Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Marcelo Yukio Iyama Silvarolla

O Lema Local de Lovász e o Algoritmo de Moser-Tardos

São Paulo
Novembro de 2016

O Lema Local de Lovász e o Algoritmo de Moser-Tardos

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Rodrigo Bissacot

São Paulo
Novembro de 2016

Resumo

O Lema Local de Lovász (LLL) é uma ferramenta útil para se provar a existência de objetos combinatórios, enquandrando-se no chamado "método probabilístico". Uma versão algorítmica foi provada por Moser e Tardos ([11]), possibilitando que se encontrem tais objetos, e de maneira mecânica. Demonstramos o LLL, juntamente com sua versão algorítmica. Também apresentamos novas versões do lema provindas da sua conexão com o Gás de Rede e suas respectivas algoritmizações. O texto é acessível a alunos de graduação de matemática e computação. **Palavras-chave:** Lema Local de Lovász, algoritmo de Moser-Tardos

Abstract

The Lovász Local Lemma (LLL) is a useful tool for proving the existence of combinatorial objects, and is part of the so-called "probabilistic method". An algorithmic version, that finds such objects computationally, was proved by Moser and Tardos ([11]). We prove the LLL and its algorithmic version. We also present the new versions of the lemma that come from its connection with the lattice gas, and their algorithmization. Our presentation is accessible to undergraduate students of mathematics and computer science.

Keywords: Lovász Local Lemma, Moser-Tardos Algorithm

Sumário

Capítulo 1

Introdução

Uma técnica importante em combinatória é o método probabilístico. O objetivo do método é provar a existência de um objeto matemático satisfazendo certas condições, por exemplo, uma bicoloração dos elos de um grafo tal que não exista subgrafo completo monocromático. Para tal, o método constrói um espaço de probabilidade tal que a probabilidade de um objeto satisfazer as condições é positiva.

Note que, embora existam eventos não-vazios com probabilidade zero, se o evento for vazio, ele, pela definição de espaço de probabilidade, tem probabilidade zero. Isso nos permite concluir, a partir do fato de a probabilidade do objeto satisfazer as condições ser positiva, que um objeto satisfazendo as condições, de fato, existe.

O Lema Local de Lovász (LLL) é uma das ferramentas disponíveis para o matemático que deseja utilizar o método probabilístico. Tal lema nos diz que, se impusermos certas restrições sobre as dependências entre diversos eventos ruins, então podemos concluir que a probabilidade de que nenhum dos eventos ruins ocorra é positiva. Um caso bem particular desse lema é aquele em que todos os eventos ruins são independentes e de probabilidade diferente de 1. Aqui basta observarmos que a probabilidade de que nenhum dos eventos ruins ocorra é o produto das probabilidades de que cada evento ruim não ocorra e tal produto é positivo, pois cada multiplicando o é.

Para encapsular as restrições sobre as dependências dos eventos, o LLL utiliza um grafo de dependência: um grafo indexado pelos eventos ruins (mais precisamente, pelo conjunto de índices dos eventos) tal que para qualquer evento ruim, os eventos não ligados a ele pelo grafo de dependência são mutuamente independentes dele. Os detalhes são apresentados no capítulo 3.

É comum, na apresentação do LLL, ignorarem-se os cálculos de probabilidade condicionadas à ocorrência de eventos de probabilidade zero. Para deixar a apresentação precisa e clara para alunos de graduação, consertamos esse problema explicitamente. Para isso, utilizamos uma convenção descrita na seção 2.

Uma crítica que se pode fazer ao LLL é que ele não nos diz quem é o objeto cuja existência se demonstra. Após vários anos de pesquisa por diversos matemáticos para melhorar o LLL nesse aspecto, encontrou-se uma versão algorítmica do lema que funciona em praticamente todas as aplicações do lema original. Moser e Tardos, em 2009, publicaram o seu *paper* ([11]) demonstrando o novo teorema.

Em diversos pontos, o artigo não apresenta detalhes de formalização. Aqui, faremos uma apresentação mais detalhada, visando o entendimento por alunos de graduação.

