbolos $u_1; u_2; \ldots$ e $s_1; s_2; \ldots$ são constantes da linguagem L, as do primeiro tipo são constantes numéricas e as do segundo tipo são instâncias da estrutura discreta. Mas L é uma linguagem com quantificadores e, assim, requer a fixação de variáveis, as quais poderão representar elementos dos dois tipos: números reais e instâncias da estrutura. As variáveis numéricas serão $r_1; r_2; \ldots$ e as variáveis que representarão instâncias da estrutura discreta serão $x_1; x_2; \ldots$

Definimos recursivamente as expressões de L que representam conjuntos da seguinte forma:

1- Cada um dos símbolos C_j, em que $j = 1, 2, 3, \ldots$, é uma expressão que representa um conjunto.

2- Se α e β são expressões que representam conjuntos, então $\alpha \cup \beta$, $\alpha \cap \beta$ e $\alpha \backslash \beta$ também são expressões que representam conjuntos.

Definimos (recursivamente) da seguinte maneira as expressões de L que representam instâncias da estrutura:

1- As constantes $s_1; s_2; \ldots$ são expressões que representam instâncias da estrutura.

2- As variáveis $x_1; x_2; \ldots$ são expressões que representam instâncias da estrutura.

3- Se $\otimes$ é uma operação n-ária da estrutura discreta e α_1, α_2, ..., α_n são expressões que representam instâncias da estrutura discreta em questão, então $\otimes(\alpha_1, \alpha_2, \ldots, \alpha_n)$ é uma expressão que representa uma instância da estrutura.

Definimos as expressões numéricas de L seguinte forma:

1- As constantes numéricas e variáveis numéricas são expressões numéricas.

2- Se ω é um parâmetro n-ário da estrutura discreta e α_1, α_2, ..., α_n são expressões que representam instâncias da estrutura discreta em questão, então $\omega(\alpha_1, \alpha_2, \ldots, \alpha_n)$ é uma expressão numérica.

As *fórmulas atômicas* de L da consistem nas seguintes expressões:

1- $\alpha = \beta$ e $\alpha \neq \beta$, onde α e β são expressões que representam conjuntos, números ou n-uplas de instâncias (para algum n natural não nulo).

2- $\alpha \in \beta$ e $\alpha \notin \beta$, onde α é uma expressão de L que representa um conjunto, número ou n-upla de instâncias (para algum n natural não nulo) e β é uma expressão que representa um conjunto.

3- $\alpha \subseteq \beta$ e $\alpha \subsetneq \beta$, onde α e β são expressões que representam conjuntos.

4- $\alpha < \beta$, $\alpha > \beta$, $\alpha \leq \beta$ e $\alpha \geq \beta$, onde α e β são expressões numéricas.

5- $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \gamma$ e $(\alpha_1, \alpha_2, \ldots, \alpha_n) \notin \gamma$, onde γ é uma relação n-ária da estrutura discreta e α_1, α_2, ..., α_n são expressões que representam instâncias da estrutura.

Até agora descrevemos apenas as expressões de L que representem proposições envolvendo os objetos de uma estrutura discreta, mas não definimos, ainda, regras de formação de expressões de L que representem os demais objetos matemáticos que definem os modelos de testabilidade: a métrica do modelo de testabilidade, as funções-consulta e a relação de invariância. É disso que nos ocuparemos a partir do parágrafo seguinte.

O símbolo $\equiv$ denotará a relação de invariância do modelo de testabilidade na linguagem L. Este símbolo será usado em notação infixa para representar a afirmação de que duas instâncias da estrutura discreta são equivalentes segundo a relação de invariância do modelo. Da mesma forma, o símbolo $\not\equiv$, que também será usado em notação infixa, será empregado para representar a negação de que duas instâncias são equivalentes sob a relação de invariância.

Agora estamos aptos a incluir na linguagem L expressões envolvendo objetos dos modelos de testabilidade, já que as expressões já descritas envolviam apenas objetos das estruturas discretas. Neste sentido, adicionalmente às classes de expressões já descritas, também são expressões da linguagem L as seguintes:

1- Se α e β são expressões que representam instâncias da estrutura discreta, então $\alpha \equiv \beta$ é uma fórmula atômica.

2- Se α e β são expressões que representam instâncias da estrutura discreta, então $\alpha \not\equiv \beta$ é uma fórmula atômica.

3- Se α é uma função-consulta do modelo de testabilidade e β é uma instância da estrutura discreta, então $\alpha(\beta)$ é uma expressão numérica.

4- Se δ é a métrica do modelo de testabilidade, β é uma instância da estrutura discreta e γ é uma propriedade ou instância da estrutura, então $\delta(\beta, \gamma)$ é uma expressão numérica.

5- Se t é a função tamanho do modelo de testabilidade e α é uma instância da estrutura discreta, então $t(\alpha)$ é uma expressão numérica.

Para concluir, as fórmulas da linguagem L são definidas recursivamente, mediante as regras abaixo:

1- As fórmulas atômicas de L são fórmulas.

2- Se α e β são fórmulas, então $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \implies \beta$ e $\alpha \iff \beta$ são fórmulas.

3- Se α é uma variável, β é um conjunto e γ é uma fórmula, então $(\forall \alpha \in \beta)(\gamma)$ e $(\exists \alpha \in \beta)(\gamma)$ são fórmulas.

Alguns breves comentários sobre aspectos semânticos destas expressões são necessários. Os significados atribuídos às fórmulas caracterizadas no item 2 do rol acima são os habituais, os quais consistem nos seguintes, respectivamente: α e β; α ou β; se α, então β; se e somente se α, então β. O significado que será atribuído a uma fórmula do tipo $(\forall \alpha \in \beta)(\gamma)$ é: Para todo α pertencente ao conjunto β, tem-se que γ é verdadeira. Afirmações análogas podem ser feitas quanto ao quantificador existencial ($\exists$), ao qual será dada a interpretação de praxe.

Se cada variável de uma fórmula de L está ligada a um, e apenas um, quantificador, então tal fórmula é uma *sentença* de L.

