Compressão de Entropia e Colorações Legítimas em Planos Projetivos

Luís Doin

Trabalho de conclusão de curso apresentado
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Bacharel em Matemática Aplicada

Curso: Matemática Aplicada
Habilitação em Métodos Matemáticos
Orientador: Prof. Rodrigo Bissacot

São Paulo, julho de 2017

Compressão de Entropia e Colorações Legítimas em Planos Projetivos

Esta versão do trabalho de formatura contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 11/07/2017. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- ◇ Prof. Dr. Rodrigo Bissacot - IME-USP
- ◇ Prof. Dr. Guilherme Oliveira Mota - CMCC-UFABC
- ◇ Prof. Dr. Fernando Mário de Oliveira Filho - IME-USP

Agradecimentos

Esse trabalho nasceu de uma ideia ousada de como orientar uma iniciação científica em matemática: ao invés de colocar livros nas mãos de graduandos, coloque artigos! O professor Bissacot vai ainda um passo adiante incentivando seus alunos a atacar conjecturas e desenvolver ideias originais. Essa busca por novidades, mesmo que como objetivo secundário, coloca o aluno verdadeiramente em contato com a vida acadêmica, experienciando problemas muitos distintos daqueles encontrados em sala de aula onde o ambiente e as questões propostas são mais controlados. Dessa forma, ao longo do curso o aluno ganha motivação e ânimo e consegue se lembrar mais facilmente porque mesmo decidiu vencer toda a burocracia que é um curso de graduação. Ao menos essa foi minha experiência e agradeço muito ao professor Bissacot pela orientação nesse caminho. Essa é a parte acadêmica, pois outro agradecimento vai por todo apoio que recebi dele de forma geral. Quem o conhece sabe a pessoa humana que é.

Além de meu professor, também recebi a ajuda de alguns amigos. Dentre eles, quero agradecer especialmente à Laïs pelo incentivo em geral e pela ajuda no apêndice de planos projetivos, ao Marcelo, por aguçar minha curiosidade quanto a teoria de compressão de entropia, ao Eric, por sua predisposição natural em ajudar e dividir conhecimento e ao João, por me ajudar a passar em análise estatística!

Gostaria também de ressaltar a contribuição dos professores Fernando Oliveira Filho e Guilherme Mota. Esse texto passou por alterações e correções importantes após a avaliação de ambos.

Não posso deixar de agradecer a CNPq e a FAPESP pelo apoio financeiro a essa pesquisa.

Finalmente, agradeço à minha família: minha mãe Márcia, minha irmã Larissa e meu irmão Leonardo, por me incentivarem desde sempre.

Resumo

Doin, L. **Colorindo Planos Projetivos Finitos via Compressão de Entropia**. 2017. Trabalho de Conclusão de Curso - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

Esse estudo teve como objetivo fornecer ao aluno experiência em pesquisa em matemática. O caminho escolhido para tanto foi pensado de forma a apresentar ao aluno ferramentas e técnicas importantes não abordadas na graduação. O aluno então estudou o artigo [3] de Noga Alon e Zoltan Füredi sobre colorações legítimas em planos projetivos finitos onde faz-se uso de diferentes técnicas contidas no Método Probabilístico dentre elas o Lema Local de Lovász (LLL) exposto em [9]. De posse de versões melhoradas do LLL ([11] e [28]) e do método de entropy compression sistematizado por Louis Esperet e Aline Parreau em [29], buscou-se uma melhora dos resultados obtidos por Noga Alon e Zoltan Füredi. Para tanto, desenvolveu-se uma variação do método de entropy compression provando-se que tal variação pode ser aplicada em todos os problemas formulados dentro da linguagem *variable version* do Lema Local de Lovász – isso é feito na seção 3.2.3. Os resultados obtidos aplicando essa variação são expostos no Capítulo 4. Os resultados desse trabalho também estão expostos em [10].

Palavras-chave: Lema Local de Lovász, Entropy Compression, Planos Projetivos Finitos.

Sumário

Introdução

Em meados de 1940 surgia uma ideia para se obter provas de existência de objetos matemáticos por meio da construção de espaços de probabilidade adequados. Muitas técnicas foram desenvolvidas com esse espírito e esse conjunto de ideias, que mais tarde ficaria conhecido como Método Probabilístico, se mostrou poderoso e aplicável em diversas áreas da matemática. Nas palavras de Béla Bollobás [12]: "*Existence results based on probabilistic ideas can now be found in many branches of mathematics, especially in analysis, the geometry of Banach spaces, number theory, graph theory, combinatorics and computer science. Probabilistic methods have become an important part of the arsenal of a great many mathematicians. Nevertheless, this is only a beginning: in the next decade or two (escrito em 1985) probabilistic methods are likely to become even more prominent.*". E posteriormente, em 2001, nas de Niranjan Balachandran [7]: *"The Probabilistic Method has now become one of the most important and indispensable tools for the Combinatorist. There have been several hundred papers written which employ probabilistic ideas and some wonderful monographs. ... Over the past 2 decades, the explosion of research material, along with the wide array of very impressive results demonstrates another important aspect of the Probabilistic Method; some of the techniques involved are subtle, one needs to know how to use those tools, more so than simply understand the theoretical underpinnings."*. Pensando no dito por Balachandran, nesse trabalho cada capítulo terá como uma de suas preocupações fornecer ao leitor novas maneiras de aplicar as ideias apresentadas.

Paul Erdős, um dos mais prolíficos matemáticos de todos os tempos, foi talvez o maior responsável por popularizar o Método Probabilístico publicando diversos trabalhos a partir de 1947 usando esse método como em [25, 26, 27, 28]. De fato, Noga Alon e Joel Spencer usam uma foto de Erdös como capa do seu livro *The Probabilistic Method* [5] que é tido como referência por compilar diferentes abordagens e aplicações do método. Justiça seja feita, um dos primeiros trabalhos de que se tem notícia onde tal método é utilizado foi escrito em 1943 por Szele em [63] enquanto estudava caminhos Hamiltonianos. A abordagem do Método Probabilístico pode ser resumida nas seguintes palavras: para assegurar a existência de um determinado ob-

jeto (um grafo com determinadas propriedades pré-estabelecidas, por exemplo), constrói-se um espaço de probabilidade adequado ao problema onde o objeto de interesse tem probabilidade positiva de ocorrer concluindo assim que tal objeto existe mesmo que não saibamos exatamente como construí-lo. Exemplos do método sendo aplicado para provar a existência de procedimentos de rotinas eficientes podem ser encontrados em [16] e [46] e problemas de coloração em grafos em [41] e [43]. Uma das ferramentas mais afiadas do Método Probabilístico é o Lema Local de Lovász (LLL) capaz de provar a existência de objetos onde outras abordagens falham. Estudaremos esse lema no próximo capítulo. O seu ponto fraco, assim como o de todas as técnicas do Método Probabilístico, é não construir o objeto para o qual prova a existência. Dada as pequenas probabilidades com as quais o LLL pode trabalhar, desenvolver um algoritmo que encontra o objeto apontado por Lovász se mostrou uma tarefa desafiadora.

Paralelamente a tudo isso, em 1948 Claude E. Shannon fundava a área de teoria da informação com seu artigo *A Mathematical Theory of Communication* [60]. No ano seguinte o artigo se transformava em livro ganhando o nome *The Mathematical Theory of Communication*, uma pequena porém relevante mudança dada a percepção da generalidade de seu trabalho.

Em 2010 Robin Moser e Gárbor Tardos em seu celebrado artigo [51] desenvolveram, para um versão do LLL chamada de *variable version*[1]– a priori mais restritiva – um algoritmo que constrói o objeto cuja existência conhecemos pelo LLL provando que tal algoritmo pára em tempo finito usando uma bonita ideia baseada em conceitos de compressão de dados estudada na teoria da informação. Essa ideia ficou conhecida como *entropy compression method*[2]. Em 2013 Jaros Grytczuc, Jakub Kozik e Piotr Micek, em [37], perceberam que essa abordagem poderia ser aplicada diretamente nos problemas obtendo dessa forma novos resultados em sequências não repetitivas usando um argumento de compressão de entropia. Também em 2013, em [29], Louis Esperet e Aline Parreau sistematizaram a abordagem proposta em [37] para um determinado conjunto de problemas. Muitos resultados foram obtidos dessa forma como em [13, 14, 24, 29, 35, 55, 56]. O método pode ser resumido da seguinte forma: desenvolve-se um algoritmo que constrói o objeto passo a passo aleatoriamente. Se em um determinado passo a construção produz uma característica que o objeto não possui então esse passo é desfeito juntamente com alguns mais. A condição para o algoritmo parar é achar um objeto com as características desejadas. A ideia então é usar um argumento de compressão de entropia para provar que o algoritmo pára. Estudamos o artigo [29] no capítulo 3 juntamente com o aparato teórico que o sustenta. Na seção 3.2.3 provamos que essa abordagem pode ser aplicada em todos os problemas que podem ser atacados utilizando-se varia-

[1] Esse nome é devido a Kolipaka e Szegedy por [44].
[2] Esse nome é em devido a Tao por [64].

ble version do LLL. Como aplicação, trabalhamos no problema de existência de colorações legítimas em planos projetivos finitos proposto por Noga Alon e Zoltan Füredi em [3] resultando em [10]. O artigo de Alon e Füredi é exposto no capítulo 2 e os resultados que obtivemos são expostos no capítulo 4.

Em tempo, no desenvolvimento desse trabalho teve-se como norte escrever um texto, se não totalmente, o mais autocontido possível. Dessa forma nos apêndices A, B e C o leitor encontrará o conteúdo necessário de planos projetivos finitos, funções geradoras e entropia de Shannon respectivamente.

Capítulo 1

O Lema Local de Lovász

"This course could be called The Probabilistic Method, The Erdős Method or, my favorite, Erdős Magic" - Joel Spencer

A frase acima, introduzindo o curso de Grafos Aleatórios na New York University em 2012, descreve bem o atual status que o Método Probabilístico obteve dentro da teoria combinatória. A palavra *magic* não é usada na frase somente por ter sido Paul Erdős quem popularizou a abordagem, mas pelo poder que este conjunto de técnicas e ideias acabou exibindo quando aplicado aos mais variados problemas.

Por vezes, a probabilidade de um objeto escolhido aleatoriamente ter as propriedades desejadas é razoavelmente alta e então muitas abordagens dentro do método probabilístico obtém êxito. Ferramentas mais sofisticadas, tais como o Lema Local de Lovász (LLL), nos permite provar a existência de objetos que ocorrem com probabilidades muito pequenas. Com o passar dos anos, versões do LLL mais poderosas e/ou mais adequadas a determinados problemas foram desenvolvidas. Nesse capítulo estudamos diferentes enunciados do Lema Local de Lovász (seção 1.1) e ganhamos familiaridade com algumas técnicas para aplicá-los (seção 1.2).

1.1 Diferentes Versões do Lema Local de Lovász

Para provar a existência de um objeto matemático, o LLL o define pelo seu complementar. Ao invés de definir as características de objetos de determinada classe, define uma coleção $(A_x)_{x\in X}$ de *eventos ruins* – num espaço de probabilidade $(\Omega, \mathcal{A}, P)$ – que englobam as características cuja negação

é condição necessária e suficiente para pertencer à classe buscada. Dentro da filosofia do Método Probabilístico, quer-se descobrir condições, as mais gerais possíveis, de modo a garantir que com probabilidade positiva nenhum destes eventos ruins ocorre. Em outras palavras, busca-se o mínimo de restrições possíveis para que

$$\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) > 0.$$

Daqui em diante, X denotará um conjunto finito e G um grafo tal que $V(G) = X$.

Exemplo 1.1. Seja $(A_x)_{x\in X}$ uma família de eventos independentes em $(\Omega, \mathcal{A}, P)$. Basta exigir que $P(A_x) = p_x < 1, \forall x \in X$, pois nesse caso:

$$\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) = \prod_{x\in X} \mathbf{P}(\overline{A_x}) = \prod_{x\in X} (1 - p_x) > 0.$$

□

Exemplo 1.2. Seja $(A_x)_{x\in X}$ uma família de eventos em $(\Omega, \mathcal{A}, P)$ tal que $\mathbf{P}(A_x) = p_x$ sobre a qual não se sabe quais são independentes. Como $\mathbf{P}(\bigcup_{x\in X} A_x) \leq \sum_{x\in X} \mathbf{P}(A_x) = \sum_{x\in X} p_x$, impondo que $\sum_{x\in X} p_x < 1$ obtemos o resultado. De fato

$$\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) = \mathbf{P}(\overline{\bigcup_{x\in X} A_x}) = 1 - \mathbf{P}(\bigcup_{x\in X} A_x) \geq 1 - \sum_{x\in X} p_x > 0.$$

□

O último exemplo não levou em consideração a dependência entre os eventos. A ideia então é que, caso seja possível usar esta informação, obteremos maiores valores para as probabilidades p_x de forma a ainda a garantir $\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) > 0$. O primeiro resultado nesta direção foi obtido em [27] por Paul Erdős e László Lovász em 1973, o qual continha a primeira versão do então chamado Lema Local de Lovász. Antes de enunciar o teorema, precisamos da

Definição 1.3. Dizemos que G é um grafo de dependência para a família de eventos $(A_x)_{x\in X}$ em um espaço de probabilidade quando, para cada $x \in X$, A_x é mutuamente independente da família $\{A_y : y \in X \setminus \Gamma^*(x)\}$, onde $\Gamma^*(x)$ se refere à vizinhança de x unida com x.

É importante ressaltar que o grafo de dependência não é único – basta observar que dado um grafo de dependência G para uma família de eventos

$(A_x)_{x\in X}$ qualquer outro grafo obtido a partir deste adicionando elos a G será um grafo de dependência para a família. O resultado obtido por Erdős e Lovász pode então ser enunciado da seguinte forma:

Teorema 1.4 (Erdős e Lovász [27]). *Seja G um grafo de dependência para uma família de eventos $(A_x)_{x\in X}$ com grau máximo Δ e $\mathbf{P}(A_x) = p_x$. se $4p_x\Delta \leq 1$, para todo $x \in X$. Então $\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) > 0$.*

Uma versão mais geral foi obtida por Joel Spencer em 1977 em [62]:

Teorema 1.5 (Spencer [62]). *Seja G um grafo de dependência para a família de eventos $(A_x)_{x\in X}$ e suponha que existam $(r_x)_{x\in X}$ números em [0,1) tais que, para cada $x \in X$,*

$$\mathbf{P}(A_x) = p_x \leq r_x \prod_{y\in\Gamma(x)} (1 - r_y).$$

Então $\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) \geq \prod_{x\in X}(1 - r_x) > 0.$

O artigo [3] de Noga Alon e Zoltan Füredi mencionado usa a seguinte versão do LLL, cuja prova pode ser encontrada em [9]:

Teorema 1.6 (Lema Local Lovász - caso simétrico). *Seja G um grafo de dependência para uma família de eventos $(A_i)_{i\in[n]}$ com grau máximo Δ e $\mathbf{P}(A_i) = p_i$. Suponha $p(\Delta + 1)e \leq 1$ e considere $p = sup_{i\in[n]}p_i$. Então $\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) > 0$.*

Pela praticidade na verificação das hipóteses, esta versão do teorema é muitas vezes preferida pelos autores. Neste caso não é necessário ter uma informação mais refinada a respeito da estrutura do grafo de dependência, apenas o valor da probabilidade dos eventos (ou o supremo destas) e o grau máximo do grafo de dependência.

O próximo resultado a ser apresentado é o Lopsided Lema Local de Lovász que leva em consideração que os eventos ruins não precisam ser independentes. No que concerne ao LLLL, eventos positivamente correlacionados (em uma certa maneira) podem ser considerados como eventos independentes. Isso possibilita a construção de um "grafo de dependência" com um grau menor para a mesma família de eventos.

Definição 1.7. Dizemos que G é um grafo de dependência assimétrico[1] para a família de eventos $(A_x)_{x\in X}$ em um espaço de probabilidade quando, para cada $x \in X$,

$$\mathbf{P}(A_x| \bigcap_{y\in Y} A_y) \leq \mathbf{P}(A_x)$$

para todo $Y \subseteq X \setminus \Gamma^*(x)$.

Este fato foi observado por Spencer e Erdős em 1991 em [28], onde perceberam a importância de trabalhar com grafos mais gerais. O novo conceito foi usado para uma nova aplicação do Lema Local de Lovász no estudo dos Latin Transversals. Estudaremos essa aplicação na próxima seção. O resultado obtido por Spencer e Erdős é o seguinte:

Teorema 1.8 (Spencer e Erdős [28])**.** *Suponha que G é um grafo de dependência assimétrico para a família de eventos $(A_x)_{x\in X}$ e suponha que existam $(r_x)_{x\in X}$ números em $[0, 1)$ tais que, para cada x,*

$$\boldsymbol{P}(A_x) = p_x \leq r_x \prod_{y\in\Gamma(x)} (1 - r_y).$$

Então $\boldsymbol{P}(\bigcap_{x\in X} \overline{A_x}) \geq \prod_{x\in X}(1 - r_x) > 0.$

A seguir, apresentamos a versão do lema fornecida por Bissacot *et al.* em [11] em 2011. Recentemente, o LLL foi relacionado a um resultado da mecânica estatística. Essa ligação surpreendente foi apontada por Scott e Sokal em [59] em 2005. Nesse artigo, Scott e Sokal concluíram que o LLL é uma reformulação do critério de Dobrushin ([22]) para a convergência da pressão da gás de caroço duro em G. Em 2007, Fernández e Procacci, em [31], forneceram um novo critério para a convergência da pressão do gás de caroço duro em G, e mostraram que esse novo critério é sempre mais eficiente que o critério de Dobrushin. Bissacot *et al.*, em [11], usaram o critério de Fernández-Procacci e os resultados obtidos em [59] para melhorar o Lema Local de Lovász. Essa versão do LLL foi usada, por exemplo, para melhorar o resultado mencionado anteriormente em Latin Transversal ([11]) e para obter novos resultados sobre colorações dos elos do grafo completo K_n ([15]). Cabe mencionar que Pegden, em [54], mostrou que o algoritmo de Moser-Tardos ([51]) vale para esse novo lema. Para enunciar o lema, precisamos da seguinte nomenclatura:

[1] A nomenclatura usual em inglês para este tipo de grafo é *lopsidependency graph*.

Sejam X um conjunto finito, $\{\mu_x\}_{x\in X}$ uma família de números não negativos e G um grafo tal que $V(G) = X$. Para cada $x \in X$ definimos a partição restrita aos vizinhos de x por:

$$Z_{\Gamma^*(x)}(\mu) = \sum_{\substack{T\in I(G)\\ T\subset\Gamma^*_G(x)}} \prod_{x\in T} \mu_x,$$

onde $I(G)$ denota a família de subconjuntos independentes em relação ao grafo G do conjunto $X = V(G)$. Definimos também a seguinte função auxiliar:

$$Z_{\Gamma(x)}(\mu) = \sum_{\substack{T\in I(G)\\ T\subset\Gamma_G(x)}} \prod_{x\in T} \mu_x.$$

Teorema 1.9 (Bissacot *et al.* [11])**.** *Suponha que G é um grafo de dependência para a família de eventos $(A_x)_{x\in X}$ em um espaço de probabilidade $(\Omega, \mathcal{A}, P)$ e que existem $\mu = \{\mu_x\}_{x\in X}$ números reais em $[0,\infty)$ tais que, para cada $x \in X$, $\mathbf{P}(A_x) = p_x$ satisfaz*

$$p_x \leq \frac{\mu(x)}{Z_{\Gamma^*(x)}(\mu)}.$$

Então $\mathbf{P}(\bigcap_{x\in X} \overline{A_x}) \geq \prod_{x\in X}(1-p_x)^{Z_{\Gamma(x)}(\mu)} > 0.$

Agora, reenunciaremos o Teorema 1.5 para esclarecer a relações entre essas duas versões do LLL.

Teorema 1.10. *Seja H um grafo de dependência para a família de eventos $\{A_x\}_{x\in X}$ cada um com $\boldsymbol{P}(A_x) = p_x$. Suponha que existam $\{\mu_x\}_{x\in X}$ números reais em $[0,\infty)$ tais que, para cada $x \in X$,*

$$p_x \leq \frac{\mu_x}{\phi_x(\mu)}, \quad \textit{onde} \quad \phi_x(\mu) = \sum_{R\subseteq\Gamma^*_H(x)} \prod_{x\in R} \mu_x.$$

Então $\boldsymbol{P}(\bigcap_{x\in X} \overline{A}_x) > 0.$

A única diferença entre os dois teoremas acima é que $\phi_x(\mu)$ soma sobre todos os subconjuntos de $\Gamma^*_H(x)$ enquanto $Z_{\Gamma(x)}(\mu)$ soma apenas sobre os subconjuntos independentes de $\Gamma^*_H(x)$. Disso resulta que

$$Z_{\Gamma(x)}(\mu) \leq \phi_x(\mu).$$

Como $\phi_x(\mu)$ não leva em consideração a estrutura de $\Gamma^*_H(x)$, o Teorema 1.10 fornece o mesmo resultado se $\Gamma^*_H(x)$ for um conjunto independente

ou um clique. Já o Teorema 1.9 fornece sua maior melhora se $\Gamma^*_H(x)$ for um clique e praticamente não fornece melhora se $\Gamma^*_H(x)$ for um conjunto independente.

A desigualdade a seguir é de grande utilidade quando pretende-se aplicar o Teorema 1.9: seja x um vértice do grafo de dependência H para a família de eventos $\{A_x\}_{x\in X}$ e suponha que $\Gamma^*_H(x)$ é a união de $c_1, \ldots, c_k$ cliques então, pela definição de $Z_{\Gamma(x)}(\mu)$,

$$Z_{\Gamma(x)}(\mu) \leq \prod_{i=1}^{k}(1 + \sum_{y\in c_i} \mu_y).$$

Assim como anteriormente, o seguinte resultado também é válido.

Teorema 1.11. *Seja G um grafo de dependência assimétrico para a família de eventos $(A_x)_{x\in X}$. Suponha que existam $\mu = \{\mu_x\}_{x\in X}$ números reais em $[0, \infty)$ tais que, para cada $x \in X$ temos que*

$$\boldsymbol{P}(A_x) = p_x \leq \frac{\mu_x}{Z_{\Gamma(x)}(\mu)}.$$

Então $\boldsymbol{P}(\bigcap_{x\in X} \overline{A}_x) > 0$.

1.2 Aplicações do Lema Local de Lovász

O objetivo dessa seção é fornecer ao leitor alguma familiaridade com as diferentes formas de se aplicar o LLL. Para tanto, selecionamos problemas que requerem o Lema em suas diferentes versões e organizamos os problemas em uma ordem didática, i.e., as proposições evoluem em nível de dificuldade e quando uma proposição usa uma versão mais antiga do LLL a proposição seguinte mostrará como o mesmo problema pode ser atacado com uma versão mais atual, obtendo assim melhores resultados[2]. Vamos a elas:

Proposição 1.12. *Sejam $G = (V, E)$ um grafo com grau máximo Δ e $V = V_1 \cup V_2 \cup \ldots \cup V_n$ uma partição de V. Suponha ainda que para cada conjunto V_i temos $|V_i| \geq 2e\Delta$, para todo $1 \leq i \leq n$. Então existe um conjunto independente $W \subseteq V$ (i.e. não existem $a, b \in W$ tais que $(a, b) \in E$) de cardinalidade n que contém exatamente um vértice de cada V_i.*

Prova: Aplicaremos o caso simétrico do LLL: seja $k = min\{|V_i| : 0 \leq i \leq n\}$. Assumiremos que $|V_i| = k$ para todo i (o caso geral segue deste

[2]Pode parecer então que não vale a pena investir tempo aprendendo a usar uma versão antiga do LLL. Acontece que existem problemas que não conseguem ser resolvidos com versões mais atuais mas aceitam versões mais antigas.

considerando a união de n subconjuntos de cardinalidade k onde cada um deles é subconjunto de algum V_i). Sejam, para cada i, ψ_i variáveis aleatórias uniformes independentes cujos domínios são os respectivos V_i. Seja W o conjunto formado pela união dos elementos gerados por uma amostragem de cada ψ_i.

Para cada elo $ab \in E$, seja W_{ab} o evento "W contém os vértices a e b". Logo $P(W_{ab}) = 0$, se a e b são elementos do mesmo V_i, e $P(W_{ab}) = 1/k^2 = p$, se $a \in V_i$ e $b \in V_j$, com $i \neq j$. Note que, se os vértices a e b pertencem a $V_i \cup V_j$, temos que o evento W_{ab} é mutuamente independente da família de eventos composta por todos os eventos W_{cd} tais que nem c e nem d pertencem a $V_i \cup V_j$. Logo, como eventos de probabilidade zero são independentes em relação a quaisquer outros eventos, podemos construir um grafo de dependência $H = (\{ab\}_{ab\in E(G)}, E(H))$ para a família de eventos $(W_{ab})_{ab\in E(G)}$ com grau máximo $2k\Delta - 1$. Com efeito, sejam $a \in V_i$ e $b \in V_j$, com $i \neq j$. Como existem k vértices em V_i e cada um deles tem grau máximo Δ então existem no máximo $k\Delta$ eventos W_{cd} com $c \in V_i$. Da mesma forma existem no máximo $k\Delta$ eventos W_{cd} com $c \in V_j$. Logo, desconsiderando o próprio elo ab, a afirmação segue. Definamos H como o grafo de dependência para a família $(W_{ab})_{ab\in E(G)}$ com menor número de elos possível (i.e. existirá um elo de H entre dois elos ab e cd se, e somente se, W_{ab} e W_{cd} forem dependentes).

Seja $\{a, b\} \subset V_i \cup V_j$. Pela definição do grafo H temos que se $cd \in \Gamma_H(ab)$, onde $\Gamma_H(ab)$ é a vizinhança do elo ab no grafo H, então $c \in V_i \cup V_j$ ou $d \in V_i \cup V_j$. Logo $|\Gamma_H(ab)| \leq 2k\Delta - 1$.

Pelo Lema Local de Lovász (caso simétrico), se $p(2k\Delta-1+1)e \leq 1$ então

$$P(\cap_{ab\in E(G)}\overline{W}_{ab}) > 0.$$

A proposição segue pois, como $p = 1/k^2$ e $k = |V_i|$, $p(2k\Delta - 1 + 1)e \leq 1$ se, e somente se $|V_i| \geq 2e\Delta$. ■

Conseguiremos agora uma melhora do resultado anterior aplicando uma versão mais refinada do LLL:

Proposição 1.13. *Sejam $G = (V, E)$ um grafo com grau máximo Δ e $V = V_1 \cup V_2 \cup \ldots \cup V_n$ uma partição de V. Suponha ainda que para cada conjunto V_i temos $|V_i| \geq 4\Delta$, para todo $1 \leq i \leq n$. Então existe um conjunto independente $W \subseteq V$ de cardinalidade n que contém exatamente um vértice de cada V_i.*

Prova: Seja H o grafo de dependência construído na proposição 1.2. Para obter essa melhora em relação à proposição 1.12, aplicaremos o teorema 1.9 usando o grafo de dependência H. Para tanto, precisamos encontrar

uma cota superior para $Z_{\Gamma^*(ab)}(\mu)$, onde ab é um elo arbitrário de G. A ideia é usar números $\mu = (\mu_{ab})_{ab \in E(G)}$ com $\mu_{ab} = \mu$ para um mesmo $\mu > 0$. Dessa forma será fácil fatorar $Z_{\Gamma^*(ab)}(\mu)$ para posterior análise. Lembrando que, dado um vértice v de um grafo de dependência, o teorema 1.9 usa como informação os conjuntos independentes formados por eventos em $\Gamma(v)$, precisamos de uma cota superior para o número de pares (pois trincas não são possíveis) de eventos $\{W_{cd}, W_{fg}\}$ independentes (em H) tais que cd e fg sejam adjacentes a ab no grafo H. Claramente o número de pares não excede $k^2\Delta^2$, pois temos no máximo $k\Delta$ eventos W_{cd} tais que cd é adjacente a ab onde ou c ou d não pertence a $V_i \cup V_j$. Como mencionado, não existem trincas $\{W_{cd}, W_{ef}, W_{gh}\}$ de eventos cujos respectivos elos $\{cd, ef, gh\}$ sejam adjacentes a ab e independentes dois a dois em H. De fato, sejam V_i e V_j tais que $\{a, b\} \subset V_i \cup V_j$. Se $cd \in \Gamma_H(ab)$, então $c \in V_i \cup V_j$ ou $d \in V_i \cup V_j$. O mesmo vale para ef e gh. Assim, cada um dos eventos do conjunto $\{W_{cd}, W_{ef}, W_{gh}\}$ deveria ter seus índices com pelo menos um vértice em $V_i \cup V_j$ e os eventos deveriam ser independentes dois a dois, o que é impossível. Assim:

$$Z_{\Gamma^*(ab)}(\mu) \leq 1 + 2k\Delta\mu + k^2\Delta^2\mu^2 = (1 + k\Delta\mu)^2.$$

Seja

$$f(\mu) = \frac{\mu}{Z_{\Gamma^*(ab)}(\mu)} \geq \frac{\mu}{(1 + k\Delta\mu)^2}.$$

Como o lado direito da desigualdade acima assume seu valor máximo em $\mu_0 = \frac{1}{k\Delta}$, pelo teorema 1.9, se $p \leq \frac{1}{4k\Delta} \leq f(\mu_0)$ então a probabilidade de nenhum dos eventos da família $(W_{ab})_{ab \in E(G)}$ ocorrer é positiva. Isso conclui a prova, pois $p = 1/k^2$ e então $p \leq 1/4k\Delta$ se, e somente se, $k \geq 4\Delta$. ■

Sobre este problema, cabe citar o fato de que é conhecida sua constante ótima. Já provamos que o resultado funciona para $|V_i| \geq c\Delta$ para $c = 2e$ e para $c = 4$ (em particular para qualquer valor maior que este). Já se sabia que c não poderia ser menor que 2 através de contra-exemplos. Em [57] Bruce Reed conjecturou que 2 era o menor valor de c. A conjectura foi provada por Haxell em [40] em 2001 usando uma abordagem alheia ao método probabilístico.

Definição 1.14. Seja $(a_{ij}) \in M_n(\mathbb{Z})$. Uma permutação $\sigma\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$ é chamada Latin Transversal de (a_{ij}) se as entradas $a_{i\sigma(i)}$ são todas distintas para $i \in \{1, \ldots, n\}$.

Proposição 1.15. *Sejam $(a_{ij}) \in M_n(\mathbb{Z})$ e $k \leq (n-1)/4e$. Suponha que nenhum inteiro aparece em mais do que k entradas de (a_{ij}). Então (a_{ij}) tem uma Latin Transversal.*

Prova: Nesse problema usaremos o teorema 1.8. Precisamos, portanto, como em todas as aplicações do LLL, criar um espaço de probabilidade apropriado e reconhecer quais são nossos eventos ruins (os quais, se forem todos evitados, teremos a situação desejada). Para tanto, considere o espaço de probabilidade $(\Omega, \mathcal{P}(\Omega), \mathbf{P})$, onde Ω é o conjunto de todas as permutações de $\{1, \ldots, n\}$ em $\{1, \ldots, n\}$, $\mathcal{P}(\Omega)$ é o conjunto das partes de Ω e $\mathbf{P}$ é a medida de probabilidade homogênea em Ω. Seja T o conjunto das quatro-uplas ordenadas $(i, j, i^{'}, j^{'})$ tais que $i < i^{'}$, $j \neq j^{'}$ e $a_{ij} = a_{i^{'}j^{'}}$. Para cada $(i, j, i^{'}, j^{'}) \in T$, seja $A_{iji^{'}j^{'}}$ o evento onde $\sigma(i) = j$ e $\sigma(i^{'}) = j^{'}$. Claramente esses são os nossos eventos ruins. Nosso objetivo é mostrar que

$$\mathbf{P}(\bigcap_{i<i^{'}, j\neq j^{'}} \overline{A}_{iji^{'}j^{'}}) > 0.$$

Comecemos pela probabilidade dos eventos ruins:

$$\mathbf{P}(A_{iji^{'}j^{'}}) = \mathbf{P}(\text{“}\sigma(i) = j\text{”} \cap \text{“}\sigma(i^{'}) = j^{'}\text{”}) =$$

$$= \mathbf{P}(\text{“}\sigma(i) = j\text{”})\mathbf{P}(\text{“}\sigma(i^{'}) = j^{'}\text{”}|\text{“}\sigma(i) = j\text{”}) = \frac{1}{n} \cdot \frac{1}{n-1} = \frac{1}{n(n-1)}.$$

Agora, seja G o grafo cujo conjunto de vértices é T onde dois vértices $(i, j, i^{'}, j^{'})$ e $(p, q, p^{'}, q^{'})$ são adjacentes se, e somente se, $\{i, i^{'}\} \cap \{p, p^{'}\} \neq \emptyset$ ou $\{j, j^{'}\} \cap \{q, q^{'}\} \neq \emptyset$. G é um grafo de dependência assimétrico para a família de eventos $A_{iji^{'}j^{'}}$. Com efeito, seja $A_{iji^{'}j^{'}}$ fixado e B o evento “*existe* $\{x, x^{'}\}$ tal que $\{x, x^{'}\} \cap \{i, i^{'}\} = \emptyset$ e $\sigma(x) = j$ ou $\sigma(x^{'}) = j^{'}$”. Seja Y o conjunto dos membros de T que não são adjacentes em G a $(i, j, i^{'}, j^{'})$. Logo, se $(p, q, p^{'}, q^{'}) \in Y$ então $\overline{A}_{pqp^{'}q^{'}}$ é o conjunto das permutações onde $\sigma(p) \neq q$ ou $\sigma(p^{'}) \neq q^{'}$. Logo, dado que $\overline{A}_{pqp^{'}q^{'}}$ ocorre, aumenta a probabilidade de que $\sigma(p)$ ou $\sigma(p^{'})$ pertença a $\{j, j^{'}\}$. Logo

$$\mathbf{P}(A_{iji^{'}j^{'}}|\overline{A}_{pqp^{'}q^{'}}) = \mathbf{P}(A_{iji^{'}j^{'}} \cap B|\overline{A}_{pqp^{'}q^{'}}) + \mathbf{P}(A_{iji^{'}j^{'}} \cap \overline{B}|\overline{A}_{pqp^{'}q^{'}}) =$$

$$= 0 + \mathbf{P}(A_{iji^{'}j^{'}} \cap \overline{B}|\overline{A}_{pqp^{'}q^{'}}) = \mathbf{P}(\overline{B}|\overline{A}_{pqp^{'}q^{'}}) \cdot \mathbf{P}(A_{iji^{'}j^{'}}|\overline{B} \cap \overline{A}_{pqp^{'}q^{'}}).$$

Mas, pelo que foi discutido acima, temos que

$$\mathbf{P}(\overline{B}|\overline{A}_{pqp^{'}q^{'}}) < \mathbf{P}(\overline{B}) \quad \text{e} \quad \mathbf{P}(A_{iji^{'}j^{'}}|\overline{B} \cap \overline{A}_{pqp^{'}q^{'}}) = \mathbf{P}(A_{iji^{'}j^{'}}|\overline{B}).$$

Como

$$\mathbf{P}(A_{iji^{'}j^{'}}) = \mathbf{P}(A_{iji^{'}j^{'}} \cap B) + \mathbf{P}(A_{iji^{'}j^{'}} \cap \overline{B}) =$$

$$= 0 + \mathbf{P}(A_{iji'j'} \cap \overline{B}) = \mathbf{P}(\overline{B})\mathbf{P}(A_{iji'j'}|\overline{B})$$

temos então que

$$\mathbf{P}(A_{iji'j'}|\overline{A}_{pqp'q'}) \leq \mathbf{P}(A_{iji'j'}).$$

Pela extensão desse raciocínio temos que

$$\mathbf{P}(A_{iji'j'}| \bigcap_{(p,q,p',q')\in Y} \overline{A}_{pqp'q'}) \leq \mathbf{P}(A_{iji'j'})$$

provando então que G é um grafo de dependência assimétrico. Ainda, G tem grau máximo $4n(k-1)$. Com efeito, para $(i, j, i^{'}, j^{'})$ fixo, podemos escolher (s, t) de $4n$ maneiras diferentes com $s \in \{i, i^{'}\}$ ou $t \in \{j, j^{'}\}$. Uma vez escolhido (s, t) , temos menos do que k escolhas para $(s^{'}, t^{'})$ diferentes de (s, t), tal que $a_{st} = a_{s't'}$, pois, por hipótese, existem no máximo k entradas de (a_{ij}) com o mesmo valor. Assim, temos menos do que $4nk$ quatro-uplas $(s, t, s^{'}, t^{'})$ tais que $s = i$ ou $s = i^{'}$ ou $t = j$ ou $t = j^{'}$, com $a_{st} = a_{s't'}$. Agora, para cada uma das quatro-uplas $(s, t, s^{'}, t^{'})$, podemos associar as quatro-uplas $(p, q, p^{'}, q^{'}) = (s, t, s^{'}, t^{'})$ se $s < s^{'}$ ou as quatro-uplas $(p, q, p^{'}, q^{'}) = (s^{'}, t^{'}, s, t)$ se $s^{'} < s$, o que completa a prova, pois a condição $p(\Delta + 1)e \leq 1$ exigida pelo caso simétrico do teorema 1.8 é equivalente a $k \leq (n-1)/4e$. ■

Proposição 1.16. *Sejam $(a_{ij}) \in M_n(\mathbb{Z})$ e $k \leq (n-1)/(256/27)$. Suponha que nenhum inteiro aparece em mas do que k entradas de (a_{ij}). Então (a_{ij}) tem uma Latin Transversal.*

Prova: Conseguiremos essa melhora em relação à proposição 1.15 aplicando o teorema 1.11. Seja G o grafo definido na proposição 1.15. Como na proposição 1.4, para facilitar a fatoração de $Z_{\Gamma^*(iji'j')}(\mu)$, usaremos $\mu = (\mu_{iji'j'})_{iji'j'\in V(G)}$ com $\mu_{iji'j'} = \mu$ para um mesmo $\mu > 0$.

O próximo passo é dar uma cota superior para $Z_{\Gamma^*(iji'j')}(\mu)$. Observe que, pela construção de G, para um vértice $iji^{'}j^{'}$ fixo, temos que $\Gamma^*(iji^{'}j^{'})$ é a união de 4 subconjuntos, cada um com cardinalidade máxima nk tais que todos os vértices em cada um desses quatro subconjuntos são adjacentes. Logo um conjunto independente de vértices de $\Gamma^*(iji^{'}j^{'})$ tem cardinalidade menor ou igual a 4. Ainda,

• Existem no máximo $4nk$ (a cardinalidade máxima de $\Gamma^*(iji^{'}j^{'})$) conjuntos independentes de vértices de $\Gamma^*(iji^{'}j^{'})$ com cardinalidade 1.

• Existem no máximo $\binom{4}{2}(nk)^2$ conjuntos independentes de vértices de $\Gamma^*(iji^{'}j^{'})$ com cardinalidade 2. Com efeito, os vértices deverão ser escolhidos de dois subconjuntos diferentes dos quatro subconjuntos que compõe

$\Gamma^*(iji'j')$. Feito isso, existem nk formas de se escolher um vértice em um subconjunto e outras nk formas de se escolher um vértice no outro subconjunto.