Nos últimos dois capítulos, explicamos como o critério do LLL tradicional pode ser melhorado e apresentamos versões algorítmicas destas melhoras.

Capítulo 2

Uma convenção para a probabilidade condicional e três consequências

Supomos a familiaridade do leitor com as definições e proposições básicas da Teoria da Probabilidade (ver [23]). No entanto, entraremos brevemente em tal território, a fim de destacarmos uma convenção incomum (Definição 2.1) e três proposições dela resultantes (Proposições 2.3, 2.4 e 2.6), utilizadas na demonstração do Lema de Lovász.

Definição 2.1. Seja $(\Omega, \mathcal{F}, P)$ um espaço de probabilidade. Definimos a *probabilidade condicional de um evento A dado um evento B* por $P(A|B) := \frac{P(A \cap B)}{P(B)}$ se $P(B) \neq 0$ e $P(A|B) := P(A)$ se $P(B) = 0$.

Observação 2.2. A definição da probabilidade condicional no caso em que $P(B) = 0$ é apenas uma convenção, mas útil, pois nos permite enunciar, sem nos preocuparmos com eventos de probabilidade zero, a proposição 2.3 a seguir.

Proposição 2.3. *Dois eventos A e B (num certo espaço de probabilidade) são independentes se, e somente se, $P(A|B) = P(A)$.*

Demonstração. Se $P(B) = 0$, então a igualdade vale e os eventos são independentes (pois $P(A \cap B) = 0 = P(A)P(B)$). Se $P(B) \neq 0$, então $P(A|B) = P(A) \iff P(A \cap B)/P(B) = P(A) \iff P(A \cap B) = P(A)P(B)$ e esta última afirmação é, por definição, equivalente à independência dos eventos A e B. □

Proposição 2.4 (Regra do produto)**.** *Se $A_1, A_2, \ldots, A_n$ ($n \geq 0$) são eventos num espaço de probabilidade $(\Omega, \mathcal{F}, P)$, então*

$$P(A_1 \cap A_2 \cap \ldots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \cdots P(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1})$$

Demonstração. Faremos indução em n. Os casos $n = 0$ e $n = 1$ são triviais. (Vale ressaltar que convencionamos intersecções sobre conjuntos de índices vazios como sendo o espaço amostral Ω, ao passo que produtórios sobre conjuntos de índices vazios são supostos iguais a 1.) Suponha $n \geq 2$. Temos $P(A_1 \cap A_2 \cap \ldots \cap A_n) = P((A_1 \cap A_2 \cap \ldots \cap A_{n-1}) \cap A_n) = P(A_1 \cap A_2 \cap \ldots \cap A_{n-1})P(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1}) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \cdots P(A_{n-1}|A_1 \cap A_2 \cap \ldots \cap A_{n-2}) \cdot P(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1})$

A segunda igualdade decorre da Definição 2.1, enquanto a terceira vem da hipótese de indução. □

Observação 2.5. Note que a regra do produto vale mesmo quando alguma intersecção da forma $A_1 \cap A_2 \cap \ldots \cap A_i$ ($i \in \{1, 2, ..., n\}$) tiver probabilidade zero.

Finalmente, um resultado análogo ao que acabamos de ver:

Proposição 2.6 (Regra do produto condicional)**.** *Se $A_1, A_2, \ldots, A_n$ ($n \geq 0$) e B são eventos num espaço de probabilidade $(\Omega, \mathcal{F}, P)$ e $P(B) \neq 0$, então*

$$P(A_1 \cap A_2 \cap \ldots \cap A_n | B) = P(A_1|B)P(A_2|A_1 \cap B)P(A_3|A_1 \cap A_2 \cap B) \cdots P(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1} \cap B)$$

Demonstração. Basta usar a regra do produto usual (2.4). □

Capítulo 3

O Lema Local de Lovász

Ao se enunciar e provar o LLL, costuma-se não levar em conta o caso em que se calcula probabilidades do tipo $P(A|B)$ com $P(B) = 0$. O problema pode ser sanado, sem grandes dificuldades, com a adoção da convenção 2.1, conforme mostramos no Teorema 3.2 a seguir.