Um exemplo de sentença da linguagem L sobre a estrutura discreta definida no Exemplo 5.1, em que C_1 é o universo dos grafos, é o seguinte.

$$(\exists x_1 \in C_1)((\forall x_2 \in C_1)(x_2 \equiv x_1))$$

A esta sentença, que é falsa, é atribuído o significado correspondente a afirmar que existe um grafo isomorfo a todos os grafos. Já a sentença seguinte significa que todo grafo tem um isomorfo, e é verdadeira:

$$(\forall x_1 \in C_1)((\exists x_2 \in C_1)(x_2 \equiv x_1))$$

6 O LEMA DA EQUIVALÊNCIA

Neste capítulo, nosso objetivo é definir e provar um teorema que chamaremos de Lema da Equivalência, o qual consiste no principal instrumento proposto neste trabalho como método para unificar teorias de testabilidade de propriedades discretas. Aqui, na Seção 6.1, enunciaremos o Lema da Equivalência logo após definirmos os conceitos indispensáveis para tal. Posteriormente, no Capítulo 6.2, exporemos uma demonstração do lema e, finalmente, na Seção 6.3, trataremos da aplicabilidade do Lema da Equivalência para o fim de derivarmos teoremas em testabilidade de certas estruturas discretas a partir de conhecimento pré-existente em testabilidade de outras estruturas discretas.

6.1 Enunciado do Lema da Equivalência

Começaremos pelas seguintes definições.

Definição 6.1. *Dizemos que uma transformação regular ψ entre as estruturas E_1 e E_2* preserva a relação de invariância, *sempre que é válido que: uma relação unária ou parâmetro unário α de E_1 é invariante se, e somente se, sua imagem $\psi(\alpha)$ também é invariante (em E_2).*

Definição 6.2. *Sejam E_1 e E_2 duas estruturas discretas. Diremos que E_1 e E_2 são duas* estruturas discretas equivalentes *se, e somente se, existe uma transformação regular ψ entre as estruturas discretas E_1 e E_2 que preserva a relação de invariância. Se ψ é uma transformação regular com esta propriedade, diremos que o par ordenado (E_1, E_2) é um* par equivalente de estruturas discretas com respeito a *ψ.*

Definição 6.3. *Sejam M o modelo de testabilidade associado à estrutura discreta E e μ uma fórmula da linguagem L. Diremos que μ* é uma fórmula de *M (ou de E) na linguagem L, se as constantes e variáveis de μ, na linguagem L, que denotam instâncias, relações, operações e parâmetros nomearem, respectivamente, instâncias, relações,* *operações e parâmetros de E. Se μ for uma sentença, diremos que μ* é uma sentença de *E (ou de M).*

Definição 6.4. *Dada uma 4-upla $\psi = (\psi_W, \psi_R, \psi_\otimes, \psi_\omega)$ em que $\psi_W : W(E_1) \longrightarrow W(E_2)$, $\psi_R : R(E_1) \longrightarrow R(E_2)$, $\psi_\otimes : F(E_1) \longrightarrow F(E_2)$, $\psi_\omega : \omega(E_1) \longrightarrow \omega(E_2)$ são funções bijetoras, dizemos que uma sentença μ_2 de $L(M2)$ é formada a partir da sentença μ_1 da linguagem $L(M1)$ por substituição uniforme segundo ψ, se o termo que denota cada objeto c de E_1 é substituído em μ_1 pelo termo que denota $\psi(c)$, gerando μ_2. Neste caso, diremos também que μ_1 gera μ_2.*

Note-se que, na Definição 6.4, empregamos $\psi(c)$ para denotar $\psi_W(c)$, caso $c \in W(E_1)$, ou para denotar $\psi_R(c)$, $\psi_\otimes(c)$, ou $\psi_\omega(c)$, caso c seja relação, operação ou parâmetro de E_1, respectivamente. Observe-se, também, que, nesta mesma definição, ψ não precisa satisfazer as condições da Definição 5.1.

É importante observar que, quando dizemos que μ_1 *gera* μ_2, não afirmamos que μ_1 implica μ_2. Tampouco afirmamos a veracidade de μ_1 ou de μ_2. A afirmação de que μ_1 gera μ_2 é simplesmente o enunciado da relação formal envolvendo μ_1 e μ_2 acima definida. Dizer que uma sentença da estrutura E_1 gera uma outra sentença, sendo esta sobre a estrutura E_2, equivale a dizer que tais sentenças têm a mesma forma, cada qual na respectiva linguagem.

O lema a seguir mostra-nos que muito se pode dizer quando a regra de associação ψ empregada para gerar sentenças por substituição uniforme é uma transformação regular.

Lema 6.1 (Lema da Equivalência). *Sejam E_1 e E_2 duas estruturas discretas equivalentes, a primeira compondo o modelo de testabilidade M_1 e a segunda compondo o modelo M_2, e seja ψ a transformação regular entre E_1 e E_2. Então toda sentença verdadeira na linguagem $L(M_1)$ sobre a estrutura discreta E_1 gera uma sentença verdadeira na linguagem $L(M_2)$ sobre a estrutura discreta E_2, mediante a substituição uniforme de cada constante c de $L(M_1)$ por $\psi(c)$ de $L(M_2)$.*

É fácil perceber a equivalência formal entre duas sentenças μ_1 e μ_2 tais que μ_1 gera μ_2, por exemplo, comparando o Lema da Regularidade para grafos (Lema 2.1) e o Lema da Regularidade para torneios (Lema 2.3): se os reescrevermos em linguagem L, adotando alguma estrutura discreta de grafos (como a mostrada na Seção 5.2) e alguma de torneios, poderemos verificar que o primeiro gera o segundo. É disso que se trata o Exemplo 6.1, que segue.

Exemplo 6.1. Para verificar que o enunciado do Lema 2.1 gera o enunciado do Lema 2.3, o primeiro passo é converter ambos para a linguagem L.

Consideremos que E_1 seja uma estrutura cujas instâncias são grafos e equipartições de grafos, que tenha como uma das suas relações o conjunto de todo par (G,Q) tal que Q é equipartição de G e que tenha como parâmetros a ordem de um grafo, a cardinalidade de uma equipartição e o ínfimo do conjunto de todo γ tal que Q é uma equipartição γ-regular de G (o que chamaremos de parâmetro "regularidade"). E consideremos que E_2 é uma estrutura cujas instâncias são torneios e equipartições de torneio, que tenha como relação o conjunto de todo par (G,Q) tal que Q é equipartição de G e que tenha como parâmetros o tamanho de um torneio, a cardinalidade de uma equipartição e o ínfimo do conjunto de todo γ tal que Q é uma equipartição γ-regular de T.