• Existem no máximo $\binom{4}{3}(nk)^3$ conjuntos independentes de vértices de $\Gamma^*(iji'j')$ com cardinalidade 3. A prova é muito similar a do caso anterior.

• Existem no máximo $\binom{4}{4}(nk)^4$ conjuntos independentes de vértices de $\Gamma^*(iji'j')$ com cardinalidade 4. A prova é muito similar a do caso anterior.

Logo

$$Z_{\Gamma^*(iji'j')}(\mu) = \sum_{\substack{H\subset\Gamma^*(x)\\ H\in I(G)}} \prod_{x\in H} \mu_x \leq (1+nk\mu)^4$$

e

$$\frac{\mu}{Z_{\Gamma^*(iji'j')}(\mu)} \geq \frac{\mu}{(1+nk\mu)^4} \equiv f(\mu).$$

Como o lado direito assume seu valor máximo em $\mu_0 = 1/3nk$, podemos usar o Teorema 1.11 na região $p = 1/n(n-1) \leq 27/256nk = f(\mu_0)$, o que é equivalente a dizer que $k \leq (n-1)/(256/27)$. ■

A seguir, definiremos o problema de coloração acíclica de elos e apresentaremos dois resultados obtidos pelo LLL. No capítulo 3, mostramos como Louis Esperet e Aline Parreau, em [29], melhoraram esses resultados usando outra abordagem.

Definição 1.17. Uma λ-coloração dos elos de um grafo $G = (V, E)$ é uma função $f\colon E \to \{1, 2, \ldots, \lambda\}$. Uma λ-coloração é dita acíclica se dois elos adjacentes não possuem a mesma cor e se todo ciclo tem pelo menos três cores.

Definição 1.18. *a'*(G) é o menor número de cores que podem formar uma coloração acíclica de G.

Proposição 1.19. *Seja $G = (V, E)$ um grafo com grau máximo $\Delta \geq 3$. Então a'(G) $\leq 16\Delta$.*

Prova: Nesse proposição aplicaremos o teorema 1.5: sejam T o conjunto dos pares $\{ab, bc\}$, onde $a, b, c \in V$ e $ab, bc \in E$, e, para $k \geq 2$, $C_{2k}(G)$ o

conjuntos de todos os ciclos de G de tamanho $2k$. Seja $X = T \cup (\cup_{k\geq 2} C_{2k})$. Considerando os elementos de C_{2k} como o conjunto de elos que formam cada ciclo temos que os elementos de X são subconjuntos de E. Seja $N = 16\Delta$ o número de cores que usaremos para colorir o grafo. Seguindo a mesma linha das proposições anteriores, definiremos um espaço de probabilidade (dessa vez de forma implícita) e reconheceremos os eventos ruins.

Para cada elo de G, associe uma cor escolhida uniformemente dentre as N cores disponíveis. Para cada $\{e_i, e_j\} \in T$ seja $A_{\{e_i,e_j\}}$ o evento em que e_i e e_j recebem a mesma cor. Para cada $c_{2k} \in C_{2k}$, seja $A_{c_{2k}}$ o evento em que o ciclo c_{2k} recebe apenas duas cores sem que dois elos adjacentes recebam a mesma cor. Repare que estamos considerando apenas os ciclos de tamanho par. Se um ciclo de tamanho ímpar for colorido com apenas duas cores obrigatoriamente teremos dois elos adjacentes com a mesma cor. Claramente se todos os eventos descritos acima forem evitados então teremos uma coloração acíclica dos elos de G.

Como antes, investiguemos a probabilidade de tais eventos ocorrerem: sejam $e_i, e_j \in T$. Se uma cor for atribuída ao elo e_i, a probabilidade de a mesma cor ser atribuída ao elo e_j é igual a $1/N$. Logo

$$\mathbf{P}(A_{\{e_i,e_j\}}) = \frac{1}{N}.$$

Agora, seja $e_1, \ldots, e_{2k}, e_1 \in C_{2k}$. Se uma cor é atribuída a e_1 e uma cor diferente é atribuída a e_2 há apenas uma maneira de se estender essa coloração de forma a construir um *evento ruim* $A_{c_{2k}}$. Como as escolhas de cores são independentes, tal extensão tem probabilidade $1/N^{2k-2}$ de ocorrer. Logo

$$\mathbf{P}(A_{c_{2k}}) \leq \frac{1}{N^{2k-2}}.$$

O próximo passo é criar um grafo de dependência para a família de eventos ruins. Como cada evento ruim está associado a um e apenas um elemento de X, construiremos um grafo onde os vértices são os elementos de X. Como as escolhas de cores são independentes, é claro que o evento $A_{\{e_i,e_j\}}$ é independente do evento $A_{\{e_r,e_s\}}$ se, e somente se, $\{e_i, e_j\} \cap \{e_r, e_s\} = \emptyset$. Assim como $A_{\{e_i,e_j\}}$ é independente de $A_{c_{2k}}$ se, e somente se, $\{e_i, e_j\} \cap c_{2k} = \emptyset$. Da mesma forma, o evento $A_{c_{2k}}$ é independente dos eventos $A_{\{e_i,e_j\}}$ e $A_{c_{2m}}$ se, e somente se, $c_{2k} \cap \{e_i, e_j\} = \emptyset$ e $c_{2k} \cap c_{2m} = \emptyset$, respectivamente. Logo, seja $H(X, F)$ o grafo de dependência para a família de eventos ruins onde $\{x, x^{'}\} \in F$ se, somente se, $x \cap x^{'} \neq \emptyset$.

Observe que cada elo e está contido em no máximo $2(\Delta - 1)$ pares $\{e_i, e_j\} \in T$ e no máximo $(\Delta - 1)^{2k-2}$ ciclos $c_{2k} \in C_{2k}$, para todo $k \geq 2$. Logo

1) para cada vértice $x = \{e_i, e_j\} \in T$ de H, $\Gamma^*_H(x)$ é a união de dois conjuntos $\Gamma^*_1(x)$ e $\Gamma^*_2(x)$ tais que, para $i \in \{1, 2\}$

$$|\Gamma^*_i(x)| \leq 2(\Delta - 1) + \sum_{k \geq 2} (\Delta - 1)^{2k-2}$$

e cada elemento z de $\Gamma^*_i(x)$ contém e_i logo o subgrafo de H induzido por $\Gamma^*_i(x)$ é um clique (esse fato só será aproveitado na próxima proposição).

2) para $k \geq 2$ e para cada vértice $y = c_{2k} = \{e_1, \ldots, e_{2k}, e_1\} \in C_{2k}$ de H, $\Gamma^*_H(y)$ é a união de $2k$ conjuntos $\Gamma^*_1(y), \ldots, \Gamma^*_{2k}(y)$ tais que, para $j \in \{1, \ldots, 2k\}$,

$$|\Gamma^*_j(y)| \leq 2(\Delta - 1) + \sum_{s \geq 2} (\Delta - 1)^{2s-2}$$

e cada elemento z de $\Gamma^*_j(y)$ contém e_j logo o subgrafo de H induzido por $\Gamma^*_j(y)$ é um clique (de novo, esse fato só será aproveitado na próxima proposição).

Dessa vez, não conseguiremos usar o caso simétrico do Lema de Lovász. Sejam, para cada $\{e_i, e_j\} \in T$ e cada $c_{2k} \in C_{2k}$, para $k \geq 2$, $r_{\{e_i,e_j\}} = 2/N$ e $r_{c_{2k}} = (2/N)^{2k-2}$. Logo a condição do teorema 1.5 está satisfeita. Com efeito, para um $\overline{c}_{2k} \in C_{2k}$, consideremos $A_{\overline{c}_{2k}}$:

$$r_{\overline{c}_{2k}} \prod_{A_j \in \Gamma(A_{\overline{c}_{2k}})} (1 - r_j) \geq (\frac{2}{N})^{2k-2} \cdot (1 - \frac{2}{N})^{4k\Delta} \cdot \prod_{l \geq 2} (1 - (\frac{2}{N})^{2l-2})^{2k\Delta^{2l-2}}.$$

Usando o fato de que, para $z \geq 2$, $(1 - 1/z)^z \geq \frac{1}{4}$, temos que, lembrando que $N = 16\Delta$,

$$\prod_{l \geq 2} (1 - (\frac{2}{N})^{2l-2})^{2k\Delta^{2l-2}} = \prod_{l \geq 2} ((1 - \frac{1}{(N/2)^{2l-2}})^{(N/2)^{2l-2}})^{2k/8^{2l-2}} \geq$$

$$\geq \prod_{l \geq 2} (\frac{1}{4})^{2k/8^{2l-2}} = (\frac{1}{4})^{2k \sum_{l \geq 2} \frac{1}{64^{l-1}}} = (\frac{1}{4})^{\frac{2k}{63}}.$$

Usando a mesma abordagem podemos mostrar que

$$(1 - \frac{2}{N})^{4k\Delta} \geq (\frac{1}{4})^{\frac{k}{2}}.$$

Logo

$$r_{\overline{c}_{2k}} \prod_{A_j \in \Gamma(A_{\overline{c}_{2k}})} (1 - r_j) \geq (\frac{2}{N})^{2k-2} \cdot (\frac{1}{4})^{\frac{k}{2}} \cdot (\frac{1}{4})^{\frac{2k}{63}}.$$

Como

$$\frac{(\frac{2}{N})^{2k-2}\cdot(\frac{1}{4})^{\frac{k}{2}}\cdot(\frac{1}{4})^{\frac{2k}{63}}}{\frac{1}{N^{2k-2}}} = 2^{2k-2}\cdot(\frac{1}{4})^{\frac{k}{2}}\cdot(\frac{1}{4})^{\frac{2k}{63}} > 1$$

temos o resultado desejado. O mesmo raciocínio pode ser usado para os eventos da forma $A_{\{e_i,e_j\}}$. ■

Proposição 1.20. *Seja $G = (V, E)$ um grafo com grau máximo $\Delta \geq 3$. Então a'(G) $\leq \lceil 9.62(\Delta - 1)\rceil$.*

Prova: Usaremos o mesmo grafo H da proposição anterior dessa vez com $N = c(\Delta - 1)$, para algum $c > 0$ a determinar. Como mencionado, a ideia é usar o teorema 1.9 para melhorar o resultado anterior. Dessa vez, como temos eventos ruins com probabilidades muito diferentes, não poderemos colocar $\mu_{\mathbf{x}} = \mu$ para todo $x \in X$. Faremos algo similar: para cada $x \in T$, seja $\mu_x = \mu_1 > 0$. Para cada $y \in C_{2k}$, seja $\mu_y = \mu_k > 0$. Então, pela desigualdade citada após o teorema 1.10, temos que

$$Z_{\Gamma^*(x)}(\mu) \leq (1 + 2(\Delta - 1)\mu_1 + \sum_{s\geq 2}(\Delta - 1)^{2s-2}\mu_s)^2$$

e

$$Z_{\Gamma^*(y)}(\mu) \leq (1 + 2(\Delta - 1)\mu_1 + \sum_{s\geq 2}(\Delta - 1)^{2s-2}\mu_s)^{2k}.$$

Logo devemos achar μ de forma que

$$\frac{1}{N} \leq \frac{\mu_1}{(1 + 2(\Delta - 1)\mu_1 + \sum_{s\geq 2}(\Delta - 1)^{2s-2}\mu_s)^2}$$

e

$$\frac{1}{N^{2k-2}} \leq \frac{\mu_k}{(1 + 2(\Delta - 1)\mu_1 + \sum_{s\geq 2}(\Delta - 1)^{2s-2}\mu_s)^{2k}}.$$

Seja $\mu_1 = q = \frac{\alpha}{(\Delta-1)}$, com $0 < \alpha < 1$, e $\mu_k = q^{2k-2}$. Lembrando que $N \geq c(\Delta - 1)$, as inequações acima tomam a forma

$$\frac{1}{c} \leq \frac{\alpha}{(1 + 2\alpha + \sum_{s\geq 2}\alpha^{2s-2})^2}$$

e

$$\frac{1}{c} \leq \frac{\alpha}{(1 + 2\alpha + \sum_{s\geq 2}\alpha^{2s-2})^{\frac{2k}{2k-2}}}.$$

Como $k \geq 2$, a primeira inequação garante a segunda. Logo a condição que garante que nenhum *evento ruim* ocorre é

$$\frac{1}{c} \leq \frac{\alpha}{(1+2\alpha+\sum_{s\geq 2}\alpha^{2s-2})^2}$$

ou seja

$$c \geq \frac{(1+2\alpha+\frac{\alpha^2}{1-\alpha^2})^2}{\alpha}.$$

Minimizando a função acima chega-se ao resultado que se $c \geq 9.62$ então todo grafo G com grau máximo Δ admite uma coloração acíclica de elos usando $N = c(\Delta - 1)$ cores. ■

◇ ◇ ◇

Capítulo 2

O Artigo de Alon e Füredi

"Na realidade trabalha-se com poucas cores. O que dá a ilusão do seu numero é serem postas em seu justo lugar." - Pablo Picasso

Nesse capítulo (P, L, R) denotará um plano projetivo finito[1] de ordem n com conjunto de pontos P, de linhas L e relação $R\colon P \to L$.

Dado $c \in \mathbb{N}$, uma c-coloração dos ponto de P é uma função $f\colon P \to \{1, \ldots, c\}$. A mesma c-coloração também pode ser representada pela c-upla $(f^{-1}(1), \ldots, f^{-1}(c))$ onde $f^{-1}(i)$ é o conjunto de todos os pontos que f colore com a cor i. Em relação a uma c-coloração, define-se o tipo de uma linha de L como sendo $t_{l,f} := (|l \cap f^{-1}(1)|, \ldots, |l \cap f^{-1}(c)|)$ (aqui l representa o conjunto que contém todos os pontos p tais que $(p, l) \in R$. A mesma representação será usada mais adiante). Se todas as linhas de um (P, L, R) tiverem tipos diferentes em relação a uma c-coloração, dá-se a essa coloração a alcunha de *coloração legítima*. Se, para $l_i, l_j \in L$, $t_{li,f} = t_{lj,f}$ então $\{l_i, l_j\}$ é dito ser um par ruim.

O objeto de estudo aqui é o número mínimo de cores necessárias – denotado por $c(P_n)$ – para que exista uma c-coloração legítima de (P, L, R). O artigo [3] de Noga Alon e Zoltan Füredi afirma que se a ordem de (P, L, R) for suficientemente grande (maior que 10^{250}), então $5 \leqslant c(P_n) \leqslant 8$. Em outras palavras: para um (P, L, R) de ordem suficientemente grande, se $c < 5$ então em toda c-coloração de (P, L, R) existem pelo menos duas linhas com o mesmo tipo e se $c = 8$ então existe uma c-coloração legítima de (P, L, R), qualquer que seja sua ordem. Que $c(P_n)$ tem uma cota inferior já era esperado. Porém, surpreende $c(P_n)$ ter uma cota superior, i.e., com 8 cores

[1] Visite o apêndice A para definições e resultados básicos de planos projetivos finitos.

é possível colorir de forma legítima planos projetivos de ordem tão grande quanto se queira. Isso é contra-intuitivo pois quanto maior a ordem do plano mais linhas temos para atribuir tipos diferente. Porém, também ganhamos mais pontos em cada linha o que faz com que consigamos produzir mais tipos com o mesmo número de cores. A dúvida seria se ganhamos mais tipos do que linhas. A resposta foi dada por Alon Füredi e é explicada a seguir.

2.1 8 Cores São Suficientes

A seguir provaremos o seguinte resultado mostrado em [3]:

Teorema 2.1. *Para um (P, L, R) de qualquer ordem, $c(P_n) \leqslant 8$.*

Prova: Durante toda a prova, n representará a ordem do (P, L, R) a ser colorido. n será considerado tão grande quanto necessário em todas as passagens.

Para essa prova usaremos os conhecimentos obtidos no capítulo 1 e aplicaremos o caso simétrico do Lema Local de Lovász: seja C uma 8-coloração aleatória uniforme de P (i.e., para cada $p \in P$, $\mathbf{P}(C(p) = i) = \frac{1}{8}$, para todo i $\in \{1, \ldots, 8\}$). Sejam $A_{li,lj}$ os eventos "l_i e l_j tem o mesmo tipo em relação a C", para todo $l_i, l_j \in L$ com $i < j$. Claramente esses $\binom{n^2+n+1}{2}$ eventos não são mutuamente independentes. Se o fossem, o teorema já estaria provado (ver [9], pág. 12). Teremos mais trabalho que isso para usar o Lema de Lovász . Buscando maior fluidez e clareza, apenas enunciarei dois lemas - que sustentarão a prova - para só depois prová-los:

Lema 2.2. *Existe $S \subset P$ tal que, para todo $l \in L$, $\ln n \leqslant |l \cap S| \leqslant 20 \ln n$.*

De agora em diante S denotará tal conjunto. Seja $f\colon P \setminus S \to \{1, \ldots, 8\}$ uma 8-coloração parcial de (P, L, R). Para todo $l_i, l_j \in L$, define-se $\{l_i, l_j\}$ como sendo um par perigoso se $d_1(t_{li,f}, t_{lj,f}) \leqslant 40 \ln n$ (sejam $x = (x_1, \ldots, x_n)$ e $y = (y_1, \ldots, y_n)$ tem-se $d_1(x, y) = \sum_{i=0}^{n} |x_i - y_i|$). Entendamos essa definição: como, pelo lema 2.2, para todo $l \in L$, $\ln n \leqslant |l \cap S| \leqslant 20 \ln n$ e $|l| = n + 1$ então $n + 1 - 20 \ln n \leqslant |(P \setminus S) \cap l| = |l \setminus S \cap l| \leqslant n + 1 - \ln n$. Logo $f\colon P \setminus S \to \{1, \ldots, 8\}$ colore, no mínimo, $n + 1 - 20 \ln n$ pontos de l. Portanto deixa de colorir, no máximo, $20 \ln n$ de l (que são os pontos de l que estão em S). Logo, para cada par $l_i, l_j \in L$, f deixa de colorir, no máximo, $40 \ln n$ pontos de $l_i \cup l_j$. Suponhamos que, para algum $c \in \{1, \ldots, 8\}$ e $l_i, l_j \in L$, $\mid (|f^{-1}(c) \cap l_i| - |f^{-1}(c) \cap l_j|) \mid = k$. Isso quer dizer que f coloriu, com a cor c, k pontos a mais em uma linha do que na outra. Quando essa coloração parcial for estendida a uma coloração total $g\colon P \to \{1, \ldots, 8\}$

(colorindo-se os pontos restantes em S) pode acontecer dessa diferença ser equilibrada (basta que a linha que recebeu menos cores c tenha, no processo de extensão da coloração parcial para a total, mais k pontos coloridos com a cor c e a outra linha não tenha mais nenhum, por exemplo). Desse modo $|g^{-1}(c)\cap l_i|-|g^{-1}(c)\cap l_j|$ seria igual a zero. Se, durante a extensão, o mesmo acontecer para todos os $|f^{-1}(c)\cap l_i|-|f^{-1}(c)\cap l_j|\neq 0$ teremos $t_{li,g}=t_{lj,g}$. Como não queremos que isso aconteça, chamamos de pares perigosos (em relação a uma coloração parcial f) os pares $\{l_i,l_j\}$ que podem se tornar ruins em alguma extensão de f. Como uma extensão de f só colorirá os pontos que f ainda não coloriu - já vimos que são no máximo $40\ln n$ em cada par de linhas de L - essa extensão poderá diminuir $d_1(t_{li,f},t_{lj,f})$ em, no máximo, $40\ln n$. Logo, se $d_1(t_{li,f},t_{lj,f})>40\ln n$, $\{l_i,l_j\}$ não se tornará um par ruim em nenhuma extensão possível de f.

Lema 2.3. *Existe uma 8-coloração $f\colon P\setminus S\to\{1,\ldots,8\}$ na qual nenhum ponto $p\in P$ pertence a mais do que 4 pares perigosos.*

Admitindo esses dois lemas, estamos aptos a aplicar o Lema Local de Lovász (caso simétrico):

Seja $f\colon P\setminus S\to\{1,\ldots,8\}$ uma coloração parcial de (P,L,R) que obedeça o Lema 2.3. Seja C uma extensão aleatória uniforme de f. Para todo par perigoso $\{l_i,l_j\}$ (em respeito a f) seja A_{l_i,l_j} o evento "$\{l_i,l_j\}$ torna-se ruim com respeito à extensão aleatória C de f". Sejam $\{l_i,l_j\}$ um par perigoso, $S_i=S\cap l_i$ e $S_j=S\cap l_j$ (S_i e S_j são os conjuntos que contém os pontos de l_i e l_j, respectivamente, que não foram coloridos por f). Pelo lema 2.2 temos que:

$$\ln n\leqslant|S_i|\leqslant 20\ln n\quad\text{e}\quad\ln n\leqslant|S_j|\leqslant 20\ln n.$$

Na prova do Lema 2.4, veremos que isso implica que:

$$\mathbf{P}(A_{l_i,l_j})\leqslant\frac{100}{(\ln n)^{\frac{7}{2}}}.$$

Não é difícil constatar que cada evento $A_{li,lj}$ é mutuamente independente a todos os eventos $A_{lm,ln}$ que satisfaçam $(S_i\cup S_j)\cap(S_m\cup S_n)=\emptyset$ (basta observar que qualquer informação sobre a coloração dos pontos de $S_m\cup S_n$ não fornece nenhuma informação sobre a coloração dos pontos de $S_i\cup S_j$).

Pelo lema 2.3 temos que não existe $p\in P$ tal que p pertença a mais do que quatro pares perigosos. Disso segue que passam no máximo quatro pares perigosos por cada $p\in S_i\cup S_j$. Como $|S_i\cup S_j|\leqslant 40\ln n$, temos que o número de pares perigosos $\{l_m,l_n\}$ com $i\neq m\neq j\neq n\neq i$ tais que $(S_i\cup S_j)\cap(S_m\cup S_n)\neq 0$ é no máximo $|S_i\cup S_j|4\leqslant 160\ln n$. Finalmente

estamos aptos a aplicar o Lema Local de Lovász (com $\Delta = 160 \ln n$, e $p \leqslant \frac{100}{(\ln n)^{\frac{7}{2}}}$) para provar que que $\mathbf{P}(\bigcap \overline{A_{li,lj}}) > 0$ pois, como estamos supondo n grande, $(160 \ln n + 1)e^{\frac{100}{(\ln n)^{\frac{7}{2}}}} < 1$. ■

2.2 Provas dos Lemas

Lema 2.2. Existe $S \subset P$ tal que, para todo $l \in L$, $\ln n \leqslant |l \cap S| \leqslant 20 \ln n$.

Prova: Para provar esse lema, usaremos o seguinte resultado provado em [48] e reenunciado em [61]:

Lema 2.4. *Sejam $X_1, \ldots, X_n$ variáveis aleatórias i.i.d com distribuição Bernoulli(p) e $f\colon X_1 \times \ldots \times X_n \to \mathbb{R}$ uma função que satisfaz $|f(x) - f(y)| = 1$ sempre que $x, y \in \{0,1\}^n$ diferirem em apenas uma coordenada e $f(x) \geq f(y)$ sempre $x, y \in \{0,1\}^n$ obedecerem $x_i \geq y_i$, para todo $i \in [n]$. Logo, para todo $a > 0$ valem as desigualdades:*

$$\boldsymbol{P}(\ f \ \geqslant \ \mathbb{E}(f) + a\ ) \leqslant \exp\{-\frac{a^2}{2(np + a/3)}\};$$

$$\boldsymbol{P}(\ f \ \leqslant \ \mathbb{E}(f) - a\ ) \leqslant \exp\{-\frac{a^2}{2np}\}.$$

Nosso desafio agora é construir um conjunto $S \subset P$ que satisfaz o lema 2.2. Para tanto, considere uma moeda com probabilidade de sair cara $c = \frac{10 \ln n}{n+1}$. Jogaremos essa moeda para cada $p \in P$. Se der cara, $p \in S$. Caso contrário, $p \notin S$. Assim, $|S|$ é uma soma de Bernoullis com parâmetro c. Como $|P| = n^2+n+1$ então $|S| \sim \text{Bin}(n^2+n+1, c)$. Seja A_l o evento "$|S \cap l|$ não satisfaz o lema 2.2". Como $|l| = n+1$ temos que $|S \cap l| \sim Bin(n+1, c)$ com $\mathbb{E}(|S \cap l|) = 10 \ln n$. Considere a função $f : \{1,0\}^{n+1} \to \mathbb{R}$ definida por $f(x) = \sum x_i$. Sejam $\widehat{x}, \overline{x} \in \{0,1\}^{n+1}$ vetores que diferem em apenas uma coordenada. Logo

$$|f(\widehat{x}) - f(\overline{x})| = 1.$$

Sejam $\widehat{y}, \overline{y} \in \{0,1\}^{n+1}$ tais que $\widehat{y}_i \geq \overline{y}_i$, para todo $i \in [n+1]$. Logo $f(\widehat{y}) \geq f(\overline{y})$. Portanto, tendo visto que f satisfaz tais propriedades podemos usar as cotas enunciadas no lema 2.4. Como $f(x_l) = |S \cap l|$, onde x_l é o vetor de amostragens da linha l, temos então que

$$\mathbf{P}(\ |S \cap l| \geqslant 20 \ln n\ ) \leqslant exp\{-\frac{300 \ln^2 n}{80 \ln n}\} < \frac{1}{n^3}$$

e

$$\mathbf{P}(\ |S \cap l| \leqslant \ln n\ ) \leqslant exp\{-\frac{81 \ln^2 n}{20 \ln n}\} < \frac{1}{n^4}.$$

Logo

$$\mathbf{P}(A_l) < \frac{1}{n^4} + \frac{1}{n^3}.$$

Sejam X_i e Y variáveis aleatórias com $X_i(\overline{A_{li}}) = 0$ e $X_i(A_{li}) = 1$, para todo $i \in \{1, \ldots, n^2 + n + 1\}$ e $Y = \sum_{i=1}^{n^2+n+1} X_i$. Logo temos

$$\mathbb{E}(Y) = \mathbb{E}(\sum_{i=1}^{n^2+n+1} X_i) = \sum_{i=1}^{n^2+n+1} \mathbb{E}(X_i) < \sum_{i=1}^{n^2+n+1} \frac{1}{n^4} + \frac{1}{n^3} =$$

$$= (n^2 + n + 1)(\frac{1}{n^4} + \frac{1}{n^3}) < 1.$$

Acima, foi usado a linearidades da esperança e que n é grande. Com isso concluímos que o evento "para S, todas as linhas satisfazem o lema 2.2" tem probabilidade maior que zero. Vejamos porque:

Seja $(\Omega, \mathscr{P}, \mathbf{P})$ um espaço de probabilidade com $\Omega = \{A_0, \ldots, A_{n^2+n+1}\}$, onde A_i é o evento "apenas i linhas não satisfazem o lema 2.2", para todo $i \in \{0, \ldots, n^2 + n + 1\}$, $\mathscr{P}$ é o conjuntos das partes de Ω e $\mathbf{P}(A_i)$ é a probabilidade de A_i sendo que $\mathbf{P}(A_{li}) < \frac{1}{n^4}$. Por construção, a variável aleatória Y conta quantas linhas não satisfazem o lema 2.2. Sendo assim temos que $\mathbb{E}(Y) = \sum_{i=0}^{n^2+n+1} i\mathbf{P}(A_i)$. Faremos um argumento por absurdo: Supondo que $\mathbf{P}(A_0) = 0$ temos que:

$$1 = \mathbf{P}(\Omega) = \mathbf{P}(\bigcup_{i=0}^{n^2+n+1} A_i) = \mathbf{P}(\bigcup_{i=1}^{n^2+n+1} A_i) = \sum_{i=1}^{n^2+n+1} \mathbf{P}(A_i) = 1$$

Agora, se $\sum_{i=1}^{n^2+n+1} \mathbf{P}(A_i) = 1$ é evidente que $\sum_{i=1}^{n^2+n+1} i\mathbf{P}(A_i) > 1$. Logo

$$\mathbb{E}(Y) = \sum_{i=0}^{n^2+n+1} i\mathbf{P}(A_i) = \sum_{i=1}^{n^2+n+1} i\mathbf{P}(A_i) > 1.$$

Absurdo! Pois já estabelecemos acima que $\mathbb{E}(Y) < 1$. Logo $\mathbf{P}(A_0) > 0$. Portanto existe um conjunto S que satisfaz o lema 2.2. ■

Observação 2.5. Com a mesma argumentação acima é possível mostrar que existe $S \subset P$ tal que, para todo $l \in L$, $\ln n \leqslant |l \cap S| \leqslant 11 \ln n$. Esse resultado será usado no capítulo 4.

Lema 2.3. Existe uma 8-coloração $f\colon\ P \setminus S \to \{1,\ldots,8\}$ na qual nenhum ponto $p \in P$ pertence a mais do que quatro pares perigosos.

Prova: Para provar esse lema, precisaremos estabelecer alguns outros resultados antes. De agora em diante F denotará o conjunto $P \setminus S$, onde S é um conjunto que obedece o lema 2.2.

Lema 2.6. *Sejam $l \in L$ e $T \subset l$ tal que $|T| = k$ e $\widehat{t} = (t_1,\ldots,t_8)$ um vetor com entradas inteiras não negativas. Então, para qualquer $g : T \to \{1,\ldots,8\}$ e $f : F \to \{1,\ldots,8\}$, temos que*

$$\boldsymbol{P}(\ t_{l,f} = \widehat{t} \mid\ f(p) = g(p),\ \forall p \in T\ ) < \frac{\binom{m-k}{\lfloor\frac{m-k}{8}\rfloor,\lfloor\frac{m-k+1}{8}\rfloor,\ldots,\lfloor\frac{m-k+7}{8}\rfloor}}{8^{m-k}},$$

onde $m = |l \cap F|$. Em particular se $k \leqslant \sqrt{n}$ a probabilidade condicional acima é menor que $\frac{100}{n^{\frac{7}{2}}}$.

Prova: Seja $s_i = |g^{-1}(i)|$. É evidente que $t_{l,f} = \widehat{t}$ se, e somente se, o número de pontos coloridos com i de $L \cap F \setminus T$ (i.e., os pontos de $L \cap F$ que sobram para f colorir após g colorir os pontos em $L \cap T$) for igual a $t_i - s_i$. Para esses $|L \cap F \setminus T|$ pontos, há 8^{m-k} possíveis colorações igualmente prováveis. Dessas, exatamente $\binom{m-k}{t_1-s_1,t_2-s_2,\ldots,t_8-s_8}$ fazem com que $t_{l,f} = \widehat{t}$. Logo a probabilidade condicional acima é calculada por $\frac{\binom{m-k}{t_1-s_1,t_2-s_2,\ldots,t_8-s_8}}{8^{m-k}}$. Esse multinomial assume valor máximo quando os valores de $t_i - s_i$ são os mais próximos possíveis. Com efeito, estabeleceremos, primeiro, o seguinte resultado: sejam a_1 e a_2 tais que $a_1 + a_2 = c$, com c constante, então $\binom{c}{a_1,a_2}$ assume seu valor máximo quando a_1 e a_2 estão o mais próximo possível. Com efeito, se c é par considere $a_1 = \frac{c}{2} = a_2$. Sendo assim, temos que $(a_1 + i)!(a_2 - i)! > a_1!a_2!$, para todo $i \in \{1,\ldots,\frac{c}{2}\}$, pois

$$(a_1 + i)!(a_2 - i)! = a_1!(a_1 + i)\ldots(a_1 + 1)(a_2 - i)! >$$

$$> a_1!(a_2)\ldots(a_2 - i + 1)(a_2 - i)! = a_1!a_2!.$$

Portanto $\binom{c}{a_1,a_2} > \binom{c}{a_1+i,a_2-i}$, para todo $i \in \{1,\ldots,\frac{c}{2}\}$. Se c é ímpar o resultado segue com um raciocínio muito similar. O resultado obtido é facilmente generalizado para $\binom{c}{a_1,a_2,\ldots,a_8}$. E com isso o lema está provado, pois $\lfloor\frac{m-k}{8}\rfloor, \lfloor\frac{m-k+1}{8}\rfloor, \ldots, \lfloor\frac{m-k+7}{8}\rfloor$ são os números mais próximos cuja soma é $m - k$. Repare que, pelo lema 2.2, $n + 1 - 20\ln n \leqslant m \leqslant n + 1 - \ln n$. Como

$$\frac{\binom{m-k}{\lfloor\frac{m-k}{8}\rfloor,\lfloor\frac{m-k+1}{8}\rfloor,\ldots,\lfloor\frac{m-k+7}{8}\rfloor}}{8^{m-k}}$$

aumenta a medida que m diminui (verifique!), temos que, sendo $N = n+1-20\ln n-k$, o maior valor que essa probabilidade condicional pode assumir é

$$\frac{\binom{N}{\lfloor \frac{N}{8}\rfloor,\lfloor \frac{N+1}{8}\rfloor,\ldots,\lfloor \frac{N+7}{8}\rfloor}}{8^N}.$$

Usando que $a! \sim (\frac{a}{e})^a\sqrt{2\pi a}$ ([12], pág. 4) e supondo, sem perda de generalidade, $n \equiv 0(mod\ 8)$, temos o resultado particular para $k \leqslant \sqrt{n}$:

$$\frac{\binom{N}{\lfloor \frac{N}{8}\rfloor,\lfloor \frac{N+1}{8}\rfloor,\ldots,\lfloor \frac{N+7}{8}\rfloor}}{8^N} = \frac{N!}{8^N(\frac{N}{8}!)^8} \sim \frac{(\frac{N}{e})^N\sqrt{2\pi N}}{8^N(\frac{(\frac{N}{8})^{\frac{N}{8}}}{e^{\frac{N}{8}}})^8(\sqrt{2\pi\frac{N}{8}})^8}$$

Mas

$$\frac{(\frac{N}{e})^N\sqrt{2\pi N}}{8^N(\frac{(\frac{N}{8})^{\frac{N}{8}}}{e^{\frac{N}{8}}})^8(\sqrt{2\pi\frac{N}{8}})^8} = \frac{N^N8^4\sqrt{2\pi N}}{8^N(\frac{N}{8})^N(2\pi N)^4} = \frac{(\frac{4}{\pi})^4\sqrt{2\pi}}{N^{\frac{7}{2}}} =$$

$$= \frac{(\frac{4}{\pi})^4\sqrt{2\pi}}{(n+1-20\ln n-\sqrt{n})^{\frac{7}{2}}} < \frac{100}{n^{\frac{7}{2}}}.$$

Conforme mencionado anteriormente, esse resultado pode ser usado para provar que $\mathbf{P}(A_{l_i,l_j}) \leqslant \frac{100}{(\ln n)^{7/2}}$. Com efeito, para qualquer coloração de l_j, a probabilidade de l_i ter o mesmo tipo é menor que

$$\frac{\binom{|S_i|-1}{\lfloor \frac{|S_i|-1}{8}\rfloor,\lfloor \frac{|S_i|}{8}\rfloor,\ldots,\lfloor \frac{|S_i|+6}{8}\rfloor}}{8^{|S_i|-1}}.$$

Como $|S_i| > \ln n$, por um raciocínio similar ao desenvolvido acima, temos que essa probabilidade é menor que $\frac{100}{(\ln n)^{7/2}}$. ■

Corolário 2.7. *Sejam $l_1, l_2 \in L$, $T \subset P$ tais que $|l_2 \cap T| \leqslant \sqrt{n}$. Para qualquer $g : T \to \{1,\ldots,8\}$ tem-se que, em relação a f,*

$$\boldsymbol{P}(\ \{l_1,l_2\} \text{ é } perigoso \mid f(p) = g(p),\ \forall p \in T\ ) < \frac{(\ln n)^9}{n^{\frac{7}{2}}}.$$

Prova: Fixada uma coloração de $l_1 \cup T$ temos que, para cada $\widehat{t}$ com $d_1(t_{l_1,f},\widehat{t}) \leqslant 40\ln n$, pelo lema 2.5, a probabilidade de $t_{l_2,f} = \widehat{t}$ é menor que $\frac{100}{n^{7/2}}$. Há menos de $(100\ln n)^8$ vetores $\widehat{t}$ que distam não mais que $40\ln n$ de $t_{l_1,f}$. Com efeito, seja $t_{l_1,f} = ((l_1)_1,\ldots,(l_1)_8)$. Considere a seguinte notação:

r símbolos $\bullet$ na i-ésima coordena de um vetor $\widehat{t}$ significa que $|(l_1)_i - \widehat{t_i}| = r$. Por exemplo, o vetor

$$(\bullet \ \bullet \ \bullet \ \bullet \ \bullet ; \ \bullet ; \ ; \ ; \ ; \ ; \ \bullet \bullet ; \ \bullet)$$

indica que $|(l_1)_1 - t_1| = 5$, $|(l_1)_2 - t_2| = 1$, $|(l_1)_3 - t_3| = 0$, ..., $|(l_1)_7 - t_7| = 2$ e $|(l_1)_8 - t_8| = 1$. Com essa notação e levando em consideração que se $|(l_1)_i - \widehat{t_i}| = r$ temos que $\widehat{t_i}$ pode ter mais ou menos r pontos coloridos com a cor i é fácil perceber que o número de vetores que distam D de l_1 é

$$2^8 \binom{D+7}{7}.$$

Logo, o número de vetores que distam menos que $40 \ln n$ de l_1 é

$$2^8 \sum_{i=0}^{40 \ln n} \binom{i+7}{7} = 2^8 \binom{40 \ln n + 8}{8} < (100 \ln n)^8.$$

Acima foi usado que $\binom{n}{n} + \binom{n+1}{n} + \ldots + \binom{n+k}{n} = \binom{n+k+1}{n+1}$ e que n é grande.