Vamos supor, a partir de agora, além de conhecimentos básicos de Teoria da Probabilidade, a familiaridade do leitor com conceitos elementares de Teoria dos Grafos (ver capítulos iniciais de [5]).

Notação 3.1. Se $G = (X, E)$ é um grafo e $x \in X$, denotamos por $\Gamma(x)$ a vizinhança de x em G e pomos $\Gamma^*(x) = \Gamma(x) \cup \{x\}$. (Note que não estamos indicando o grafo em relação ao qual se toma a vizinhança. Isso não é problema, pois estará claro de que grafo se trata em cada contexto em que utilizarmos a notação. O mesmo vale para várias outras notações que usamos no texto.)

A nossa demonstração do LLL é uma adaptação daquela feita em [5], a mesma feita em diversas fontes. (Em [5], demonstra-se, na verdade, o que aqui é o Teorema 3.8.)

Teorema 3.2 (Lema Local de Lovász)**.** *Seja $G = (X, E)$ um grafo finito. Seja $(A_x)_{x\in X}$ uma família de eventos em algum espaço de probabilidade fixado $(\Omega, \mathcal{F}, P)$. Suponha que existam $(r_x)_{x\in X}$ em $[0, 1)$ tais que, para todo $x \in X$,*

$$P(A_x|\cap_{y\in Y} A_y^c) \leq r_x \prod_{z\in\Gamma(x)} (1 - r_z) \tag{3.3}$$

para todo $Y \subseteq X \setminus \Gamma^*(x)$. Então $P(\cap_{x\in X} A_x^c) \geq \prod_{x\in X}(1 - r_x) > 0$.

Demonstração. O caso $X = \emptyset$ é trivial. Para fixar as ideias, podemos supor, portanto, que $X \neq \emptyset$. Além disso, para facilitar a escrita, vamos supor $X = \{1, 2, \ldots, n\}$ onde $n \in \mathbb{N}$. Considere a seguinte

Afirmação 3.4. *Para todo $x \in X$ e $S \subseteq X$ com $x \notin S$,*

$$P(A_x|\bigcap_{s\in S} A_s^c) \leq r_x \tag{3.5}$$

Observe que basta demonstrar tal afirmação, pois daí, pela regra do produto, $P(\cap_{x\in X} A_x^c) = P(\cap_{i=1}^n A_i^c) = P(A_1^c)P(A_2^c|A_1^c)P(A_3^c|A_1^c \cap A_2^c) \cdots P(A_n^c|A_1^c \cap A_2^c \cap \ldots \cap A_{n-1}^c) \geq (1-r_1)(1-r_2)(1-r_3)\cdots(1-r_n) = \prod_{x\in X}(1-r_x) > 0$

Demonstração. Observemos, primeiramente, que se $S \cap \Gamma^*(x) = \emptyset$, isto é, $S \subseteq X \setminus \Gamma^*(x)$, então 3.5 segue imediatamente de 3.3, pois daí $P(A_x|\bigcap_{s\in S} A_s^c) \leq r_x \prod_{z\in\Gamma(x)}(1-r_z) \leq r_x$.

Vamos proceder por indução em $|S|$. Se for $|S|= 0$, então $S \cap \Gamma^*(x) = \emptyset$ e, assim, recaímos no caso que acabamos de mencionar. Agora, supondo que a afirmação valha para todos os conjuntos de cardinalidade menor que $|S|$, vamos mostrar que vale para $S \subseteq X$.