Lembrando que o Lema 2.1 diz que "para todo $\gamma \in (0,1)$ e todo inteiro positivo m, existe um inteiro positivo M tal que todo grafo G com ordem maior do que M admite uma equipartição γ-regular de cardinalidade maior ou igual a m e menor ou igual a M", podemos escrevê-lo simbolicamente, sem perda nem ganho de generalidade, assim:

$$(\forall \gamma \in (0,1))(\forall m \in \mathbb{N}^*)(\exists M \in \mathbb{N}^*)(\forall G \in W_1)(\alpha)$$

em que α representa a fórmula

$$|G| > M \Rightarrow (\exists Q \in W_2)((G,Q) \in R \wedge \omega_1(G,Q) \leq \gamma \wedge m < \omega_2(Q) < M)$$

onde W_1 é o conjunto de grafos do universo de E_1, W_2 é o conjunto de equipartições de grafos do universo de E_1, R é a propriedade binária dos pares compostos por um grafo e uma equipartição sua, ω_1 é o parâmetro "regularidade" de uma equipartição em relação a um grafo e o parâmetro unário parcial ω_2 é a cardinalidade de uma equipartição.

Em linguagem L, esta sentença fica assim:

$$(\forall r_{1,1} \in C_{1,1})(\forall r_{2,1} \in C_{2,1})(\exists r_{3,1} \in C_{2,1})(\forall x_{1,1} \in C_{3,1})(\alpha_1 \Rightarrow \beta_1)$$

$$\alpha_1 : \omega_{3,1}(x_{1,1}) > r_{3,1}$$

$$\beta_1 : (\exists x_{2,1} \in C_{4,1})((x_{1,1}, x_{2,1}) \in R_{1,1} \wedge \omega_{1,1}(x_{1,1}, x_{2,1}) \leq r_{1,1} \wedge r_{2,1} < \omega_{2,1}(x_{2,1}) < r_{3,1})$$

onde $C_{1,1}$ é o intervalo (0,1), $C_{2,1}$ é o conjunto dos inteiros positivos, $C_{3,1}$ é o universo dos grafos, $C_{4,1}$ é o universo das equipartições de grafos, $R_{1,1}$ é a relação entre duas instâncias tais que a primeira seja um grafo e a segunda seja uma equipartição de tal grafo, etc...

Se executarmos esse mesmo procedimento para converter o Lema 2.3 para a linguagem L, este passará a ficar assim:

$$(\forall r_{1,2} \in C_{1,2})(\forall r_{2,2} \in C_{2,2})(\exists r_{3,2} \in C_{2,2})(\forall x_{1,2} \in C_{3,2})(\alpha_2 \Rightarrow \beta_2)$$

$$\alpha_2 : \omega_{3,2}(x_{1,2}) > r_{3,2}$$

$$\beta_2 : (\exists x_{2,2} \in C_{4,2})((x_{1,2}, x_{2,2}) \in R_{1,2} \wedge \omega_{1,2}(x_{1,2}, x_{2,2}) \leq r_{1,2} \wedge r_{2,2} < \omega_{2,2}(x_{2,2}) < r_{3,2})$$

onde $C_{1,2}$ é o intervalo (0,1), $C_{2,2}$ é o conjunto dos inteiros positivos, $C_{3,2}$ é o universo dos torneios, $C_{4,2}$ é o universo das equipartições de torneios, $R_{1,2}$ é a relação entre duas instâncias tais que a primeira seja um torneio e a segunda seja uma equipartição de tal torneio, etc...

Comparando os dois lemas acima na linguagem L, levando em conta a Definição 6.4, podemos concluir que o Lema 2.1 gera o Lema 2.3, quando realizamos a substituição uniforme do parâmetro ordem de um grafo pelo parâmetro tamanho de um torneio, da relação "...ser equipartição do grafo..." pela relação "...ser equipartição do torneio..." e do parâmetro "regularidade de uma equipartição de grafos" pelo parâmetro "regularidade de uma equipartição de torneios". (Lembremo-nos de que, conforme a Definição 6.4, o fato de uma sentença gerar outra nada tem a ver com a veracidade ou falsidade de qualquer uma delas.)

Na Seção 6.2 provaremos o Lema da Equivalência de maneira rigorosa. Veremos que, para uma prova do teorema da equivalência, basta provar a equivalência lógica entre fórmulas atômicas da linguagem L de estruturas discretas equivalentes que guardem a relação de que uma gera a outra e, posteriormente, provar recursivamente a equivalência lógica para sentenças equivalentes formalmente mais complexas desta mesma linguagem. Veremos isso em detalhes.

6.2 Prova do Lema da Equivalência

No que segue, adotaremos a seguinte terminologia: $C_{1,1}$ é o universo da estrutura discreta do modelo de testabilidade M_1. $s_{1,1}$, $s_{2,1}$, $s_{3,1}$,..., são constantes da estrutura discreta de M_1; $R_{1,1}, R_{2,1}$, $R_{3,1}$, ... são relações da estrutura discreta E_1 de M_1. $\otimes_{1,1}$,$\otimes_{2,1}\otimes_{3,1}$,... são operações da estrutura discreta de M_1. $\omega_{1,1}, \omega_{2,1}, \ldots$ são parâmetros da estrutura discreta de M_1.

Também empregaremos, de maneira análoga, os termos seguintes para denotar objetos da estrutura discreta do modelo M_2. $C_{1,2}$ é o universo da estrutura discreta do modelo de testabilidade M_2. $s_{1,2}$, $s_{2,2}$, $s_{3,2}$,..., são constantes da estrutura discreta E_2 de M_2. $R_{1,2}, R_{2,2}$, $R_{3,2}$, ... são relações da estrutura discreta de

M_2. $\otimes_{1,2}$,$\otimes_{2,2}\otimes_{3,2}$,... são operações da estrutura discreta de M_2. $\omega_{1,2}, \omega_{2,2}, \omega_{3,2}, \ldots$ são parâmetros da estrutura discreta de M_2.

Afirmação 6.1. *Se (E_1, E_2) é um par equivalente de estruturas discretas com respeito à transformação regular ψ e se μ_1 é uma sentença que seja fórmula atômica sem quantificadores na linguagem L sobre objetos da estrutura E_1 e μ_2 é uma sentença que seja fórmula atômica sem quantificadores na linguagem L sobre objetos da estrutura E_2 tal que μ_1 gera μ_2 por substituição uniforme de cada objeto pela imagem de ψ, então $\mu_1 \Leftrightarrow \mu_2$.*

Demonstração. Pensemos no caso em que μ_1 é da forma $\alpha \in R_{j,1}$, onde $R_{j,1}$ é uma relação de E_1. Supondo que $\mu 1$ gera μ_2 por substituição uniforme segundo ψ, tem-se que μ_2 será da forma $\beta \in R_{j,2}$, em que $R_{j,2} = \psi\left(R_{j,1}\right)$ e $\beta = \psi(\alpha)$. Assim, conforme a equivalência lógica enunciada no item 1 da Definição 5.1, $\mu_1 \Leftrightarrow \mu_2$.