Portanto,

$$\mathbf{P}(\ \{l_1, l_2\} \text{ é } \textit{perigoso} \mid f(p) = g(p),\ \forall p \in T\ ) < (100 \ln n)^8 \frac{100}{n^{7/2}} < \frac{(\ln n)^9}{n^{7/2}}.$$

■

Lema 2.8. *A probabilidade de existirem $l_1, l_2, l_3 \in L$ tais que $\{l_1, l_2\}$ e $\{l_1, l_3\}$ sejam perigosos (com respeito a f) é menor que $\frac{(\ln n)^{18}}{n}$.*

Prova: Sejam $l_1, l_2, l_3 \in L$. Pelo corolário 2.5 temos que

$$\mathbf{P}(\ \{l_1, l_2\} \text{ é } \textit{perigoso}\ ) < \frac{(\ln n)^9}{n^{\frac{7}{2}}}.$$

De novo pelo corolário temos que

$$\mathbf{P}(\ \{l_1, l_3\} \text{ é } \textit{perigoso} \mid \{l_1, l_2\} \text{ é } \textit{perigoso}\ ) < \frac{(\ln n)^9}{n^{\frac{7}{2}}}.$$

Sendo A_{l_i, l_j} o evento "$\{l_i, l_j\}$ é perigoso" temos que

$$\mathbf{P}(\ A_{l_1,l_2} \cap A_{l_1,l_3}\ ) = \mathbf{P}(\ A_{l_1,l_2}\ )\ \mathbf{P}(\ A_{l_1,l_3} \mid A_{l_1,l_2}\ ) < \frac{(\ln n)^{18}}{n^7}.$$

Como existem $(n^2+n+1)\binom{n^2+n}{2} < n^6$ diferentes formas de escolher o trio $l_1, l_2, l_3 \in L$ então, sendo A_{l_1,l_2,l_3} o evento "$\{l_1, l_2\}$ e $\{l_1, l_3\}$ são perigosos":

$$\mathbf{P}(\bigcup_{1\leqslant i\leqslant n^2+n+1,\ i\neq j<k\neq i} A_{l_i,l_j,l_k}\ ) <$$

$$< \sum_{1\leqslant i\leqslant n^2+n+1,\ i\neq j<k\neq i} \mathbf{P}(\ A_{l_i,l_j,l_k}) \ < n^6\frac{(\ln n)^{18}}{n^7} = \frac{(\ln n)^{18}}{n}.$$

■

Lema 2.9. *A Probabilidade de existirem $p \in P$ e $\{l_1,\ldots,l_5,l'_1,\ldots,l'_5\}$ tais que $p \in l_1 \cap \ldots \cap l_5$ e $\{l_i,l'_i\}$ é perigoso para todo $i \in \{1,\ldots,5\}$ é menor que $\frac{(\ln n)^{45}}{\sqrt{n}}$.*

Prova: Sejam $p \in P$, $l_1,\ldots,l_5,l'_1,\ldots,l'_5 \in L$, A_{l_i,l_j} o evento "$\{l_i,l_j\}$ é um par perigoso". Pelo Corolário 2.5, temos que

$$\mathbf{P}(\ A_{l_1,l'_1}\ ) < \frac{(\ln n)^9}{n^{\frac{7}{2}}} \qquad \text{e} \qquad \mathbf{P}(\ A_{l_{i+1},l'_{i+1}} \mid \bigcap_{k=1}^{i} A_{l_i,l'_i}\ ) < \frac{(\ln n)^9}{n^{\frac{7}{2}}}.$$

Portanto $\mathbf{P}(\ \bigcap_{k=1}^{5} A_{l_i,l'_i}\ ) < \frac{(\ln n)^{45}}{n^{35/2}}$. Como há menos que $(n^2+n+1)\binom{n+1}{5}(n^2+n+1)^5 < n^{17}$ formas de escolher $p,l_1,\ldots,l_5,l'_1,\ldots,l'_5$ temos que (por um raciocínio análogo da prova do lema 2.7) a probabilidade de existirem tais $p,l_1,\ldots,l_5,l'_1,\ldots,l'_5$ é menor que

$$n^{17}\frac{(\ln n)^{45}}{n^{\frac{35}{2}}} = \frac{(\ln n)^{45}}{\sqrt{n}}.$$

■

Com esses resultados, finalmente estamos aptos a provar o lema 2.3. Para melhor compreensão, o reenunciaremos de forma mais precisa:

Lema 2.3. A Probabilidade de nenhum ponto $p \in P$ estar em mais de quatro pares perigosos é pelo menos $1-(\frac{(\ln n)^{18}}{n}+\frac{(\ln n)^{45}}{\sqrt{n}})$ (em particular, existe uma 8-coloração f de $F = P\backslash S$ na qual nenhum ponto $p \in P$ pertence a mais do que quatro pares perigosos).

Prova: Esse lema é uma consequência direta dos lemas 2.7 e 2.8. Pelo lema 2.8 a probabilidade de não existirem $l_1,\ldots,l_5 \in L$ e $p \in P$ como na figura 1 tais que existam $l'_1,\ldots,l'_5 \in L$ com $\{l_i,l'_i\}$ par perigoso para todo $i \in \{1,\ldots,5\}$ é pelo menos $1-\frac{(\ln n)^{45}}{\sqrt{n}}$.

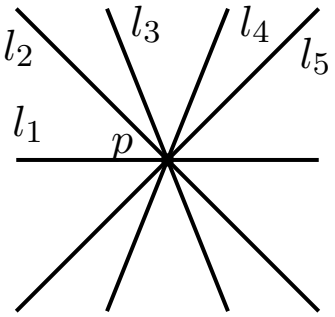

Figura 1

Porém, ainda poderiam existir

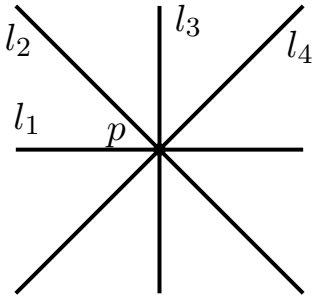

Figura 2

$l_1, \ldots, l_4 \in L$ e $p \in P$ como na figura 2 tais que existissem $l_1', l_1'', l_2', l_3', l_4' \in L$ com $\{l_1, l_1'\}$, $\{l_1, l_1''\}$, $\{l_i, l_i'\}$ pares perigosos para todo $i \in \{2,3,4\}$. Mas, pelo lema 2.7, temos que a probabilidade de isso não acontecer é pelo menos $1 - \frac{(\ln n)^{18}}{n}$. Logo, a probabilidade de não acontecer nenhuma das duas situações acima (o que resulta no lema) é pelo menos $1 - (\frac{(\ln n)^{18}}{n} + \frac{(\ln n)^{45}}{\sqrt{n}})$. ■

Observação 2.10. Em [3], é conjecturado[2] que $c(P_n) \leq 7$. Não é difícil verificar que, com pequenas mudanças, quase todos os resultados dessa seção podem ser provados para $c = 7$. A única exceção é o lema 2.8. Sem esse resultado não conseguimos provar aquilo que tínhamos como única finalidade: o lema 2.3, que efetivamente tira a dependência do grau máximo do gráfico de dependência com n. Portanto, para provar tal conjectura, se

[2]Mais precisamente, é conjecturado que $6 \leq c(P_n) \leq 7$.

faz necessária outra abordagem. Nesse trabalho, apresentamos melhorias no resultado no sentido de achar valores para $c(P_n)$ para n's pequenos e diminuir a exigência para n's grandes. A conjectura $c(P_n) \leq 7$ segue um problema em aberto. □

◇ ◇ ◇

Capítulo 3

O Método de Entropy Compression

"If something cannot go on forever, it must stop." - Herbert Stein

Como visto no capítulo 2, o Lema Local de Lovász prova a existência de um determinado objeto porém não o constrói. Para um caso especial do LLL chamado de *variable version*[1], tal problema foi resolvido por Moser e Tardos em [51] pelo uso do chamado algoritmo de Moser-Tardos que busca no espaço de probabilidade por um ponto que evite todos os eventos ruins. O algoritmo foi posteriormente refinado e estendido para outras situações como em [1, 19, 44, 54]. O argumento principal em [51] é o chamado *entropy compression method*[2]. Grytczuk, Kozik, e Micek, em [37], perceberam que aplicando o método de entropy compression diretamente pode-se, na maioria dos casos, conseguir melhores resultados do que aplicando o LLL. Muitos resultados foram obtidos dessa forma como em [13, 14, 24, 29, 35, 55, 56]. A próxima seção estuda como tal método funciona baseando-se no que foi exposto em [64] por Terence Tao.

3.1 O Argumento de Entropy Compression

Uma estratégia comum para garantir que determinado algoritmo pára é explorar alguma propriedade de monotonicidade, isto é, mostrar que alguma

[1] Esse nome é devido a Kolipaka e Szegedy por [44].
[2] Esse nome é devido a Tao por [64].

quantidade limitada superiormente/inferiormente continua
crescendo/diminuindo a cada passo do algoritmo. O método de entropy compression se enquadra nesse tipo de estratégia. Nele, é mostrado que um algoritmo probabilístico pára fazendo-se uso da entropia de Shannon das variáveis do algoritmo. Mais precisamente, suponha que um algoritmo probabilístico faz uso de um vetor de bits aleatório F e a cada passo do algoritmo toma um objeto A (que pode ter sido inicializado aleatoriamente) e alguma porção do vetor F para deterministicamente criar um novo objeto A' e um vetor F' formado pelo vetor F menos as entradas utilizadas no passo em questão. Se, de posse A', F' e mais um pedaço de informação R' formos capazes de resgatar A e F usando menos espaço, então estamos realizando uma compressão sem perdas de $A + F$ em $A' + F' + R'$. Porém, como exposto no apêndice C, a probabilidade de comprimirmos sem perdas $A + F$ em um vetor de tamanho menor que sua entropia de Shannon tende a zero. Logo se tivermos um limite inferior para a entropia de $A + F$ e se mostrarmos que a cada passo[3] diminuímos significativamente em tamanho o vetor $A' + F' + R'$ então o algoritmo deve parar depois de um número finito de iterações. Ainda pode-se levantar a seguinte questão: digamos que $|F| = t$. É possível que, após t iterações, o vetor $A + F$ ainda seja comprimível. Se aumentarmos o tamanho de F, o algoritmo terá mais iterações para continuar a compressão mas, em contrapartida, terá uma mensagem maior para comprimir, podendo assim ainda ficar acima do limite que é a entropia de $A + F$. O que garante que existe um t grande o suficiente para o qual o algoritmo pára antes de t iterações? Para responder essa questão precisamos ser um pouco menos simplistas: quando aumentamos o valor de t, aumentamos o tamanho do vetor comprimido (pois aumentamos o tamanho da mensagem original) mas aumentamos também a entropia da mensagem original. A questão é o que aumentamos mais. Se conseguirmos argumentar que a entropia tem um aumento maior que o tamanho do vetor comprimido, aí sim podemos garantir que existe um t para o qual, se o algoritmo chegasse a t iterações, a entropia da mensagem original seria maior que o tamanho da mensagem comprimida. Um absurdo, pois, como t é grande o suficiente, isso contradiz o teorema C.26. Digamos que, para algum $c \in \mathbb{N}$, as entradas de F são coletadas de variáveis aleatórias i.i.d. discretas Uniforme(c) (como é o caso em aplicações do método de entropy compression). Logo, para cada unidade aumentada em t, temos um incremento de $\log_2 c$ bits de entropia em F. Para armazenar essa informação extra, inicialmente são necessários $\lceil \log_2 c \rceil$ bits. Mas estamos armazenando essa informação de forma comprimida, logo precisaremos de menos do que $\log_2 c$ bits extras de armazenamento. Portanto o incremento de entropia supera o incremento no tamanho da mensagem comprimida e o algoritmo

[3] Precisamos na verdade mostrar que existe um número $M \in \mathbb{N}$ para o qual após M passos diminuímos significativamente em tamanho o vetor $A' + F' + R'$.

deve parar antes de t iterações. Vejamos um exemplo exposto em [34]:

Exemplo 3.1. Considere o conjunto $\{x_1, \ldots, x_n\}$ de variáveis booleanas e sua negações $\{\neg x_1, \ldots, \neg x_n\}$, chamadas de literais. Fixado um natural $k > 1$, definimos uma cláusula (de tamanho k) como a disjunção de k literais. Por exemplo

$$x_1 \vee x_3 \vee \neg x_2$$

é uma cláusula de tamanho 3, falsa somente se x_1 e x_3 são falsos e x_2 é verdadeiro. Definimos o suporte de uma cláusula pelo conjunto de variáveis que a formam. Por exemplo a cláusula acima tem suporte $\{x_1, x_2, x_3\}$. Para evitar casos degenerados nós consideraremos apenas as cláusulas que tenham suporte de tamanho k. Note que se uma cláusula é falsa então sabemos os valores de todas as suas variáveis. Isso será um fato importante mais adiante.

O problema enunciado a seguir é conhecido como problema de k-satisfabilidade booleana: dado um conjunto S de cláusulas de tamanho k envolvendo n variáveis booleanas $x_1, \ldots, x_n$, existe alguma configuração de valores de tais variáveis tal que todas as cláusulas de S são satisfeitas?

Esse problema é simples para $k = 2$ e, segundo o teorema de Cook-Levin, NP-Completo para $k \geq 3$. Mas o problema se torna mais simples se impusermos restrições em S. Por exemplo, se todas as cláusulas de S têm suportes disjuntos então o problema é resolvido facilmente (basta que uma variável de cada cláusula tenha o valor requerido pela mesma). Supondo agora que cada cláusula de S não intersecta a maioria das outras cláusulas de S, nós estamos mais próximos de colocar o problema na linguagem do Lema Local de Lovász.

Teorema 3.2. *Seja S um conjunto de cláusulas de tamanho k tal que o suporte de cada cláusula $s \in S$ intersecta no máximo 2^{k-C} suportes de cláusulas de S, onde C é uma constante suficientemente grande. Então as cláusulas de S são simultaneamente satisfazíveis.*

Prova: Atribuindo independente e uniformemente um valor a cada variável, temos que a probabilidade de cada cláusula não ser satisfeita é 2^{-k}. Para cada $s \in S$, seja A_s o evento "a cláusula s não é satisfeita". Como cada cláusula não intersecta mais que 2^{k-C} cláusulas temos que o grafo de dependência dessa família de eventos tem grau máximo 2^{k-C}. Logo, pelo caso simétrico do lema de Lovász, temos que se $C > \log_2 e$ então todos os eventos ruins podem ser evitados. ■

Como dito anteriormente, apesar de bastante simples, essa prova não nos dá valores para as variáveis para os quais todas as cláusulas estão satisfeitas.

Usaremos agora um argumento de compressão de entropia para o mesmo problema:

Prova: De novo, começamos a prova atribuindo aleatoriamente valores $a_1, \ldots, a_n$ para as variáveis $x_1, \ldots, x_n$. Seja $A = (a_1, \ldots, a_n)$. Se A satisfizer todas as cláusulas S, não há nada mais a ser feito. Suponha que exista algum subconjunto não vazio $T \subset S$ cujas cláusulas não sejam satisfeitas por A.

Nossa tarefa agora é modificar A de maneira a reduzir $|T|$. Se, por exemplo, sempre fosse possível achar uma modificação A' de A cujo conjunto T' fosse menor que T então nós poderíamos repetir esse processo até que todas as cláusulas em S fossem satisfeitas. Uma maneira de tentar atingir esse objetivo é escolher uma cláusula s não satisfeita por A e mudar os valores de seu suporte. Basta que uma variável de seu suporte mude de valor para que s passe a ser satisfeita. Como ficará claro mais adiante, é desejável maximizar a quantidade de entropia do sistema no método de entropy compression. Portanto nós escolheremos aleatoriamente essa modificação de A. Nós escolheremos k novos valores para o suporte de s aleatoriamente para criar A' (dessa forma há uma probabilidade de 2^{-k} de que essa cláusula continue a não ser satisfeita porém o argumento fica ligeiramente mais simples se não nos preocuparmos com isso).

Se todas as cláusulas tiverem suportes disjuntos, essa estratégia funciona sem dificuldade. Mas, quando não podemos garantir isso, temos um problema: toda vez que modificamos A para satisfazer uma clausula s podemos estar fazendo com que outras cláusulas que intersectam o suporte de s deixem de ser satisfeitas podendo assim estar aumentando o tamanho de T. A observação chave de Moser é que cada falha de uma cláusula s para um conjunto A nos revela k bits de informação - os valores das variáveis do suporte de s em A. O plano é usar cada falha de uma cláusula para comprimir A (e mais alguma informação) sem perdas. O ponto principal é que, a cada passo do processo, a cláusula que estamos consertando seja quase sempre uma que intersecta uma cláusula consertada no passo anterior. Dessa forma, o número total de possibilidades para cada cláusula a ser consertada, dada a cláusula consertada no passo anterior, é 2^{k-C}, o que requer somente $k-C$ bits de armazenamento[4] contra k bits de entropia eliminados. É justamente isso que nos garante que o processo pára em um número finito de passos.

Para construir essa prova de maneira mais detalhada, nós precisaremos do seguinte:

- Uma n-upla $A = \{a_1, \ldots, a_n\}$, que será iniciada aleatoriamente e será

[4]Cada bite armazena 0 ou 1. Portanto n bits podem representar 2^n objetos distintos. Logo, para armazenar x objetos precisamos de $\log_2 x$ bits.

modificada ao longo do algoritmo (como mencionado acima);

• Um longo vetor de bits aleatório F, que será usado, em ordem, sempre que o algoritmo pedir uma entrada aleatória;

• Um algoritmo recursivo Fix(s) que modifique a n-upla A para satisfazer uma cláusula $s \in S$ (e que pode ocasionalmente fazer com que A obedeça outras cláusulas de S anteriormente não satisfeitas) definido recursivamente da seguinte maneira:

Passo 1.a: Se A satisfaz s não faça nada.

Passo 1.b: Caso contrário, ordene o suporte de s de maneira arbitrária. Leia os primeiros k bits de F ainda não usados (reduzindo F de k bits) e use a j-ésima entrada desse passo para redefinir o valor da j-ésima variável do suporte de s.

Passo 2: Encontre todas as cláusulas $s' \in S$ cujo suporte intersecta s e que agora não estejam satisfeitas por A - há no máximo 2^{k-C} cláusulas dessa forma. Ordene essas cláusulas arbitrariamente e execute Fix(s') para cada cláusula em ordem - assim, o algoritmo original Fix(s) é colocado "em espera" em alguma pilha de uma CPU enquanto todos os processos Fix(s') são executados. Fix(s) termina assim que todos os Fix(s') terminem.

Por um argumento de indução é fácil mostrar que se Fix(s) termina então a n-upla resultante A satisfará s e qualquer outra cláusula s' já satisfeita por A antes de Fix(s) ser executado. Logo Fix(s), quando termina, devolve (implicitamente) um conjunto de cláusulas não satisfeitas T' menor do que o conjunto T anterior. Podemos, então, consertar todas as cláusulas de S chamando o algoritmo Fix para cada cláusula ainda não satisfeita no início de cada passo dado que o algoritmo Fix termina para cada uma dessas chamadas.

Cada vez que o passo 1.b do algoritmo Fix é chamado, a n-upla A é transformada em uma nova n-upla A' e o vetor aleatório F é reduzido a um vetor F'. Note que esse processo é reversível. Com efeito, dado que sabemos $\{A', R' := s\}$ então A pode ser reconstruido mudando os valores do suporte de s para os únicos valores que violam s enquanto F pode ser reconstruido adicionando ao início de F' os bits de A no suporte de s. Esse tipo de reversibilidade não parece ser muito útil em um argumento de compressão de entropia porque enquanto F' é menor que F por k bits, são necessários $\log_2 |S|$ bits para armazenar a cláusula s (veja a nota de rodapé da página anterior). Então $F + A \mapsto F' + A' + R'$ só é uma compressão se $\log_2 |S| < k$, o que não é o que foi assumido nesse exemplo. Porém, enquanto são necessários $\log_2 |S|$ bits para armazenar uma cláusula s, em uma aplicação

recursiva do algoritmo Fix são necessário $k - C + O(1)$ bits para armazenar a cláusula s' que está sendo consertada. Se C for grande o suficiente temos que $k - C + O(1) < k$ o que torna possível nosso argumento de compressão de entropia. Com efeito, as cláusulas s para as quais o algoritmo Fix(s) é chamado podem ser divididas em duas categorias: 1) aquelas nas quais s veio da lista T original de cláusulas não satisfeitas. Cada uma delas requer $O(\log_2 |S|)$ bits de armazenamento. Como há no máximo $|T|$ chamadas desse tipo e $|T| \leq |S|$ temos que o total de bits necessários para armazena tais cláusulas é no máximo $O(|S| \log_2 |S|)$. Ou 2) aquelas para as quais Fix(s) é chamado recursivamente a partir de uma chamada anterior Fix(s'). Nesse caso s é uma entre as, no máximo, 2^{k-C} cláusulas de S cujo suporte intersecta o suporte de s'. Logo podemos nos referir a s usando s' e um número entre 1 e 2^{k-C} representando a posição de s (com respeito a uma ordenação arbitrária de S) na lista de todas as cláusulas de S cujo suporte intersecta o suporte de s'. Chamaremos esse número de índice da chamada Fix(s).

Como mencionado acima, a ideia é mantermos um registro R com o qual possamos reconstruir os vetores originais. Seja R o vetor que armazena 1) s cada vez que uma das $|T|$ Fix(s) originais for chamada 2) o índice das chamadas Fix(s) subsequentes pelo processo recursivo do algoritmo Fix e 3) um símbolo de parada sempre que um algoritmo Fix terminar. Dessa forma, podemos deduzir qual cláusula s está sendo consertada para cada chamada de Fix apenas analisando o registro R até aquele ponto. Logo em qualquer momento do algoritmo é possível reconstruir $A + F$ pelo estado atual $A' + F' + R'$ do processo.

Agora suponha, por contradição, que S não é satisfazível. Logo a pilha de chamadas da função Fix deve ser infinita. Nós caminharemos nessa pilha dando M passos, onde M é algum número grande a ser escolhido. Nesse ponto do algoritmo, o vetor F já foi reduzido de Mk bits e o vetor R' tem no máximo $O(|S| \log_2 |S|) + M(k - C + O(1))$ bits. Logo, nós temos uma compressão sem perdas $A + F \mapsto A' + F' + R'$ de $n + |F|$ bits aleatórios para $n + |F| - Mk + O(|S| \log_2 |S|) + M(k - C + O(1))$ bits. Mas, como $n + |F|$ bits aleatórios não podem ser comprimidos em um espaço menor que $n + |F|$, nós temos o limite[5]

[5]Por esse motivo, é desejável maximizar a entropia do sistema em um argumento de compressão de entropia.

$$n + |F| - Mk + O(|S| \log_2 |S|) + M(k - C + O(1)) \geq n + |F| \iff$$

$$\iff O(|S| \log_2 |S|) + M(O(1) - C) \geq 0.$$

Logo, se M for grande o suficiente, chegamos à contradição desejada. O teorema 3.2 segue. ■

□

3.2 O Método de Esperet e Parreau

Em [29], em 2013, Louis Esperet e Aline Parreau descreveram um método usando um argumento de compressão de entropia para resolver problemas de coloração onde algumas configurações são proibidas. Seguindo o proposto no artigo, estudaremos tal método pelo problema de

3.2.1 Coloração Acíclica de Elos

Definição 3.3. Uma λ-coloração dos elos de um grafo $G = (V, E)$ é uma função $f : E \to \{1, 2, \ldots, \lambda\}$. Uma λ-coloração é dita acíclica se dois elos adjacentes não possuem a mesma cor e se todo ciclo tem pelo menos três cores.

Definição 3.4. a'(G) é o menor número de cores que podem formar uma coloração acíclica de G.

Fazendo uso do Lema Local de Lovász, Michael Molloy e Bruce Reed, em [50], mostraram que se G tem grau máximo Δ então a'$(G) \leq 64\Delta$. Usando uma versão mais recente do Lema Local devido a Bissacot *el al.* ([11]), Ndreca *el al.*, em [53], mostraram que a'$(G) \leq \lceil 9,62(\Delta - 1) \rceil$[6]. Do método de entropy compression [29] resulta que a'$(G) \leq 4(\Delta - 1)$.

Para tanto, faz-se uso de um simples algoritmo: seja $\gamma > 0$ e $K = \lceil (2 + \gamma)(\Delta - 1) \rceil$. Ordene os elos de G em $e_1, \ldots, e_m$ (onde $m = |E|$). Ao começo de cada passo, selecione o elo e_i não colorido de menor índice e sorteie (uniformemente, por exemplo) uma cor em $\{1, \ldots, K\}$ que não apareça em

[6] Conforme mostrado na proposição 1.20.

um elo adjacente a e_i. Se algum ciclo for colorido com apenas duas cores, descolora esse ciclo. Com o uso do método de entropy compression será mostrado que, dado que K é grande o suficiente, um algoritmo muito similar a esse pára.

O Algoritmo

Para a análise do algoritmo, construiremos aleatoriamente o vetor $F \in \{1, \ldots, \lceil \gamma(\Delta - 1) \rceil\}^t$, para um t grande. No i-ésimo passo do algoritmo, a i-ésima entrada F_i de F será usada para atribuir uma cor ao elo não colorido de menor índice e_j da seguinte forma: seja $e_j = uv$ e $S = \{1, \ldots, K\} \setminus S'$, onde S' é o conjunto das cores atribuídas aos elos $xy \neq uv$ tais que:

(1) $x = u$ ou $x = v$

ou

(2) elos ux e vy existem e tem a mesma cor.

Agora atribuímos o F_i-ésimo elemento de S ao elo e_j. A condição (1) impede que, ao colorir e_j, formem-se dois elos adjacentes com a mesma cor. A condição (2) impede que se forme um ciclo com quatro elos com apenas duas cores, como mostra a figura a seguir.

Figura 1: Representação da condição (2).

Repare que essa instrução está bem definida pois $|S| \geq \lceil \gamma(\Delta - 1) \rceil$. Com efeito, pela figura, é fácil notar que para cada cor contada em (2) uma cor é contada pelo menos duas vezes em (1). Logo S' não contém mais cores do que o número de elos adjacentes a e_j. Logo $|S'| \leq 2(\Delta - 1)$. A condição (1) também impede que se formem ciclos de tamanho ímpar coloridos com apenas duas cores. Se, ao atribuirmos uma cor a e_j, forme-se um ciclo com duas cores (de tamanho 6 ou mais), digamos $e_{i_1}, \ldots, e_{i_{2k}}, e_{i_1}$ com $e_{i_1} = e_j$ e $i_2 < i_{2k}$ então descolorimos todos os elos desse ciclo exceto e_{i_2} e e_{i_3}.

A próxima seção se dedica à análise do algoritmo. Para tanto, definimos o vetor R com t entradas da seguinte forma: Suponha que, no i-ésimo passo do algoritmo, o elo e_j seja colorido e um ciclo de tamanho $2k$ receba apenas duas cores. Como existe um número finito de possíveis ciclos de tamanho $2k$ que contém o elo e_j (no máximo $(\Delta-1)^{2k-2}$), podemos ordená-los em $C_1, \ldots, C_s$ com $s \leq (\Delta-1)^{2k-2}$. Nesse caso, à i-ésima entrada R_i de R será atribuído o par (k,l) onde $l \leq s$ é o índice do ciclo de tamanho $2k$ que contém e_j. Se, no i-ésimo passo do algoritmo, nenhum ciclo for formado então a R_i nada será atribuído. Será de fundamental importância que, com o vetor R e a coloração parcial ao passo i, sejamos capazes de reconstruir o vetor F até sua i-ésima entrada.

A ideia é mostrar que o conjunto de todos os vetores F para os quais o algoritmo não pára antes do passo t é menor que o conjunto de todas as possibilidades de pares de vetor R com a coloração parcial ao passo t. Se mostrarmos que esse último é $o(\lceil\gamma(\Delta-1)\rceil^t)$ então provaremos que o algoritmo pára com probabilidade maior que zero, pois $\lceil\gamma(\Delta-1)\rceil^t$ é justamente o número de escolhas possíveis para o vetor F implicando que o algoritmo pára para algum vetor F.

Análise do Algoritmo

Seja X_i o conjunto dos elos não coloridos após o passo i e ϕ_i a coloração parcial de G depois do passo i. Seja F um vetor para o qual o algoritmo retorna o par (R,ϕ_t). Os próximos dois lemas mostram que o par (R,ϕ_t) determina o vetor F:

Lema 3.5. *A cada passo i, o conjunto X_i é unicamente determinado pelo vetor $(R_j)_{j\leq i}$.*

Prova: Provaremos por indução em i. É imediato que X_1 é unicamente determinado por $(R_j)_{j\leq 1}$ pois $X_1 = E \setminus \{e_1\}$.

Agora, supomos que X_{i-1} é unicamente determinado por $(R_j)_{j\leq i-1}$. Ao passo i, será colorido o elo e_j onde $j = \min\{j : e_j \in X_{i-1}\}$. Se R_i é vazio então $X_i = X_{i-1} \setminus e_j$. Caso contrário, lendo R_i sabemos exatamente quais vértices foram descoloridos nesse passo. Logo, também conhecemos o conjunto X_i. ■

Lema 3.6. *A cada passo i, a função que, a cada vetor $(F_j)_{j\leq i}$, associa o par $((R_j)_{j\leq i}, \phi_i)$ é injetiva.*

Prova: Novamente o argumento será por indução. Após o primeiro passo, ϕ_1 é a configuração em que apenas o primeiro vértice está colorido

e com a cor F_1. Supondo, para $i > 1$, que o lema é verdadeiro para $i-1$ provaremos que ele também vale para i. Pelo lema 3.5, nós conhecemos X_i e X_{i-1}. Em particular nós conhecemos o elo e_j que foi colorido no passo i.

Se R_i é vazio então ϕ_{i-1} é obtido através de ϕ_i apenas descolorindo o elo e_j. Logo, por indução, nós conhecemos o vetor $(F_j)_{j<i}$ bastando determinar a entrada F_i. Seja c a cor atribuída ao elo $e_j = uv$ e seja a o número de diferentes cores $\{i : i < c\}$ que aparecem em ϕ_{i-1} em (1) elos adjacentes a e_j e (2) em elos xy tais que xu e yv são elos de G e possuem a mesma cor. Então $F_i = c - a$.

Agora, seja $R_i = (k, l)$. Então nós conhecemos o ciclo $e_{i_1}, \ldots, e_{i_{2k}}, e_{i_1}$, com $e_{i_1} = e_j$, que foi colorido com duas cores no passo i. Nesse caso, ϕ_{i-1} é obtido a partir de ϕ_i apenas colorindo os elos $e_{i_5}, e_{i_7}, \ldots, e_{i_{2k-1}}$ com a cor $\phi_i(e_{i_3})$ e os elos $e_{i_4}, e_{i_6}, \ldots, e_{i_{2k}}$ com a cor $\phi_i(e_{i_2})$. Além disso, sabemos que, no passo i, e_j recebeu a cor $\phi_i(e_{i_3})$ logo antes de ser descolorido. Pelo mesmo raciocínio acima, $(F_j)_{j \leq i-1}$ está determina pela hipótese de indução e F_i está determinado pois conhecemos a cor que o elo e_j recebeu no passo i. ■

Seja $\mathcal{F}_t$ o conjunto dos vetores F tais que, ao passo t do algoritmo, o grafo G não está completamente colorido. Seja $\mathcal{R}_t$ o conjunto dos vetores R que são gerados a partir de vetores de $\mathcal{F}_t$. Como há no máximo $(K+1)^m$ possibilidades de colorações parciais ϕ_t de G, os dois lemas anteriores implicam que

Lema 3.7. $|\mathcal{F}_t| \leq (K+1)^m |\mathcal{R}_t|$.

O desafio agora é achar uma cota para $|\mathcal{R}_t|$ a fim de mostrar que, para t grande o suficiente, $|\mathcal{F}_t|$ é menor que o conjunto de todos os vetores F possíveis. Implicando que existe um vetor F para o qual o algoritmo pára. Para tanto, transformaremos $|\mathcal{R}_t|$ em um conjunto mais familiar:

Considere uma palavra $w = w_1, \ldots, w_{2k-2}$ no alfabeto $\mathbf{A} = \{1, \ldots, \Delta - 1\}$, e a função $\theta_k(w) = 1 + \sum_{i=1}^{2k-2}(w_i - 1)(\Delta - 1)^{i-1}$. A função θ_k tem imagem igual a $\{1, \ldots, (\Delta-1)^{2k-2}\}$ e é injetiva. Com efeito, seja γ e ψ tais que $\theta_k(\gamma) = \theta_k(\psi)$. Logo

$$\theta_k(\gamma) - \theta_k(\psi) = \sum_{i=1}^{2k-2}(\gamma_i - 1)(\Delta-1)^{i-1} - \sum_{j=1}^{2k-2}(\psi_i - 1)(\Delta-1)^{j-1} =$$

$$= \sum_{i=1}^{2k-2}((\gamma_i - 1)(\Delta-1)^{i-1} - (\psi_i - 1)(\Delta-1)^{i-1}) =$$

$$= \sum_{i=1}^{2k-2} ((\gamma_i - 1) - (\psi_i - 1))(\Delta - 1)^{i-1} = \sum_{i=1}^{2k-2} (\gamma_i - \psi_i)(\Delta - 1)^{i-1}.$$

Mas $|(\gamma_i - \psi_i)| < (\Delta - 1)$, $\forall i \in \{1, \ldots, 2k-2\}$. Assim $\sum_{i=1}^{2k-2}(\gamma_i - \psi_i)(\Delta - 1)^{i-1}$ é uma combinação linear de potências de $(\Delta - 1)$ com coeficientes inteiros e menores, em modulo, que $(\Delta - 1)$. Portanto $\sum_{i=1}^{2k-2}(\gamma_i - \psi_i)(\Delta - 1)^{i-1} = 0 \iff \gamma_i = \psi_i$, $\forall i \in \{1, \ldots, 2k-2\}$ pois esse somatório é uma soma de números na base $(\Delta - 1)$ onde as parcelas são da forma $a00\ldots000$ e existe apenas uma parcela não nula com i zeros à direta para cada i. Logo $\theta_k(w)$ é injetiva. É imediato que $\theta_k(w)$ assume seu valor mínimo quando $w_i = 1$, $\forall i \in \{1, \ldots, 2k-2\}$ - quando $\theta_k(w)$ é igual a 1 - e assume seu valor máximo quando $w_i = (\Delta - 1)$, $\forall i \in \{1, \ldots, 2k-2\}$ - quando $\theta_k(w)$ é igual a $(\Delta-1)^{2k-2}$. Como existem exatamente $(\Delta-1)^{2k-2}$ palavras w diferentes e a função $\theta_k(w)$ é injetiva segue que a função tem imagem $\{1, \ldots, (\Delta-1)^{2k-2}\}$.

Seja $R \in \mathcal{R}_t$ e $R^* = (R_i^*)_{i \leq t}$ uma sequência de t palavras no alfabeto $\mathbf{A}^* = \mathbf{A} \cup \{0\}$ assim definida: para todo $1 \leq i \leq t$, se R_i é vazio então $R_i^* = 0$. Caso contrário, existem k e l tais que $R_i = (k, l)$ e então R_i^* será a concatenação de 0 com $\theta_k^{-1}(l)$ (R_i^* está bem definido pois, lembremos, $l \leq (\Delta - 1)^{2k-2}$). Agora consideramos a sequência de palavras R^* como uma palavra $R^\bullet$ (a concatenação de todas as palavras de R^*) e definimos R° como a palavra em $\{0, 1\}$ onde $R_i^\circ = 0$ se $R_i^\bullet = 0$ e $R_i^\circ = 1$ caso contrário. Por exemplo, se $\Delta = 4$ e

$R = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, (3, 4), \emptyset, \emptyset, \emptyset, (3, 15))$, então teremos
$R^* = (0, 0, 0, 0, 0, 01211, 0, 0, 0, 03221)$,
$R^\bullet = 000000121100003221$, e
$R^\circ = 00000111100001111$.

Observe que $R^* \to R^\bullet$ é uma injeção já que cada entrada de R^* começa com um 0 e não há outros zeros nas palavras de R^*. Logo $R \to R^\bullet$ também é uma injeção. Seguimos agora com algumas observações a respeito das palavras R° que provém de algum $R \in \mathcal{R}_t$.

Definição 3.8. Uma palavra Dyck parcial[7] é uma palavra w no alfabeto $\{0, 1\}$ tal que qualquer prefixo de w contém pelo menos o mesmo número de zeros do que de uns. Uma palavra Dyck de tamanho $2t$ é uma palavra Dyck parcial com t zeros e t uns. Uma descida[8] em uma palavra Dyck (parcial) é uma sequência de uns consecutivos maximal.

[7] Traduzido de *partial Dyck word*.
[8] Traduzido de *descent*.

Lema 3.9. *Para qualquer $R \in \mathcal{R}_t$, a palavra R° é uma palavra Dyck parcial com t zeros e $t-r$ uns, onde r é o número de elos coloridos após o passo t do algoritmo. Ainda, todas as descidas de R° são pares e, se todos os ciclos de G tem tamanho pelos menos $2l+1$, para algum $l > 0$, então todas as descidas em R° têm tamanho pelo menos max(4,2l).*

Prova: Quando lemos R° da esquerda para a direita, cada 0 corresponde a um elo ao qual foi atribuído uma cor e cada 1 corresponde a um elo que foi descolorido. Como não se pode descolorir mais elos do que o número de elos coloridos, a primeira parte do lema segue. A segunda parte vem do fato que se todos os ciclos têm tamanho maior ou igual a $2l+1$ então todos os ciclos pares (os que têm a possibilidade de ser coloridos com apenas duas cores pelo algoritmo) têm tamanho maior ou igual a $2l+2$. Um ciclo colorido com apenas duas cores tem tamanho maior ou igual a 6. Logo cada descida é par e tem tamanho maior ou igual a max(4,2l). ■

Seja $R \in \mathcal{R}_t$. Se a palavra R° tem $t-r$ uns então a pré-imagem de R° pela função $R \to R^\circ$ tem, no máximo, cardinalidade $(\Delta-1)^{t-r}$. Com efeito, como $R \to R^*$ e $R^* \to R^\bullet$ são injeções e cada 1 em R° corresponde a um elemento de $\{1,\ldots,\Delta-1\}$ em $R^\bullet$ o resultado segue.

Seja $\mathcal{R}_t^\circ = \{R^\circ | R \in \mathcal{R}_t\}$. A última afirmação junto com o fato que em uma palavra R° o número de uns não é maior que o número de zeros (lema 3.9) implicam que $|\mathcal{R}_t| \leq (\Delta-1)^t |\mathcal{R}_t^\circ|$. Então, pelo lema 3.7:

Lema 3.10. $|\mathcal{F}_t| \leq (K+1)^m (\Delta-1)^t |\mathcal{R}_t^\circ|$.

Nosso objetivo agora é contar palavras Dyck parciais que tenham as propriedades citadas no lema 3.9. Para facilitar, contaremos, na verdade, palavras Dyck com essas propriedades. O próximo lema mostra que contar esses dois objetos é quase equivalente, dado que r (a diferença entre o número de zeros e o número de uns em uma palavra Dyck parcial) não é muito grande.

Lema 3.11. *Sejam $t, r \in \mathbb{N}$, com $r \leq t$, e $E \neq \{1\}$ um conjunto não vazio de números naturais. Seja $C_{t,r,E}$ (resp. $C_{t,E}$) o número de palavras Dyck parciais com t zeros, $t-r$ uns (resp. palavras Dyck com tamanho $2t$) e todas as descidas tendo tamanho em E. Então $C_{t,r,E} \leq C_{t+r(s-1),E}$, onde $s = min(E \setminus \{1\})$.*

Prova: Seja $\mathbf{D}_{t,r,E}$ (resp. $\mathbf{D}_{t,E}$) o conjunto das palavras Dyck parciais com t zeros, $t-r$ uns (resp. palavras Dyck com tamanho $2t$) e todas as descidas tendo tamanho em E. Seja $\psi : \mathbf{D}_{t,r,E} \to \mathbf{D}_{t+r(s-1),E}$ a função que

anexa ao final de uma palavra a palavra $(0^{s-1}1^s)^r$. É imediato que a função está bem definida e é injetiva o que termina a prova. ■

Existem várias formas de calcular cotas para $C_{t,E}$. Uma delas é construir uma bijeção com alguma estrutura bem conhecida. Nós usaremos uma bijeção com árvores planas[9].