Seja $x \in X$ tal que $x \notin S$. Se $P(\bigcap_{s\in S} A_s^c) = 0$, então $P(A_x|\bigcap_{s\in S} A_s^c) = P(A_x) = P(A_x|\Omega) = P(A_x|\cap_{y\in\emptyset} A_y^c) \leq r_x \prod_{z\in\Gamma(x)}(1-r_z) \leq r_x$, aplicando 3.3 para $Y = \emptyset$. Caso contrário, isto é, $P(\bigcap_{s\in S} A_s^c) \neq 0$, pondo, para facilitar a escrita, $m := |S \cap \Gamma^*(x)|\geq 0$ e chamando de $j_1, j_2, \ldots, j_m$ os elementos de $S \cap \Gamma^*(x)$, temos o seguinte:

$$
\begin{aligned}
P(A_x|\bigcap_{s\in S} A_s^c) =& P(A_x| \bigcap_{s\in S\cap\Gamma^*(x)} A_s^c \ \cap \bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c) \\
=^{2.6}& \frac{P(A_x \cap \bigcap_{s\in S\cap\Gamma^*(x)} A_s^c|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c)}{P(\bigcap_{s\in S\cap\Gamma^*(x)} A_s^c|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c)} \\
\leq& \frac{P(A_x|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c)}{P(\cap_{l=1}^m A_{j_l}^c|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c)} \\
\leq^{3.3,2.6}& \frac{r_x \prod_{z\in\Gamma(x)}(1-r_z)}{\prod_{l=1}^m P(A_{j_l}^c|\bigcap_{k=1}^{l-1} A_{j_k}^c \cap \bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c)} \\
\leq^{3.5}& \frac{r_x \prod_{z\in\Gamma(x)}(1-r_z)}{\prod_{l=1}^m (1-r_{j_l})} \\
\leq& r_x
\end{aligned}
$$

Note que os denominadores são todos diferentes de zero, pois $P(\bigcap_{s\in S\cap\Gamma^*(x)} A_s^c|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c) = P(\cap_{l=1}^m A_{j_l}^c|\bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c) =^{2.6} \prod_{l=1}^m P(A_{j_l}^c|\bigcap_{k=1}^{l-1} A_{j_k}^c \cap \bigcap_{s\in S\cap(\Gamma^*(x))^c} A_s^c) \geq^{3.5} \prod_{l=1}^m (1 - r_{j_l}) > 0$. □

□

A maneira como apresentamos o LLL acima não é a tradicional, que envolve grafos de dependência. Apresentaremos esta agora, mas, antes, precisaremos de algumas definições.

Definição 3.6. Fixe um espaço de probabilidade $(\Omega, \mathcal{F}, P)$. Um evento A é *mutuamente independente* de uma família finita de eventos $\{A_x : x \in X\}$ se A é independente de $\cap_{y\in Y} A_y$ para todo $Y \subseteq X$.

Definição 3.7. Um grafo $G = (X, E)$ é um *grafo de dependência para a família de eventos (indexada pelos vértices de G)* $(A_x)_{x\in X}$ (de um espaço de probabilidade fixado) se, para todo $x \in X$, o evento A_x é mutuamente independente da família de eventos $\{A_y : y \in X \setminus \Gamma^*(x)\}$

É natural pensar que os eventos ligados por elos no grafo possuem alguma dependência, mas isso não é necessariamente verdade, já que o grafo completo é um grafo de dependência para qualquer família de eventos, mesmo que independentes.

É melhor pensar em termos de ausência de elos, ao invés de presença: intuitivamente, um evento não tem "nada a ver"com os eventos com os quais ele não está ligado pelo grafo.

Teorema 3.8 (LLL com grafos de dependência)**.** *Seja $G = (X, E)$ um grafo de dependência para uma família $(A_x)_{x\in X}$ de eventos em algum espaço de probabilidade fixado $(\Omega, \mathcal{F}, P)$. Suponha que existam $(r_x)_{x\in X}$ em $[0, 1)$ tais que, para todo $x \in X$,*

$$P(A_x) \leq r_x \prod_{z\in\Gamma(x)} (1 - r_z) \tag{3.9}$$

Então $P(\cap_{x\in X} A_x^c) \geq \prod_{x\in X}(1 - r_x) > 0$.