Supondo que μ_1 seja da forma $\otimes_{j,1}\left(s_{i_1,1}, s_{i_2,1}, \ldots, s_{i_n,1}\right) = s_{i_n+1,1}$, então μ_2 é da forma $\otimes_{j,2}\left(s_{i_1,2}, s_{i_2,2}, \ldots, s_{i_{n(j)},2}\right) = s_{i_{n(j)}+1,2}$, e o item 2 da Definição 5.1 assegura que $\mu_1 \Leftrightarrow \mu_2$, $\forall j \forall i_1 \forall i_2 \ldots \forall i_n$.

Se μ_1 é da forma $\omega_{j,1}\left(s_{i_1,1}, s_{i_2,1}, \ldots, s_{i_{n(j)},1}\right) = u_{k,1}$, para que μ_1 gere μ_2, μ_2 será da forma $\omega_{j,2}\left(s_{i_1,2}, s_{i_2,2}, \ldots, s_{i_{n(j)},2}\right) = u_{k,2}$. Neste caso, o item 3 da Definição 5.1 garante-nos que $\mu_1 \Leftrightarrow \mu_2$, $\forall j \forall k \forall i_1 \forall i_2 \ldots \forall i_n$.

Podemos continuar analisando todos os tipos de fórmulas atômicas da linguagem L, um por um, como fizemos com os três casos acima. Em todos os casos, poderemos verificar que $\mu_1 \Leftrightarrow \mu_2$. □

Afirmação 6.2. *Se (E_1, E_2) é um par equivalente de estruturas discretas com respeito à transformação regular ψ e se μ_1 é uma sentença sem quantificadores na linguagem L sobre objetos da estrutura E_1 e μ_2 é uma sentença sem quantificadores na linguagem L sobre objetos da estrutura E_2 tal que μ_1 gera μ_2 por substituição uniforme de cada objeto pela imagem de ψ, então $\mu_1 \Leftrightarrow \mu_2$.*

Demonstração. Partindo da Afirmação 6.1, o que precisamos mostrar é que aquilo que é expresso na referida afirmação sobre as fórmulas atômicas não quantificadas também é válido para conjunções, disjunções e negações de sentenças que sejam fórmulas atômicas não quantificadas, ou seja, sobre a classe definida recursivamente por meio da aplicação dos mencionados conectivos lógicos sobre sentenças cada vez maiores, começando com as fórmulas atômicas não quantificadas que sejam sentenças.

Tal prova se dá por indução finita. Chamemos de S_0 a classe das fórmulas atômicas da linguagem L. E chamemos de S_n a classe das fórmulas da linguagem L que sejam conjunções ou disjunções de duas fórmulas de S_{n-1}, a negação de uma fórmula de S_{n-1}, ou, ainda, uma fórmula de S_{n-1}. Queremos mostrar que, se $\mu_1 \in S_n$ e $\mu_2 = \psi(\mu_1)$, então $\mu_1 \Leftrightarrow \mu_2$. A Afirmação 6.1 garante que, se $\mu_1 \in S_0$, então a afirmação de que μ_1 gera μ_2 segundo ψ implica que $\mu_1 \Leftrightarrow \mu_2$. O que nos cabe agora é dar o passo de indução para concluir a prova.

Nesta prova, o passo de indução consiste em mostrar que:

1- Se μ_1 é uma sentença sem quantificadores na linguagem L sobre objetos da estrutura E_1 tal que μ_1 gera μ_2 segundo ψ, então $\sim \mu_1 \Leftrightarrow \sim \mu_2$.

2- Se μ_1 e τ_1 são sentenças sem quantificadores na linguagem L sobre objetos da estrutura E_1 tal que μ_1 gera μ_2 segundo ψ e τ_1 gera τ_2 segundo ψ, então $(\mu_1 \wedge \tau_1) \Leftrightarrow (\mu_2 \wedge \tau_2)$.

A primeira destas duas teses listadas acima é decorrência trivial da Afirmação 6.1, pois, $\mu_1 \Leftrightarrow \mu_2$ se, e somente se, $\sim \mu_1 \Leftrightarrow \sim \mu_2$, conforme mostram os manuais de lógica matemática.

A segunda das duas teses acima decorre da Afirmação 6.1 e da análise do seguinte argumento:

$$\mu_1 \Leftrightarrow \mu_2$$

$$\tau_1 \Leftrightarrow \tau_2$$

$$\therefore (\mu_1 \wedge \tau_1) \Leftrightarrow (\mu_2 \wedge \tau_2)$$

Esse argumento é válido, o que pode ser demonstrado por tabelas de verdade. E μ_1 gera μ_2 por substituição uniforme de cada objeto pela imagem de ψ (a transformação regular entre μ_1 e μ_2) se, e somente se, $(\mu_1 \wedge \tau_1)$ gera $(\mu_2 \wedge \tau_2)$.

A classe de sentenças sobre as quais provamos que a Afirmação 6.2 é verdadeira inclui os conectivos lógicos $\sim$, $\vee$ e $\wedge$. Os conectivos lógicos $\Rightarrow$ e $\Leftrightarrow$ podem ser definidos em função dos conectivos $\sim$ e $\wedge$, o que nos permite concluir a demonstração da Afirmação 6.2. $\square$

Agora demonstraremos o Lema da Equivalência em toda a sua generalidade.