Lema 3.12. *O número $C_{t,E}$ de palavras Dyck com tamanho $2t$ e todas as descidas em E é igual ao número de árvores planas com $t+1$ vértices tais que o grau de cada vértice está em $E \cup \{0\}$.*

Prova: Existem bijeções entre os três seguintes objetos para qualquer $t \in \mathbb{N}$:

1. Árvores planas com $t+1$ vértices tais que o grau de cada vértice está em $E \cup \{0\}$;

2. Palavras Dyck com tamanho $2t$ nas quais o tamanho de qualquer sequência maximal de zeros consecutivos está em E;

3. Palavras Dyck de tamanho $2t$ tais que o tamanho de cada descida está em E.

A bijeção entre 1 e 2 pode ser construida da seguinte maneira: Em um passeio DFS na árvore, escreva, concatenadamente, cada vértice (exceto o último do passeio) tendo i filhos como a palavra 0^i1. É de fácil observação que a palavra obtida é uma palavra Dyck de tamanho $2t$ e com cada sequência maximal de zeros consecutivos em E. Também é fácil observar que essa é uma relação de bijeção. A bijeção entre 2 e 3 se dá pelo processo de pegar o espelho de uma palavra em 2 e trocar os zeros por uns e os uns por zeros. ■

Agora, usamos funções geradoras e o método descrito por Drmota em [23] para estimar $C_{t,E}$. Seja $X_E(z)$ a função geradora ordinária (ver apêndice B) associada ao número de árvores planas com $t+1$ vértices tais que o grau de cada vértice está em $E \cup \{0\}$. Pelo lema anterior, $X_E(z) = z\sum_{t\in\mathbb{N}} C_{t,E} z^t$. Mas uma árvore plana como descrita acima é ou um único vértice ou, para algum $i \in E$, um vértice ligado a uma sequência de i árvores planas tais que o grau de cada vértice está em $E \cup \{0\}$. Logo, pelo método simbólico descrito no apêndice B, $X_E(z) = z(1 + \sum_{i\in E} X_E(z)^i) = z\phi_E(X_E(z))$, com

[9]Uma árvore plana é uma árvore enraizada na qual uma ordem é dada para os filhos de cada vértice. Tal nome vem do fato que atribuir uma ordem aos filhos em uma árvore enraizada t é equivalente a fixar t ao plano.

$\phi_E(x) = 1 + \sum_{i \in E} x^i$. O próximo lema é um corolário direto do teorema B.8.

Lema 3.13. *Sejam $E \neq \{1\}$ um conjunto não vazio de números inteiros não negativos e $\phi_E(x) := 1+\sum_{i \in E} x^i$ tais que a equação $\phi_E(x)-x\phi_E^{'}(x) = 0$ tem uma solução $x = \tau$, com $0 < \tau < R$, onde R é o raio de convergência de ϕ_E. Então τ é a única solução da equação no intervalo aberto $(0, R)$ e existe uma constante c_E tal que $C_{t,E} \leq c_E \gamma^t t^{-\frac{3}{2}}$, onde $\gamma = \phi_E^{'}(\tau) = \phi_E(\tau)/\tau$.*

Agora estamos aptos a estabelecer cotas para a'(G). Lembre que a cintura de um grafo G é o tamanho de seu ciclo mais curto (se G não tem ciclos então G tem cintura infinita).

Teorema 3.14. *Sejam $l \in \mathbb{N}$, com $l \geq 1$, $k = max(2, l)$. Então o polinômio $P(x) = (2k-3)x^{2k+2} + (1-2k)x^{2k} + x^4 - 2x^2 + 1$ tem uma única raiz τ no intervalo aberto $(0, 1)$ e todo grafo com grau máximo Δ e cintura maior ou igual a $2l + 1$ tem uma coloração acíclica de elos com no máximo $\lceil(2 + \gamma)(\Delta - 1)\rceil$ cores, onde $\gamma = (\tau^{2k} - \tau^2 + 1)/(\tau - \tau^3)$.*

Prova: Seja $E = 2\mathbb{N}+2k$. Então $\phi_E(x) = 1+\sum_{i \in E} x^i = 1+\frac{x^{2k}}{1-x^2}$. Logo $\phi_E^{'}(x) = (2kx^{2k-1}-(2k-2)x^{2k+1})/(1-x^2)^2$, e a equação $\phi_E(x)-x\phi_E^{'}(x) = 0$ é equivalente a $P(x) = 0$. O raio de convergência de ϕ_E é 1 e como $P(0) = 1$ e $P(1) = -2$ o polinômio P tem uma raiz τ no intervalo aberto $(0, 1)$. Pelo lema 3.13, essa é a única raiz em $(0, 1)$. Também pelo lema 3.13, sabemos que existe uma constante c_E tal que $C_{t,E} \leq c_E \gamma^t t^{-\frac{3}{2}}$, onde $\gamma = \phi_E^{'}(\tau) = \phi_E(\tau)/\tau = (\tau^{2k} - \tau^2 + 1)/(\tau - \tau^3)$.

Para provar o teorema, precisamos mostrar a existência de um vetor $F \in \{1, \ldots, \lceil\gamma(\Delta - 1)\rceil\}^t$ tal que o algoritmo, recebendo G e F como entradas, devolve uma coloração acíclica dos elos de G. Pelo lema 3.10, $|\mathcal{F}_t| \leq (\lceil(2+\gamma)(\Delta-1)\rceil+1)^m(\Delta-1)^t|\mathcal{R}_t^\circ|$. Observe que, para cada $R \in \mathcal{R}_t$, o número de zeros e uns em cada prefixo de R° difere de no máximo $m - 1$ (onde m é o número de elos de G) já que no máximo $m - 1$ elos estão coloridos a cada passo do algoritmo. Pelos lemas 3.9 e 3.11, isso implica que $|\mathcal{R}_t^\circ| \leq \sum_{r=0}^{m-1} C_{t+r(2k-1),E} \leq c_E^{'}\gamma^{t+m(2k-1)}t^{-\frac{3}{2}}$, onde $c_E^{'} = c_E/(\gamma^{2k-1} - 1)$. Isso implica que $|\mathcal{F}_t| \leq c_E^{'}(\lceil(2 + \gamma)(\Delta - 1)\rceil + 1)^m(\Delta - 1)^t\gamma^{t+m(2k-1)}t^{-\frac{3}{2}}$, e $|\mathcal{F}_t|/\lceil\gamma(\Delta - 1)\rceil^t$ tende a zero quando t tende a infinito. Em particular, para t grande o suficiente $|\mathcal{F}_t| < \lceil\gamma(\Delta - 1)\rceil^t$, o que significa que existe um vetor F para o qual o algoritmo para em menos de t passos e devolve uma coloração acíclica dos elos de G com no máximo $\lceil(2 + \gamma)(\Delta - 1)\rceil$ cores. ■

Muthu *et al.* em [52] provaram, em 2007, que grafos com grau máximo não maior que Δ e cintura no mínimo 9 têm uma coloração acíclica de elos

com no máximo 6Δ cores e grafos com cintura no mínimo 220 essa cota foi melhorada para 4.52Δ. Ndreca *et al.* em [53] mostraram as seguintes cotas para grafos com grau máximo Δ e cintura g: a'$(G) \leq \lceil 9.62(\Delta - 1)\rceil$, a'$(G) \leq \lceil 6.42(\Delta - 1)\rceil$, se $g \geq 5$, a'$(G) \leq \lceil 5.77(\Delta - 1)\rceil$, se $g \geq 7$ e a'(G) $\leq \lceil 4.52(\Delta - 1)\rceil$, se $g \geq 53$. O próximo corolário segue diretamente do teorema 3.12 e melhora significativamente todas essas cotas.

Corolário 3.15. *Seja G um grafo de grau máximo Δ e cintura g. Então*

1. a'$(G) \leq 4\Delta - 4$;
2. se $g \geq 7$, a'$(G) \leq \lceil 3.74(\Delta - 1)\rceil$;
3. se $g \geq 53$, a'$(G) \leq \lceil 3.14(\Delta - 1)\rceil$;
4. se $g \geq 220$, a'$(G) \leq \lceil 3.05(\Delta - 1)\rceil$.

A tabela 3.1 mostra os valores de τ e γ bem como E e $P(x)$ referentes aos casos mencionados no corolário 3.13.

g	E	$P(x)$	τ	γ
3	$2\mathbb{N}+4$	$x^6 - 2x^4 - 2x^2 + 1$	$\frac{1}{2}(\sqrt{5}-1)$	2
7	$2\mathbb{N}+6$	$3x^8 - 5x^6 + x^4 - 2x^2 + 1$	0.66336	1.73688
53	$2\mathbb{N}+52$	$49x^{54} - 51x^{52} + x^4 - 2x^2 + 1$	0.89610	1.13481
220	$2\mathbb{N}+218$	$215x^{220} - 217x^{218} + x^4 - 2x^2 + 1$	0.96341	1.04225

Tabela 3.1: Valores referentes ao corolário 3.15.

3.2.2 O Caso Geral de [29]

O método apresentado na seção anterior pode ser aplicado a qualquer problema de coloração de vértices (ou elos) que:

i possa ser definido como colorações que evitam um conjunto de configurações proibidas de cores – uma configuração é um par (H, c) onde H é um grafo com uma coloração c.

ii para cada configuração ruim (H, c), para cada vértice v de H exista um número k de vértices diferentes de v em H para os quais se soubermos suas cores existe uma única extensão dessa coloração parcial de H para uma coloração congruente a c.

Seja o conjunto das configurações proibidas indexado por um conjunto $I \in \mathbb{N}$. O objetivo é encontrar uma coloração C do grafo G tal que, para qualquer $i \in I$, e qualquer cópia H de um grafo H_i em G, a restrição de C a H não seja congruente a c_i (duas colorações do mesmo grafo são ditas congruentes se uma pode ser obtida da outra por uma simples permutação de nomes das cores).

Ainda, para cada i, sejam $l_i = |V(H_i)| - k_i$ e $E = \{l \in \mathbb{N} | \exists\ i, l_i = l\}$. Para $l \in E$, seja d_l o máximo, entre os vértices v de G, do número de subgrafos que contém o vértice v e são isomorfos a algum H_i com $l_i = l$. Seja γ definido como no lema 3.13 da seção anterior usando esse conjunto E. Analisando da mesma forma que anteriormente, podemos provar que existe uma coloração de G com $\gamma \cdot \sup_{l \in E} d_l^{1/l}$ cores, tal que nenhuma cópia de H_i tem uma coloração congruente a c_i, para qualquer i.

Prova do Caso Geral

Considere um problema de coloração de vértices de V de um grafo G com conjunto $\mathcal{H} = \{H_1, \ldots, H_n\}$ de configurações proibidas. Para cada H_i, sejam k_i, l_i, E, d_l e γ como definidos acima.

O Algoritmo

Considere o seguinte algoritmo: colete t amostras aleatórias da distribuição discreta Uniforme($\lceil \gamma \sup_{l \in E} d_l^{1/l} \rceil$) e armazene-as no vetor F. Como anteriormente, esse vetor fornecerá as entradas do algoritmo. Estabeleça uma ordem para o conjunto de vértices V. Ao i-ésimo passo do algoritmo, atribua a i-ésima entrada do vetor F ao vértice v_j descolorido de menor índice. Se, ao fazê-lo, todos os vértices de V ficam coloridos e nenhuma configuração proibida é criada, o algoritmo pára. Caso contrário, se nenhuma configuração proibida é criada mas ainda restam vértices a serem coloridos, o algoritmo segue para o próximo passo. Finalmente, se ao colorir o vértice no i-ésimo passo uma configuração ruim H_s é criada, descolore-se os l_s vértices de H_s. Ainda, define-se uma ordem aos no máximo d_{l_s} subgrafos de G que contém v_j e são isomorfos a algum H_r com $l_r = l_s$ e atribui-se a i-ésima entrada de um vetor R um par contendo l_s e a ordem do subgrafo observado em tal passo. O algoritmo então segue para o próximo passo.

Análise do Algoritmo

Seja X_i o conjunto dos vértices não coloridos após o passo i e ϕ_i a coloração parcial de G depois do passo i. Seja F um vetor para o qual o

algoritmo retorna o par (R,ϕ_t). Os próximos dois lemas mostram que o par (R,ϕ_t) determina o vetor F.

Lema 3.16. *A cada passo i, o conjunto X_i é unicamente determinado pelo vetor $(R_j)_{j\leq i}$.*

Prova: Provaremos por indução em i. É imediato que X_1 é unicamente determinado por $(R_j)_{j\leq 1}$ pois $X_1 = V \setminus \{v_1\}$.

Agora, supomos que X_{i-1} é unicamente determinado por $(R_j)_{j\leq i-1}$. Ao passo i, será colorido o vértice v_j onde $j = min(j : v_j \in X_{i-1})$. Se R_i é vazio então $X_i = X_{i-1} \setminus \{v_j\}$. Caso contrário, com as entradas de R_i nós sabemos exatamente qual configuração proibida foi gerada e em qual subgrafo de G. Portanto sabemos quais vértices foram descoloridos. Logo X_i também está unicamente determinado. ■

Lema 3.17. *A cada passo i, a função que, a cada vetor $(F_j)_{j\leq i}$, associa o par $((R_j)_{j\leq i}, \phi_i)$ é injetiva.*

Prova: Novamente o argumento será por indução. Após o primeiro passo, ϕ_1 é a configuração em que apenas o primeiro vértice está colorido e com a cor F_1. Supondo, para $i > 1$, que o lema é verdadeiro para $i-1$ provaremos que ele também vale para i. Pelo lema 3.16 nós conhecemos X_i e X_{i-1}. Em particular nós conhecemos o vértice v_j que foi colorido no passo i.

Se R_i é vazio então ϕ_{i-1} é obtido através de ϕ_i apenas descolorindo o vértice v_j. Logo, por indução, nós conhecemos o vetor $(F_j)_{j<i}$ bastando determinar a entrada F_i. Claramente, F_i é a cor atribuída a v_j em ϕ_i.

Agora, seja $R_i = (l, q)$. Logo nós sabemos qual configuração proibida H_s envolvendo v_j foi criada no passo i. Nesse caso, ϕ_{i-1} é obtido a partir de ϕ_i apenas colorindo os l_s vértices descoloridos no passo i da única forma que gera H_s. Pelo mesmo raciocínio acima, $(F_j)_{j\leq i-1}$ está determinado pela hipótese de indução e F_i está determinado pois conhecemos a cor que o vértice v_j recebeu no passo i (pois só há uma maneira de colorir os l_s vértices de forma a gerar H_s). ■

Seja $\mathcal{F}_t$ o conjunto dos vetores F tais que, ao passo t do algoritmo, o grafo G não está completamente colorido. Seja $\mathcal{R}_t$ o conjunto dos vetores R que são gerados pelo vetores de $\mathcal{F}_t$. Como há no máximo $(\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil + 1)^{|V|}$ possibilidades de colorações parciais ϕ_t de G, os dois lemas anteriores implicam que

Lema 3.18. $|\mathcal{F}_t| \leq (\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil + 1)^{|V|} |\mathcal{R}_t|$.

O desafio agora é achar uma cota para $|\mathcal{R}_t|$ a fim de mostrar que, para t grande o suficiente, $|\mathcal{F}_t|$ é menor que o conjunto de todos os vetores F possíveis implicando que existe um vetor F para o qual o algoritmo pára. Novamente, transformaremos $|\mathcal{R}_t|$ em um conjunto mais familiar:

Considere uma palavra $w = w_1, \ldots, w_l$ no alfabeto $\mathbf{A} = \{1, \ldots, d_l^{1/l}\}$, e a função $\theta_l(w) = 1 + \sum_{i=1}^{l}(w_i - 1)(d_l^{1/l})^{i-1}$. A função θ_l tem imagem igual a $\{1, \ldots, d_l\}$ e é injetiva (com prova muito similar a realizada anteriormente).

Seja $R \in \mathcal{R}_t$ e $R^* = (R_i^*)_{i \leq t}$ uma sequência de t palavras no alfabeto $\mathbf{A}^* = \mathbf{A} \cup \{0\}$ assim definida: para todo $1 \leq i \leq t$, se R_i é vazio então $R_i^* = 0$. Caso contrário, seja $R = (l, q)$ com $1 \leq q \leq d_l$. Então R_i^* será a concatenação de 0 com $\theta_l^{-1}(q)$. Agora consideramos a sequência de palavras R^* como uma palavra $R^\bullet$ (a concatenação de todas as palavras de R^*) e definimos R° como a palavra em $\{0, 1\}$ onde $R_i^\circ = 0$ se $R_i^\bullet = 0$ e $R_i^\circ = 1$ caso contrário. Observe que $R^* \to R^\bullet$ é uma injeção já que cada entrada de R^* começa com um 0 e não há outros zeros nas palavras de R^*. Logo $R \to R^\bullet$ também é uma injeção. Usaremos novamente o conceito de palavras Dyck para analisar $|R^\circ|$.

Lema 3.19. *Para qualquer $R \in \mathcal{R}_t$, a palavra R° é uma palavra Dyck parcial com t zeros e $t - r$ uns, onde r é o número de vértices coloridos após o passo t do algoritmo. Ainda, E é o conjunto dos tamanhos de descidas de R°.*

Prova: Quando lemos R° da esquerda para a direita, cada 0 corresponde a um vértice ao qual foi atribuído uma cor e cada 1 corresponde a um vértice que foi descolorido. Como não se pode descolorir mais vértices do que o número de vértices coloridos, a primeira parte do lema segue. A segunda parte segue da definição do conjunto E. ■

Seja $R \in \mathcal{R}_t$. Se a palavra R° tem $t - r$ uns então a pré-imagem de R° pela função $R \to R^\circ$ tem, no máximo, cardinalidade $(d_l^{1/l})^{t-r}$. Com efeito, como $R \to R^*$ e $R^* \to R^\bullet$ são injeções e cada 1 em R° corresponde a um elemento de $\{1, \ldots, d_l^{1/l}\}$ em $R^\bullet$ o resultado segue.

Seja $\mathcal{R}_t^\circ = \{R^\circ | R \in \mathcal{R}_t\}$. A última afirmação junto com o fato que em uma palavra R° o número de uns não é maior que o número de zeros (lema 3.19) implicam que $|\mathcal{R}_t| \leq (\sup_{l \in E} d_l^{1/l})^t |\mathcal{R}_t^\circ|$. Então, pelo lema 3.18:

Lema 3.20. $|\mathcal{F}_t| \leq (\lceil \gamma \sup_{l \in E} d_l^{1/l} \rceil + 1)^{|V|} (\sup_{l \in E} d_l^{1/l})^t |\mathcal{R}_t^\circ|$.

Nosso objetivo agora é contar palavras Dyck parciais que tenham as propriedades citadas no lema 3.19. Nós seguiremos a mesma metodologia

usada anteriormente. Em particular, usaremos o lema 3.13 supondo que existe τ solução da equação $\phi_E(x) - x\phi_E^{'}(x) = 0$ para finalmente provar o

Teorema 3.21. *Existe uma coloração dos vértices de G com $\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil$ cores que não contém subgrafos congruentes a alguma configuração proibida.*

Prova: Para provar o teorema, precisamos mostrar a existência de um vetor $F \in \{1, \ldots, \gamma \sup_{l\in E} d_l^{1/l}\}^t$ tal que o algoritmo, recebendo G e F como entradas, devolve uma coloração de G que evita configurações proibidas. Pelo lema 3.20, $|\mathcal{F}_t| \leq (\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil + 1)^{|V|} (\sup_{l\in E} d_l^{1/l})^t |\mathcal{R}_t^{\circ}|$. Observe que, para cada $R \in \mathcal{R}_t$, o número de zeros e uns em cada prefixo de R° difere de no máximo $|V| - 1$ já que no máximo $|V| - 1$ vértices estão coloridos a cada passo do algoritmo. Pelos lemas 3.19 e 3.11, isso implica que $|\mathcal{R}_t^{\circ}| \leq \sum_{r=0}^{|V|-1} C_{t+r(s-1),E} \leq c_E^{'} \gamma^{t+|V|(s-1)} t^{-\frac{3}{2}}$, onde $s = min(E \setminus \{1\})$ e $c_E^{'} = c_E/(\gamma^{s-1} - 1)$, com c_E e γ como definidos na seção anterior. Isso implica que $|\mathcal{F}_t| \leq c_E^{'} (\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil + 1)^{|V|} (\sup_{l\in E} d_l^{1/l})^t \gamma^{t+|V|(s-1)} t^{-\frac{3}{2}}$ logo $|\mathcal{F}_t| / \lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil^t$ tende a zero quando t tende a infinito. Em particular, para t grande o suficiente, $|\mathcal{F}_t| < \lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil^t$, o que significa que existe um vetor F para o qual o algoritmo para em menos de t passos e devolve uma coloração dos vértices de G que evita todas as configurações proibidas com no máximo $\lceil \gamma \sup_{l\in E} d_l^{1/l} \rceil$ cores. ■

Exemplo 3.22. Coloquemos o problema de coloração acíclica de elos na linguagem que acabamos de apresentar: as configurações proibidas são H_1 = "*um caminho com dois elos com uma única cor*" e, para cada $i \geq 2$, H_i = *um ciclo com 2i elos colorido com apenas duas cores (sem que ocorra* H_1). Logo $k_1 = l_1 = 1$ e, para cada $i \geq 2$, $k_i = 2$ e $l_i = 2i - 2$. Isso implica que $E = \{1\} \cup 2\mathbb{N} + 2$ e $\phi_E(x) = 1 + x + \frac{x^2}{1-x^2}$. Assim obtemos $\gamma = 3.6$. Nós temos $d_1 \leq 2\Delta$ e, para cada $i \geq 2$, $d_{2i-2} \leq \Delta^{2i-2}$. Então todo grafo com grau máximo Δ tem uma coloração acíclica de elos com $3.6 \cdot 2\Delta = 7.2\Delta$ cores. Tal cota é pior do que a alcançada pelo teorema 3.12 pois nele configurações menores foram tratadas de outra maneira de forma a diminuir sua influência no resultado final. □

3.2.3 Variable Version via Compressão de Entropia

Nessa seção, desenvolvemos uma variação do método descrito em [29] e provamos que essa variação se aplica a todo problema na forma variable version do LLL. Na sequência, relaxamos as exigências da seção anterior

não requisitando a restrição ii. Isto é, provaremos que o método de entropy compression pode ser aplicado a todo problema de coloração em um grafo $G = (V, E)$ tal que

i existem *colorações ruins* a serem evitadas. Por coloração ruim, nos referimos a um par (Z, c) onde $Z \subset V$ e c é uma dada coloração de Z.

Provaremos que todo problema na linguagem variable version do Lema Local de Lovász se enquadra nessa categoria. Para tanto precisamos do seguinte vocabulário:

Seja G um grafo (ou a hipergrafo) com conjunto de vértices finito V para o qual estamos buscando uma coloração que evite um conjunto de *colorações ruins*. Uma coloração ruim é um par (Z, c) onde $Z \subset V$ e c é uma dada coloração de Z. Sejam:

- $\mathcal{Z} := \{Z :$ Existe c para o qual (Z,c) é uma coloração ruim$\}$ indexado por um conjunto finito$I \subset \mathbb{N}$.
- Para cada $i \in I$:
  - $C_i := \{c : (Z_i, c)$ é uma coloração ruim$\}$.
  - $\mathscr{Z}_i := \{(Z_i, c) : c \in C_i\}$.
  - Para cada v de Z_i, escolheremos k_i vértices fixos diferentes de v para os quais, se soubermos suas cores, existem no máximo m_i maneiras de estender essa coloração parcial a uma coloração de Z_i a alguma coloração $c \in C_i$. O que sempre é possível pois trabalharemos com um número finito de cores.
  - $l_i := |Z_i| - k_i$.
- $E := \{l \in \mathbb{N} : \exists\, i \in I$ tal que $l_i = l\}$.
- Para cada $l \in E$ e $v \in G$, seja $Q_l(v)$ o conjunto de subconjuntos de V que contém v e são iguais a algum $Z_j \in \mathcal{Z}$ com $l_j = l$. Seja ainda

$$d_l := \max_{v \in V} |Q_l(v)|.$$

O Algoritmo

Sejam $\Psi = \{\psi_1, \ldots, \psi_n\}$ um conjunto de variáveis aleatórias discretas independentes com imagem finita e $\mathcal{A} = \{A_x : x \in X\}$ uma família de eventos

(ruins) determinados por tais variáveis[10]. Para cada $x \in X$, seja $c(A_x)$ o conjunto de configurações das variáveis de $\mathrm{vbl}(A_x)$[11] que fazem A_x ocorrer. Para aplicar o entropy compression basta definirmos as classes de colorações ruins. Seja G um grafo com $V(G) = \Psi$. Nós olharemos para o problema de achar uma configuração de Ψ que evite todos os evento de $\mathcal{A}$ como o problema de achar uma coloração de $V(G)$ que evite todas as colorações ruins, onde dizemos que se para algum $i \in [n]$ uma variável ϕ_i é amostrada então o valor amostrado é a cor de ϕ_i. Logo, $\mathcal{Z} = \mathcal{A}$, $I = X$ e $\mathscr{Z}_x = \{(vbl(A_x), c) : c \in c(A_x)\}$. Agora, sejam $\Psi_1, \ldots, \Psi_n$ vetores que armazenam t amostragens das variáveis $\psi_1, \ldots, \psi_n$, respectivamente e $x_{\psi_1}, \ldots, x_{\psi_n}$ variáveis que podem assumir valores em $Im(\psi_1) \cup \{\circ\}, \ldots, Im(\psi_n) \cup \{\circ\}$, respectivamente. O elemento $\circ$ é um caractere com o qual todas as variáveis $x_{\psi_1}, \ldots, x_{\psi_n}$ são inicializadas. Para cada $j \in [n]$ e cada $l \in E$, defini-se uma ordem no conjunto $\{vbl(A_x) : x \in X,\ \psi_j \in \mathrm{vbl}(A_x),\ l_x = l\}$ – cuja cardinalidade é no máximo d_l. Para cada $x \in X$, definimos também uma ordem para as m_x possíveis formas de reamostrar as l_x variáveis de $vbl(A_x)$ recriando uma coloração ruim de $\mathcal{H}_x$. No i-ésimo passo do algoritmo, atribua a primeira entrada ainda não utilizada do vetor Ψ_j à variável x_{ψ_j} com valor $\circ$ de menor índice. Se, ao fazê-lo, todas as variáveis ficam diferentes de $\circ$ e nenhuma coloração ruim é criada, o algoritmo pára. Caso contrário, se nenhuma coloração ruim é criada mas ainda restam variáveis com o valor $\circ$ atribuído, o algoritmo segue para o próximo passo. Finalmente, se uma coloração ruim $(vbl(A_y), c)$ é criada, atribuí-se o valor $\circ$ às l_y variáveis de $vbl(A_y)$ (note que, desta forma, a cor de ψ_s é o atual valor da variável x_s, para cada $s \in [n]$). Ainda, considere um vetor R para o qual atribui-se a i-ésima entrada uma terna contendo $\alpha = l_y$, $\beta \in [d_{l_y}]$ – que nos fala a ordem de $vbl(A_y)$ em $\{vbl(A_y) : x \in X,\ \psi_j \in \mathrm{vbl}(A_x),\ l_x = l_y\}$ e $\gamma \in [m_y]$ – que nos fala quais eram os valores das l_y variáveis de $vbl(A_y)$. O algoritmo então segue para o próximo passo. O vetor R servirá de registro dado que será de fundamental importância para a análise do algoritmo guardar a informação de o que ocorreu durante sua execução.

Observação 3.23. A diferença em relação à abordagem original de Esperet e Parreau [29] é a adição de uma nova informação no registro R: a etiqueta γ de configurações parciais. Dado que em geral trabalharemos com um número finito de parâmetros (vértices, cores, etc.), sempre podemos inicialmente fixar uma ordem para as configurações ruins parciais de forma a evitar a restrição ii da seção anterior evitando o número $k_{(H,c)}$ da abordagem original.

[10] Para aplicar o LLL, a partir desse ponto, basta construir o grafo de dependência G para a família de eventos $\mathcal{A}$ e encontrar os números $(r_x)_{x \in X}$ em [0,1) tais que, para cada $x \in X$,

$$\mathbf{P}(A_x) = p_x \leq r_x \prod_{y \in \Gamma(x)} (1 - r_y).$$

[11] Ver [51].

□

Análise do Algoritmo

Seja X_i o conjunto das variáveis com valor $\circ$ após o passo i e ϕ_i a configuração das variáveis no passo i. Sejam $\Psi_1, \ldots, \Psi_n$ vetores para os quais o algoritmo retorna o par (R,ϕ_t). Essa seção se limitará a provar os próximos dois lemas que mostram que o par (R,ϕ_t) determina os vetores $\Psi_1, \ldots, \Psi_n$.

Lema 3.24. *A cada passo i, o conjunto X_i é unicamente determinado pelo vetor $(R_j)_{j\leq i}$.*

Prova: Provaremos por indução em i. É imediato que X_1 é unicamente determinado por $(R_j)_{j\leq 1}$ pois $X_1 = \{x_{\psi_2}, \ldots, x_{\psi_n}\}$.

Agora, supomos que X_{i-1} é unicamente determinado por $(R_j)_{j\leq i-1}$. Ao passo i, será atribuído um valor à variável x_{ψ_j}, onde $j = min(j : x_{\psi_j} \in X_{i-1})$. Se R_i é vazio então $X_i = X_{i-1} \setminus \{x_{\psi_j}\}$. Caso contrário, com as entradas de R_i nós sabemos exatamente qual coloração ruim foi gerada e em qual conjunto de v.a.. Portanto sabemos quais variáveis receberam o valor $\circ$. Logo X_i também está unicamente determinado. ■

Lema 3.25. *$((R_j)_{j\leq i}, \phi_i)$ armazena informação suficiente para concluir quais cores foram usadas em cada passo do algoritmo até o passo i.*

Prova: Novamente o argumento será por indução. Após o primeiro passo, ϕ_1 é a configuração em que apenas a primeira variável tem o valor diferente de $\circ$ e vale $(\Psi_1)_1$. Supondo, para $i > 1$, que o lema é verdadeiro para $i-1$ provaremos que ele também vale para i. Pelo lema 3.23, nós conhecemos X_{i-1}. Em particular, nós conhecemos a variável x_{ψ_j} que recebeu um valor no passo i. Se R_i é vazio então ϕ_{i-1} é obtido através de ϕ_i apenas atribuindo o valor $\circ$ à variável x_{ψ_j}. Logo, pela hipótese de indução, nós sabemos quais cores foram utilizadas em cada passo até o passo $i-1$. Claramente, a cor utilizada no passo i é o valor de x_{ψ_j} em ϕ_i. Agora, seja $R_i = (\alpha, \beta, \gamma)$. Então nós sabemos qual coloração ruim $(vbl(A_y), c)$ envolvendo ψ_j foi criada no passo i e quais eram os valores de suas l_y variáveis. Nesse caso, ϕ_{i-1} é obtido a partir de ϕ_i apenas atribuindo esses valores às l_y variáveis que receberam o valor $\circ$ no passo i. Pelo mesmo raciocínio acima, Portanto novamente conhecemos as cores utilizadas até o passo $i-1$ pela hipótese de indução. Por γ nós conhecemos a cor atribuída à ψ_j no passo i.
■

A partir desse ponto a análise segue de perto a análise feita na seção anterior e resulta no seguinte teorema:

Teorema 3.26. *Seja G um grafo (ou a hipergrafo) com conjunto de vértices finito V para o qual estamos buscando uma coloração que evite um conjunto de colorações ruins. Uma coloração ruim é um par (Z, c) onde $Z \subset V$ e c é uma dada coloração de Z. Sejam:*

- $\mathcal{Z} := \{Z :$ *Existe c para o qual (Z,c) é uma coloração ruim*$\}$ *indexado por um conjunto finitoI* $\subset \mathbb{N}$.
- *Para cada* $i \in I$*:*
  - $C_i := \{c : (Z_i, c)$ *é uma coloração ruim*$\}$.
  - *Para cada v de Z_i, escolheremos k_i vértices fixos diferentes de v para os quais, se soubermos suas cores, existem no máximo m_i maneiras de estender essa coloração parcial a uma coloração de Z_i a alguma coloração $c \in C_i$. O que sempre é possível pois trabalharemos com um número finito de cores.*
  - $l_i := |Z_i| - k_i$.
- $E := \{l \in \mathbb{N} : \exists\, i \in I$ *tal que* $l_i = l\}$.
- *Para cada $l \in E$ e $v \in G$, seja $Q_l(v)$ o conjunto de subconjuntos de V que contém v e são iguais a algum $Z_j \in \mathcal{Z}$ com $l_j = l$. Seja ainda* $d_l := \max_{v \in V} |Q_l(v)|$.
- $\phi_E(x) := 1 + \sum_{i \in E} x^i$ *e* $\gamma := \phi_E'(\tau)$.

Então existe uma coloração dos vértices de G usando $\lceil \gamma \sup_{i \in I} (d_{l_i} m_i)^{1/l_i} \rceil$ cores que evita todos os eventos ruins.

3.2.4 O Código do Entropy Compression

Como visto, o método descrito em [29] faz uso da teoria da informação – em particular, da teoria de compressão de dados – mas não explicita o código[12] utilizado. Isso será feito nessa seção.

A mensagem a ser codificada é o vetor F das entradas do algoritmo[13]. F é formado pela coleção de t amostragens independentes da variável aleatória

[12] O leitor interessado é convidado a visitar o apêndice C.

[13] Em geral, a mensagem é composta também pela configuração inicial do sistema mas esta, em [29], não é aleatória (o sistema começa com todos os vértices/elos descoloridos).

discreta X com distribuição Uniforme(c), onde c é o número de cores disponíveis. Aqui fica clara a principal dificuldade na hora de definir o código: seu domínio é a imagem de X. O problema é que, se atribuirmos uma palavra para cada cor, cada vez que uma cor for usada pelo algoritmo ela vai ser codificada pela mesma palavra. Porém são duas situações distintas atribuir uma cor para um elo e formar ou não uma configuração proibida, logo esse evento deve ser codificado de maneira diferente (lembrando que nossa missão é definir um código univocamente decodificável). Precisaremos de uma estrutura um pouco mais complexa para definir tal código.

Sem perda de generalidade, suponha que estamos colorindo os vértices de um grafo $G = \{V, E\}$. Sejam C o conjunto das c cores disponíveis , Φ o conjunto de todas as configurações possíveis, $\mathbf{R}$ o conjunto de todas as entradas possíveis do registro R (incluindo o vazio), D o alfabeto binário e D^* o conjunto de todas sequências finitas de símbolos do alfabeto D.

Considere a variável aleatória $Y : \Phi \times C \to \mathbb{R}$ que associa: o número 1 a todos os pares que não formam uma configuração proibida se a cor for associada ao vértice não colorido de menor índice da configuração deixando ainda vértices a serem coloridos e aos pares em que não há vértices não coloridos na configuração (ou seja, aos pares em que a configuração já atingiu uma coloração que evita todos os pares ruins[14]); a todos os pares que geram um registro distinto, associa um elemento distinto do conjunto $\mathbb{N} \setminus \{1\}$ – chamemos o conjunto dos números utilizados para os registros de $\mathcal{R}$; a todos os pares que colorem por completo de forma distinta o conjunto V evitando configurações ruins associe um elemento distinto do conjunto $\mathbb{N} \setminus (\{1\} \cup \mathcal{R})$. Digamos, por simplicidade, que $Im(Y) = [K]$, para algum $K \in \mathbb{N}$.

Agora, usando as informações do problema de coloração em questão, devemos calcular as probabilidades da v.a. Y assumir cada valor de $[K]$. Chamemos esses valores de p_i, para cada $i \in [K]$. Conforme exposto na seção C.3, para definir o melhor código univocamente decodificável para Y devemos achar a distribuição q(Y) 2-ádica que mais se aproxime, no sentido da entropia relativa, da distribuição de p(Y), isto é, devemos achar os valores das probabilidades q_i que minimizem

$$D(p||q) = \sum_{i=1}^{K} p_i \log \frac{p_i}{q_i} = E_p[\log \frac{p(Y)}{q(Y)}].$$

Agora podemos definir o código $C_Y : [K] \to D^*$, onde, para cada $i \in [K]$, os comprimentos das palavras são dados por

$$l_i = -\log_D q_i.$$

[14]Incluímos os pares em que não há vértices não coloridos na configuração por uma questão de formalidade.

Com o código C_Y definido, basta um pouco de formalidade para chegarmos ao nosso código: para cada $i \in [t]$, sejam X_i a variável aleatória que amostrará a i-ésima entrada do vetor F e ϕ_i a configuração no i-ésimo passo do algoritmo. Sejam C_{X_i} os códigos definidos por

$$\begin{aligned} C_{X_i} : [c] &\to D^* \\ x &\mapsto C_Y(\phi_i, x) \end{aligned}$$

O código desejado é dado por

$$\begin{aligned} C_F : [c] \times \overbrace{\ldots}^{\text{t vezes}} \times [c] &\to D^* \times \overbrace{\ldots}^{\text{t vezes}} \times D^* \\ (x_1, \ldots, x_t) &\mapsto C_{X_1}(x_1) \ldots C_{X_t}(x_t) \end{aligned}$$

$\diamond \ \diamond \ \diamond$

Capítulo 4

Novos Resultados para $c(P_n)$

"Pois não é o esforço da memória que constitui o verdadeiro conhecimento na matemática, ele sobretudo restringe suas competências, não as aumenta. ... fazem bem os que abdicam desse trabalho tedioso que enfraquece o espírito de invenção e de pesquisa." -

Sylvestre François Lacroix

Como explicado no capítulo 2, sendo (P, L, R) um plano projetivo finito de ordem n, o objeto de estudo desse capítulo é o número mínimo de cores necessárias – denotado por $c(P_n)$ – para que exista uma c-coloração legítima de P_n. Em [3], Noga Alon e Zoltan Füredi impõe que se a ordem de P_n for suficientemente grande (maior do que $3 \cdot 10^{250}$)[1], então $5 \leqslant c(P_n) \leqslant 8$. Em outras palavras, para um P_n de ordem suficientemente grande, se $c < 5$ então em toda c-coloração de P_n existem pelo menos duas linhas com o mesmo tipo e se $c = 8$ então existe uma c-coloração legítima de P_n.

No capítulo 2 mostramos que existe um conjunto $S \subset P$ e uma 8-coloração $f\colon\ P \setminus S \to \{1, \ldots, 8\}$ na qual nenhum ponto $p \in P$ pertence a mais do que quatro linhas envolvidas em pares perigosos e nenhuma linha forma um par perigoso com outras duas linhas, para um n grande. Fazendo pequenas alterações nos lemas de [3], não é difícil mostrar que, a partir de um determinado n, existe uma c-coloração de $P \setminus S$ na qual nenhum ponto $p \in P$ pertence a mais do que b linhas envolvidas em pares perigosos e nenhuma linha forma um par perigoso com outras a linhas. Como ilustrado na Figura 1 isso implica que nenhum ponto pertence a mais do que $a \cdot b$ pares perigosos.

[1] Esse número é obtido do lema 2.8.