Demonstração. Basta usarmos o LLL que provamos acima (Teorema 3.2), observando que vale 3.3, pois A_x é independente de $\cap_{y\in Y} A_y^c$ e, portanto, pela Proposição 2.3, temos $P(A_x|\cap_{y\in Y} A_y^c) = P(A_x) \leq r_x \prod_{z\in\Gamma(x)}(1 - r_z)$. □

Capítulo 4

O LLL algorítmico

O Lema de Lovász provado no capítulo anterior nos garante a existência de (pelo menos) um objeto matemático que evita todos os eventos "ruins", mas não nos diz como encontrá-lo. Moser e Tardos ([11]) conseguiram, sob certas condições além das do LLL, um algoritmo[1] que faz exatamente isso. Essencialmente, necessita-se supor que os eventos "ruins"sejam determinados por um conjunto finito de variáveis aleatórias independentes discretas, conforme detalhamos nesta seção.

Com o objetivo de definir o algoritmo de Moser-Tardos, definiremos vbl(A) para eventos (ruins) $A \in \mathcal{A}$. Para definirmos vbl, estabeleceremos o que significa um conjunto ser determinado por um conjunto de funções.

4.1 σ-álgebras e conjuntos determinados por funções

Notação 4.1. Se $\Omega \neq \emptyset$ e $\psi_1, \psi_2, \ldots, \psi_n$ forem funções $\Omega \to \mathbb{R}$, vamos escrever

$$[\psi_1, \ldots, \psi_n \text{ satisfazem certa propriedade}]$$

no lugar de

$$\{\omega \in \Omega : \psi_1(\omega), \ldots, \psi_n(\omega) \text{ satisfazem a tal propriedade}\}.$$

Por exemplo,

$$[\psi_1 \leq 5] := \{\omega \in \Omega : \psi_1(\omega) \leq 5\},$$

$$[\psi_1 < 9, \psi_2 \in \{1, 3, 15, 42\}] := \{\omega \in \Omega : \psi_1(\omega) < 9 \text{ e } \psi_2(\omega) \in \{1, 3, 15, 42\}\}.$$

As vírgulas na expressão entre colchetes devem ser interpretadas como e's lógicos.

Além disso, se estivermos trabalhando com um espaço de probabilidade $(\Omega, \mathcal{F}, P)$ e tivermos

$$[\psi_1, \ldots, \psi_n \text{ satisfazem a tal propriedade}] \in \mathcal{F},$$

então poderemos escrever

$$P(\psi_1, \ldots, \psi_n \text{ satisfazem certa propriedade})$$

[1]Conforme o restante da literatura, não provemos uma definição formal para o termo *algoritmo*. Supomos a familiaridade do leitor com o conceito, dando apenas a seguinte definição informal: "Um algoritmo é uma n-upla de instruções. Uma instrução pode consistir em, dentre outra coisas: atribuir um valor a uma variável ('x recebe $7+y$'), pular para outra instrução ('vá para a terceira instrução se $x > 5$') e fazer um sorteio ('sorteie um valor para a variável aleatória ψ de acordo com sua distribuição')."Note que *loops* podem ser traduzidos para saltos condicionais para o algoritmo ficar condizente com a definição.

no lugar de

$$P([\psi_1, \ldots, \psi_n \text{ satisfazem a tal propriedade}]).$$

Notação 4.2. Fixe um conjunto não-vazio Ω. A σ-álgebra gerada por um conjunto $\mathcal{C} \subseteq \mathcal{P}(\Omega)$ é denotada por $\sigma(\mathcal{C})$. Vamos sobrecarregar a notação e denotar a σ-álgebra gerada por um conjunto Ψ de funções $\psi : \Omega \to \mathbb{R}$ por $\sigma(\Psi)$. Supomos que o leitor já esteja familiarizado com esses conceitos, mas lembramos que

$$\sigma(\mathcal{C}) := \bigcap_{\mathcal{G} \text{ é } \sigma\text{-álgebra de } \Omega \text{ e } \mathcal{C} \subseteq \mathcal{G}} \mathcal{G}$$

é a menor sigma-álgebra de Ω que contém $\mathcal{C}$ (i.é., que torna os elementos de $\mathcal{C}$ mensuráveis), e que

$$\sigma(\Psi) := \sigma(\{[\psi < \alpha] : \psi \in \Psi, \alpha \in \mathbb{R}\})$$

é a menor sigma-álgebra de Ω que torna todas as funções de Ψ variáveis aleatórias (i.é., mensuráveis).