Afirmação 6.3. *Se (E_1, E_2) é um par equivalente de estruturas discretas com respeito à transformação regular ψ, μ_1 é uma sentença na linguagem L sobre objetos da estrutura E_1 e μ_2 é uma sentença na linguagem L sobre objetos da estrutura E_2 tal que μ_1 gera μ_2 por substituição uniforme de cada objeto de E_1 pela imagem de ψ sobre o mesmo, então $\mu_1 \Leftrightarrow \mu_2$.*

As sentenças na linguagem L de que trata a Afirmação 6.2 são sem variáveis, somente com constantes, pois são sentenças não quantificadas. Podemos construir o conjunto de todas as sentenças que se iniciam com o quantificador universal mediante a conjunção de sentenças sem quantificador, e podemos construir o conjunto de todas as sentenças que se iniciam com o quantificador existencial mediante a disjunção inclusiva de sentenças sem quantificador. Isso é feito da seguinte forma:

$$\Big((\forall j \in [h])\big(\theta_j\big)\Big) \Leftrightarrow \bigwedge_{j=1}^{h} \theta_j$$

$$\Big((\exists j \in [h])\big(\theta_j\big)\Big) \Leftrightarrow \bigvee_{j=1}^{h} \theta_j$$

É sabido que toda sentença pode ser escrita como uma sentença na *forma normal prenex*, em que apenas há quantificadores no início da sentença, sendo tal expressão seguida de uma fórmula livre de quantificadores. Consequentemente, podemos ampliar a tese da Afirmação 6.2 para sentenças quantificadas da linguagem L na forma normal prenex, e por extensão, para sentenças quaisquer de L, conforme estabelece a Afirmação 6.3.

Retornemos ao Lema da Equivalência enunciado na Seção 6.1. Se partirmos da Afirmação 6.3 tomando uma sentença verdadeira sobre E_1 como μ_1, então teremos que a sentença gerada por μ_1, a qual chamaremos de μ_2, é tal que $\mu_1 \Leftrightarrow \mu_2$. Como μ_1 é uma sentença verdadeira, então μ_2 também é. Desta forma, o Lema da Equivalência está provado.

6.3 Aplicações do Lema da Equivalência

Todo este sexto capítulo tem sido dedicado ao Lema da Equivalência. Primeiramente, na Seção 6.1, fixamos uma série de definições para habilitar-nos a enunciar o Lema, o que fizemos logo após. Posteriormente à apresentação do Lema da Equivalência, expusemos o Exemplo 6.1, que será de grande utilidade aqui, na Seção 6.3, bem como no Capítulo 7. Depois disso demonstramos o Lema da Equivalência, por um processo em que deduzimos sucessivamente as afirmações 6.1 a 6.3. No presente momento da nossa exposição, estamos preocupados com a aplicação do Lema da Equivalência. Queremos organizar algumas vertentes a serem trilhadas para fazer gerar conhecimento novo com a ajuda do Lema da Equivalência. Este é o propósito primordial do capítulo.

Neste ponto, é conveniente fixarmos formalmente uma estrutura discreta de grafos e uma estrutura discreta de torneios para tomarmos como referência nas nossas considerações. Para isso, adotaremos as estruturas discretas fixadas nos Exemplos 5.1 e 5.2.

Retomemos o Exemplo 6.1. No mencionado exemplo, trabalhamos numa estrutura discreta cujo universo é a união do conjunto de grafos com o conjunto de equipartições de grafos. Consideremos que as estruturas E_1 do Exemplo 6.1 é resultante da inclusão das equipartições de grafos à estrutura discreta de grafos definida no Exemplo 5.1. E, da mesma forma, consideremos que a estrutura E_2 do Exemplo 6.1 resulta da inclusão das equipartições de torneios à estrutura definida no Exemplo 5.2. Sobre estas duas estruturas discretas enunciamos, no Exemplo 6.1, os Lemas 2.1 e 2.3 em linguagem L. Naquele exemplo, comparamos os Lemas 2.1 e 2.3 escritos em linguagem L e verificamos que o primeiro gera o segundo, nos termos da Definição 6.4. Se excluirmos as variáveis do enunciado do Lema 2.1 em linguagem L, conforme vimos naquele exemplo, os termos que sobram são os conjuntos $C_{1,1}$, $C_{2,1}$, $C_{3,1}$ e $C_{4,1}$, a relação $R_{1,1}$ e os parâmetros $\omega_{1,1}$ e $\omega_{2,1}$. Semanticamente, $C_{1,1}$ denota o intervalo $(0,1)$, $C_{2,1}$ denota o conjunto dos inteiros positivos, $C_{3,1}$ é o universo dos grafos, $C_{4,1}$ é o universo das equipartições de grafos, $R_{1,1}$ é a relação de duas instâncias tais que a primeira seja um grafo e a segunda seja uma equipartição de tal grafo, o parâmetro ω_1 é a "regularidade" de uma equipartição em relação a um grafo e o parâmetro ω_2 é a cardinalidade de uma equipartição.

O Exemplo 6.1 mostra que o enunciado do Lema 2.1 gera o enunciado do Lema 2.3 por substituição uniforme nos termos da Definição 6.4, para alguma 4-upla ψ, mas não nos habilita a provar este a partir daquele. Como poderíamos provar o Lema 2.3 por meio da aplicação do Lema da Equivalência ao Lema 2.1? Em primeiro lugar, devemos estar certos de que alguma subestrutura de E_1 composta por termos mencionados no parágrafo anterior forme um par equivalente com a subestrutura correspondente de E_2. Tais termos são $C_{1,1}$, $C_{2,1}$, $C_{3,1}$ e $C_{4,1}$, a relação $R_{1,1}$ e os parâmetros $\omega_{1,1}$ e $\omega_{2,1}$, e seus correspondentes em E_2 são $C_{1,2}$, $C_{2,2}$, $C_{3,2}$ e $C_{4,2}$, a relação $R_{1,2}$ e os parâmetros $\omega_{1,2}$ e $\omega_{2,2}$. Precisamos selecionar os objetos de E_1 e de E_2 que formem o maior par de subestruturas discretas destas estruturas. Para sermos mais precisos, uma condição suficiente para demonstrar o Lema 2.3 mediante a aplicação do Lema da Equivalência ao Lema 2.1 é provar que o seguinte par de estruturas discretas seja um par equivalente de estruturas discretas:

$$((C_{3,1} \cup C_{4,1}, C_{3,1}, C_{4,1}, R_{1,1}, \omega_{1,1}, \omega_{2,1}), (C_{3,2} \cup C_{4,2}, C_{3,2}, C_{4,2}, R_{1,2}, \omega_{1,2}, \omega_{2,2}))$$