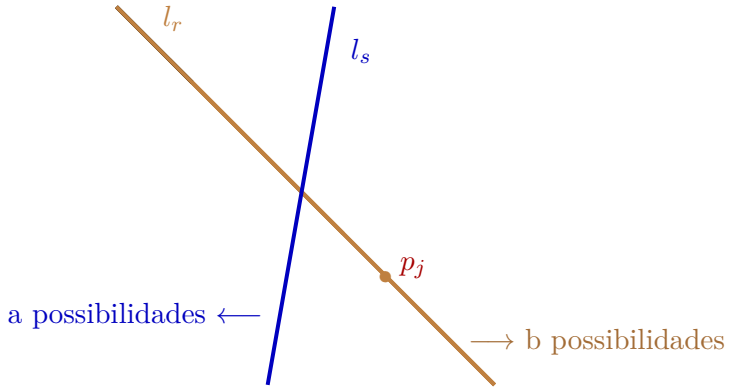

Figura 1: Representação dos pares perigosos que contém p_j nos quais p_j não está na intersecção das linhas.

Usaremos tais resultados combinados com o Teorema 3.26 para mostrar que, fixado um P_n, existe uma coloração legítima dos pontos de P_n com um número limitado de cores. Seja $P_n = (P, L)$ um plano projetivo finito com $P = \{p_1, \ldots, p_{n^2+n+1}\}$ e $L = \{\ell_1, \ldots, \ell_{n^2+n+1}\}$. Fixemos um conjunto $S \subset P$ e uma coloração parcial $f\colon P \setminus S \to [\mathsf{d}]$ para a qual nenhum ponto pertence a mais do que $a \cdot b$ pares perigosos. Definiremos nossas colorações ruins da seguinte maneira:

- $\mathcal{Z} = \{Z \subset P : \exists\, \ell_\kappa, \ell_j, \ell_\kappa \neq \ell_j$ tais que $Z = (\ell_\kappa \cup \ell_j) \setminus (\ell_\kappa \cap \ell_j)\}$ indexado por um conjunto finito $I \subset \mathbb{N}$.
- Para cada $i \in I$:
  - $C_i = \{c : Z_i = (\ell_\kappa \cup \ell_j) \setminus (\ell_\kappa \cap \ell_j)$ e $t_{\ell_\kappa, c} = t_{\ell_j, c}\}$.
  - Para cada ponto p de Z_i, considere o conjunto Y_i contendo p e outros $\underline{m} - 1$ pontos de menor índice da linha contendo p, onde $\underline{m} = \arg\min_m \frac{m}{m-1}(m!a \cdot b(m-1))^{(1/m)}$ (essa escolha se tornará clara mais adiante). Fixando os pontos de $Z_i \setminus Y_i$ é fácil mostrar que $m_i = \underline{m}!$.
  - $l_i = |Z_i| - |Z_i \setminus Y_i| = \underline{m}$.
- $E = \{\underline{m}\}$.
- $d_{\underline{m}} = a \cdot b$.

Então o Teorema 3.26 nos permite provar o seguinte resultado:

Proposição 4.1. *Seja P_n um plano projetivo finito de ordem n. Existe uma coloração legítima de P_n com* $\max(\mathsf{d}(n, a, b), \lceil \min_m \frac{m}{m-1}(m!a \cdot b(m-1))^{1/m} \rceil)$

cores, onde $\mathsf{d}(n,a,b)$ *é o número de cores necessárias para que exista uma coloração parcial* $f : P \setminus S \to [\mathsf{d}(n,a,b)]$ *na qual nenhum ponto pertença a mais do que* $a \cdot b$ *pares perigosos.*

Prova: Nosso primeiro passo é fixar uma coloração parcial f de P_n na qual nenhum ponto pertença a mais do que $a \cdot b$ pares perigosos usando $\mathsf{d}(n,a,b)$ cores, nós estimares esse número em um segundo momento. Como $E = \{\underline{m}\}$ nós temos que $\phi_E(x) = 1 + x^{\underline{m}}$ e $\phi_E(\tau) - \tau\phi_E^{'}(\tau) = 0$ sse $\tau = (\underline{m}-1)^{-1/\underline{m}}$. Então $\phi_E^{'}(\tau) = \underline{m}(\underline{m}-1)^{(1-\underline{m})/\underline{m}}$. Portanto, usando $\lceil \min_m \frac{m}{m-1}(m!a{\cdot}b(m-1))^{1/m} \rceil$ cores é possível estender f para uma coloração C na qual nenhum par ruim é criado. ■

A seguir cotaremos a cardinalidade do conjunto S para deduzir uma relação entre d, n, a e b.

Lema 4.2. *A cardinalidade de* S *é cotada superiormente por* $\frac{(n^2+n+1)11\ln n}{n+1}$.

Prova: Como existem $n+1$ linhas passando por cada ponto de S e $|L| = n^2+n+1$, nós temos que, em média, cada linha $\ell \in L$ terá $\frac{|S|(n+1)}{n^2+n+1}$ pontos em S. Portanto $\max_{\ell \in L} |\ell \cap S| \geqslant \frac{|S|(n+1)}{n^2+n+1}$. Como, para cada linha $\ell \in L$ nós temos que $|\ell \cap S| \leqslant 11\ln n$, então $|S| \leqslant \frac{(n^2+n+1)11\ln n}{n+1}$. ■

Agora seja

$$K := 2^{\mathsf{d}} \binom{22\log n + \mathsf{d}}{\mathsf{d}} \frac{\mathsf{d}^{\frac{\mathsf{d}}{2}}}{[2\pi(n+1-11\log n - n^{1/2})]^{\frac{\mathsf{d}-1}{2}}}.$$

Generalizando o lema 2.7, descobre-se que a probabilidade P_a de que exista uma linha em L que forme pares perigosos com outras $a+1$ linhas (ao invés de 2) obedece

$$P_a \leq (K)^{a+1}(n^2+n+1)\binom{n^2+n}{a+1}.$$

De forma similar, generalizando o lema 2.8, descobre-se que a probabilidade P_b de que exista um ponto de P que pertença a mais do que $b+1$ linhas envolvidas em pares perigosos com outras linhas (ao invés de 5) obedece

$$P_b \leq \frac{(K)^{b+1}11\log n(n^2+n+1)\binom{n+1}{b+1}(n^2+n-(b+1))^{b+1}}{n+1}.$$

A condição que precisa ser satisfeita para que o conjunto f tenha as propriedades mencionadas é

$$1 - (P_a + P_b) > 0.$$

Como visto no capítulo 2, K é uma melhor estimativa do que a feita no corolário 2.5 de [3] e está provada no corolário 2.5 do capítulo 2. Derivar P_a e P_b de K e a condição $1-(P_a+P_b)>0$ é trivial. Temos então o seguinte problema de otimização:

$$\begin{aligned} min \quad & f(a,b,m,n)=n \\ s.a. \quad & \begin{cases} \frac{m}{m-1}(m!ab(m-1))^{1/m} \leq c \\ P_a+P_b<1 \end{cases} \end{aligned}$$

que busca a menor ordem de um plano projetivo que pode ser colorido legitimamente com c cores.

Observação 4.3. Nos últimos anos o Método de Entropy Compression tem vencido do Método Probabilístico em inúmeros problemas. O problema de otimização acima serve como altar para o inesperado matrimônio entre os dois métodos ministrado pela aplicação aqui exposta. De fato a primeira restrição do problema nasce do Método de Entropy Compression e a segunda do Método Probabilístico. □

A figura a seguir mostra alguns resultados obtidos manualmente do problema acima:

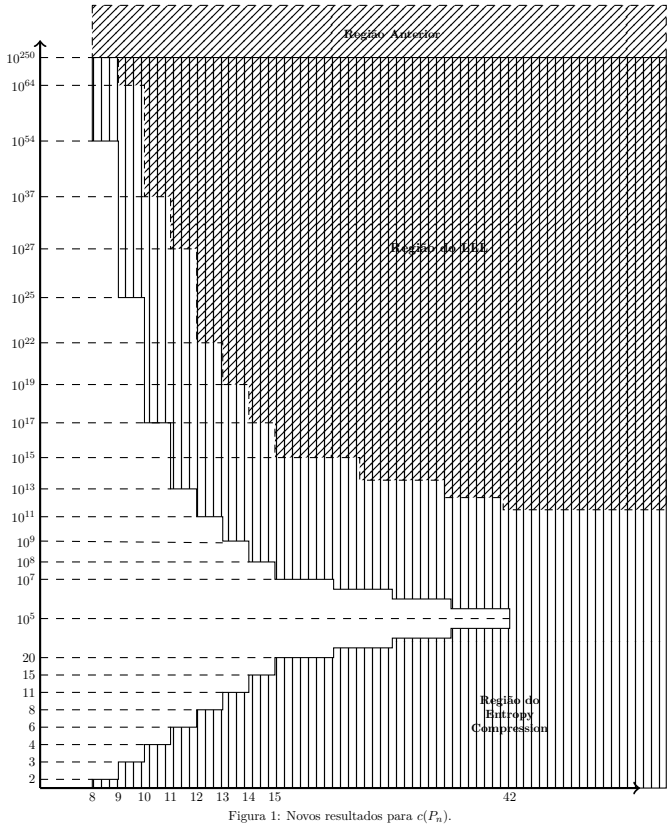

Figura 1: Novos resultados para $c(P_n)$.

Observação 4.4. Exemplos onde a região obtida pelo entropy compression é contida pela região obtida pelo lema local de lovász fogem ao meu conhecimento - possivelmente por inexistência dos mesmos. Logo, o fato de isso ter ocorrido em nosso problema não é surpreendente. O que surpreende é a região essencialmente diferente encontrada: não apenas conseguimos resultados melhores via compressão de entropia como também obtivemos resultados para valores pequenos de n, o que era inacessível para Lovász. □

Observação 4.5. A figura 1 levanta a seguinte questão:

Questão: Pode-se colorir de forma legítima qualquer plano projetivo finito usando 8 cores?

A pergunta, a priori sem cabimento, agora se faz natural pois, fixado um número de cores, conseguimos colorir de forma legítima planos projetivos até uma determinada ordem, depois deixamos de conseguir e voltamos a conseguir após uma ordem mais elevada. Isso levanta a questão se essa região é intrínseca ao problema ou se é possível preenche-la provando que com 8 cores consegue-se colorir de forma legítima um plano projetivo de qualquer ordem. Uma possível explicação para a região pertencer ao problema é a falta de complexidade do plano projetivo quando sua ordem é baixa. Após um ganho considerável de complexidade é preciso produzir mais tipos com o mesmo número de cores conforme a ordem cresce, como elucidado no capítulo 2. Porém tal região pode ser um ponto cego do método que usamos. Na tentativa de responder essa questão, desenvolvi um algoritmo que toma como entradas a ordem n de um plano projetivo e o numero c de cores a serem usadas para formar uma coloração legítima. O programa retorna, após um processo de exaustão no conjunto das colorações do plano de ordem n usando c cores, se existe uma coloração legítima para as entradas dadas. O código em python é o seguinte:

```
import numpy as np
import itertools as itt

def Compara_tipos(c, t_L1, t_L2):

    k = 0

    for i in range(c):

        if( t_L1[i] == t_L2[i] ):

            k += 1

    if( k == c ):
```

```
15          return 1
16      else:
17          return 0
18
19
20
21  def Busca_Coloracao_Legitima(q,c):
22
23
24      MI = MatrizIncidencia(q) #Essa funcao esta definida no
        apendice A.
25
26      k = 0
27
28      for co in itt.product( range(1,c+1), repeat = q*q + q + 1 ):
29
30          if( k == 1 ): #Se os tipos de duas linhas sao iguais, na
         proxima iteracao comparamos primeiramente essas duas linhas.
31
32              t_L1 = np.zeros(c)
33
34              t_L2 = np.zeros(c)
35
36              for i in range(q*q + q + 1):
37
38                  if( MI[i][parr[0]] == 1 ):
39
40                      t_L1[ co[i]-1 ] += 1
41
42              for i in range(q*q + q + 1):
43
44                  if( MI[i][parr[1]] == 1 ):
45
46                      t_L2[ co[i]-1 ] += 1
47
48              if( Compara_tipos(c, t_L1, t_L2) ):
49                  continue
50
51
52          for par in itt.combinations( range(q*q + q + 1), 2 ):
53
54              t_L1 = np.zeros(c)
55
56              t_L2 = np.zeros(c)
57
58              for i in range(q*q + q + 1):
59
60                  if( MI[i][par[0]] == 1 ):
61
62                      t_L1[ co[i]-1 ] += 1
63
64              for i in range(q*q + q + 1):
65
66                  if( MI[i][par[1]] == 1 ):
```

```
67
68                      t_L2[ co[i]-1 ] += 1
69
70              if( Compara_tipos(c, t_L1, t_L2) ):
71                  k = 1
72                  parr = par
73                  break
74          else:
75
76            print("Existe uma coloracao legitima!\n\n ", co,"\n\n
    para a seguite matriz de incidencia\n\n",MI)
77
78              break
79
80      else:
81
82          print("Nao existe uma coloracao legitima para o plano
    projetivo de ordem",q,"usando",c,"cores!")
```

Infelizmente esse código não tem sentido prático pois seu tempo de execução é $O(c^n)$. Um caminho natural para melhorar o algoritmo é desconsiderar previamente um número considerável de colorações trivialmente não-legítimas. Dado a complexidade do plano projetivo tal tarefa se mostrou nada trivial. □

◇ ◇ ◇

Conclusão

Neste trabalho utilizamos diferentes versões do Lema Local de Lovász (LLL) [9, 11] – contido no arsenal do *Método Probabilístico* [5] – bem como o recente método chamado *entropy compression* [29, 64], para atacar o problema de existência de colorações legítimas em planos projetivos finitos proposto por Noga Alon e Zoltan Füredi em [3]. Obtemos em [10] uma variação do método sistematizado por Louis Esperet e Aline Parreau em [29] e provamos que esta variação pode ser aplicada em todos os problemas formulados dentro da linguagem *variable version* do LLL [64, 51], onde se aplica o algoritmo de Moser-Tardos. Como aplicação, mostramos que todo plano projetivo com ordem maior que 10^{54} admite uma coloração legítima com 8 cores – o resultado original dos autores em [3] exige uma ordem maior que 10^{250}. Além disso, obtemos colorações legítimas com 8 cores para planos projetivos de ordem pequena. Permitindo o uso de mais cores, obtemos progressivamente melhores resultados para ordens grandes e pequenas. De fato, provamos que com 42 cores pode-se colorir legitimamente planos projetivos finitos de qualquer ordem. A prova da existência de colorações legítimas para planos de ordens grandes e pequenas e com diferentes números de cores levanta a questão se todo plano projetivo pode ser colorido de forma legítima usando-se poucas cores independente de sua ordem. Com base nos resultados obtidos, conjecturamos que pode-se colorir legitimamente qualquer plano projetivo usando-se somente 8 cores (e não 42). Os resultados obtidos estão contidos na próxima figura:

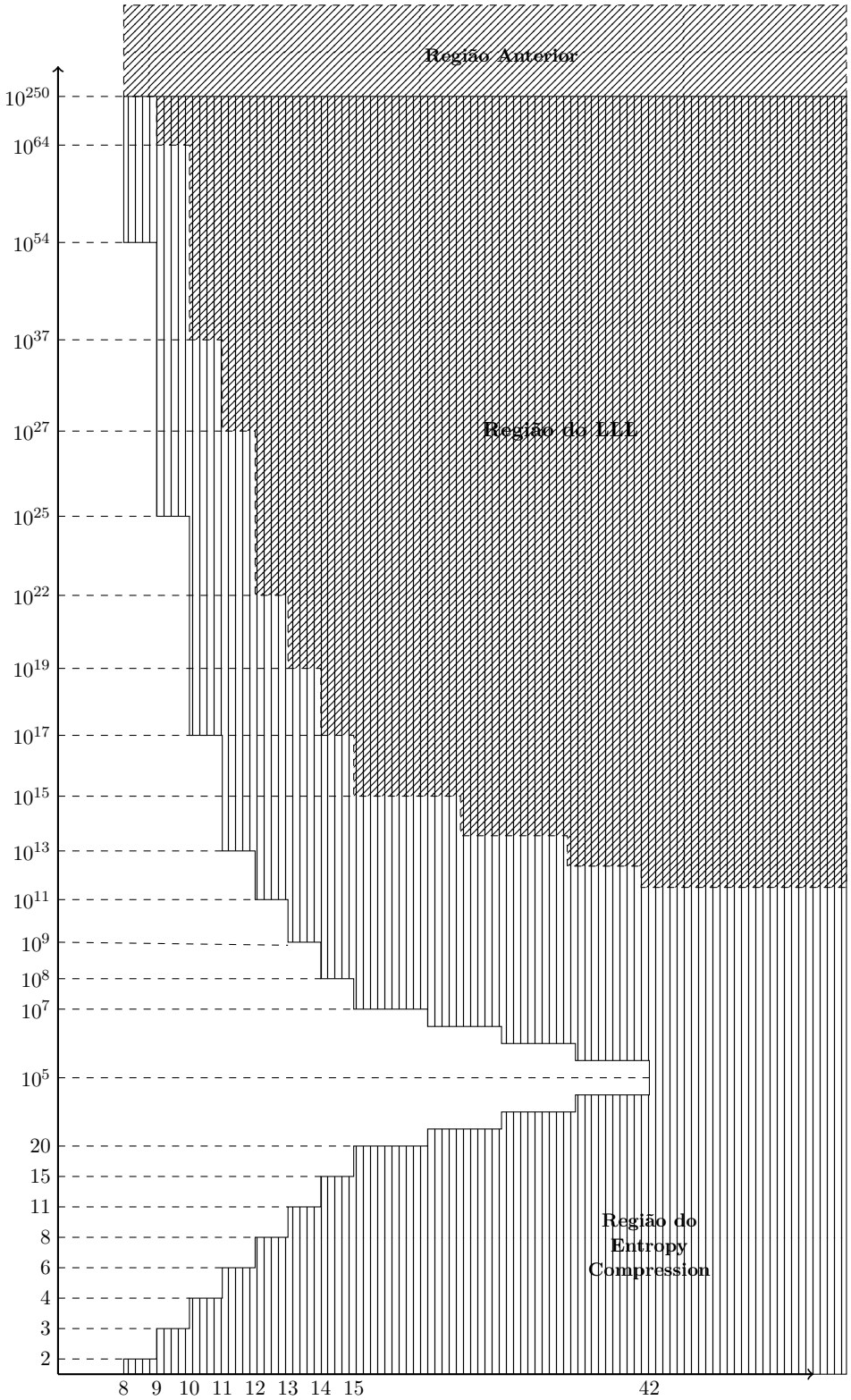

Figura 1: Novos resultados para $c(P_n)$.

Apêndice A

Planos Projetivos Finitos

Esse capítulo se propõe a expor de forma sucinta as definições e resultados básicos relacionados a planos projetivos finitos. Espera-se com isso municiar o leitor para a leitura e entendimento dos capítulos 2 e 4. Para um estudo mais aprofundado no tema, indico ao leitor interessado os artigos [**?**] e [65].

Um plano projetivo pode ser obtido pela expansão do plano euclidiano por meio de dois acréscimos:

1. Um ponto para cada família de retas paralelas. Sendo esse a interseção entre elas.

2. Uma reta que contém somente os novos pontos adicionados.

Note que, com o acréscimo desses pontos, não existe mais o conceito de retas paralelas. Mais rigorosamente, podemos definir um plano projetivo de forma axiomática: sejam P um conjunto cujos elementos são chamados *pontos*, L um conjunto cujos elementos são chamados *retas* ou *linhas* e R: $P \to L$ uma relação entre esses dois conjuntos. A terna (P, L, R) é um plano projetivo se:

Axioma 1 - Dados b, $c \in P$, $b \neq c$, existe e é único $a \in L$ tal que (b, a), $(c, a) \in R$.

Axioma 2 - Dados b, $c \in L$, $b \neq c$, existe e é único $a \in P$ tal que (a, b), $(a, c) \in R$.

Axioma 3 - Existem a_1, a_2, a_3, $a_4 \in P$ tais que não existem $b \in L$ e i, j, $k \in \{1, 2, 3, 4\}$ tais que $(a_i, b), (a_j, b)(a_k, b) \in R$, com $k \neq i \neq j \neq k$.

Duas observações são necessárias:

1 - O Axioma 1 se assemelha ao primeiro axioma de Euclides que diz "*pode-se traçar uma única reta ligando quaisquer dois pontos*". Porém os dois diferem em algo essencial: o conceito de reta. Para nós, uma reta é apenas um elemento do conjunto L. A reta de Euclides, ou o segmento de reta entre dois pontos, é um comprimento de dimensão 1. Tais conceitos nem fazem sentido em nossa construção. Essa diferença ficará mais clara, mais adiante, com a análise do Plano de Fano (figura 1).

2 - O Axioma 2 exclui o conceito de retas paralelas.

Feitas as observações, uma convenção: se $(p, l) \in R$ diremos que l passa por p ou é incidente a p. Outras vezes, diremos que p é incidente a l ou p pertence a l (muitas vezes l será visto como um conjunto). Mais uma vez, cuidado para não confundir com a noção euclidiana de ponto pertencente a uma reta. A relevância do Axioma 3 será comentada mais adiante.

Se P e L são conjuntos com cardinalidade finita então (P, L, R) é dito ser um plano projetivo finito. Pode-se provar que, para um plano projetivo finito:

Resultado 1: Existe $n \in \mathbb{N}$ tal que $|P| = |L| = n^2 + n + 1$ (a esse n, dá-se o nome de ordem do plano projetivo finito).

Resultado 2: Se $l \in L$ então existem somente $p_1, \ldots, p_{n+1} \in P$ tais que $(p_i, l) \in R$, para todo $i \in \{1, ..., n + 1\}$ (n fixado no Resultado 1).

Resultado 3: Seja $p \in P$ então existem somente $l_1, \ldots, l_{n+1} \in L$ tais que $(p, l_i) \in R$, para todo $i \in \{1, ..., n + 1\}$ (n fixado no Resultado 1).

Daqui em diante, nos referiremos a um plano projetivo finito pela terna (P, L, R).

Exemplo A.1. Um bom exemplo para nos afastar de nossa intuição geométrica euclidiana e assimilar melhor essa construção axiomática é o plano projetivo finito de ordem 2 ou Plano de Fano:

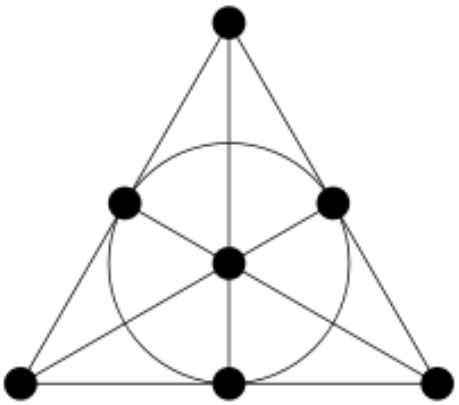

Figura 1 - Plano de Fano.

Repare que:

• (Axioma 1) Cada par de pontos tem exatamente um reta de incidência em comum.

• (Axioma 2) Cada par de retas tem exatamente um ponto de incidência em comum (em nossa construção, uma reta não é um conjunto infinito de pontos. Não é porque duas retas se cruzam no desenho que deve ter um ponto incidindo com essas duas retas).

• (Axioma 3) Existem quatro pontos tais que nenhuma reta é incidente a mais de dois desses pontos (os três vértices mais o ponto central, por exemplo).

• (Resultado 1) Como o (P, L, R) é de ordem 2, há exatamente sete pontos e sete retas (você contou seis?).

• (Resultado 2) Cada ponto é incidente a três $(= n + 1)$ retas.

• (Resultado 3) Cada reta é incidente a três $(= n + 1)$ pontos. □

Analisar um (P, L, R) por um desenho que o represente pode gerar confusões. Isso se dá pela dificuldade de desvencilhar-nos dos axiomas da geometria euclidiana. Uma abordagem, na minha opinião, mais adequada é a construção de uma matriz de incidência.

Exemplo A.2. A figura a seguir é a representação de um (P, L, R) de ordem três por sua matriz de incidência:

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	l_{13}
p_1	1	1	1	1									
p_2	1				1	1	1						
p_3	1							1	1	1			
p_4	1										1	1	1
p_5		1			1			1			1		
p_6		1				1			1			1	
p_7		1					1			1			1
p_8			1		1				1				1
p_9			1			1				1	1		
p_{10}			1				1	1				1	
p_{11}				1	1					1		1	
p_{12}				1		1		1					1
p_{13}				1			1		1		1		

Figura 2 - Matriz de incidência de (P, L, R) de ordem 3.

Onde $M_{i,j} = 1$ se $(p_i, l_j) \in R$ e $M_{i,j} = 0$ (aparece em branco na matriz) se $(p_i, l_j) \notin R$. Repare que, considerando que p_i é incidente a l_j se $M_{i,j} = 1$:

• (Axioma 1) Cada par de linhas tem exatamente uma coluna de incidência em comum.

• (Axioma 2) Cada par de colunas tem exatamente uma linha de incidência em comum.

• (Axioma 3) Existem quatro linhas tais que não existe coluna incidente a mais do que duas dessas linhas (p_1,p_4,p_5 e p_8, por exemplo).

• (Resultado 1) Como o (P, L, R) é de ordem 3, a matriz tem exatamente 13 linhas e 13 colunas.

- (Resultado 2) Cada linha é incidente a 4 $(= n + 1)$ colunas.
- (Resultado 3) Cada coluna é incidente a 4 $(= n + 1)$ linhas. □

No capítulo 4, a observação 4.8 apresenta um código que faz uso da matriz de incidência de um plano projetivo de ordem n primo. Com base em resultados obtidos em [8], o código a seguir monta tal matriz:

```
import numpy as np


def Posicao (a,A): #Acha a matriz posicao de A.

    B = np.zeros((len(np.transpose(A)[0]),len(A[0])*len(a)))
    for i in range(len(np.transpose(A)[0])):

        for j in range(len(A[0])*len(a)):

            if A[i][j % len(A[0])] == a[j//len(A[0])]:

                B[i][j] = 1

    return B


def MatrizA (u,q): #Cria os blocos para montar-se a matrix A
    mencionada no teorema 3.1 de [7].


    B = np.zeros((q,q))

    for i in range(q):

        for j in range(q):

            if -j%q == i:

                B[i][j] = q
            else:

                B[i][j] = u*(i+j)%q

    return B

def Oq (q): #Monta a matriz O_q mencionada em [7].

    B = np.zeros((q,q))

    for i in range(q):

```

```
42          for j in range(q):
43
44              B[i][j] = i + 1
45
46      return B
47
48  def OQ (q): #Monta a matriz O_{q+1,q} mencionada em [7].
49
50      B = np.zeros((q + 1,q))
51
52      for i in range(q + 1):
53
54          for j in range(q):
55
56              B[i][j] = i + 1
57
58      return B
59
60  def Ot (q): #Monta as ultimas colunas da matriz de incidencia.
61
62      a = np.zeros((q,1))
63      b = np.ones((q,1))
64
65
66  def MatrizIncidencia(q):
67
68
69
70      B = np.zeros((q-1, q, q)) #Sera usado para armazenar os
        blocos da matriz A.
71
72      P = np.zeros((q-1, q, q**2)) #Sera usado para armazenar as
        matrizes posicoes.
73
74      a = list(range(1,q+1)) #Armazena os simbolos do campo de
        Galois.
75
76      o = Oq(q)
77
78      O = OQ(q)
79
80      I = np.identity(q)
81
82      AI = np.identity(q)
83
84      j = np.ones(q + 1)
85
86      h = np.zeros(q**2)
87
88      for u in range(1,q):
89
90          B[u-1] = MatrizA(u,q)
91
92      for u in range(q-1):
```

```
93
94          P[u] = Posicao(a,B[u])
95
96      MI = P[0]
97
98      for i in range(1, q-1):
99
100         MI = np.vstack((MI,P[i]))
101
102     for i in range(q-1):
103
104         AI = np.hstack((AI,I))
105
106     PO = Posicao(a,o)
107
108     MI = np.vstack((MI,AI))
109
110     MI = np.vstack((MI,PO))
111
112     MI = np.vstack((MI,h))
113
114     Ot = np.transpose(Posicao(list(range(1,q+2)), O))
115
116     Ot = np.vstack((Ot, j))
117
118     MI = np.hstack((MI,Ot)) #E isso completa nossa matriz de
        incidencia.
119
120     return MI
```

O código acima, apesar de bastante didático, não é muito eficiente quanto ao uso da memória. O próximo código inverte os valores sendo em certa medida confuso mas usando apenas a memória necessária:

```
1  import numpy as np
2
3  def Posicao (a,A): #Acha a matriz posicao de A.
4
5      B = np.zeros((len(np.transpose(A)[0]),len(A[0])*len(a)))
6      for i in range(len(np.transpose(A)[0])):
7
8          for j in range(len(A[0])*len(a)):
9
10             if A[i][j % len(A[0])] == a[j//len(A[0])]:
11
12                 B[i][j] = 1
13
14     return B
15
16 def PSE_MI(a, q, MI): #Monta a parte superior esquerda da matriz
       de incidencia.
17
```

```
18      for i in range(q * (q - 1)):
19
20
21          for  j in reversed(range(q * q)):
22
23              if( MI[i][j % q] == a[j//q] ):
24                  MI[i][j] = 1 ;
25              else:
26                  MI[i][j] = 0 ;
27
28      return MI
29
30  def MatrizA (MI, q): #Cria os blocos para montar-se a matrix A
        mencionada em [7].
31
32
33      for i  in range( q * (q - 1) ):
34
35          for j in  range( q ):
36
37              if -j % q ==  i % q :
38                  MI[i][j] = q
39              else:
40                  MI[i][j] =  ( (i//q + 1) * (i + j) ) % q
41
42      return MI
43
44  def Oq (q): #Monta a matriz O_q mencionada em [7].
45
46      B = np.zeros((q,q))
47
48      for i in range(q):
49
50          for j in range(q):
51
52              B[i][j] = i + 1
53
54      return B
55
56  def OQ (q): #Monta a matriz O_{q+1,q} mencionada em [7].
57
58      B = np.zeros((q + 1,q))
59
60      for i in range(q + 1):
61
62          for j in range(q):
63
64              B[i][j] = i + 1
65
66      return B
67
68  def MatrizIncidencia(q):
69
70      a = list(range(1,q+1)) #Armazena os simbolos do campo de
```

```
    Galois.
71
72      MI = np.zeros((q*(q-1), q*q))
73
74      o = Oq(q)
75
76      O = OQ(q)
77
78      I = np.identity(q)
79
80      AI = np.identity(q)
81
82      j = np.ones(q + 1)
83
84      h = np.zeros(q**2)
85
86      MI = MatrizA(MI, q)
87
88      MI = PSE_MI(a, q, MI)
89
90      for i in range(q-1):
91
92          AI = np.hstack((AI,I))
93
94      PO = Posicao(a,o)
95
96      MI = np.vstack((MI,AI))
97
98      MI = np.vstack((MI,PO))
99
100     MI = np.vstack((MI,h))
101
102     Ot = np.transpose(Posicao(list(range(1,q+2)), O))
103
104     Ot = np.vstack((Ot, j))
105
106     MI = np.hstack((MI,Ot)) #E isso completa nossa matriz de
        incidencia.
107
108     return MI
```

Agora nos resta entender a relevância do axioma 3. Observe a figura a seguir:

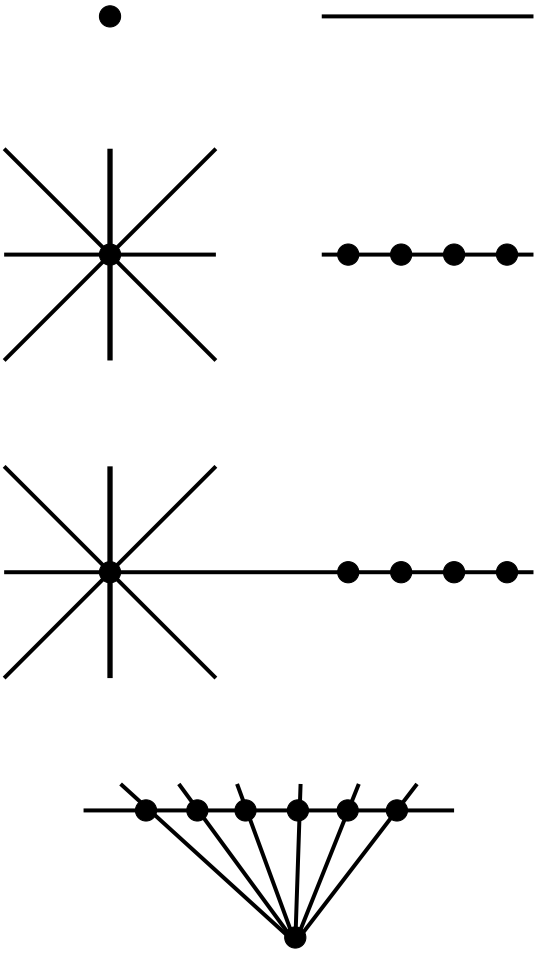

Figura 3: Planos projetivos degenerados.

Os seis "planos projetivos" acima mais o espaço vazio formam os sete tipos diferentes de planos projetivos degenerados. Eles obedecem aos dois primeiros axiomas (verifique!) mas não ao terceiro. A inclusão do axioma 3 exclui os casos degenerados. Desse forma sobram apenas os planos projetivos com estrutura mais complexa.

Finalmente, repare que o resultado 1 garante que para cada (P, L, R) existe um $n \in \mathbb{N}$ tal que $|P| = |L| = n^2 + n + 1$. Porém, não garante que para cada $n \in \mathbb{N}$ existe um (P, L, R) de ordem n. Quais $n \in \mathbb{N}$ são ordem de um (P, L, R) ainda é uma questão em aberto. Em [18], é provado que se n é uma potência de um primo então existe um (P, L, R) de ordem n. Outro resultado importante é o obtido por Bruck e Ryser em [17] que impõe que se $n = 1(\text{mod}4)$ ou $n = 2(\text{mod}4)$ então n deve ser a soma de dois quadrados

para ser ordem de algum (P, L, R) (isso elimina $n = 6$ como ordem possível, por exemplo). Os únicos (P, L, R) conhecidos são aqueles cuja ordem é uma potência de um primo. Porém é sabido que não existe um (P, L, R) de ordem 10 conforme provado em [45], mostrando-se, computacionalmente, que não é possível construir uma matriz de incidência de ordem 10.

Apêndice B

Funções Geradoras

Como visto no capítulo 3, o método descrito em [29] faz uso de funções geradoras para contar o número de árvores planas com $t+1$ vértices tais que o grau de cada vértice está contigo em um conjunto E. Esse capítulo se baseia, em sua maior parte, em [67] e se propõe a mostrar as definições e propriedades básicas das funções geradoras ordinárias e elucidar como tal ferramenta pode ser usada em problemas de contagem. É importante ressaltar, porém, que esse método é muito mais abrangente do que esse capítulo foi capaz de mostrar. Apenas citando algumas aplicações possíveis: na próxima seção, resolveremos relações de recorrência; em [66] o leitor poderá ver essa técnica aplicada a problemas computacionais (como ordenação, procura, algoritmos dinâmicos, etc.); em [39] funções geradoras são usadas para contar grafos; e em [33] e [36] o leitor encontrará muitas outras aplicações não citadas nesse capítulo. O leitor interessado em uma abordagem mais completa desde a base da teoria é convidado a consultar [20]. Por sua vez, o leitor interessado em se aprofundar na questão poderá consultar [36].

Uma função geradora é um varal onde pendura-se uma sequência de números. A próxima seção procura entender o sentido dessa frase. Enquanto isso, suponha que tenhamos um problema cuja resposta é uma sequência $a_0, a_1, a_2, \ldots$. Encontrar uma fórmula simples para a_n seria a melhor maneira de resolver o problema. Se descobrirmos que $a_n = n^2 + 3$ para cada $n = 0, 1, 2, \ldots$, então não há dúvidas que respondemos a questão. Mas e se não houver uma fórmula simples para o membros da sequência desconhecida? Afinal, algumas sequências são complicadas. Suponha, por exemplo, que a sequência desconhecida seja 2, 3, 5, 7, 11, 13, 17, 19, ... , onde a_n é o n-ésimo número primo. Nesse caso não podemos esperar por uma fórmula simples. Funções geradoras oferecem uma outra abordagem para o problema: ao invés de procurar por uma fórmula simples para os membros da sequência podemos encontrar uma fórmula simples para a soma da série de potência que tem tal sequência como coeficientes.

B.1 Relações de Recorrência

Exemplo B.1. Considere a sequência $a_0, a_1, \ldots$, satisfazendo

$$a_{n+1} = 2a_{n+1} \ , \ n \geq 0 \ ; \ a_0 = 0.$$

Nosso desafio é descobrir quais são os elementos da sequência. Para ganharmos alguma intuição, calculemos alguns elementos primeiro. A sequência começa com 1, 3, 7, 15, 31, ... , indicando que a sequência deve obedecer $2^n - 1, (n \geq 0)$. De fato não é difícil provar tal fato por um argumento de indução. Vejamos, porém, como é a abordagem usando funções geradoras: seja

$$A(x) = \sum_{n \geq 0} a_n x^n.$$

Para achar $A(x)$, basta multiplicar ambos os lados da relação de recorrência por x_n e somar nos valores para os quais a recorrência é válida, i.e., $n \geq 0$ e tentar relacionar tais somas à função geradora $A(x)$. Do lado esquerdo ficamos com

$$\sum_{n \geq 0} a_{n+1} x^n.$$

Já quase encontramos $A(x)$. O índice de 'a' é uma unidade maior do que deveria. Porém

$$\sum_{n \geq 0} a_{n+1} x^n = a_1 + a_2 x + a_3 x^2 + \ldots =$$

$$= \frac{(a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots) - a_0}{x} = \frac{A(x)}{x},$$

já que, nesse problema, $a_0 = 0$. O lado direito da ralação de recorrência nos fornece

$$\sum_{n \geq 0} (2a_n + 1) x^n = 2A(x) + \sum_{n \geq 0} x^n = 2A(x) + \frac{1}{1 - x}.$$

Logo, nós temos que

$$\frac{A(x)}{x} = 2A(x) + \frac{1}{1 - x}.$$

E

$$A(x) = \frac{x}{(1-x)(1-2x)}.$$

Essa é a função geradora para o problema. os elementos desconhecidos da sequências estão dispostos no varal: a_n é o coeficiente de x^n na expansão em série de $A(x)$. Nós temos que

$$\frac{x}{(1-x)(1-2x)} = x(\frac{2}{1-2x} - \frac{1}{1-x} = (2x + 2^2x^2 + 2^3x^3 + \ldots) -$$

$$-(x + x^2 + x^3 + \ldots) = (2-1)x + (2^2-1)x^2 + (2^3-1)x^3 + \ldots.$$

E de fato temos que $a_n = 2^n - 1$, para $n \geq 0$. □

Esse exemplo gera algum desconforto. De fato, usamos um canhão para matar uma formiga. O interessante desse método é que ele é de fato um canhão! A mesma abordagem pode ser aplicada para problemas bem mais complicados. Para tanto, dada uma relação de recorrência para uma sequência $(a_n)_{n\in\mathbb{N}}$, basta seguir o seguinte método:

• Assegure-se de que o conjunto dos índices para os quais a relação é válida está bem definido.