Definição 4.3. Sejam Ω um conjunto não-vazio e Ψ um conjunto (possivelmente vazio) de funções $\psi : \Omega \to \mathbb{R}$. Uma *configuração de* Ψ (ou *das funções em* Ψ) é uma função $f : \Psi \to \mathbb{R}$ tal que $f(\psi) \in \mathrm{Im}(\psi)\ \forall \psi \in \Psi$.

Por exemplo, suponha que $\Psi = \{\psi_1, \psi_2, \psi_3\}$ e que as ψ assumem valores em $\{0, 1, 2\}$. Então, para cada $\omega \in \Omega$, é possível, em princípio, que $(\psi_1(\omega), \psi_2(\omega), \psi_3(\omega))$ seja igual a $(0, 0, 0)$ ou $(0, 0, 1)$ ou $\ldots (2, 2, 1)$ ou $(2, 2, 2)$. Esses vetores são as configurações de Ψ. (Estritamente falando, teríamos de escrever os vetores em termos de funções $\Psi \to \mathbb{R}$.)

Definição 4.4. Sejam Ω um conjunto não-vazio e Ψ um conjunto (possivelmente vazio) de funções $\psi : \Omega \to \mathbb{R}$. Dizemos que $A \subseteq \Omega$ *é determinado por* Ψ (ou *pelas funções em* Ψ) se para toda configuração $f : \Psi \to \mathbb{R}$ de Ψ, o conjunto $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\}$ está contido em A ou em A^c.

Intuitivamente, isso significa que sempre que soubermos os valores que as $\psi \in \Psi$ assumem (i.é., conhecemos os $\psi(\omega)$), nós consequentemente saberemos se A ocorre ou não (i.é., se $\omega \in A$ ou não).

Note que $\emptyset$ e Ω são sempre determinados por Ψ e que eles são os únicos quando $\Psi = \emptyset$.

Alguns resultados que nos serão úteis:

Proposição 4.5. *A é determinado por Ψ se, e somente se, A é uma união (arbitrária) de conjuntos da forma $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\}$, em que f é configuração.*

Demonstração. Basta usar o fato de que a família de todos os conjuntos da forma $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\}$ é uma partição de Ω (estamos permitindo, aqui, que uma partição tenha o vazio como elemento). □

Proposição 4.6. *Todo elemento de $\sigma(\Psi)$ é determinado por Ψ.*

Demonstração. (Note que o caso $\Psi = \emptyset$ é trivial, então podemos esquecê-lo para facilitar o pensamento.)

Como $\sigma(\Psi)$ é a menor σ-álgebra que torna todas as $\psi \in \Psi$ mensuráveis, basta mostrarmos que a família $\mathcal{F}$ de conjuntos determinados por Ψ forma uma σ-álgebra que torna todas as $\psi \in \Psi$ mensuráveis.

Pela proposição anterior, $\mathcal{F}$ é a família de conjuntos A que podem ser escritos como uma união (arbitrária) de conjuntos da forma $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\}$, com f configuração de Ψ.