Os conjuntos $C_{1,1}$, $C_{2,1}$, $C_{1,2}$ e $C_{2,2}$ não são conjuntos de instâncias, de maneira que não integram as estruturas discretas. O primeiro objeto da primeira subestrutura é $C_{3,1} \cup C_{4,1}$, que é o seu universo. Da mesma forma, o primeiro objeto da segunda é $C_{3,2} \cup C_{4,2}$, seu universo. Os conjuntos $C_{3,1}$ e $C_{4,1}$ são subconjuntos do universo da primeira subestrutura, enquanto os conjuntos $C_{3,2}$ e $C_{4,2}$ são subconjuntos do universo da segunda subestrutura. Estes quatro conjuntos são, pois, relações unárias das respectivas estruturas, assim como $R_{1,1}$ e $R_{1,2}$. Chamemos a primeira destas estruturas de E_3 e a segunda de E_4. E_3 é uma subestrutura de E_1, E_4 é uma subestrutura de E_2, e:

$$E_3 = (C_{3,1} \cup C_{4,1}, C_{3,1}, C_{4,1}, R_{1,1}, \omega_{1,1}, \omega_{2,1})$$

$$E_4 = (C_{3,2} \cup C_{4,2}, C_{3,2}, C_{4,2}, R_{1,2}, \omega_{1,2}, \omega_{2,2})$$

Todos os termos que figuram no Lema 2.1 são objetos de E_3, e todos os termos que figuram no Lema 2.3 são objetos de E_4. O Exemplo 6.1 nos mostra que o Lema 2.1 gera o Lema 2.3. Para que possamos aplicar o Lema da Equivalência para provar o Lema 2.3 partindo do Lema 2.1, é suficiente que E_3 e E_4 sejam estruturas discretas equivalentes. E para que E_3 e E_4 sejam estruturas discretas equivalentes, é suficiente e necessário que exista uma transformação regular entre E_3 e E_4 que preserva a relação de invariância, o que decorre da Definição 6.2.

A seguir definiremos uma bijeção entre as estruturas discretas E_1 e E_2 dos Exemplos 5.1 e 5.2 para fazermos algumas considerações.

Definição 6.5. *Seja W_j o universo da estrutura discreta E_j, j=1,2, tal como nos Exemplos 5.1 e 5.2. Definiremos como $\phi_1 : W_1 \longrightarrow W_2$ a função que leva um grafo G a um torneio $\phi_1(G)$ da seguinte maneira: empregando os conceitos de arco crescente e arco decrescente do Exemplo 5.2, $V(G) = V(\phi_1(G))$, $\{u,v\} \in A(G)$ implica que u e v formam um arco crescente de $\phi_1(G)$ e $\forall u \in V(G) \forall v \in V(G)$, $\{u,v\} \notin A(G)$ implica que u e v formam um arco decrescente de $\phi_1(G)$. É fácil ver que a função ϕ_1 é única com tais atributos e que é uma bijeção.*

Definição 6.6. *Adotemos $C_{3,1}$, $C_{4,1}$ tal como no Exemplo 6.1 e fixemos as estruturas discretas E_3 e E_4 tal como feito acima. Definiremos como $\phi_2 : W(E_3) \longrightarrow W(E_4)$ a função tal que $i \in C_{3,1} \Rightarrow (\phi_2(i) = \phi_1(i))$ e $i \in C_{4,1} \Rightarrow (\phi_2(i) = i)$.*

É imediato que ϕ_2 é uma bijeção que leva grafos a torneios e equipartições de grafos a equipartições de torneios.

Proporemos um método decorrente do Lema da Equivalência para exportar conhecimento de uma teoria de testabilidade referente a um tipo de estrutura discreta para outra teoria, correspondente a outro tipo de estrutura. Este método é composto por três passos, como segue. Para dadas estruturas discretas E_1 e E_2:

1º Passo: Definir uma bijeção $\phi : W_1 \longrightarrow W_2$, o que permite obter a transformação regular ψ gerada por ϕ.

2º Passo: Verificar se é possível provar que (E_1, E_2) é um par equivalente de estruturas discretas com respeito à transormação ψ.

3º Passo: Caso se tenha verificado, no 2º passo, que (E_1, E_2) é um par equivalente de estruturas discretas, então para *qualquer* sentença verdadeira μ_1 de E_1, gerar a sentença μ_2 de E_2, a qual será verdadeira.

Neste método, se for provado no seu segundo passo que (E_1, E_2) é um par equivalente de estruturas discretas, o terceiro passo pode ocorrer diversas vezes de modo independente. Não é para toda bijeção definida no 1º passo que haverá uma transformação regular, mas, se as estruturas discretas E_1 e E_2 formarem um par equivalente (conforme prova do segundo passo), qualquer sentença μ_1 de E_1 que gerar a sentença μ_2 de E_2 é tal que, se μ_1 é verdadeira, então μ_2 também é verdadeira. Uma vez provado o fato a que se refere o 2º passo deste método, todas as sentenças verdadeiras de E_1 geram sentenças verdadeiras de E_2 por substituição uniforme segundo a transformação regular encontrada.

A seguir, mostraremos que, se $S \lhd G$, então $\phi_1(S) \subseteq \phi_1(G)$ e, com isso, encontraremos um exemplo de uma transformação regular.

O que queremos provar é que, dados os grafos G e S tais que $S \lhd G$, os torneios $\phi_1(S)$ e $\phi_1(G)$ são tais que $\phi_1(S) \subseteq \phi_1(G)$. Comecemos observando que $V(\phi_1(G)) = V(G)$ e $V(\phi_1(S)) = V(S)$, conforme Definição 6.6. Reparemos, agora, nas relações de ser subgrafo induzido e de ser subtorneio. No Capítulo 2 definimos estas duas relações. Sabemos que $V(S) \subseteq V(G)$ e S contém todas as arestas de G incidentes em dois vértices de S, o que significa que $A(S) = A(G) \cap [V(S)]^2$. Se provarmos que estas premissas implicam que $\phi_1(S) \subseteq \phi_1(G)$, dar-nos-emos por satisfeitos. Já sabemos quais são os conjuntos de vértices de $\phi_1(S)$ e de $\phi_1(G)$, bem como os conjuntos de vértices e arestas de S e G. Assim, para verificarmos se $\phi_1(S) \subseteq \phi_1(G)$, precisamos encontrar os conjuntos de arcos de $\phi_1(S)$ e de $\phi_1(G)$.