• Dê um nome para a função geradora a ser descoberta e a escreva em função dos termos da sequência ($A(x)$, por exemplo, e $A(x) = \sum a_n x^n$).

• Multiplique ambos os lados da relação de recorrência por x^n e some em todos os valores de n para os quais a relação é válida.

• Expresse ambos os lados da equações oriunda do passo anterior em função da função geradora $A(x)$.

• Resolva a equação na incógnita $A(x)$.

• Escolha algum método conveniente para expandir $A(x)$ em sua série de potência.

Exemplo B.2. Agora, apliquemos tal método à sequência de Fibonacci:

$$F_{n+1} = F_n + F_n - 1 \quad , \quad (n \geq 1; F_0 = 0; F_1 = 1).$$

Chamemos nossa função geradora de $F(x) = \sum F_n x^n$.. Agora, devemos multiplicar ambos os lados da relação de recorrência por x^n e somar para $n \geq 1$. O lado esquerdo nos dá

$$F_2 x + F_3 x^2 + F_4 x^3 + \ldots = \frac{F(x) - x}{x},$$

e o lado direito

$$(F_1 x + F_2 x^2 + F_3 x^3 + \ldots) + (F_0 x + F_1 x^2 + F_2 x^3 + \ldots) = F(x) + xF(x).$$

Logo

$$(F(x) - x)/x = F(x) + xF(x) \quad \therefore \quad F(x) = \frac{x}{1 - x - x^2}.$$

Sejam $r\pm = (1 \pm \sqrt{5})/2$. Logo

$$1 - x - x^2 = (1 - xr_+)(1 - xr_-)$$

e

$$\frac{x}{1 - x - x^2} = \frac{x}{(1 - xr_+)(1 - xr_-)} = \frac{1}{r_+ - r_-}\left(\frac{1}{1 - xr_+} - \frac{1}{1 - xr_-}\right) =$$

$$= \frac{1}{\sqrt{5}}\left(\sum_{j \geq 0} r_+^j x^j - \sum_{j \geq 0} r_-^j x^j\right).$$

Logo

$$F_n = \frac{1}{\sqrt{5}}(r_+^n - r_-^n) \ , \ n \geq 0.$$

Esse exemplo nos mostra, inclusive, um pouco mais o tipo de informação que podemos adquirir usando funções geradoras. Nós não obtivemos somente a resposta exata mas também uma aproximação, tão mais próxima da verdade quanto maior for o valor de n. Para valores grandes de n, uma ótima aproximação para F_n é

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1 + \sqrt{5}}{2}\right)^n.$$

Ainda cabe a pergunta: de que serve uma aproximação quando se tem o valor exato? Um resposta possível é que, as vezes, a resposta exata é muito complicada e a aproximada é mais informativa. Mesmo nesse caso, em que a resposta exata não é muito complicada, ainda podemos aprender algo com a aproximação. De fato, desconsiderando r_- em F_n comete-se um erro inferior a 0.5. Consequentemente, F_n é exatamente igual a

$$\lceil \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n \rceil.$$

E assim obtivemos uma fórmula mais simples para F_n. □

No capítulo 3, aplicamos funções geradoras para contar árvores com determinada propriedade. Árvores são definidas como grafos conexos sem ciclos. Suas propriedades são básicas na teoria de grafos. Por exemplo, um grafo conexo é uma árvore se, e somente se, o número de elos é igual ao número de vértices menos 1. Ou cada par de vértices é conectado por um único caminho, etc..

Se marcarmos um vértice r de uma árvore T, a transformamos em uma *árvore enraizada*. Árvores enraizadas são organizadas em *gerações*. A raiz (r) é a geração zero. Seus vizinhos formam a primeira geração e em geral vértices que distam k da raiz formam a k-ésima geração. Se um vértice v da k-ésima geração tem vizinhos na $(k+1)$-ésima geração, então esses vizinhos são chamados de *sucessores* de v. Logo, se v é um vértice da uma árvore enraizada T, então v pode ser considerado como a raiz da subárvore T_v composta por todos os sucessores iterado de v. Isso significa que uma árvore enraizada pode ser definida de forma recursiva. Essa propriedade de árvores enraizadas as tornam objetos mais fáceis de se trabalhar em problemas de contagem em comparação com árvores não-enraizadas. Existem várias aplicações de árvores enraizadas em problemas de computação. Elas aparecem naturalmente em estrutura de dados - a estruturas recursiva de pastas em um computador é basicamente uma árvore enraizada, por exemplo. Algoritmos como quicksort e o algoritmo de compressão de dados de Lampel-Zig têm uma relação direta com árvores enraizadas. Como citado, árvores enraizadas também aparecem em teoria da informação, abordada no apêndice C. Por exemplo, códigos instantâneos em um alfabeto m-ário são facilmente codificados como as folhas[1] de uma árvore m-área[2].

Funções geradoras são especialmente apropriadas para problemas de contagem de árvores enraizadas pois sua estrutura recursiva se traduz facilmente em uma relação de recorrência, como mostra o seguinte exemplo:

Exemplo B.3. Uma árvore binária é uma estrutura definida recursivamente como sendo um único vértice ou um vértice conectado com duas outras árvores binárias - uma sub-árvore esquerda e uma sub-árvore direita. Nosso problema é descobrir quantas árvores binárias com n folhas existem. Seja T_n o número de árvores binárias com $n+1$ folhas. A próxima figura representa as primeiras árvores dessa sequência:

[1] Vértices que não possuem sucessores

[2] Uma árvore m-área é uma generalização de árvore binária onde cada vértice está ligado a no máximo m sub-árvores.

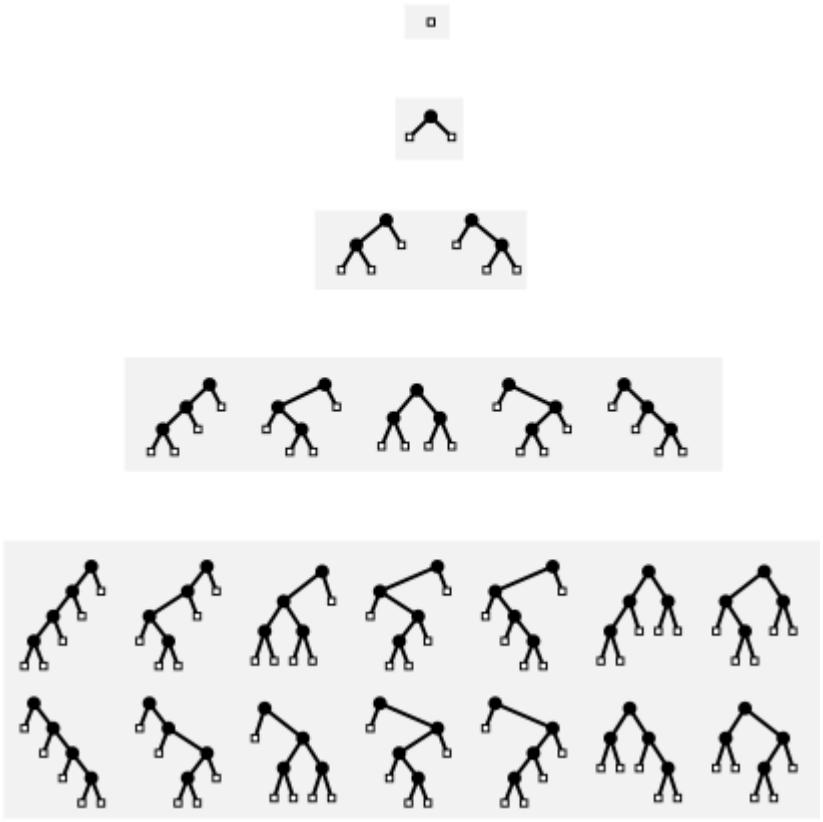

Figura 1: Árvores binárias com $n \in \{1, \ldots, 5\}$ folhas.

Temos, então, $T_0 = 1$, $T_1 = 1$, $T_2 = 2$, $T_3 = 5$ e $T_4 = 14$. Calculemos T_n: se, em uma árvore binária com $n+1$ folhas, a sub-árvore esquerda tem k folhas (existem T_{k-1} sub-árvores possíveis dessa forma) então a sub-árvore direita deverá ter $n-k+1$ (existem T_{n-k} sub-árvores possíveis dessa forma). Assim, T_n deve satisfazer

$$T_n = \sum_{1 \leq k \leq n} T_{k-1} T_{n-k} \ , \ \textit{para } n > 0, \ \textit{com } T_0 = 1.$$

Multiplicando por z^n e somando em n temos

$$\sum_{n \geq 1} T_n z^n = \sum_{n \geq 1} (\sum_{1 \leq k \leq n} T_{k-1} T_{n-k}) z^n \therefore$$

$$\sum_{n \geq 1} T_n z^n + T_0 - T_0 = \sum_{n \geq 1} (\sum_{0 \leq k \leq n-1} T_k T_{n-k-1}) z^{n-1} z \therefore$$

$$\sum_{n \geq 0} T_n z^n - T_0 = z \sum_{n-1 \geq 0} (\sum_{0 \leq k \leq n-1} T_k T_{n-k-1}) z^{n-1}.$$

Pondo $m = n - 1$ e considerando a regra do produto de Cauchy temos

$$\mathbf{T}(z) - T_0 = z\mathbf{T}^2(z) \therefore \mathbf{T}(z) = z\mathbf{T}^2(z) + 1.$$

O que se reduz a uma equação do segundo grau onde

$$\mathbf{T}(z) = \frac{1 \pm \sqrt{1-4z}}{2z}.$$

Se escolhermos o sinal positivo, o limite do lado direito para $z \to 0$ será ∞. Mas $T(0) = 1$. Se escolhermos o sinal negativo e fizermos uso da regra de L'Hospital, veremos, aliviados, que o lado direito atinge 1 no limite para $z \to 0$. Para extrair os outros coeficientes usaremos o teorema binomial generalizado para o expoente $\frac{1}{2}$:

$$z\mathbf{T}(z) = -\frac{1}{2}\sum_{n\geq 1}\binom{\frac{1}{2}}{n}(-4z)^n.$$

E, ajustando os coeficientes, temos

$$T_n = -\frac{1}{2}\binom{\frac{1}{2}}{n+1}(-4)^{n+1} = -\frac{1}{2}\frac{\frac{1}{2}(\frac{1}{2}-1)\ldots(\frac{1}{2}-n)(-4)^{n+1}}{(n+1)!} =$$

$$= \frac{1\cdot 3\cdot 5\ldots(2n-1)\cdot 2^n}{(n+1)!} = \frac{1}{n+1}\frac{1\cdot 3\cdot 5\ldots(2n-1)}{n!}\frac{2\cdot 4\cdot 6\ldots 2n}{1\cdot 2\cdot 3\ldots n} =$$

$$= \frac{1}{n+1}\binom{2n}{n}.$$

Esses números são conhecidos como os números de Catalan. □

Para mais exemplos (mais complicados), o leitor pode consultar [67].

B.2 Métodos de Obtenção de Funções Geradoras

A seção anterior nos mostrou um método para se obter funções geradoras. Dependendo da natureza do problema, outros métodos podem ser mais adequados. Essa seção apresenta ao leitor dois outros métodos.

B.2.1 O Método Simbólico

O método simbólico, utilizado no capítulo 4, é uma ferramenta poderosa para traduzir definições formais de objetos combinatórios em equações de funções geradoras. Essa seção apenas dá a primeira pincelada em uma grande tela. Para uma abordagem mais completa e profunda, consulte [33].

Quando contamos com funções geradoras, nós consideramos classes de objetos combinatórios com uma noção de tamanho definida para cada objeto. Para uma classe $\mathcal{A}$, nós denotamos o número de membros da classe de tamanho n por a_n. E a função geradora toma a forma

$$\mathbf{A}(z) = \sum_{n \geq 0} a_n z^n = \sum_{a \in \mathcal{A}} z^{|a|}.$$

Dadas duas classes $\mathcal{A}$ e $\mathcal{B}$ de objetos combinatórios, podemos criar outras classes como $\mathcal{A} + \mathcal{B}$, que consiste na classe composta pela união disjunta dos elementos das duas classes ou $\mathcal{A} \times \mathcal{B}$, que consiste na classe composta pelos pares ordenados de elementos de $\mathcal{A}$ e $\mathcal{B}$. Não é difícil mostrar que essas novas classes terão funções geradoras dadas por $\mathbf{A}(z) + \mathbf{B}(z)$ e $\mathbf{A}(z)\mathbf{B}(z)$, respectivamente.

O método simbólico consiste em combinar classes elementares para construir uma classe desejada. As classes elementares são: a classe vazia $\emptyset$, com função geradora 0, o objeto nulo ϵ, com função geradora 1 e objetos atômicos como 0 ou 1 em problemas binários, $\circ$ representando vértices em problemas de grafos, com função geradora z.

Exemplo B.4. Seja G a classe de sequências binárias que não possuem dois zeros consecutivos. Tais sequências podem ser ϵ, um único 0, ou um 1 ou um 01 seguido de uma sequência sem dois zeros consecutivos. O que se traduz simbolicamente em

$$G = \epsilon + \{0\} + \{1, 01\} \times G.$$

Então

$$\mathbf{G}(z) = 1 + z + (z + z^2)\mathbf{G}(z) \ \therefore \ \mathbf{G}(z) = \frac{1+z}{(1-z-z^2)}.$$

Logo, pelo resultado do exemplo B2, o número de sequências binárias de tamanho n que não possuem dois zeros consecutivos obedece a sequência de Fibonacci $G_n + G_{n+1} = G_{n+2}$. □

Exemplo B.5. Mostraremos que o número p_n de árvores enraizadas planares[3] com $n \geq 1$ vértices é dado por

$$p_n = \frac{1}{n}\binom{2n-2}{n-1}.$$

Seja $\mathcal{P}$ o conjunto de árvores enraizadas planares. Logo, pela definição de árvores enraizadas planares,

$$\mathcal{P} = \circ + \circ \times \mathcal{P} + \circ \times \mathcal{P}^2 + \circ \times \mathcal{P}^3 + \ldots.$$

Sendo $P(x) = \sum p_n x^n$ temos então

$$P(x) = x + xP(x) + xP(x)^2 + xP(x)^3 + \ldots = \frac{x}{1 - P(x)}.$$

Logo

$$P(x) = \frac{1 - \sqrt{1-4x}}{2} = xT(x).$$

Portanto

$$p_n = T_{n-1} = \frac{1}{n}\binom{2n-2}{n-1}.$$

□

Observação B.6. A relação $p_n = T_{n-1}$ tem um significado mais profundo. Existe uma bijeção entre árvores enraizadas planares com n vértices e árvores binárias com n folhas. Comecemos com uma árvore enraizada planar com n vértices e apliquemos o seguinte procedimento:

1. Delete a raiz e todos os elos envolvendo a raiz.

2. Se um vértice possui sucessores, delete todos os elos envolvendo o elo e os sucessores exceto o elo mais a esquerda.

3. Ligue todos os sucessores do passo anterior com um caminho (horizontal).

4. Rotacione esses novos caminhos 90º.

5. Os $n-1$ vértices restantes são agora considerados vértices internos de uma árvore binária bastando anexar as $n+1$ folhas que faltam.

Não é difícil verificar que esse procedimento é uma bijeção. □

[3] Árvores enraizadas onde cada vértice possui um número arbitrário de sucessores com uma ordem natural da esquerda para a direita.

Exemplo B.7. O problema de contar árvores binárias com n folhas também pode ser atacado por esse procedimento:

Uma árvore binária é composta por ou apenas um vértice ou um vértice ligado a duas sub-árvores que são, por sua vez, árvores binárias. Logo, sendo τ a classe das árvores binárias, temos

$$\tau = \epsilon + \circ \times \tau \times \tau$$

Como a condição inicial é $T_0 = 1$ temos, como esperado

$$\mathbf{T}(z) = 1 + z\mathbf{T}^2(z).$$

□

B.2.2 Árvores Simplesmente Geradas

Árvores simplesmente geradas[4] foram introduzidas por Meir e Moon em [49] e são generalizações de vários tipos de árvores enraizadas. Nessa seção, definimos esse objeto e provamos o teorema B.8 usado no capítulo 4. Seja

$$\phi(x) = \phi_0 + \phi_1 x + \phi_2 x^2 + \dots$$

uma série de potências com nenhum coeficiente negativo. Em particular, com $\phi_0 > 0$ e $\phi_j > 0$ para algum $j \geq 2$. Seja T uma árvore enraizada finita. Definimos o peso $w(T)$ dessa árvore por

$$w(T) = \prod_{j \geq 0} \phi_j^{D_j(T)},$$

onde $D_j(T)$ é o número de vértices de T com j sucessores. Definindo

$$y_n = \sum_{|T|=n} w(T),$$

então y_n denota um número ponderado de árvores de tamanho n. Por exemplo, se $\phi_j = 1$ para todo $j \geq 0$ (ou seja, se $\phi(x) = 1/(1-x)$) então todas as árvores enraizadas tem o mesmo peso $w(T) = 1$ e $y_n = p_n$ é o número de árvores planas. Se $\phi(x) = 1 + x + x^2$ então apenas árvores enraizadas cujos vértices tem menos do que três sucessores recebem um peso diferente de zero dado por $w(T) = 1$. Tais árvores são conhecidas como árvores de Motzkin e, nesse caso, y_n é o número de árvores de Motzkin de tamanho n. As árvores binárias também podem ser expressas dessa maneira: se

[4] Traduzido de simply generated trees.

$\phi(x) = 1 + 2x + x^2 = (1 + x)^2$ então os vértices com apenas um sucessor recebem peso 2. Isso se dá pois, em uma árvore binária, existem dois tipos de vértices com apenas um sucessor, aqueles com uma sub-árvore esquerda mas sem sub-árvore direita e aqueles com uma sub-árvore direita mas sem sub-árvore esquerda. De forma similar, uma árvore m-ária, uma generalização da árvore binária onde cada vértice está ligado a no máximo m sub-árvores, são contadas com o uso da função $\phi(x) = (1 + x)^m$.

Logo a função geradora

$$y(x) = \sum_{n \geq 1} y_n x^n$$

satisfaz a equação

$$y(x) = x\phi(y(x)).$$

Com efeito, dada estrutura recursiva das árvores simplesmente geradas, usaremos o método simbólico para atingir a equação desejada. A dificuldade consiste no fato de que não estamos trabalhando com uma estrutura fixa e sim com uma classe de objetos. Para driblar esse problema usaremos os coeficientes ϕ_i. Pelas definições e exemplos dados acima, é possível entender a informação "$\phi_i = a$" como "*os vértices com i filhos possuem a formas de dispor seus descendentes*". Portanto, para uma árvore simplesmente gerada T temos a relação recursiva ponderada

$$T = \phi_0 \cdot \circ + \phi_1 \cdot \circ \times T + \phi_2 \cdot \circ \times T^2 + \cdots .$$

Portanto $y(x) = x\phi(y(x))$

Provaremos agora o resultado que dá sentido a esse apêndice pois sustenta o lema 3.13:

Teorema B.8. *Seja R o raio de convergência de $\phi(t)$ e suponha que existe τ com $0 < \tau < R$ tal que $\tau\phi'(\tau) = \phi(\tau)$. Seja $d = mdc\{j > 0 : \phi_j > 0\}$. Então*

$$y_n = d\sqrt{\frac{\phi(\tau)}{2\pi\phi''(\tau)}}\frac{\phi'(\tau)^n}{n^{3/2}}(1 + O(n^{-1}) \ , \ n \equiv 1(mod\ d),$$

e $y_n = 0$ se $n \not\equiv 1(mod\ d)$.

Prova Aplicaremos o teorema B.38 para $F(x, y) = x\phi(y)$. Assumindo $d = 1$ as condições do teorema B.38 são satisfeitas e o resultado segue. Para $d > 1$ temos que $y_n = 0$ se $n \not\equiv 1(\text{mod } d)$. Logo $y(x) = \widehat{y}(x^d)/x^{d-1}$ e $\phi(x) = \widehat{\phi}(x^d)$ com $\widehat{y}$ e $\widehat{\phi}$ funções analíticas satisfazendo $\widehat{y}(x) = x\widehat{\phi}(\widehat{y}(x))$ com mdc $\widehat{d} = 1$. Logo estamos novamente nas condições do teorema B.38 e o resultado segue. ■

Observação B.9. Em tempo, cabe ressaltar que existe uma surpreendente relação entre árvores simplesmente geradas e o processo de ramificação de Galton-Watson. O leito interessado é convidado a consultar [23]. □

B.3 Séries de Potência Formais

Discutir a teoria formal de série de potências, e não a teoria analítica, significa observar tais séries somente como objetos algébricos em sua função de varal, i.é., com as potências servindo apenas como índices para seus coeficientes sem se importar com a função para a qual essa série possa estar convergindo. O objetivo de se estudar séries sob esse ponto de vista é conseguir fazer manipulações como as feitas na seção B.1 sem se preocupar com questões de convergência podendo, por exemplo, calcular a derivada de uma função geradora sem saber se ela converge para uma função. Essa seção se propõe a mostrar que não há falta de rigor ao se trabalhar dessa forma. Tais manipulações podem ser feitas no anel de séries de potências formais onde não existe o conceito de convergência. Nós podemos executar o método por completo, achar a função geradora em sua forma de série de potência, e só então descobrir se tal série converge ou não. Caso não convirja, ainda haverá muita informação a ser coletada da série formal mas não teremos acesso a nenhuma informação analítica, como fórmulas assintóticas para os coeficientes por exemplo. A série

$$f = 1 + x + 2x^2 + 6x^3 + 24x^4 + 120x^5 + \ldots + n!x^n + \ldots \ ,$$

por exemplo, pertence ao anel das séries de potências formais mas não converge para nenhum valor de x. Apesar disso, essa série desempenha um papel importante em alguns problemas de contagem.

Definição B.10. Uma série de potência formal é uma expressão da forma

$$a_0 + a_1x + a_2x^2 + \ldots,$$

onde a sequência $(a_n)_{n\in\mathbb{N}}$ é chamada de sequência dos coeficientes. Ainda, dizer que duas séries de potência são iguais significa dizer que os coeficientes são iguais.

Algumas operações podem ser feitas entre séries de potências formais como adição e subtração:

$$\sum a_nx^n \pm \sum b_nx^n = \sum(a_n \pm b_n)x^n.$$

E multiplicação, usando a regra do produto de Cauchy:

$$\sum a_n x^n \sum b_n x^n = \sum c_n x^n \ , \ onde \ c_n = \sum_k a_k b_{n-k}.$$

Essa regra para o produto confere uma aplicabilidade de séries de potências a muitos problemas de combinatória. De fato, é frequente conseguirmos construir todos os objetos a_n de objetos do tipo n de uma família escolhendo um objeto do tipo k e outro do tipo $n-k$ e colocando-os juntos para forma um objeto do tipo n (isso é feito no exemplo B.3). O número de maneiras de se fazer isso será $a_k a_{n-k}$, e se somarmos em k nós veremos que o produto de Cauchy é de fundamental importância para o problema em questão.

Usando a propriedade distributiva, nós obtemos

$$(1-x)(1+x+x^2+x^3+\ldots) = 1.$$

Logo, podemos dizer que a série $(1-x)$ tem um inverso multiplicativo dado por $1+x+x^2+\ldots$ (e vice-versa).

Proposição B.11. *Uma série de potência formal $f = \sum a_n x^n$ tem uma inverso multiplicativo se, e somente se, $a_0 \neq 0$. Nesse caso o inverso multiplicativo é único.*

Prova: Seja f uma série de potência com inverso multiplicativo $1/f = \sum b_n x^n$. Então $f(1/f) = 1$ e $a_0 b_0 = 1$. Logo $a_0 \neq 0$. Além disso, sabemos que, para $n \geq 1$, $\sum_k a_k b_{n-k} = 0$, de onde tiramos

$$b_n = \frac{-1}{a_0} \sum_{k \geq 1} a_k b_{n-k}.$$

Logo $b_1, b_2, \ldots$ estão unicamente determinados. Reciprocamente, suponha que $a_0 \neq 0$. Então podemos determinar $b_0, b_1, b_2, \ldots$ como acima e a série resultante é o inverso multiplicativo de F. ■

As séries de potencia formais com as regras aritméticas descritas formam um anel cujos elementos invertíveis são as séries com termo independente diferente de zero. É importante ressaltar que o conceito de inverso multiplicativo de uma série difere do conceito de inverso de uma série. O inverso de uma série f, se existir, é uma série g que obedece $f(g(x)) = g(f(x)) = x$, para todo x. Vejamos quando uma série possui um inverso.

Definição B.12. se $f = \sum a_n x^n$ então $f(g(x)) = \sum a_n g(x)^n$.

Se a série $g(x)$ tiver elemento independente igual a zero, então nós sabemos calcular os coeficientes de $f(g(x))$. De fato, se quisermos, para algum k, calcular o coeficiente de x^k, podemos observar que em

$$a_n g(x)^n = a_n(g_1 x + g_2 x^2 + \ldots)^n = a_n x^n (g_1 + g_2 x + \ldots)^n$$

se $n > k$ só existirão potências de x maiores que k e não precisamos nos preocupar com eles. Logo teremos um processo bem definido (com número finito de passos) para calcular o coeficiente de x^k. Se $g_0 \neq 0$, o mesmo não pode ser feito e, se não for polinomial, o processo para calcular cada coeficiente é infinito e dependente da convergência da série para estar bem definido. Como no anel de séries formais não existe o conceito de convergência, $f(g(x))$ só está definido se f for polinomial ou se $g_0 = 0$.

No anel das séries de potência formais existem outras operações análogas às operações definidas no cálculo (porém sem o conceito de limite). Por exemplo, a derivada de uma série de potência formal $f = \sum a_n x^n$ é a série $f' = \sum n a_n x^{n-1}$. A derivação segue as regras usuais do calculo tais como soma, produto e cociente. Muitas delas, inclusive, são mais fáceis de provar nesse contexto. Por exemplo:

Proposição B.13. *Se $f' = 0$ então $f = a_0$ é constante.*

Prova: $f' = 0$ significa que a série de potência formal f' é idêntica à série de potência formal 0, o que significa que todos os coeficientes de f' são iguais á zero. Mas os coeficientes de f' são $a_1, 2a_2, 3a_3, \ldots$. Logo $a_j = 0$ para todo $j \geq 1$ e $f = a_0$ é constante. ■

Proposição B.14. *Se $f' = f$ então $f = ce^x$.*

Prova: Como $f' = f$, os coeficientes de x^n devem ser os mesmo em f e em f', para todo $n \geq 0$. Logo $(n+1)a_{n+1} = a_n$, para todo $n \geq 0$ e $a_{n+1} = a_n/(n+1), (n \geq 0)$. Por indução em n, $a_n = a_0/n!$, para todo $n \geq 0$. Portanto $f = a_0 e^x$. ■

B.4 Funções Geradoras Ordinárias

Operações com séries formais implicam em operações correspondentes em seus coeficientes. Se a série convergir para uma função, então operações com tal função correspondem a operações nos coeficientes da série de potência obtida ao expandir a função. Nessa seção, nós exploraremos algumas dessas relações cujo entendimento facilita o manuseio e entendimento de como se usar as funções geradoras. Usaremos a seguinte definição:

Definição B.15. O símbolo $f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$ significa que f é a função geradora ordinária para a sequência $(a_n)_{n\in\mathbb{N}}$, i.e., que

$$f = \sum a_n x^n.$$

B.4.1 Propriedades das Funções Geradoras Ordinárias

Sejam $f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$. Procuraremos agora a função que gera $(a_{n+1})_{n\in\mathbb{N}}$. Para tanto, precisamos fazer alguns cálculos:

$$\sum a_{n+1}x^n = \frac{1}{x}\sum_{m\geq 1} a_m x^m = \frac{f(x)-f(0)}{x}.$$

Então

$$f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}} \Rightarrow (\frac{f-a_0}{x}) \overset{\text{fgo}}{\longleftrightarrow} (a_{n+1})_{n\in\mathbb{N}}.$$

Se transladarmos os índices em duas unidades apenas precisaremos repetir o cálculo feito acima para concluir que

$$(a_{n+2})_{n\in\mathbb{N}} \overset{\text{fgo}}{\longleftrightarrow} \frac{((f-a_0)/x)-a_1}{x} = \frac{f-a_0-a_1x}{x^2}.$$

Note que esse fato facilita entender a relação de recorrência de Fibonacci $F_{n+2} = F_{n+1}+F_n$, para $n \geq 0$, $F_0 = 0$ e $F_1 = 1$. De fato, a partir da relação, sendo $f(x) = \sum F_n x^n$, temos que

$$\frac{f-x}{x^2} = \frac{f}{x} + f$$

.

A próxima proposição – cuja prova é deixada para o leitor – generaliza o que foi feito acima.

Proposição B.16. *sejam* $f \overset{fgo}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$ *e* $h > 0$. *Então*

$$(a_{n+h})_{n\in\mathbb{N}} \overset{fgo}{\longleftrightarrow} \frac{f-a_0-\ldots-a_{h-1}x^{h-1}}{x^h}.$$

Agora, vejamos o que acontece quando multiplicamos uma série por uma potência de n. De novo, sejam $f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$. Para achar a função geradora de $(na_n)_{n\in\mathbb{N}}$ basta observar que $\sum(na_n)x^n = xf'$. Logo, multiplicar o n-ésimo elemento da sequência por n implica em uma 'multiplicação' por $x(d/dx)$ em sua função geradora cuja notação será xD. i.e.:

$$f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}} \Rightarrow (xDf) \overset{\text{fgo}}{\longleftrightarrow} (na_n)_{n\in\mathbb{N}}.$$

Exemplo B.17. Considere a seguinte relação de recorrência

$$(n+1)a_{n+1} = 3a_n + 1 \ , \ n \geq 0, \ a_0 = 1.$$

Se f é a função geradora ordinária da sequência $(a_n)_{n\in\mathbb{N}}$, então, pelo o que foi feito acima e pela proposição B.16, temos que

$$f' = 3f + \frac{1}{1-x},$$

uma equação diferencial de primeira ordem na incógnita f. □

Para achar a função que gera $(n^2 a_n)_{n\in\mathbb{N}}$ basta aplicarmos o operador xD novamente obtendo $(xD)^2 f$ como resposta. Em geral

$$(xD)^k f \overset{\text{fgo}}{\longleftrightarrow} (n^k a_n)_{n\in\mathbb{N}}.$$

O que deve gerar $((3-7n^2)a_n)_{n\in\mathbb{N}}$? De novo, basta aplicarmos ao operador xD o mesmo que é aplicado a n. i.e., a resposta é $(3-7(xD)^2)f$. O caso geral – também deixado como exercício para o leitor – é dado por

Proposição B.18. *Sejam $f \overset{fgo}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$ e P um polinômio. Então*

$$P(xD)f \overset{fgo}{\longleftrightarrow} (P(n)a_n)_{n\in\mathbb{N}}.$$

Exemplo B.19. Vamos usar o que foi discutido para achar uma fórmula fechada para a soma dos quadrados dos primeiros N inteiros positivos. Para tanto, comecemos com a identidade

$$\sum_{n=0}^{N} x^n = \frac{x^{N+1}-1}{x-1}$$

Se aplicarmos $(xD)^2$ em ambos os lados e fixarmos $x = 1$, o lado esquerdo será a soma dos quadrados enquanto o lado direito nos dará a resposta:

$$\sum_{n=1}^{N} n^2 = (xD)^2 \left(\frac{x^{N+1}-1}{x-1}\right)\Big|_{x=1}.$$

Após duas diferenciações e alguma álgebra, chegamos na resposta

$$\sum_{n=1}^{N} n^2 = \frac{N(N+1)(2N+1)}{6} \ , \text{ para } N \in \{1, 2, \ldots\}.$$

De fato essa é uma fórmula já muito conhecida. O que é importante ressaltar aqui é que as funções geradoras resolvem de forma mecânica muitos problemas complexos envolvendo somas[5]. □

A próxima proposição, cuja prova é facilmente derivada do produto de Cauchy, aborda novamente a multiplicação de funções geradoras.

Proposição B.20. *se* $f_k \overset{fgo}{\longleftrightarrow} ((a_k)_n)_{n\in\mathbb{N}}$, *para todo* $k \in \{1,\ldots,K\}$. *então*

$$\prod_{k=1}^{K} f_k \overset{fgo}{\longleftrightarrow} \Big(\sum_{n_1+n_2+\ldots+n_K=n} ((a_1)_{n_1}(a_2)_{n_2}\ldots(a_K)_{n_K})_{n\in\mathbb{N}}\Big).$$

Exemplo B.21. Seja $f(n,k)$ o número de maneiras que um inteiro não negativo n pode ser escrito como uma soma ordenada de k inteiros não negativos. Por exemplo, $f(4,2) = 5$ pois $4 = 4+0 = 3+1 = 2+2 = 1+3 = 0+4$. Como $1/(1-x) \overset{\text{fgo}}{\longleftrightarrow} 1$[6], nós temos que

$$1/(1-x)^k \overset{\text{fgo}}{\longleftrightarrow} (f(n,k))_{n\in\mathbb{N}}.$$

Logo, após algum trabalho algébrico, chegamos a $f(n,k) = \binom{n+k-1}{n}$. □

Agora, sejam $f \overset{\text{fgo}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$. Procuraremos a sequência gerada por $f(x)/(1-x)$. Observe que

$$\frac{f(x)}{(1-x)} = (a_0 + a_1x + a_2x^2 + \ldots)(1 + x + x2 + \ldots) =$$

$$= a_0 + (a_0+a_1)x + (a_0+a_1+a_2)x^2 + (a_0+a_1+a_2+a_3)x^3 + \ldots$$

que indica que

Proposição B.22. *Se* $f \overset{fgo}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$, *então*

$$\frac{f}{(1-x)} \overset{fgo}{\longleftrightarrow} (\sum_{j=0}^{n} a_j)_{n\in\mathbb{N}}$$

[5]Se algo não parece estar certo no exemplo B19, veja o exemplo B27.
[6]Pois $\sum_{n=0}^{\infty} x^n = 1/(1-x)$.

Exemplo B.23. Os números harmônicos $(H_n)_{n\in\mathbb{N}}$ obedecem

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} \ , \ \textit{para } n \geq 1.$$

Pela proposição B.22, a função geradora de $(H_n)_{n\in\mathbb{N}}$ é o resultado da multiplicação de $1/(1-x)$ pela função geradora f da sequência $(1/n)_{n\in\mathbb{N}}$. Como $f = \sum_{n\geq 1} x^n/n$ temos que $f' = 1/(1-x)$. Logo $f = -\log(1-x)$. Portanto

$$\sum_{n\geq 1}^{\infty} H_n x^n = \frac{1}{1-x}\log(\frac{1}{1-x}).$$

□

Exemplo B.24. Mostremos que os números de Fibonacci satisfazem

$$F_0 + F_1 + F_2 + \ldots + F_n = F_{n+2} - 1 \ , \ \textit{para } n \geq 0 :$$

Pela proposição B.22, se F for a função geradora da sequência de Fibonacci, temos que a sequência geradora da sequência do lado esquerdo da igualdade é dada por $F/(1-x)$. Como visto no exemplo B2, $F = x/(1-x-x^2)$. Ainda, pela proposição B9, a função geradora da sequência do lado direito da igualdade é

$$\frac{F-x}{x^2} - \frac{1}{1-x},$$

restando um trabalho algébrico para verificar a igualdade. □

Observação B.25. Essa seção abordou os conceitos básicos de funções geradoras ordinárias. Existem muitos outros tipo de funções geradoras como funções geradoras de Poisson, séries de Lambert, séries de Bell, funções geradoras exponenciais, etc.. Toda sequência com o conjunto de índices começando em 1 possui todos os tipos de funções geradoras a diferença sendo que cada problema é melhor abordado por uma ou mais funções específicas. Esse capítulo não se propõe a definir muitas funções geradoras e, consequentemente, também não abordaremos a questão de como saber qual função geradora se usar em diferentes problemas. O leitor interessado pode consultar as referências dadas no início do capítulo. Vamos, contudo, definir e dar um exemplo de funções geradoras exponenciais. Não enunciaremos porém tudo o que foi provado nessa seção para função geradoras ordinárias possui um análogo para funções geradoras exponenciais.

Definição B.26. O símbolo $f \overset{\text{fge}}{\longleftrightarrow} (a_n)_{n\in\mathbb{N}}$ significa que f é a função geradora exponencial para a sequência $(a_n)_{n\in\mathbb{N}}$, i.e., que

$$f = \sum \frac{a_n}{n!} x^n.$$

Exemplo B.27. Vamos usar funções geradoras exponenciais e a proposição B.16 para calcular $\sum(n^2+4n+5)/n!$. Claramente, a resposta é o valor em $x = 1$ da série

$$((xD)^2 + 4(xD) + 5)e^x = (x^2 + x)e^x + 4xe^x + 5e^x = (x^2 + 5x + 5)e^x.$$

Portanto a resposta é $11e$. Mas nós realizamos um movimento ilegal. Você sabe dizer qual? Nós calculamos a função geradora para um valor de x. Tal operação não existe no anel das séries de potência formais. Nesse anel, as séries não assumem valores para diferentes valores de x. A letra x é apenas um símbolo representando um pregador no varal. O que pode ser calculado para um valor de x é a série de potência que converge naquele ponto, uma ideia analítica portanto. A ideia é que, se após calcularmos a função geradora usando a teoria formal nós percebermos que a série converge para uma função analítica em um disco no plano complexo, então toda a dedução formal que fizemos é também válida do ponto de vista analítico dentro desse disco. Então podemos mudar o ponto de vista se assim nos convier. A próxima seção se propõe a estudar um pouco a teoria de funções analíticas com o objetivo de provar o teorema B.8 usado no problema de contagem do capítulo 3. □

□

B.5 A Teoria Analítica Aplicada a Problemas de Contagem

A teoria formal de séries de potência nos permite manipular relações de recorrência e resolver equações funcionais – como funções diferenciais – para séries de potências sem nos preocupar com questões de convergência. Porém, se a série convergir para uma função nós ainda conseguiremos obter informações analíticas possivelmente difíceis de serem obtidas de outra maneira. Esse capítulo se propõe a estudar a teoria analítica necessária para provar o teorema B.8. O leitor interessado em se aprofundar no tema é indicado a consultar [33].

B.5.1 Propriedades Analíticas

Nosso objetivo nessa seção é desenvolver a teoria necessária para provar o teorema B.38. Nós provaremos a existência do raio de convergência para serie de potência em $\mathbb{C}$ depois generalizaremos nosso espaço para $\mathbb{C}^n$ e provaremos em parte o teorema de preparação de Weierstrass, o teorema da função implícita e, finalmente, o teorema B.38.

Considere a série

$$f = \sum a_n z^n,$$

onde usamos a letra z para encorajar o entendimento de que estamos agora no plano complexo. Nosso desafio agora é descobrir para quais valores complexos a série f converge e expressá-los em termos dos coeficientes $(a_n)_{n\in\mathbb{N}}$.