Como o vazio pode ser escrito como uma união sobre um conjunto vazio de índices, segue que $\emptyset \in \mathcal{F}$. Além disso, uma união arbitrária (em particular, uma enumerável) de conjuntos A que, por sua vez, são uniões arbitrárias de conjuntos da forma $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\}$ pode ser escrita como uma união arbitrária (uma só) de conjuntos dessa forma. Daí $\mathcal{F}$ é fechado em relação a uniões arbitrárias e, em particular, uniões enumeráveis. Para concluirmos que $\mathcal{F}$ é uma σ-álgebra, basta provarmos que o complemento de todo $A \in \mathcal{F}$ está em $\mathcal{F}$. De fato, escreva $A = \bigcup_{i\in I}\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\ \forall \psi \in \Psi\}$. Daí $A^c = \bigcup_{j\in J}\{\omega \in \Omega : \psi(\omega) = g_j(\psi)\ \forall \psi \in \Psi\}$, em que $\{g_j\}_{j\in J}$ é o conjunto de todas as configurações de Ψ que não estão em $\{f_i\}_{i\in I}$

Agora falta mostrarmos que $\mathcal{F}$ torna todas as $\psi \in \Psi$ mensuráveis. Ora, dados $\psi_0 \in \Psi$ e $\alpha \in \mathbb{R}$, temos $[\psi_0 < \alpha] = \bigcup_{k\in K}\{\omega \in \Omega : \psi(\omega) = h_k(\psi)\ \forall \psi \in \Psi\} \in \mathcal{F}$, onde $\{h_k\}_{k\in K}$ é o conjunto de todas as configurações h de Ψ tais que $h(\psi_0) < \alpha$.

(Acidentalmente, $\mathcal{F}$ também é uma topologia para Ω.) □

A contenção oposta à da proposição anterior vale, com duas hipóteses adicionais.

Proposição 4.7. *Se Ψ é enumerável e o conjunto das configurações de Ψ também, então todo conjunto $A \subseteq \Omega$ determinado por Ψ está em $\sigma(\Psi)$. Em particular, se Ψ é finito e a imagem de cada $\psi \in \Psi$ é enumerável, então todo conjunto $A \subseteq \Omega$ determinado por Ψ está em $\sigma(\Psi)$.*

Demonstração. (Note que o caso $\Psi = \emptyset$ é trivial, então podemos esquecê-lo para facilitar o pensamento.)

Seja $A \subseteq \Omega$ determinado por Ψ. Escreva, através da Proposição 4.5, $A = \bigcup_{i\in I}\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\ \forall \psi \in \Psi\}$, em que as f_i são configurações de Ψ. Como o conjunto das configurações é enumerável, podemos supor I enumerável. Como $\sigma(\Psi)$ é fechado em relação a uniões enumeráveis, basta mostrarmos que cada $\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\ \forall \psi \in \Psi\}$ está em $\sigma(\Psi)$.

Mas $\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\ \forall \psi \in \Psi\} = \bigcap_{\psi\in\Psi}\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\}$. Esta intersecção é enumerável, pois Ψ o é, donde, pelo fato de $\sigma(\Psi)$ ser fechado em relação a intersecções enumeráveis, basta mostrarmos que para cada $i \in I$ e cada $\psi \in \Psi$, temos $\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\} \in \sigma(\Psi)$.

De fato, $\{\omega \in \Omega : \psi(\omega) = f_i(\psi)\} = \psi^{-1}(f_i(\psi))$ é a pré-imagem de um ponto e, portanto, a pré-imagem de um boreliano de $\mathbb{R}$, de modo que $\psi^{-1}(f_i(\psi)) \in \sigma(\Psi)$ é uma condição necessária para que $\sigma(\Psi)$ realmente torne as $\psi \in \Psi$ mensuráveis.

O caso particular vem do fato de que, quando Ψ é finito, o conjunto de configurações de Ψ está, da maneira natural, em bijeção com o produto cartesiano (finito) das imagens das $\psi \in \Psi$. O produto cartesiano finito de conjuntos enumeráveis é, como se sabe, enumerável. Portanto, as hipóteses do caso particular implicam as do caso geral. □

4.2 Definindo vbl

Definição 4.8. Sejam Ω um conjunto não-vazio e Ψ um conjunto de funções $\psi : \Omega \to \mathbb{R}$. Dizemos que uma configuração $f : \Psi \to \mathbb{R}$ é *factível* se $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in \Psi\} \neq \emptyset$.