Ainda segundo a Definição 6.5, para cada aresta de G, há o correspondente arco crescente de $\phi_1(G)$, e para cada par de vértices não conectados por aresta de G, há o correspondente arco decrescente de $\phi_1(G)$. Ou seja,

$$A(\phi_1(G)) = \Big\{(u,v) : (\{u,v\} \in A(G) \wedge u < v) \vee \Big(\{u,v\} \in [V(G)]^2 \backslash A(G) \wedge u > v\Big)\Big\}$$

Analogamente,

$$A(\phi_1(S)) = \Big\{(u,v) : (\{u,v\} \in A(S) \wedge u < v) \vee \Big(\{u,v\} \in [V(S)]^2 \backslash A(S) \wedge u > v\Big)\Big\}$$

Todos os vértices de $\phi_1(S)$ são vértices de $\phi_1(G)$, e cada par de vértices distintos de $\phi_1(S)$ compõe um arco crescente ou um arco decrescente de $\phi_1(S)$ e um arco crescente ou decrescente em $\phi_1(G)$. Selecionemos dois vértices de $\phi_1(S)$ e verifiquemos se eles compõem um arco de $\phi_1(S)$ presente também em $\phi_1(G)$. Sejam u e v os vértices selecionados. Se $\{u,v\} \in A(S)$, então o arco crescente contendo u e v é arco de $\phi_1(S)$. Além disso, se $\{u,v\} \in A(S)$, então $\{u,v\} \in A(G)$ e o arco crescente contendo u e v é arco de $\phi_1(G)$. Se $\{u,v\} \notin A(S)$, então o arco decrescente contendo u e v é arco de $\phi_1(S)$ e de $\phi_1(G)$. Em qualquer caso, todo par de vértices de $\phi_1(S)$ compõe um arco de $\phi_1(S)$ presente também em $\phi_1(G)$. Portanto $S \lhd G$ implica que $\phi_1(S) \subseteq \phi_1(G)$, como queríamos provar. O exemplo de transformação regular que segue abaixo decorre da prova que acabamos de concluir:

Exemplo 6.2. As estruturas discretas $(W_1, C_1, R_1, F_1, \omega_1)$ e $(W_2, C_2, R_2, F_2, \omega_2)$, onde W_1 é o universo da estrutura definida no Exemplo 5.1, W_2 é o universo da estrutura definida no Exemplo 5.2, $\lhd$ é a relação binária de ser subgrafo induzido, $\subseteq$ é a relação binária de ser subtorneio e $C_1 = F_1 = \omega_1 = C_2 = F_2 = \omega_2 = \emptyset$ formam um par equivalente com respeito à transformação regular $(\psi_w, \psi_R, \psi_\otimes, \psi_\omega)$, em que $\psi_W = \phi_1$, a bijeção fixada na Definição 6.5, e $\psi_R, \psi_\otimes, \psi_\omega$ são as bijeções que respeitam as condições da Definição 5.1.

7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Iniciamos este trabalho com uma revisão da literatura existente sobre algumas das teorias de testabilidade de propriedades. Procuramos expor os resultados mais gerais em cada ramificação deste estudo. Em seguida, propusemos uma generalização destas teorias para, então, discorrermos sobre a possibilidade de aprendermos mais sobre testabilidade de objetos matemáticos de determinada classe a partir de um conhecimento pré-existente de outra classe de objetos matemáticos. Um instrumento que identificamos para este fim é o Lema da Equivalência, deduzido no capitulo 6, após o que esboçamos algumas aplicações. Apesar de termos buscado prezar pela generalidade, focamo-nos mais nas teorias dos grafos e dos torneios.

Adotando a terminologia do Capítulo 6, vimos no Exemplo 6.1 que, sob determinada formulação, o Lema da Regularidade para grafos gera o Lema da Regularidade para torneios por substituição uniforme segundo certa lei de associação entre os objetos da estrutura discreta de grafos e a estrutura discreta de torneios. Na Seção 6.3 contextualizamos melhor este resultado, definindo uma condição suficiente para provarmos um destes lemas a partir do outro mediante a aplicação do Lema da Equivalência. Para aplicarmos o Lema da Equivalência, precisamos ter como premissa uma sentença escrita em linguagem L sobre a estrutura discreta de partida, de maneira que, preservando as relações formais, geramos uma sentença logicamente equivalente e formalmente idêntica sobre a estrutura discreta de chegada. Nada nos garante que, se mudarmos um pouco a definição da estrutura de partida ou da estrutura de chegada, conseguiremos estabelecer a desejada equivalência lógica entre as versões para grafos e para torneios do Lema da Imersão ou do Lema da Remoção.

O método sucintamente exposto na Seção 6.3 pode ser empregado para fazerem-se derivar teoremas de estruturas de permutações a partir de estruturas de dígrafos, ou para demonstrarem-se teoremas sobre grupos finitos a partir de teoremas de orientações de grafos. São métodos de propósito geral para importar teoremas de uma estrutura a partir de outra estrutura discreta. Nossa escolha pelo exemplo citado no parágrafo anterior deu-se por mera comodidade, sem que isso represente uma limitação dos métodos.

Por outro lado, dadas duas estruturas discretas, definir um par de subestruturas equivalentes nem sempre é uma tarefa simples. Concluímos, no final do capítulo anterior, que uma condição suficiente para exportarmos o Lema da Regularidade para grafos para o universo dos torneios, em dada estrutura de torneios, é provarmos que as estruturas que ali chamamos de E_3 e E_4 são equivalentes. Entretanto não demonstramos a mencionada equivalência. Sem entrar no mérito desta dificuldade, restringimo-nos a observar que a função fixada na Definição 6.6 não gera uma transformação regular ψ tal que (E_3, E_4) seja um par equivalente com respeito a ψ.

A similaridade entre as estruturas de grafos e de torneios é intuitiva, assim como a analogia entre os Lemas da Regularidade, da Imersão e da Remoção para grafos e suas versões para torneios. O Lema da Equivalência faz derivarem teoremas sobre uma estrutura discreta a partir de teoremas que tenham a mesma forma na linguagem L, mas que figurem no contexto de outra estrutura discreta. Isso pode parecer uma limitação da aplicabilidade do Lema da Equivalência, no sentido de que a igualdade formal é requisito necessário de qualquer processo dedutivo que explore o lema. Mas isso não corresponde à verdade. Podemos expandir o método esboçado na Seção 6.3 para munir-nos para a capacidade de deduzirmos teoremas formalmente distintos dos teoremas originais. Tome-se como exemplo uma situação em que estamos trabalhando com uma transformação "próxima" de ser regular à luz de certa métrica de funções. Para que se possa assegurar validade do ponto de vista lógico, convém que as sentenças geradas por tal transformação mantenham o mesmo valor de verdade das sentenças originais, como reza o Lema da Equivalência. Neste caso, o que distingue tais transformações das transformações regulares é a diferença formal entre as sentenças originais e as sentenças geradas. Com isso, conseguimos deduzir proposições que não sejam formalmente idênticas àquelas das quais partimos. Destas ideias pode surgir uma extensa variedade de métodos em futuros trabalhos, tendo o Lema da Equivalência como fundamento.