Antes, precisaremos da seguinte definição:

Definição B.28. Seja L um elemento da reta real estendida. Dizemos que L é o limite superior de uma sequência $(x_n)_{n\in\mathbb{N}}$ se

(a) L é finito e

(i) para todo $\epsilon > 0$, todos com exceção de um número finito de termos da sequência satisfazem $x_n < L + \epsilon$, e

(ii) para todo $\epsilon > 0$, um número infinito de termos da sequência satisfaz $x_n > L - \epsilon$, ou

(b) $L = +\infty$ e, para todo $M > 0$, Existe um n para o qual $x_n > M$, ou

(c) $L = -\infty$ e para todo x, existe somente um número finito de termos que obedecem $x_n > x$.

Se L é o limite superior de uma sequência $(x_n)_{n\in\mathbb{N}}$ então dizemos que $L = limsup(x_n)$.

O limite superior tem as seguinte propriedades:

• Toda sequência de números reais tem um e apenas um limite superior na reta real estendida.

• Se uma sequência tem limite L, então L é também o limite superior da sequência.

• Se S é o conjunto dos pontos de aderência de $(x_n)_{n\in\mathbb{N}}$, então $limsup(x_n)$ é o sup do conjunto S.

Agora estamos prontos para o

Teorema B.29. *Existe um número $0 \leq R \leq +\infty$, chamado de raio de convergência da série f tal que a série converge para todo z com $|z| < R$ e diverge para todo z com $|z| > R$. Ainda, R é dado por*

$$R = \frac{1}{limsup|a_n|^{1/n}} \ , \ (1/0 = \infty \ e \ 1/\infty = 0).$$

Prova: Suponha que $0 < R < \infty$. Seja z tal que $|z| < R$. Nós mostraremos que a sequência converge em z. Seja $\epsilon > 0$ tal que

$$|z| < \frac{R}{1 + \epsilon R}.$$

Agora, pela definição de $limsup$, existe um N para o qual para todo $n > N$ nós temos

$$|a_n|^{1/n} < \frac{1}{R} + \epsilon.$$

Logo, para todo $n > N$,

$$|a_n z||z|^n < (|z|(\frac{1}{R} + \epsilon))^n.$$

Seja $\alpha = |z|(\frac{1}{R} + \epsilon)$. Então, pela nossa escolha de ϵ, nós temos $\alpha < 1$. Portanto a série $\sum a_n z^n$ converge absolutamente pela comparação com os termos de uma série geométrica convergente.

Agora provaremos que a série diverge para $|z| > R$. De fato, como $|z| > R$, existe $\epsilon > 0$ para o qual $\theta = |(z/R) - \epsilon z| > 0$. Pela definição de $limsup$, para infinitos valores de n nos temos $|a_n|^{1/n} > (1/R) - \epsilon$. Logo, para tais valores de n,

$$|a_n z^n| > |(\frac{1}{R} - \epsilon)z|^n = \theta^n$$

que não tem limite superior pois $\theta > 1$. Logo, essa subsequência dos termos da série de potência não converge para zero e portanto a série diverge. Isso completa a prova para o caso $0 < R < \infty$. Os casos onde $R = 0$ e $R = +\infty$ são provados similarmente. ■

Nosso desafio agora é provar dois dos resultados centrais da teoria de análise complexa: o teorema de preparação de Weierstrass e o teorema da função implícita. Para tanto, precisamos desenvolver um pouco nosso linguajar:

Definição B.30. O conjunto de séries de potência formais com n entradas em $\mathbb{C}$ denotado por

$$\mathbb{C}[[X_1, \ldots, X_n]] = \mathbb{C}[[X]]$$

consiste em todas as expressões da forma

$$P := \sum_{v \in \mathbb{N}^n} a_v X^v = \sum a_{v_1 \ldots v_n} X_1^{v_1} \ldots X_n^{v_n},$$

com $a_v \in \mathbb{C}$ para todo $v \in \mathbb{N}^n$.

As operações de soma, multiplicação e multiplicação por escalar são definidas de maneira natural, i.e., dado $P_1, P_2 \in \mathbb{C}[[X_1, \ldots, X_n]]$ com $P_1 = \sum_{v \in \mathbb{N}^n} a_v X^v$ e $P_2 = \sum_{v \in \mathbb{N}^n} b_v X^v$ e $\lambda \in \mathbb{C}$ temos

$$P_1 + P_2 := \sum_{v \in \mathbb{N}^n} (a_v + b_v) X^v,$$

$$P_1 \cdot P_2 := \sum_m (\sum_{\alpha+\beta=m} a_\alpha b_\beta) X^m \text{ e}$$

$$\lambda P_1 := \sum_{v \in \mathbb{N}^n} \lambda a_v X^v.$$

Nós queremos introduzir o conceito de convergência no conjunto $\mathbb{C}[[X]]$. Para tanto, definiremos uma pseudonorma (i.e., uma norma que pode assumir o valor ∞).

Proposição B.31. *Para cada $r \in \mathbb{R}^n_{>0}$, a função*

$$||\cdot||_r : \mathbb{C}[[X]] \to \mathbb{R} \cup \{\infty\}, \ \sum_{v \in \mathbb{N}^n} (a_v) X^v \mapsto \sum_{v \in \mathbb{N}^n} (|a_v|) r^v$$

é uma pseudonorma.

A prova da proposição B.31 segue do Lema 21.5 de [42].

Definição B.32. Uma série de potência formal $P \in \mathbb{C}[[X_1, \ldots, X_n]]$ é convergente se existe um $r \in \mathbb{R}^n_{>0}$ para o qual $||P|| < \infty$. O conjunto de séries de potência formais convergentes é denotado por $\mathbb{C}\{X_1, \ldots, X_n\} = \mathbb{C}\{X\}$.

Usaremos as notações $\mathbb{C}[[X, Y]] := \mathbb{C}[[X_1, \ldots, X_n, Y]]$ e $\mathbb{C}\{X, Y\} := \mathbb{C}\{X_1, \ldots, X_n, Y\}$, para $X \in \mathbb{C}^n$, para algum $n \in \mathbb{N}$ e $Y \in \mathbb{C}$.

Definição B.33. Uma série de potência $P \in \mathbb{C}[[X,Y]]$ é dita:

i) distinguida em Y com ordem b se $P(0,\ldots,0,Y) = Y^b \cdot e$ para alguma unidade[7] $e \in \mathbb{C}[[Y]]$.

ii) um polinômio de Weierstrass de grau b em Y se $P = Y^b + \sum_{j=1}^{b} a_j Y^{b-j}$ com $a_j \in \mathfrak{m}_{[X]}$, para cada $j \in \{1,\ldots,b\}$, onde $\mathfrak{m}_{[X]} = \{P \in \mathbb{C}[[X]] : P(0) = 0\}$.

Finalmente possuímos o vocabulário para enunciar o

Teorema B.34. ***Teorema de Preparação de Weierstrass.*** *Se $P \in \mathbb{C}[[X,Y]]$ é distinguida em Y com grau b então existe um polinômio de Weierstrass $\omega \in \mathbb{C}[[X,Y]]$ de grau b e uma unidade $e \in \mathbb{C}[[X,Y]]$ para os quais $P = \omega \cdot e$. Ainda mais, ω e e são únicos e se $P \in \mathbb{C}\{X,Y\}$ então o mesmo vale para ω e e.*

Prova: Para $b = 0$ o resultado é trivial, pois a serie de potência P é uma unidade. Seja $b \geq 1$. No caso $n = 0$ de novo o resultado segue facilmente: Y^b é o polinômio de Weierstrass de grau b e existe exatamente uma unidade $e \in \mathbb{C}[[Y]]$ tal que $P = eY^b$. Para $b \geq 1$ e $n \geq 1$ nós definimos a transformação

$$\phi : \mathbb{C}[[X,Y]] \to \mathbb{C}[[S,T]],\ X_j \mapsto S_j T^b,\ Y \mapsto T,$$

onde $S \in \mathbb{C}^n$ e $T \in \mathbb{C}$. Para analisar as propriedades de ϕ nós definiremos a seguinte função peso γ: para cada monômio $aS^\sigma T^\tau$, com $a \neq 0$, definimos

$$\gamma(aS^\sigma T^\tau) := \tau - b|\sigma| \in \mathbb{Z},$$

e escrevemos $\gamma(P) \geq m$, para $P \in \mathbb{C}[[S,T]]$, para indicar que os únicos monômios que aparecem em P tem peso pelo menos m. Também usamos os símbolos $\leq, <$ e $>$ de forma correspondente. Em particular, $\gamma(0)$ satisfaz todas as desigualdades. Com isso podemos enunciar o

Lema B.35. *i) Im $\phi = \{P \in \mathbb{C}[[S,T]] : \gamma(P) \geq 0\}$.*

ii) P é distinguida em Y com ordem b se, e somente se, $\phi(P) = eT^b$ para alguma unidade $e \in \mathbb{C}[[S,T]]$.

iii) P é um polinômio de Weierstrass de grau b em Y se, e somente se, $\phi(P) = (1+u)T^b$ para algum $u \in \mathbb{C}[[S,T]]$ tal que $\gamma(u) < 0$ (em particular, $u \in \mathfrak{m}_{[S,T]}$.

iv) Cada unidade $e \in \mathbb{C}[[S,T]]$ tem uma única decomposição da forma $e = (1+u)\widehat{e}$ tal que $\widehat{e}$ é uma unidade, $\gamma(\widehat{e}) \geq 0$ e $\gamma(u) < 0$. Se e é convergente então $\widehat{e}$ e u também o são.

[7] Um elemento é uma unidade se possui um inverso multiplicativo.

Este lema está provado em [42] (lema 22.5).

Agora, seja P distinguida em Y com ordem b. Logo, pelo Lema B.35 ii), existe exatamente uma única unidade $e^{'} \in \mathbb{C}[[S,T]]$ para a qual $\phi(P) = e^{'}T^b$ que implica na existência da seguinte decomposição de P:

$$\phi(P) = e^{'}T^b \overset{\text{iv}}{=} (1+u)\widehat{e}T^b \overset{\text{i}}{=} \phi(e)(1+u)T^b,$$

onde $e \in \mathbb{C}[[X,Y]]$ é uma unidade. Logo, pelo lema B.35 iii, a série de potência $\omega := e^{-1}P \in \mathbb{C}[[X,Y]]$ é um polinômio de Weierstrass de grau b. Para a unicidade e resultados complementares, indico novamente a referência [42]. ■

Provaremos agora, como consequência do teorema de preparação de Weierstrass, o teorema da função implícita, outro resultado central da teoria analítica. Cabe comentar, porém, que esses teoremas são de fato equivalente, i.e., o teorema de preparação de Weierstrass pode ser visto como uma consequência do teorema da função implícita.

Teorema B.36. ***Teorema da Função Implícita.*** *Sejam $X \times Y \subset \mathbb{C}^n \times \mathbb{C}^m$ e $f : X \times Y \to \mathbb{C}^m$ holomorfa com $f = (f_1, \ldots, f_m)$ tal que, para cada $j \in \{1, \ldots, m\}$, $f_j \in \mathfrak{m}_{[X,Y]}$ e $\frac{\partial f}{\partial y}(0) \neq 0$. Então existe uma vizinhança aberta de $(0,0)$, $U \times W$, tal que existe uma função $g : U \to W$ com $g = (g_1, \ldots, g_m)$ tal que, para cada $j \in \{1, \ldots, m\}$, $g_j \in \mathfrak{m}_{[U]}$ e $f(u,w) = 0 \iff w = g(u)$.*

Prova: Após uma mudança de variáveis, podemos assumir que $A := \frac{\partial f}{\partial y}(0)$ é a matriz I_m (como a condição "$f(x, A^{-1}y) = 0 \iff y = g(x)$" implica "$f(x,y) = 0 \iff y = (A^{-1}g(x)$" e, pela regra da cadeia, $\frac{\partial (fA^{-1}}{\partial y}(0) = I_m$). Então, em particular, f_m é distinguida com ordem 1 em Y_m. Logo, pelo teorema de preparação de Weierstrass, temos uma única decomposição $f_m = e(y_m - g_0)$ com $g_0 \in \mathfrak{m}_{[X,Y']}$. É fácil de ver que podemos escolher $e = 1$. Usaremos agora um argumento de indução em m. O caso $m = 1$ é trivial. Suponha agora que o teorema seja válido para $m-1$ para algum $m > 1$. Mostremos que o mesmo vale para m: seja $h = (h_1, \ldots, h_{m-1})$ tal que, para $j \in \{1, \ldots, m-1\}$, $h_j \in \mathfrak{m}_{[X,Y']}$ definida por

$$h(x, y') = f'(x, y', g_0(x, y')).$$

Logo $\frac{\partial h}{\partial y'}(0) = I_{m-1}$, e a hipótese de indução garante a existência de uma única $g' = (g_1, \ldots, g_{m-1})$ tal que, para $j \in \{1, \ldots, m-1\}$, $h_j \in \mathfrak{m}_{[X]}$ e

$$h(x, y') = 0 \iff y' = g'(x).$$

Logo $g := (g_1, \ldots, g_m)$ tem as propriedades desejadas. Com efeito, $f(x, g(x)) = 0$ pois $f_m(x, g(x)) = g_m(x) - g_0(x, g'(x)) = 0$ e $f'(x, g'(x), g_m(x)) = h(x, g'(x)) = 0$. Por outro lado, se $f(x,y) = 0$ então o fato de que $f_m(x,y) =$

0 implica que $y_m = g_0(x, y')$ e a igualdade $0 = f'(x, y', g_0(x, y')) = h(x, y')$ implica que $y' = g'(x)$. Em particular, $y_m = g_0(x, g'(x)) = g_m(x)$. ■

Estamos a um lema de estarmos aptos a provar o resultado alvo dessa seção:

Lema B.37. *Seja* $A(x) = \sum_{n\in\mathbb{N}} a_n x^n$ *uma função analítica na região*

$$\Delta = \{x : |x| < x_0 + \eta, |arg(x - x_0)| > \delta\},$$

onde $x_0, \eta \in \mathbb{R}^+$ *e* $0 < \delta < \pi/2$. *Suponha ainda que existe* $\alpha \in \mathbb{R}$ *para o qual*

$$A(x) = O((1 - x/x_0)^{-\alpha}), \quad \textit{para todo } x \in \Delta.$$

Então

$$a_n = O(x_0^{-n} n^{\alpha-1}).$$

O lema B.37 está provado em [23] (lema 1). Finalmente estamos prontos para o

Teorema B.38. *Seja* $F(x, y)$ *uma função analítica ao em uma vizinhança de (0,0) tal que* $F(0, y) = 0$ *e todos seus coeficientes de Taylor ao redor de zero são reais não negativos. Então existe uma única solução analítica* $y = y(x)$ *para a equação*

$$y = F(x, y) \tag{1}$$

cujos coeficientes de Taylor ao redor de zero são todos reais não negativos e $y(0) = 0$.

Se a região de convergência de $F(x, y)$ *possuir uma solução positiva* (x_0, y_0) *do sistema*

$$y = F(x, y)$$

$$1 = F_y(x, y),$$

com $F_x(x_0, y_0) \neq 0$ *e* $F_{yy}(x_0, y_0) \neq 0$ *então* $y(x)$ *é analítica para* $|x| < x_0$ *e existem funções* $g(x)$ *e* $h(x)$ *analíticas ao redor de* $x = x_0$ *para as quais* $y(x)$ *tem uma representação da forma*

$$y(x) = g(x) - h(x)\sqrt{1 - \frac{x}{x_0}}, \tag{2}$$

em uma vizinhança de x_0. *Temos também que*

$$g(x_0) = y(x_0), \text{ e } h(x_0) = \sqrt{\frac{2x_0F_x(x_0,y_0)}{F_{yy}(x_0,y_0)}}.$$

Ainda, se assumirmos que $[x^n]y(x)$[8] > 0 *para* $n > n_0$ *então* x_0 *é a única singularidade de* $y(x)$ *no círculo* $|x| = x_0$ *e obtemos uma expansão assintótica para* $[x^n]y(x)$ *da forma*

$$[x^n]y(x) = \sqrt{\frac{x_0F_x(x_0,y_0)}{2\pi F_{yy}(x_0,y_0)}}x_0^{-n}n^{-3/2}(1+O(n^{-1}). \tag{3}$$

Prova Primeiro mostraremos que existe uma solução (analítica) $y = y(x)$ de $y = F(x, y)$ com $y(0) = 0$. Como $F(0, y) = 0$ temos que a função

$$y(x) \mapsto F(x, y(x))$$

é uma contração para valores pequenos de x. Logo as funções definida iterativamente $y_0(x) :\equiv 0$ e, para $m \geq 0$, $y_{m+1}(x) = F(x, y_m(x))$ convergem uniformemente para uma função $y(x)$ solução de (1). Temos ainda que, para todo $m \geq 0$, y_m é analítica ao redor de zero e tem coeficientes de Taylor reais não negativos. Portanto seu limite uniforme goza das mesmas propriedades.

Seja x_0 o raio de convergência de $y(x)$. Então x_0 é uma singularidade de $y(x)$. Supondo $y(x)$ regular, a função

$$x \mapsto F_y(x, y(x))$$

é estritamente crescente em $\mathbb{R}^+$. Note que $F_y(0, y(0)) = 0$. Supondo $F_y(x, y(x)) < 1$ temos, pelo teorema da função implícita, que $y(x)$ é regular em uma vizinhança de x. Logo existe um limite x_0 para o qual $\lim_{x\to x_0^-} y(x) = y_0$ é finito e satisfaz $F_y(x_0, y_0) = 1$. Se $y(x)$ fosse regular em x_0 então

$$y^{'}(x_0) = F_x(x_0, y(x)) + F_y(x_0, y(x_0))y^{'}(x_0)$$

implicaria que $F_x(x_0, y(x_0)) = 0$, o que sabemos não ser verdade. Logo $y(x)$ é singular em x_0 (i.e., x_0 é o raio de convergência de $y(x)$) e $y(x_0)$ é finito.

Considere agora a equação $y - F(x, y) = 0$ ao redor de (x_0, y_0). Nós temos que $1 - F_y(x_0, y_0) = 0$ mas $-F_{yy}(x_0, y_0) \neq 0$. Logo, pelo teorema de preparação de Weierstrass, existem funções $H(x, y)$, $p(x)$ e $q(x)$ analíticas ao redor de (x_0, y_0) que satisfazem $H(x_0, y_0) \neq 0$, $p(x_0) = q(x_0) = 0$ e

$$y - F(x, y) = H(x, y)((y - y_0)^2 + p(x)(y - y_0) + q(x))$$

[8] $[x^n]y(x)$ é o coeficiente de x^n em $y(x)$.

ao redor de (x_0, y_0). Como $F_x(x_0, y_0) \neq 0$, nós também temos que $q_x(x_0) \neq 0$. O que significa que qualquer função analítica y que satisfaça $y(x) = F(x, y(x))$ em um subconjunto de uma vizinhança de x_0 com x_0 em sua fronteira é dada por

$$y(x) = y_0 - \frac{p(x)}{2} \pm \sqrt{\frac{p(x)^2}{4} - q(x)}.$$

Como $p(x_0) = 0$ e $q_x(x_0) \neq 0$, temos que

$$\frac{\partial}{\partial x}\left(\frac{p(x)^2}{4} - q(x)\right)_{x=x_0} \neq 0.$$

Portanto existe uma função analítica $K(x)$ tal que $K(x_0) \neq 0$ e

$$\frac{p(x)^2}{4} - q(x) = K(x)(x - x_0)$$

ao redor de x_0. Chegamos então em uma representação de $y = y(x)$ da forma

$$y(x) = g(x) - h(x)\sqrt{1 - \frac{x}{x_0}}, \tag{4}$$

onde $g(x)$ e $h(x)$ são analíticas ao redor de x_0 e satisfazem $g(x_0) = y_0$ e $h(x_0) < 0$. Para calcular $h(x_0)$, usaremos o teorema de Taylor:

$$0 = F(x, y(x)) = F_x(x_0, y_0)(x - x_0) + \frac{1}{2}F_{yy}(x_0, y_0)(y(x) - y_0)^2 + \ldots =$$

$$= F_x(x_0, y_0)(x - x_0) + \frac{1}{2}F_{yy}(x_0, y_0)h(x_0)^2(1 - x/x_0) + O(|x - x_0|^{3/2}).$$

Comparando os coeficientes de $(x - x_0)$ nós obtemos

$$h(x_0) - \sqrt{\frac{2x_0F_x(x_0, y_0)}{F_{yy}(x_0, y_0)}}.$$

Com o objetivo de aplicar o lema B.37, precisamos mostrar que $y(x)$ pode ser estendida analiticamente para uma região da forma Δ. A representação (4) nos fornece tal extensão em uma vizinhança de x_0. Agora, suponha que $|x_1| = x_0$ e $|arg(x_1)| \geq \delta$. Então, supondo que existe $n_0 \in \mathbb{N}$ para o qual para $y_n > 0$ para $n > n_0$, temos que $|y(x_1) < y(|x_1| = y(x_0)$ e portanto

$$|F_y(x_1, y(x_1))| \leq F_y(|x_1|, |y(x_1)|) < F_y(|x_1|, y(|x_1|)) = F_y(x_0, y_0) = 1.$$

Logo, $F_y(x_1, y(x_1)) \neq 1$ e o teorema da função implícita mostra que existe uma solução analítica $\widehat{y}(x)$ em uma vizinhança de x_1. Para $|x| < x_0$, essa solução se iguala à serie de potência $y(x)$ e, para $|x| \geq x_0$ ela fornece uma extensão analítica para a região Δ. Finalmente, podemos aplicar o lema B.37 para provar (3). ■

Apêndice C

Entropia de Shannon

Esse apêndice se propõe a estabelecer os conceitos básicos da entropia de Shannon com ênfase aos conceitos necessários para compreender o método de entropy compression. Ele está baseado, em sua maior parte, nos primeiros capítulos de [6] e [21].

Entropia é uma medida da imprevisibilidade de um pedaço de informação. Para se ter uma ideia intuitiva desse conceito, considere um lançamento de moeda. Quando a moeda é honesta, ou seja, quando a probabilidade de cara e coroa é a mesma, então a entropia desse evento (lançamento de uma moeda) assume seu valor máximo. Isso porque não temos nenhum indício de qual será o resultado do evento. Por outro lado, o lançamento de uma moeda que possui duas caras e nenhuma coroa tem entropia zero. Isso porque sabemos, de antemão, exatamente qual será o resultado do evento.

O desafio então é criar uma medida para a imprevisibilidade de eventos. Isso será feito exigindo que essa medida obedeça axiomas apropriados. Estabelecidos os axiomas, provaremos que apenas uma família de funções consegue satisfazer a todos eles.

C.1 Os Axiomas da Medida de Imprevisibilidade

Suponha que um experimento envolva a observação de uma variável aleatória discreta X com imagem $\{x_1, \ldots, x_M\}$ e probabilidades $p_1, \ldots p_M$, respectivamente. Suponha ainda que $p_i > 0$, para todo i. Na tentativa de associar uma incerteza (imprevisibilidade) à variável X, nós criaremos as funções h e H. A função h denotará a incerteza associada a um evento de probabilidade p (logo será definida no intervalo (0,1]). Ou seja, se o evento $\{X = x_i\}$ tem probabilidade p_i, então a incerteza removida (ou informação fornecida)

ao revelar que X amostrou o valor x_i é $h(p_i)$. Para cada M, definiremos H_M das M variáveis $p_1, \ldots, p_M$ como $H_M(p_1, \ldots, p_M) = \sum_{i=1}^{M} p_i h(p_i)$. Logo $H_M(p_1, \ldots, p_M)$ é o valor esperado de incerteza removida ao se revelar o valor que X assume. Em prol de uma notação menos carregada denotaremos $H_M(p_1, \ldots, p_M)$ por $H(p_1, \ldots, p_M)$ ou $H(X)$.

A seguir, estabeleceremos axiomas que traduzirão o que esperamos de uma medida de incerteza. Seja $f(M)$ o valor esperado de incerteza associado a M eventos equiprováveis. Ou seja, $f(M) = H(1/M \ldots, 1/M)$. Claramente, a incerteza é maior ao se escolher uma pessoa ao acaso em uma multidão do que ao se lançar uma moeda. Logo, nossa primeira exigência sobre a função $H(X)$ é

- $f(M) = H(1/M, \ldots, 1/M)$ é uma função estritamente crescente de M.

Agora, sejam X e Y duas variáveis aleatórias uniformes independentes com imagens $\{x_1, \ldots, x_M\}$ e $\{y_1, \ldots, y_L\}$ respectivamente. Logo o experimento envolvendo X e Y tem $M \cdot L$ resultados igualmente prováveis e a incerteza esperada desse experimento deve ser $f(M \cdot L)$. Se o valor de X é revelado, o valor esperado de incerteza de Y não deve mudar pois supusemos independência entre X e Y. Logo, o valor esperado de incerteza associado ao experimento envolvendo X e Y menos o valor esperado de incerteza removida ao revelar-se o valor que X assume deve ser igual ao valor esperado de incerteza de Y. Portanto a segunda exigência que fazemos é

- $f(ML) = f(M) + f(L)$, para $M, L \in \mathbb{N}$.

Considere novamente a variável aleatória X (não necessariamente uniforme) com imagem $\{x_1, \ldots, x_M\}$. Seja $A = \{x_1, \ldots, x_r\}$ e $B = \{x_{r+1}, \ldots, x_M\}$. Construiremos um experimento da seguinte maneira: primeiro selecionamos A ou B, escolhendo A com probabilidade $p_1 + \ldots + p_r$ e B com probabilidade $p_{r+1} + \ldots + p_M$. Se o grupo A for escolhido então, para $0 \leq i \leq r$, selecionamos x_i com probabilidade $p_i/(p_1 + \ldots + p_r)$ (a probabilidade condicional de x_i dado que o valor de X está em A). Da mesma forma, se o grupo B for escolhido então, para $r + 1 \leq i \leq M$, x_i é escolhido com probabilidade $p_i/(p_{r+1} + \ldots + p_M)$. A próxima figura representa esse experimento:

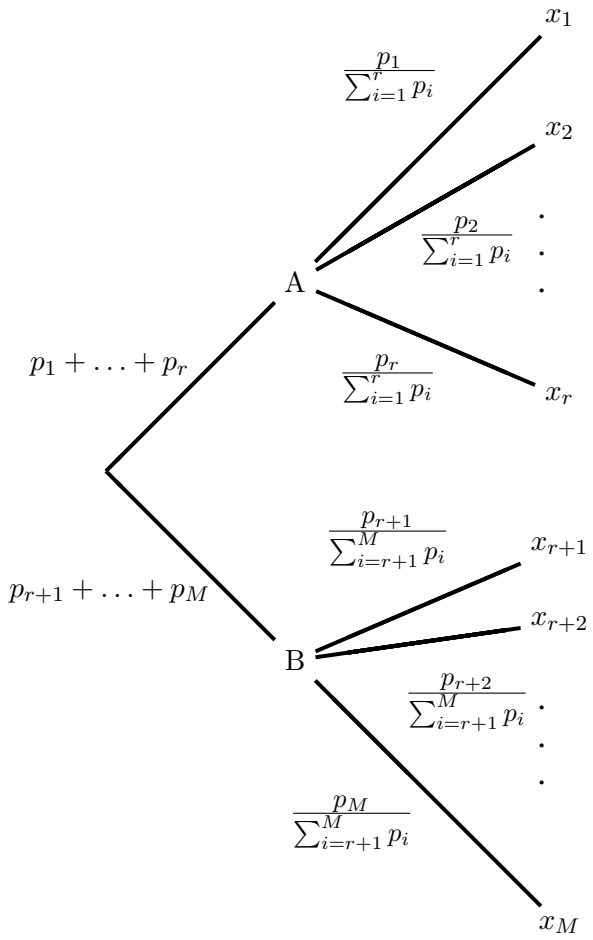

Figura 1: Representação do experimento.

O evento descrito é equivalente a X. Com efeito, seja Y a variável aleatória associada ao experimento. Seja $x_i \in A$. Logo

$$\mathbf{P}(Y = x_i) = \mathbf{P}(\text{“}A \quad \text{é} \quad escolhido \quad e \quad x_i \quad \text{é} \quad selecionado\text{”}) =$$

$$= \mathbf{P}(\text{“}A \quad \text{é} \quad escolhido\text{”})\mathbf{P}(\text{“}x_i \quad \text{é} \quad selecionado\text{”}|\text{“}A \quad \text{é} \quad escolhido\text{”}) =$$

$$= (\sum_{i=1}^{r} p_i)\frac{p_i}{\sum_{i=1}^{r} p_i} = p_i.$$

O mesmo raciocínio se aplica se $x_i \in B$. Logo X e Y têm a mesma distribuição. A incerteza associada a X é $H(p_1, \ldots, p_M)$. Se for revelado

qual dos grupos será escolhido no experimento, o valor esperado de incerteza que será removido é $H(p_1 + \ldots + p_r, p_{r+1} + \ldots p_M)$. Ao se escolher o grupo A, a incerteza restante é

$$H(\frac{p_1}{\sum_{i=1}^{r} p_i}, \ldots, \frac{p_r}{\sum_{i=1}^{r} p_i}).$$

Ao se escolher o grupo B, a incerteza restante é

$$H(\frac{p_{r+1}}{\sum_{i=r+1}^{M} p_i}, \ldots, \frac{p_M}{\sum_{i=r+1}^{M} p_i}).$$

Logo o valor esperado de incerteza removida depois que o grupo for escolhido é

$$(p_1 + \ldots + p_r)H(\frac{p_1}{\sum_{i=1}^{r} p_i}, \ldots, \frac{p_r}{\sum_{i=1}^{r} p_i}) +$$

$$+(p_{r+1} + \ldots + p_M)H(\frac{p_{r+1}}{\sum_{i=r+1}^{M} p_i}, \ldots, \frac{p_M}{\sum_{i=r+1}^{M} p_i}).$$

É desejável que o valor esperado de incerteza relacionado ao experimento menos o valor esperado de incerteza removida ao se revelar o grupo escolhido seja igual o valor esperado de incerteza restante após o grupo ser selecionado. Logo o terceiro axioma é

- $H(p_1, \ldots, p_M) = H(p_1 + \ldots + p_r, p_{r+1} + \ldots + p_M) +$

$$+(p_1 + \ldots + p_r)H(\frac{p_1}{\sum_{i=1}^{r} p_i}, \ldots, \frac{p_r}{\sum_{i=1}^{r} p_i}) +$$

$$+(p_{r+1} + \ldots + p_M)H(\frac{p_{r+1}}{\sum_{i=r+1}^{M} p_i}, \ldots, \frac{p_M}{\sum_{i=r+1}^{M} p_i}).$$

Finalmente, nós exigiremos, por conveniência matemática, que $H(p, 1 - p)$ seja uma função contínua em p. Intuitivamente isso significa uma pequena variação nas probabilidades dos valores de X representam uma pequena variação da incerteza de X.

Teorema C.1. *A única função que satisfaz os quatro axiomas é*

$$H(p_1, \ldots, p_M) = -C \sum_{i=1}^{M} p_i log(p_i),$$

onde C é um número positivo arbitrário e a base do logaritmo é qualquer número maior que 1.

Prova: Não é difícil verificar que a função acima satisfaz os quatro axiomas. Provaremos que se uma função satisfaz os quatro axiomas citados então essa função pertence à família de funções citada acima. Nessa prova, os logaritmos estão em uma mesma base fixa. Isso é feito sem perda de generalidade pois mudar a base do logaritmo é equivalente a mudar a constante C ($\log_a x = log_a b \cdot log_b x$). Por conveniência, dividiremos a prova em algumas partes:

a) É fácil provar, usando o axioma 2 e indução em k, que $f(M^k) = kf(M)$, para $M, k \in \mathbb{N}^*$.

b) $f(M) = C \log M$, para $M \in \mathbb{N}^*$ e $C \in \mathbb{R}^+$. Com efeito, seja $M = 1$. Pelo segundo axioma , $f(1) = f(1 \cdot 1) = f(1) + f(1)$. Logo $f(1) = 0$. O que está de acordo com a noção intuitiva de que um evento com probabilidade 1 não tem incerteza associada. Agora, sejam M um inteiro positivo maior que 1 e $r > 0$ um número natural fixo. Existe algum número k para o qual $M^k \leq 2^r < M^{k+1}$. Logo, pelo primeiro axioma, $f(M^k) \leq f(2^r) < f(M^{k+1})$ e, por a), $kf(M) \leq rf(2) < (k+1)f(M)$ ou $k/r \leq f(2)/f(M) < (k+1)/r$. O logaritmo é uma função estritamente crescente (pois sua base é maior que 1), logo $\log M^k \leq \log 2^r < \log M^{k+1}$ ou $k/r \leq \log 2/\log M < (k+1)/r$. Como ambos $f(2)/f(M)$ e $\log 2/\log M$ estão entre k/r e $(k+1)/r$, temos que

$$|\frac{\log 2}{\log M} - \frac{f(2)}{f(M)}| < \frac{1}{r}.$$

Como r foi escolhido de maneira arbitrária temos que

$$\frac{\log 2}{\log M} = \frac{f(2)}{f(M)}$$

ou $f(M) = C \log M$, onde $C = f(2)/\log 2$. Note que C é positivo já que $f(1) = 0$ e f é estritamente crescente.

c) $H(p, 1-p) = -C(p \log p + (1-p)\log(1-p))$, se $p \in \mathbb{Q}$. Com efeito, seja $p = r/s$, com $r, s \in \mathbb{N}^*$. Pelo terceiro axioma temos que

$$f(s) = H(\frac{r}{s}, \frac{s-r}{s}) + \frac{r}{s}f(r) + \frac{s-r}{s}f(s-r)$$

Por b, temos

$$C \log s = H(p, 1-p) + Cp \log r + C(1-p)\log(s-r).$$

Logo

$$H(p, 1-p) = -C(p \log r - \log s + (1-p)\log(s-r)) =$$

$$= -C(p\log r - p\log s + p\log s - \log s + (1-p)\log(s-r)) =$$

$$= -C(p\log\frac{r}{s} + (1-p)\log\frac{s-r}{s}) = -C(p\log p + (1-p)\log(1-p)).$$

d) $H(p, 1-p) = -C(p\log p + (1-p)\log(1-p))$, para todo p. Esse resultado segue de c) e do axioma 4. Com efeito, seja $p \in [0,1]$. Por continuidade

$$H(p, 1-p) = \lim_{p' \to p} H(p', 1-p').$$

Em particular, podemos nos aproximar de p por uma sequência de números racionais. Logo

$$\lim_{p' \to p} H(p', 1-p') = \lim_{p' \to p} -C(p'\log p' + (1-p')\log(1-p')) =$$

$$= -C(p\log p + (1-p)\log(1-p)).$$

e) $H(p_1, \ldots, p_M) = -C\sum_{i=1}^{M} p_i \log p_i$, para $M \in \mathbb{N}^*$. Com efeito, por indução em M: O resultado já foi provado para $M = 1$ e $M = 2$ (lembre que $\sum_{i=1}^{M} p_i = 1$). Seja $M > 2$. Pelo terceiro axioma temos

$$H(p_1, \ldots, p_M) = H(p_1 + \ldots + p_{M-1}, p_M) +$$

$$+(p_1, \ldots, p_{M-1}) H(\frac{p_1}{\sum_{i=1}^{M-1} p_i}, \ldots, \frac{p_{M-1}}{\sum_{i=1}^{M-1} p_i}) + p_M H(1).$$

Assumindo que a formula vale para inteiros menores que M temos:

$$H(p_1, \ldots, p_M) = -C((p_1 + \ldots + p_{M-1})\log(p_1 + \ldots + p_{M-1}) + p_M \log p_M) -$$

$$-C(p_1 + \ldots + p_{M-1})(\frac{p_1}{\sum_{i=1}^{M-1} p_i}\log\frac{p_1}{\sum_{i=1}^{M-1} p_i} + \ldots + \frac{p_{M-1}}{\sum_{i=1}^{M-1} p_i}\log\frac{p_{M-1}}{\sum_{i=1}^{M-1} p_i}) =$$

$$= -C((\sum_{i=1}^{M-1} p_i \log(\sum_{i=1}^{M-1} p_i) + p_M \log p_M) - C(\sum_{i=1}^{M-1} p_i \log p_i -$$

$$-(\sum_{i=1}^{M-1} p_i)\log\sum_{i=1}^{M-1} p_i) = -C\sum_{i=1}^{M} p_i \log p_i.$$

■

Portanto $H(p_1, \ldots, p_M) = -C \sum_{i=1}^{M} p_i log(p_i)$ para algum C. Vimos que diferentes valores de C equivalem a diferentes bases do logaritmo. Assim, diferentes unidades de entropia são definidas para cada base. Shannon ou bit para base 2, nat para base e e hartley para base 10 são alguns exemplos.

Voltemos ao exemplo do lançamento de uma moeda. Seja $H(X) = -\sum_{i=1}^{2} p_i \log_2 p_i$. Considere uma moeda com valores conhecidos para as probabilidades de cara e coroa. A situação de máxima entropia, nesse evento, é quando essas probabilidades são iguais (esse resultado vale para o caso geral, conforme provado no teorema C.11). Nesse caso temos que $H(X) = 1$. Se as probabilidades não forem iguais, então temos mais condições de prever o resultado logo a entropia deve ser menor (até o caso extremo onde uma das probabilidades é igual a 1). A próxima figura mostra a evolução da entropia e da função h com relação a probabilidade de sair cara. Também constam as parcelas $p(-\log_2(p))$ e $(1-p)(-\log_2(1-p))$ que compõe a entropia.

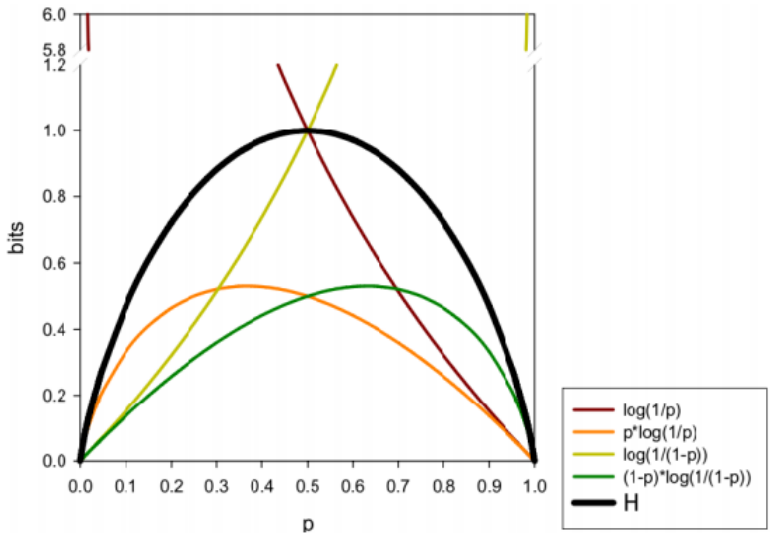

Figura 2: Evolução da entropia.

Observação C.2. A convenção $0log0 = 0$ é usual na teoria da informação. Essa convenção é facilmente justificada por continuidade, uma vez que $xlog(x)$ tende a zero quando x tende a zero (ver Figura 2). Assim, a adição de termos com probabilidade zero não muda a entropia.

Observação C.3. Os axiomas para a medida de incerteza aqui apresentados são os axiomas de Shannon [60]. Fadiev, em [30], mostrou uma versão mais fraca desses axiomas determinando a mesma função de entropia.

C.2 Entropia Relativa e Propriedades Básicas

Tendo em mente que o objetivo principal desse apêndice é dar ao leitor a base necessária para entender o método de entropy compression, vamos definir entropia relativa sem definir outros conceitos mais básico e/ou relacionados como entropia conjunta, entropia condicional e informação mútua. O leitor interessado pode consultar [6] e [21].

Definição C.4. A entropia relativa (ou distância de Kullback–Leibler) entre duas distribuições de probabilidade $p(x)$ e $q(x)$ é dada por

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} = E_p[\log\frac{p(X)}{q(X)}].$$

Na definição acima, usamos as convenções $0\log\frac{0}{0} = 0$, $0\log\frac{0}{q} = 0$ e $p\log\frac{p}{0} = \infty$. Logo, se existir $x \in \mathcal{X}$ para o qual $p(x) > 0$ e $q(x) = 0$, então $D(p||q) = \infty$. A seguir, mostraremos que a entropia relativa é não-negativa e é zero se, e somente se, $p = q$. Por essas propriedades , apesar de não ser simétrica nem respeitar a desigualdade triangular, a entropia relativa é geralmente entendida como a distância entre duas distribuições de probabilidade.

Exemplo C.5. Sejam $\mathcal{X} = \{0, 1\}$ e p e q distribuições de probabilidade em $\mathcal{X}$ com

$$p(0) = 1 - r, \quad p(1) = r, \quad q(0) = 1 - s \quad e \quad q(1) = s.$$

Então

$$D(p||q) = (1-r)\log\frac{1-r}{1-s} + r\log\frac{r}{s}$$

e

$$D(q||p) = (1-s)\log\frac{1-s}{1-r} + s\log\frac{s}{r}.$$

Se $r = s$, então $D(p||q) = D(q||p) = 0$. Se $r = \frac{1}{2}$, $s = \frac{1}{4}$, então

$$D(p||q) = \frac{1}{2}\log\frac{1/2}{3/4} + \frac{1}{2}\log\frac{1/2}{1/4} = 1 - \frac{1}{2}\log 3 = 0.2075 \ \textit{bit.}$$

Enquanto

$$D(q||p) = \frac{3}{4}\log\frac{3/4}{1/2} + \frac{1}{4}\log\frac{1/4}{1/2} = \frac{3}{4}\log 3 - 1 = 0.1887 \ \textit{bit}.$$

□

Agora provaremos algumas propriedades básicas de entropia de entropia relativa. De especial importância para nós, provaremos que a entropia relativa é não-negativa. Esse fato será usado mais adiante para provar o teorema C.26 – o resultado de maior importância para nós nessa seção.

Primeiro, observamos que a entropia é sempre positiva, i.e., $H(X) \geq 0$. Isso segue diretamente do fato que $-p_i \log p_i \geq 0$, para todo i.

Para provar as próximas propriedades, precisaremos das seguintes ferramentas:

Definição C.6. Uma função $f(x)$ é convexa num intervalo (a, b) se, para todo $x_1, x_2 \in (a, b)$ e $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2).$$

Uma função f é estritamente convexa se acima temos igualdade somente se $\lambda = 0$ ou $\lambda = 1$.

Definição C.7. Uma função é (estritamente) côncava se $-f$ é (estritamente) convexa.

Intuitivamente, uma função é convexa/côncava se sempre está abaixo/acima de qualquer secante. x^2, $|x|$, e^x, $x \log x$ são exemplos de funções convexas (para $x \geq 0$). $\log x$ e $\sqrt{x}$ são exemplos de funções côncavas (para $x \geq 0$). Funções lineares são côncavas e convexas.

Teorema C.8. *Seja f uma função com segunda derivada não-negativa/positiva num intervalo (a, b). Então f é convexa/estritamente convexa nesse intervalo.*

Prova: Seja $x_0 \in (a, b)$. Considere a expansão de Taylor de f ao redor de x_0:

$$f(x) = f(x_0) + f^{'}(x_0)(x - x_0) + \frac{f^{''}(x^*)}{2}(x - x_0)^2,$$

Onde $x^* \in B_{|x-x_0|}(x_0)$. Por Hipótese, $f^{''}(x^*) \geq 0$. Logo, o último termo é não negativo para todo $x \in (a, b)$.

Sejam $x_1, x_2 \in (a, b)$ e $\lambda \leq 1$ tais que $x_0 = \lambda x_1 + (1 - \lambda)x_2$. Logo

$$f(x_1) \geq f(x_0) + f^{'}(x_0)(1 - \lambda)(x_1 - x_2).$$

e

$$f(x_2) \geq f(x_0) + f^{'}(x_0)\lambda(x_2 - x_1).$$

Multiplicando a primeira desigualdade por λ, a segunda por $1 - \lambda$ e somando os resultados, obtemos

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

A prova para convexidade estrita segue a mesma ideia. ■

Esse teorema nos permite verificar facilmente a convexidade estrita de x^2, e^x e $x \log x$ (para $x \geq 0$) e a concavidade estrita de $\log x$ e $\sqrt{x}$ (para $x \geq 0$).

O próximo resultado é de suma importância para derivar as propriedades básicas da teoria da informação. Abaixo, $E[X]$ denota o valor esperado da v.a. X.

Teorema C.9 (Desigualdade de Jensen - Caso Discreto)**.** *Seja f uma função convexa e X uma variável aleatória discreta. Então*

$$E[f(X)] \geq f(E[X]).$$

Além disso, se f é estritamente convexa, a igualdade na expressão acima implica que $X = E[X]$ com probabilidade 1 (i.e., X é constante).

Prova: A prova será por indução na cardinalidade de $\mathcal{X}$. Para $|\mathcal{X}| = 2$, a desigualdade toma a forma

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2)$$

que segue diretamente da definição de função convexa. Supondo que a desigualdade vale para $|\mathcal{X}| = k - 1$ concluiremos que também vale para $|\mathcal{X}| = k$. Escrevendo $p_i = p_i/(1 - p_k)$ para $i \in \{1, 2, \ldots, k - 1\}$ temos:

$$E[f(X)] = \sum_{i=1}^{k} p_i f(x_i) = p_k f(x_k) + (1 - p_k) \sum_{i=1}^{k-1} p_i^{'} f(x_i) \geq$$

$$\geq p_k f(x_k) + (1 - p_k) f(\sum_{i=1}^{k-1} p_i^{'} x_i) \geq f(p_k x_k + (1 - p_k) \sum_{i=1}^{k-1} p_i^{'} x_i) = f(\sum_{i=1}^{k} p_i x_i),$$

onde a primeira desigualdade segue da hipótese de indução e a segunda da definição de convexidade. ■

O teorema acima pode ser estendido para distribuições contínuas por argumentos de continuidade. De posse dessa ferramenta, enfim somos capazes de demonstrar o que buscávamos:

Teorema C.10. *Sejam $p(x)$, $q(x)$, $x \in \mathcal{X}$, duas distribuições de probabilidade. Então*

$$D(p||q) \geq 0,$$

com igualdade se, e somente se, $p(x) = q(x)$ para todo $x \in \mathcal{X}$.

Prova: Seja $A = \{x : p(x) > 0\}$ o suporte de $p(x)$. Então

$$-D(p||q) = -\sum_{x \in A} p(x) \log \frac{p(x)}{q(x)} = \sum_{x \in A} p(x) \log \frac{q(x)}{p(x)} \leq$$

$$\leq \log \sum_{x \in A} p(x) \frac{q(x)}{p(x)} = \log \sum_{x \in A} q(x) \leq \log \sum_{x \in X} q(x) = \log 1 = 0,$$

onde a primeira desigualdade segue da desigualdade de Jensen. Como $\log t$ é uma função estritamente côncava, temos igualdade na primeira desigualdade se, e somente se, $q(x)/p(x)$ é constante (i.e., $q(x) = cp(x)$ para todo $x \in \mathcal{X}$). Então $\sum_{x \in A} q(x) = c \sum_{x \in A} p(x)$. Temos igualdade na segunda desigualdade se, e somente se, $\sum_{x \in A} q(x) = \sum_{x \in \mathcal{X}} q(x) = 1$, o que implica que $c = 1$. Logo, $D(p||q) = 0$ se, e somente se, $p(x) = q(x)$ para todo $x \in \mathcal{X}$. ■

A seguir, provaremos a intuitiva propriedade de entropia que diz que, fixado $\mathcal{X}$, a distribuição de probabilidade que resulta na maior entropia é a uniforme. O conceito de entropia relativa nos fornece uma prova simples de tal fato. Para uma prova alternativa veja o teorema 1.4.2 de [6].

Teorema C.11. *$H(X) \leq log|\mathcal{X}|$, com igualdade se, e somente se, X tem distribuição uniforme.*

Prova: Sejam $u(x) = \frac{1}{|\mathcal{X}|}$ a distribuição uniforme sobre $\mathcal{X}$ e $p(x)$ a distribuição de probabilidade de X. Então

$$D(p||u) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{u(x)} = \log |X| - H(X).$$

Logo, pela não-negatividade da entropia relativa, temos que

$$0 \leq D(p||u) = \log |X| - H(X) \therefore H(X) \leq \log |X|.$$

■

C.3 Compressão de Dados

Como o nome indica, comprimir dados é o ato de representar um pedaço de informação usando menos espaço (bits). Isso pode ser feito com ou sem perdas de informação. Aqui estudaremos a compressão sem perdas.

Esse objetivo pode ser alcançado designando símbolos curtos para frações mais frequentes do pedaço de informação (e, consequentemente, símbolos longos para frações menos frequentes). Por exemplo, no código Morse, o símbolo mais frequente é representado por um único ponto. Nosso objetivo nessa seção é descobrir até quanto e como conseguimos comprimir um pedaço de informação.

C.3.1 Códigos

Definição C.12. Um código C para uma variável aleatória X é uma função que leva $\mathcal{X}$, a imagem de X, em D^*, o conjunto de todas sequências finitas de símbolos de um alfabeto D. Para cada $x \in \mathcal{X}$, denota-se por $l(x)$ o comprimento (em número de símbolos de D) da palavra $C(x)$.

Definição C.13. O comprimento esperado $L(C)$ de um código C para uma variável aleatória X com distribuição de probabilidade $p(x)$ é dado por $L(C) = \sum_{x\in\mathcal{X}} p(x)l(x)$.

Sem perdas de generalidade, podemos assumir que o alfabeto D-ário (o alfabeto com D símbolos) é dado por $D = \{0, 1, \ldots, D-1\}$. Seguem alguns exemplos de códigos:

Exemplo C.14. Seja X uma variável aleatória com a seguinte distribuição de probabilidade e código:

$$\mathbf{P}(X = 1) = \frac{1}{2}, \quad C(1) = 0$$

$$\mathbf{P}(X = 2) = \frac{1}{4}, \quad C(2) = 10$$

$$\mathbf{P}(X = 3) = \frac{1}{8}, \quad C(3) = 110$$

$$\mathbf{P}(X = 4) = \frac{1}{8}, \quad C(4) = 111.$$

A entropia $H(X)$ é igual a 1.75 bits assim como o comprimento esperado $L(C)$. Nota-se também que qualquer sequência de bits pode ser univocamente decodificada. Por exemplo, a sequência 0110111100110 é decodificada como 134213. □

Exemplo C.15. Considere agora a seguinte distribuição de probabilidade e código:

$$\begin{aligned}\mathbf{P}(X=1) &= \frac{1}{3}, \quad C(1)=0\\ \mathbf{P}(X=2) &= \frac{1}{3}, \quad C(2)=10\\ \mathbf{P}(X=3) &= \frac{1}{3}, \quad C(3)=11.\end{aligned}$$

De novo temos um código univocamente decodificável. A diferença é que nesse exemplo temos $H(X) = \log 3 (\approx 1.58)$ bits e $L(X) = 1.66$ bits. □

Exemplo C.16 (Código Morse)**.** O Código Morse é um código razoavelmente eficiente para o alfabeto inglês usando um alfabeto de quatro símbolos: um ponto, um traço, um espaço de letra e um espaço de palavra. Sequências pequenas representam letras frequentes (por exemplo, um ponto representa a letra e) ao passo que sequências grandes representas letras pouco frequentes (por exemplo, Q é representado por "traço,traço,ponto,traço"). Esse não é o código ótimo do alfabeto em quatro símbolos. Muitas palavras não são usadas pois uma palavra representando uma letra não contém espaços a não ser por um espaço de letra ao final e nenhum espaço segue outro espaço. É um problema interessante calcular o número de sequências que podem ser construídas obedecendo essas restrições. Tal problema foi resolvido por Shannon em [60]. O problema também está relacionado com estabelecer códigos para gravação magnética onde longas sequências de 0's são proibidas como em [2] e [47]. □

Os códigos são classificados de acordo com propriedades que obedecem quanto a decodificação. Vamos a elas:

Definição C.17. Um código é não-singular se todo elemento de $\mathcal{X}$ está associado a uma sequência diferente de D^*. Ou seja:

$$x \neq x' \Rightarrow C(x) \neq C(x').$$

Não-singularidade implica em não ambiguidade na tradução de uma palavra dada. Mas, em geral, uma mensagem contém uma sequência de valores de $\mathcal{X}$. Nesse caso, nós poderíamos garantir que a mensagem será decodificada univocamente acrescentando um símbolo (exclusivo para esse fim) entre duas palavras. Claramente não é uma abordagem muito eficiente. Esse problema motiva as seguintes definições:

Definição C.18. A extensão C^* de um código C é a função que leva sequências finitas de $\mathcal{X}$ em sequências finitas de D, definida por

$$C(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n),$$

onde $C(x_1)C(x_2)\ldots C(x_n)$ é a concatenação das palavras correspondentes.

Definição C.19. Um código é univocamente decodificável se sua extensão é não-singular (invertível).

Em outras palavras, cada sequência de palavras pode ser gerada por apenas uma sequência em $\mathcal{X}$. Note, porém, que é possível que se tenha que ler a sequência inteira para descobrir quem a gerou.

Definição C.20. Um código é dito ser instantâneo se nenhuma palavra é um prefixo para outra palavra.

Um código instantâneo pode ser decodificado sem a leitura de palavras futuras pois o fim de cada palavra é imediatamente reconhecido. Por exemplo, a sequência binária 01011111010 produzida pelo código do exemplo C.14 é lida como 0,10,111,110,10. Abaixo, a figura 3 ilustra a relação entre os diferentes códigos aqui mencionados. A tabela C.1 mostra um exemplo de códigos distintos usando o mesmo alfabeto para o mesmo número de eventos. Para o código não-singular a palavra 010 tem três fontes possíveis: 2, 14 e 31 (logo não se trata de um código univocamente decodificável). O código univocamente decodificável não é instantâneo (pois $C(3)$ é um prefixo de $C(4)$).

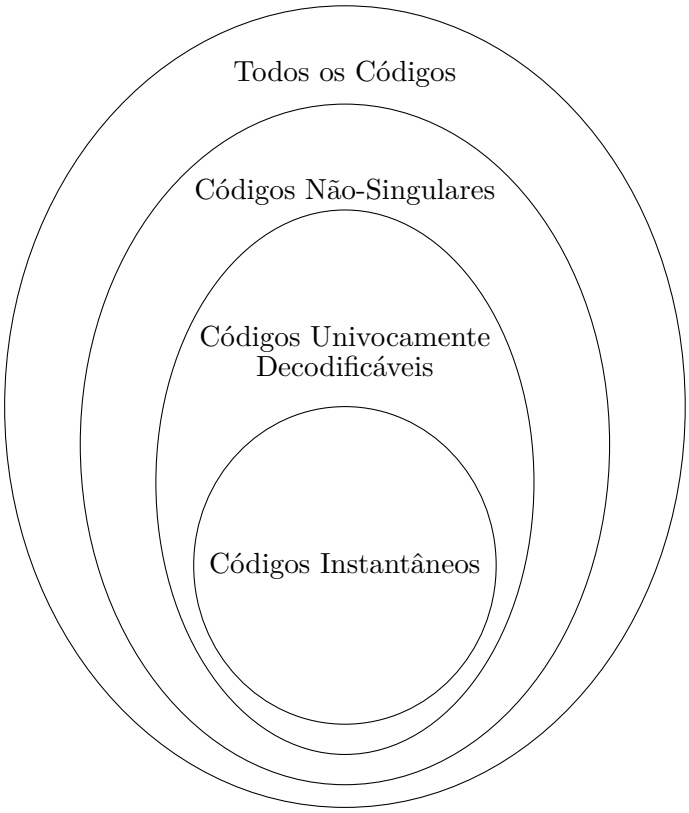

Figura 3: Relação de contenção entre códigos.

X	Singular	Não-Singular mas não Univoc. Decod.	Univoc. Decod. mas não Instantâneo	Instantâneo
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

Tabela C.1: Exemplos de códigos.

Para verificar que se trata de um código univocamente decodificável, considere uma sequência de letras: se os primeiros dois bits forem 00 ou 10 eles podem ser decodificados imediatamente. Se os primeiros bits forem 11 é preciso ler o próximo bit. Se for lido 1, lemos $C(3)$. Se a sequência de 0's imediatamente após a dupla 11 for ímpar então lemos $C(4)$. Finalmente, se a sequência de 0's for par, lemos $C(3)$. Esse argumento pode ser repetido

para ler cada palavra dessa sequência. Sardinas e Patterson em [58] criaram um teste finito para univoca-decodificabilidade. O último código da tabela C.1 é claramente instantâneo.

C.3.2 Desigualdade de Kraft

Como é de se esperar, nosso objetivo é, dado X, construir um código instantâneo com o menor comprimento esperado possível. Na próximo seção indicaremos como fazer isso. Antes precisamos de alguns alicerces:

Teorema C.21 (Desigualdade de Kraft)**.** *Sejam X uma v.a. e $C(X)$ um código instantâneo usando um alfabeto D-ário. Então $l_1, l_2, \ldots, l_m$, os comprimentos das (finitas) palavras de $C(X)$, devem satisfazer*

$$\sum_i D^{-l_i} \leq 1.$$

Reciprocamente, dado um conjunto de comprimentos satisfazendo a desigualdade acima então existe um código cujas palavras têm tais comprimentos.

Prova: Considere a árvore D-ária onde cada vértice tem D filhos. Assim seus galhos representam as palavras de $C(X)$. Por exemplo, os D filhos da raiz representam as D possibilidades para a primeira letra de uma palavra e assim por diante. Logo, o conjunto dos caminhos finitos a partir da raiz é uma representação do conjunto D^*. Um exemplo usando a árvore binária é mostrado na figura 4 abaixo. A condição do código ser instantâneo implica que nenhuma palavra é ancestral de outra na árvore. Seja l_{max} o comprimento da maior palavra de $C(X)$. Considere todos os $D^{l_{max}}$ vértices de profundidade l_{max}. Eles são ou parte de uma palavra ou descendentes de uma palavra ou nenhuma das duas. Uma palavra com tamanho l_i tem $D^{l_{max}-l_i}$ descendentes no nível l_{max}. Os conjuntos de descendentes de palavras distintas de profundidade l_{max} são disjuntos. Logo, somando em todas as palavras nós temos

$$\sum_i D^{l_{max}-l_i} \leq D^{l_{max}}.$$

Ou, como desejávamos,

$$\sum_i D^{-l_i} \leq 1.$$

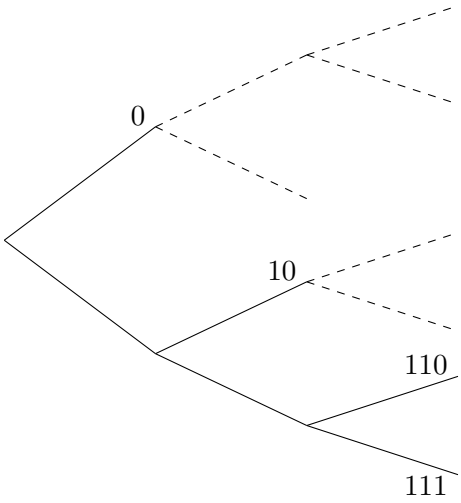

Figura 4: Código binário respeitando a desigualdade de Kraft.

Reciprocamente, seja $L = \{l_1, l_2, \ldots, l_m\}$ um conjunto que satisfaz a desigualdade de Kraft. Defina o primeiro (seguindo alguma ordem arbitrária) galho de tamanho l_1 como a primeira palavra de $C(X)$ e remova seus descendentes da árvore. Defina o primeiro galho remanescente de tamanho l_2 como a segunda palavra e assim por diante. Claramente, ao final desse processo, teremos um código instantâneo cujas palavras tem comprimento $l_1, l_2, \ldots$ e l_m. ■

O próximo teorema é uma generalização da desigualdade de Kraft.

Teorema C.22. *Para qualquer conjunto contável de palavras de um código instantâneo usando um alfabeto D-áreo, seus comprimentos devem satisfazer a seguinte desigualdade:*

$$\sum_{i}^{\infty} D^{-l_i} \leq 1.$$

Reciprocamente, dados $l_1, l_2, \ldots$ satisfazendo a desigualdade acima, existe um código instantâneo cujas palavras tem esses comprimentos.

Prova: Sejam $y^i = y_1 y_2 \ldots y_{l_i}$ a i-ésima palavra de $C(X)$ e $\alpha(y^i) = 0.y_1 y_2 \ldots y_{l_i}$ o número real dado pela expansão D-ária

$$0.y_1 y_2 \ldots y_{l_i} = \sum_{j=1}^{l_i} y_j D^{-j}.$$

O conjunto I^i de todas as expansões D-áreas de palavras que começam com $y_1y_2\dots y_{l_i}$ é um subconjunto de $[0.y_1y_2\dots y_{l_i}, 0.y_1y_2\dots y_{l_i}+\frac{1}{D^{l_i}})$ (para entender melhor essa construção veja os dois exemplos subsequentes à prova). Com efeito, a palavra que começa por $y_1y_2\dots y_{l_i}$ e tem a maior expansão D-ária (em módulo) é dada por $y_1y_2\dots y_{l_i}(D-1)(D-1)\dots$. Tal palavra obedece

$$0.y_1y_2\dots y_{l_i}(D-1)\dots - 0.y_1y_2\dots y_{l_i} = \sum_{j=1}^{\infty} y_jD^{-j} - \sum_{j=1}^{l_i} y_jD^{-j} =$$

$$= \sum_{j=li+1}^{\infty} (D-1)D^{-j} = \frac{(D-1)D^{-(l_i+1)}}{1-\frac{1}{D}} = \frac{1}{D^{l_i}}.$$

Na verdade esses conjuntos são iguais. Com efeito, o conjunto de todas as expansões D-árias de palavras que começam com $y_1y_2\dots y_{l_i}$ é o conjunto de todos os número na base D entre $0.y_1y_2\dots y_{l_i}$ e $0.y_1y_2\dots y_{l_i}+\frac{1}{D^{l_i}}$.

Como o código é instantâneo, para palavras distintas os intervalos aos quais sua expansões pertencem são disjuntos. Como tais intervalos estão contidos em $[0,1]$, temos que

$$1 = |[0,1]| \geq |\bigcup_i I^i| = \sum_i |I^i| = \sum_i D^{-l_i}.$$

Assim como no caso finito, podemos construir um código cujas palavras tem comprimentos $l_1, l_2, \dots$ satisfazendo a desigualdade de Kraft. Primeiro, ordene os comprimentos de forma que $l_1 \leq l_2, \leq \dots$. O próximo passo é associar os intervalos às palavras em sequência do mais próximo ao zero até o mais próximo ao 1. Por exemplo, se desejamos construir um código binário com $l_1 = 1, l_2 = 2, \dots$ nós associamos o intervalo $[0, 1/2), [1/2, 1/4), \dots$ às palavras correspondentes $0, 10, \dots$. ■

Os seguintes exemplos foram retirados de [32].

Exemplo C.23. Considere o caso de um código 10-ário (ou decimal). Seja $y^i = 2738$. Assim, $\alpha(y^i) = 0.2738$ (na habitual escrita em base 10) e $I^i = [0.2738, 0.2739[$, o qual é o intervalo de todos os números reais cuja escrita decimal começa por 0.2738; por exemplo $0.273845 \in [0.2738, 0.2739[$. □

Exemplo C.24. No caso de um código binário, seja $y^i = 100101$. Nesse caso, $\alpha(y^i) = 0.100101$ (em base 2, ou seja, traduzindo para base 10, $\alpha(y^i) = 1/2 + 1/16 + 1/64 = 0.5781$). O intervalo correspondente é $I^i = [0.100101, 0.10011[$ (pois, em base 2, 0.100101 + 0.000001 = 0.10011), o

qual contém todos os números cuja escrita em base 2 começa por 0.100101; por exemplo 0.100101101 ∈ [0.100101, 0.10011[. □

Observação C.25. Os comprimentos das palavras em um código univocamente decodificável também obedecem a desigualdade de Kraft. Para uma prova desse fato veja [21]. □

C.3.3 Códigos Ótimos

Como mencionado, essa seção se dedicará a, dado X e D, achar o código instantâneo de menor comprimento esperado. Da seção anterior, sabemos que isso é equivalente a achar os comprimentos $l_1, l_2, \ldots, l_m$ que satisfazem a desigualdade de Kraft e que minimizam $L = \sum p_i l_i$. Esse é um típico problema de otimização: minimize L $= \sum_i p_i l_i$ sujeito a $l_1, l_2, \ldots, l_m \in \mathbb{N}^*$ e $\sum_i D^{-l_i} \leq 1$. Nós descartaremos a primeira restrição e assumiremos igualdade na segunda. Assim, podemos escrever o problema usando multiplicadores de Lagrange como a minimização de

$$J = \sum_i p_i l_i + \lambda(\sum_i D^{-l_i}).$$

Diferenciando com respeito a l_i obtemos

$$\frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \log_e D.$$

Calculando as derivadas em 0, obtemos

$$D^{-l_i} = p_i \lambda \log_e D.$$

Substituindo esse resultado na restrição descobrimos que $\lambda = 1/\log_e D$, logo $p_i = D^{-l_i}$ resultando nos comprimentos ótimos

$$l_i^* = -\log_D p_i.$$

Essa escolha não inteira para os comprimentos resulta no comprimento esperado

$$L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H_D(X).$$

Como os comprimentos devem ser inteiros, nem sempre seremos capazes de obedecer as igualdades acima. A ideia é procurar por valores inteiros próximos aos otimais. Em vez de demonstrar por cálculo que $l_i^* = -\log_D p_i$ é um mínimo global, verificaremos a otimalidade na prova do seguinte teorema:

Teorema C.26. *O comprimento esperado L de um código instantâneo D-ário para uma v.a. X é maior ou igual à entropia $H_D(X)$. Isto é*

$$L \geq H_D(X),$$

com igualdade se, e somente se, $D^{-l_i} = p_i$.

Prova: Podemos escrever a diferença entre o comprimento esperado e a entropia como

$$L - H_D(X) = \sum_i p_i l_i - \sum_i p_i \log_D \frac{1}{p_i} = -\sum_i p_i \log_D D^{-l_i} + \sum_i p_i \log_D p_i.$$

Sendo $r_i = D^{-l_i} / \sum_j D^{-l_j}$ e $c = \sum_i D^{-l_i}$, obtemos

$$L - H = \sum_i p_i \log_D \frac{p_i}{r_i} - \log_D c = D(p||r) + \log_D \frac{1}{c} \geq 0,$$

pela não negatividade da entropia relativa (teorema C.10) e pela desigualdade de Kraft. Logo $L \geq H$ com igualdade se, e somente se $p_i = D^{-l_i}$ (i.e., se e somente se $-\log_D p_i$ é inteiro para todo i). ■

Definição C.27. Uma distribuição de probabilidade é dita ser D-ádica se cada probabilidade é igual a D^{-n} para algum n.

Logo, temos igualdade no teorema anterior se, e somente se, a distribuição de X é D-ádica. A prova do teorema C.26 também indica um procedimento para achar um código ótimo: achar a distribuição D-ádica que mais se aproxima (na métrica da entropia relativa) da distribuição de X. Essa distribuição nos fornecerá os comprimentos das palavras do código. Já provamos, de forma construtiva, que, de posse de comprimentos satisfazendo a desigualdade de Kraft, podemos definir um código instantâneo cujas palavras têm os comprimentos dados. Assim temos um código ótimo para X. Ainda há um problema: a tarefa de achar a distribuição D-ádica mais próxima da distribuição de X não é trivial. Há rotinas bem sucedidas para esse fim. O Código de Shannon-Fano mostra, além disso, que pode-se construir um código com comprimento médio tão próximo (pela direita) quanto se queira da entropia de X. O código de Huffman prova-se otimal, i.é., nenhum outro código instantâneo tem comprimento esperado menor que o obtido pelo código de Huffman. Para aprofundar no que acabamos de expor, o leitor pode consultar [21].

Referências Bibliográficas

[1] D. Achlioptas e F. Iliopoulos, *Random Walks That Find Perfect Objects and the Lovász Local Lemma*, Proceedings of the 55th Annual Symposium on Foundations of Computer Science (2014), 494–503.

[2] R. Adler, D. Coppersmith e M. Hassner, *Algorithms for sliding block codes: an application of symbolic dynamics to information theory*, IEEE Transactions on Information Theory **29(1)** (1983), 5-22.

[3] N. Alon e Z. Füredi, *Legitimate colorings of projective planes*, Graphs and Combinatorics **5** (1989), 95-106.

[4] N. Alon, C. J. H. McDiarmid e B. Reed, *Acyclic coloring of graphs*, Random Structures & Algorithms **2** (1991), 277–288.

[5] N. Alon e J. Spencer, *The Probabilistic Method, 4th Edition*, Wiley (2016).

[6] R. B. Ash, *Information Theory*, Dover Publications (2012).

[7] N. Balachandran, *The Probabilistic Method in Combinatorics*, Lectures by Niranjan Balachandran, Caltech (2010-2011).

[8] C. Balbuena, *Incidence matrices of projective planes and of some regular bipartite graphs of girth 6 with few vertices*, SIAM Journal on Discrete Mathematics **22(4)** (2008), 1351–1363.

[9] R. Bissacot, *O Lema Local de Lovász: Do Método Mágico de Erdős à Teoria Dos Gases de Rede*, 2° Colóquio de Matemática da Região Sul, (2012).

[10] R. Bissacot e L. Doin, *Entropy Compression Method and Legitimate Colorings in Projective Planes*, arXiv: 1710.06981 (2017).

[11] R. Bissacot, R. Fernández, A. Procacci e B. Scoppola, *An Improvement of the Lovász Local Lemma via Cluster Expansion*, Combinatorics, Probability and Computing **20** (2011), 709-719.

[12] B. Bollobás, *Random Graphs*, Cambridge University Press (2001).

[13] B. Bosek, S. Czerwiński, J. Grytczuk e P. Rzążewski, *Harmonious coloring of uniform hypergraphs*, Applicable Analysis and Discrete Mathematics **10** (2016), 73-87.

[14] J. Cai, X. Li e G. Yan, *Improved upper bound for the degenerate and star chromatic numbers of graphs*, Journal of Combinatorial Optimization **34(2)** (2017), 441–452.

[15] J. Böttcher, Y. Kohayakawa e A. Procacci, *Properly coloured copies and rainbow copies of large graphs with small maximum degree*, Random Structures & Algorithms **40(4)** (2012), 425–436.

[16] A. Broder, A. Frieze e E. Upfal, *Static and Dynamic Path Selection on Ezpander Graphs: A Random Walk Approach*, Proceedings of the 29th ACM Symposium on Theory of Computing (1997), 531-539.

[17] R. Bruck e H. J. Ryser, *The nonexistence of certain finite projective planes*, Canadian Journal of Mathematics **1** (1949), 88-93.

[18] W. Bussey e O. Veblen, *Finite projective geometries*, Transactions of the American Mathematical Society **7** (1906), 241-259.

[19] K. Chandrasekaran, N. Goyal e B. Haeupler, *Deterministic Algorithms for the Lovasz Local Lemma*, SIAM Journal on Discrete Mathematics **42(6)** (2013), 2132–2155.

[20] L. Comtet, *Advanced Combinatorics; The art of finite and infinite expansions*, Springer Verlag, Berlin (2014).

[21] T. Cover e J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York (1991).

[22] R. Dobrushin, *Perturbation methods of the theory of Gibbsian fields*, P. Bernard(Ed.), Lectures on Probability Theory and Statistics, Ecole d'Eté de Probabilité de Saint-Flour XXIV (1994), Lectures Notes in Mathematics, Springer Verlag, Berlin **1648** (1996), 1–66.

[23] M. Drmota, *Combinatorics and Asymptotics on Trees*, Cubo Journal **6(2)** (2004).

[24] V. Dujmović, G. Joret, J. Kozik e D. R. Wood , *Nonrepetitive Colouring via Entropy Compression*, Combinatorica (2015), 1-26.

[25] P. Erdős, *Some remarks on the theory of graphs*, Bulletin of the AMS - American Mathematical Society **53** (1947), 292-294.

[26] P. Erdős, *On Problem in Graph Theory*, The Mathematical Gazette **47** (1963), 220-223.

[27] P. Erdős e L. Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions in infinite and finite sets. Vol. II*, Colloq. János Bolyai Mathematical Society, North Holland, Amsterdam **11** (1975), 609-627.

[28] P. Erdős e J. Spencer, *Lopsided Lovász Local Lemma and latin transversals*, Discrete Applied Mathematics, **30(2–3)** (1991), 151-154.

[29] L. Esperet e A. Parreau, *Acyclic edge-coloring using entropy compression*, European Journal of Combinatorics **34(6)** (2013), 1019–1027.

[30] D. Fadiev, *On the Notion of Entropy of Finite Probability Space (in russian)*, Uspekhi Matematicheskikh Nauk **11** (1956), 227-231.

[31] R. Fernández e A. Procacci, *Cluster expansion for abstract polymer models. New bounds from an old approach*, Communications in Mathematical Physics **274(1)** (2007), 123–140.

[32] M. Figueiredo, *Elementos de Teoria da Informação*, Instituto Superior Técnico, Departamento de Engenharia Electrotécnica e de Computadores, Portugal, Lisboa (2007).

[33] P. Flajolet e R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press (2009).

[34] L. Fortnow, *A Kolmogorov Complexity Proof of the Lovász Local Lemma*, Computational Complexity (2009).

[35] D. Gonçalves, M. Montassier e A. Pinlou, *Entropy compression method applied to graph colorings*, 9th International Colloquium on Graph Theory and Combinatorics (2014).

[36] I. Goulden e D. Jackson, *Combinatorial enumeration*, John Wiley and Sons, New York (1983).

[37] J. Grytczuk, J. Kozik e P. Micek, *New approach to nonrepetitive sequences*, Random Structures & Algorithms **42(2)** (2013), 214-225.

[38] M. Hall, *Projective planes*, Transactions of the American Mathematical Society **54 (2)** (1943), 229-277.

[39] F. Harary e E. Palmer, *Graphical Enumeration*, Academic Press, New York (1973).

[40] P. Haxell, *A note on vertex list colouring*, Combinatorics, Probability and Computing **10** (2001), 345-347.

[41] J. Kahn, *Asymptotically good list-colorings*, Journal of Combinatorial Theory (A) **73** (1996), 1-59.

[42] B. Kaup e L. Kaup, *Holomorphic Functions of Several Variables*, De Gruyter Studies in Mathematics, Berlin (1983).

[43] J. Kim, *On Brook's Theorem for sparse graphs*, Combinatorics, Probability and Computing **4** (1995), 97-132.

[44] K. Kolipaka e M. Szegedy, *Moser and Tardos meet Lovász*, Proceedings of the forty-third annual ACM symposium on Theory of computing (2011), 235–244.

[45] C. Lam, *The search for finite projective plan of order 10*, Journal American Mathematical Monthly **98(4)** (1991), 305-318.

[46] F. Leighton, B. Maggs e S. Rao, *Packet routing and job-shop scheduling in O(congestion + dilation) steps*, Combinatorics **14** (1994), 167-180.

[47] B. Marcus, *Sofic systems and encoding data*, IEEE Transactions on Information Theory, **31(3)** (1985), 366-377.

[48] C. McDiarmid, *On the method of bounded differences*, Surveys in combinatorics, London Mathematical Society Lecture Note Series, Cambridge University Press **141** (1989) 148-188.

[49] A. Meir e J. Moon, *On the altitude of nodes in random trees*, Canadian Journal of Mathematics **30** (1978), 997-1015.

[50] M. Molloy e B. Reed, *Further algorithmic aspects of the local lemma*, Proceedings of the 30th Annual ACM Symposium on Theory of Computing (1998), 524–529.

[51] R. Moser e G. Tardos, *A constructive proof of the general Lovász Local Lemma*, Journal of the ACM **572** (2010), 11:1-11:15.

[52] R. Muthu, N. Narayanan e C. Subramanian, *Improved bounds on acyclic edge colouring*, Discrete Mathematics **307** (2007), 3063–3069.

[53] S. Ndreca, A. Procacci e B. Scoppola, *Improved bounds on coloring of graphs*, European Journal of Combinatorics **33(4)** (2012), 592–609.

[54] W. Pegden, *An extension of the Moser–Tardos Algorithmic Local Lemma*, SIAM Journal on Discrete Mathematics **28(2)** (2014), 911–917.

[55] J. Przybylo, *On the facial Thue choice index via entropy compression*, Journal of Graph Theory **77** (2014), 180-189.

[56] J. Przybylo, J. Schreyer e E. Škrabul'áková, *On the facial Thue choice number of plane graphs via entropy compression method*, Graphs and Combinatorics **32** (2016), 1137-1153.

[57] B. Reed, *The list colouring constants*, Journal of Graph Theory **31** (1999), 149–153.

[58] I. Sanov, *On the probability of large deviations of random variables*, Matematicheskii Sbornik, **42** (1957), 11-44. Para uma tradução para o inglês veja: Selected Translations in Mathematical Statistics and Probability **1** (1961), 213-244.

[59] A. Scott e A. D. Sokal, *The repulsive lattice gas, the independent-set polynomial, and the Lovász Local Lemma*, Journal of Statistical Physics **118(5–6)** (2005), 1151–1261.

[60] C. Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal **27** (1948), 379–423; 623–656.

[61] M. Šileikis, *Inequalities for Sums of Random Variables: a combinatorial perspective*, PhD thesis - Department of Mathematics and Computer Science, Adam Mickiewicz University, Poznan (2012).

[62] J. Spencer, *Asymptotic Lower Bounds for Ramsey Functions*, Discrete Mathematics **20** (1977), 69-76.

[63] T. Szele, *Kombinatorikai vizsgalatok az iranyitott teljes graffal kapcsolatban*, Középiskolai Matematikai és Fizikai Lapok Informatika rovattal **50** (1943), 223-256. Para uma tradução para o alemão veja: T. Szele, Publicationes Mathematicae Debrecen, **14** (1966), 145-168.

[64] T. Tao, *Moser's entropy compression argument*, What's New (2009).

[65] O. Veblen e J. Wedderburn, *Non-Desarguesian and non-Pascalian geometries*, Transactions of the American Mathematical Society **8** (1907), 379-388.

[66] J. Vitter e P. Flajolet, *Average-Case Analysis of Algorithms and Data Structures*, Handbook of Theoretical Computer Science, vol. A (1991), 431-524.

[67] H. S. Wilf, Generatingfunctionology, Academic Press Inc. (1990).