Temos muito a aprender sobre torneios com os grafos, e vice-versa. Uma interessante linha de investigação surge, quando nos perguntamos como podemos simular teste de propriedades de grafos por meio do teste de propriedades de torneios.

Não é possível esgotar um tema como o desta dissertação, nem foi esta a intenção deste trabalho. Todas essas ideias esboçadas no presente capítulo são campos em potencial a serem explorados em futuras pesquisas. Claro que não há um caminho pré-determinado a ser trilhado, o qual inevitavelmente garanta sucesso no desenvolvimento de qualquer uma das possíveis técnicas aqui mencionadas, no entanto, como vimos, há uma grande variedade de caminhos a serem seguidos dentro do que se propôs no Capítulo 1. Em suma: acreditamos que sejam necessários novos estudos para desenvolver-se a matéria desta dissertação.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ALON, N; FISCHER, E.; KRIVELEVICH, M.; SZEGEDY, M. **Efficient testing of large graphs**. 4th ed. Combinatorica, Bolyai Society–Springer-Verlag, v. 20, p.451–476, 2000.

[2] ALON, N. **Testing subgraphs in large graphs**, Random Structures Algorithms. 21 (2002), 359–370.

[3] ALON, N.; SHAPIRA, A. **Every monotone Graph Property is Testable**. Association for Computing Machinery. New York. 2005.

[4] ALON, N.; SHAPIRA, A. **Homomorphisms in Graph Property Testing**. Topics in Discrete Mathematics, Springer, Berlin, p. 281–313, 2006.

[5] ALON, N; SHAPIRA, A. **A characterization of easily testable induced subgraphs**, Combin. Probab. Comput 15, p. 791–805, 2006.

[6] ALON, N.; SHAPIRA, A. **A Characterization of the (natural) Graph Properties Testable with One-Sided Error**. SIAM Journal on Computing (Special Issue of FOCS'05), p. 1703–1727, 2008.

[7] ALON, N.; FOX, J. **Easily Testable Graph Properties**, Cambridge University Press, Cambridge, p. 646–657, 2015.

[8] BENDER, M. A.; RON, D. **Testing properties of directed graphs: acyclicity and connectivity**. Random Structures & Algorthms - Wiley. v. 20, Issue 2, p. 184–205, 2002.

[9] BLUM, M.; LUBY, M.; RUBINFELD R. **Self-testing/correcting with applications to numerical problems**, Journal of Computer and System Science, v. 47, No. 3, p. 549–595, 1993.

[10] BORGS, C.; CHAYES, J.; LOVÁSZ, L.; SÓS T.; VESZTERQOMBI, K. **Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing**, Advances in Mathematics, 219(6), p. 1801–1851, 2008.

[11] CONLON D.; FOX, J. **Bounds for graph regularity and removal lemmas**, Geom. Funct. Anal. 22, p. 1191–1256, 2012.

[12] CONLON D.; FOX, J. **Graph removal lemmas**, in Surveys in Combinatorics, 2013, London Mathematical Society Lecture Note Series, Cambridge University Press, Cambridge, v. 409, p. 1–49, 2013.

[13] COREGLIANO, L. N.; RAZBOROV, A. A. **Semantic limits of dense combinatorial objects**. Uspekhi Mat. Nauk, p. 45–152, 2020.

[14] DIACONIS, P.; GRAHAM R. L. **Spearman's futrule as a measure of disarray**, J. Roy. Statist. Soc. Ser. B 39, p. 262–268, 1977.

[15] FOX, J.; LOVÁSZ, L. M. **A tight bound for Green' s arithmetic triangle removal lemmas in vector spaces**, Advances in Mathematics 321, p. 287–297, 2017.

[16] FOX, J.; WEI, F. **Fast property testing and metrics for permutations**. Combinatorics Probability and Computing, p. 1–41, 2018.

[17] GOLDREICH O.; TREVISAN L. **Three Theorems regarding Testing Graph Properties**. Random Structures & Algorithms- Wiley Online Library. p. 23–57, 2003.

[18] GOLDREICH, O.; RON, D. **Property testing in bounded degree graphs**. In Procedings of the Thirty-First Annual ACM Symposium on the Theory of Computing, p. 406–415, 1997.

[19] GOLDREICH, O.; GOLDWASSER, S.; RON, D. **Property testing and its connection to learning and approximation**. Journal of the Association for Computing Machinery, p. 653–750, 1998.

[20] GOLDREICH, O. **Introduction to Property Testing**. New York: Cambridge University Press, 2017.

[21] HALEVI, S.; LACHISH, O.; NEWMAN, I.; TSUR, D. **Testing Orientation Properties**. Electronic Colloquium on Computational Complexity, Report No. 153, 2005.

[22] HEGENBERG, L. **Lógica**: Cálculo Sentencial, Cálculo de predicados, Cálculo com Igualdade. 3. ed. Rio de Janeiro: Forense Universitária, 2012.

[23] HOPPEN, C.; KOHAYAKAWA, Y.; MOREIRA, C. G.; SAMPAIO, R. M. **Testing permutation properties through subpermutations**. THEORETICAL COMPUTER SCIENCE, v. 412, p. 3555–3567, 2011.

[24] MORTARI, C. **Introdução à Lógica**. São Paulo: Editora UNESP, 2001.

[25] MOSHKOVITZ, G.; SHAPIRA, A. **A short proof of Gowers´ lower bound for the regularity lemma**. Combinatorica 36, p. 187–194, 2016.

[26] RUBINFIELD, R.; SUDAN, M. **Robust characterization of polynomials with application to program testing**, SIAM J. Comput., p. 256–271, 1996.

[27] STAGNI, H. **Teste de Propriedades em Torneios**. Dissertação (Mestrado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo. São Paulo. 2015.

[28] SZEMERÉDI, E. **Regular partitions of graphs**, Colloques Internationaux C.N.R.S. No 260 - Problèmes Combinatoires et Théorie des Graphes, Orsay, 399–401, 1976.