Lema 4.9. *Sejam Ω um conjunto não-vazio e S, T conjuntos (possivelmente vazios ou infinitos) de funções $\psi : \Omega \to \mathbb{R}$. Suponha que toda configuração de $S \cup T$ é factível. Se A é determinado por S e por T, separadamente, então A é determinado por $S \cap T$.*

Demonstração. Queremos mostrar que para toda configuração $f : S \cap T \to \mathbb{R}$ de $S \cap T$, temos que $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\}$ está contido em A ou em A^c.

Se $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\} \subseteq A^c$, acabamos. Suponhamos, então $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\} \cap A \neq \emptyset$, ou seja, que existe $\omega_A \in \{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\} \cap A$. Mostremos que $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\} \subseteq A$.

Observe que $B := \{\omega \in \Omega : \psi(\omega) = \psi(\omega_A)\ \forall \psi \in S\} = \{\omega \in \Omega : \psi(\omega) = \omega_A^*(\psi)\ \forall \psi \in S\}$, onde $\omega_A^* : S \to \mathbb{R}$ é dada por $\omega_A^*(\psi) = \psi(\omega_A)\ \forall \psi \in S$. Daí, como A é determinado por S e $\omega_A \in B$, temos $B \subseteq A$.

Seja $\omega_0 \in \{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in S \cap T\}$. Mostremos que $\omega_0 \in A$.

Defina a configuração $g : S \cup T \to \mathbb{R}$ por $g(\psi) = \psi(\omega_A)$ se $\psi \in S$ e $g(\psi) = \psi(\omega_0)$ se $\psi \in T \setminus S$. Essa configuração é, por hipótese, factível, então existe $\omega_1 \in \Omega$ tal que $\psi(\omega_1) = g(\psi)\ \forall \psi \in S \cup T$.

Note que $\omega_0, \omega_1 \in \{\omega \in \Omega : \psi(\omega) = g|_T(\psi)\ \forall \psi \in T\}$. Como T determina A, segue que, para concluirmos que $\omega_0 \in A$, basta mostrarmos que $\omega_1 \in A$.

Ora, $\omega_1 \in B \subseteq A$, concluindo a demonstração.

□

Lema 4.10. *Sejam Ω um conjunto não-vazio e $T_1, T_2, \ldots, T_n$ ($n \geq 1$) conjuntos (possivelmente vazios ou infinitos) de funções $\psi : \Omega \to \mathbb{R}$. Suponha que toda configuração de $T_1 \cup T_2 \cup \ldots \cup T_n$ seja factível. Se A é determinado por $T_1, T_2, \ldots, T_n$ separadamente, então A é determinado por $T_1 \cap T_2 \cap \ldots \cap T_n$.*

Demonstração. O caso $n = 1$ é trivial e o caso $n = 2$ segue do lema anterior. Suponha que valha para $n - 1$; mostremos que vale para $n \geq 3$. Note, primeiramente, que se $X \subseteq T_1 \cup T_2 \cup \ldots \cup T_n$, então toda configuração de X é factível, porque, dada uma configuração $f : X \to \mathbb{R}$, temos $\{\omega \in \Omega : \psi(\omega) = f(\psi)\ \forall \psi \in X\} \supseteq \{\omega \in \Omega : \psi(\omega) = f^*(\psi)\ \forall \psi \in T_1 \cup \ldots \cup T_n\} \neq \emptyset$, onde $f^* : T_1 \cup \ldots \cup T_n \to \mathbb{R}$ é definida por $f^*(\psi) = f(\psi)$ se $\psi \in X$ e $f^*(\psi) :=$ "uma escolha de valor em $\mathrm{Im}(\psi)$"se $\psi \in T_1 \cup \ldots \cup T_n \setminus X$ (estamos usando o axioma da escolha se $T_1 \cup \ldots \cup T_n \setminus X$ é infinito).

Aplicando a observação acima para $X := T_1 \cap \ldots \cap T_{n-1}$, podemos usar a hipótese de indução, obtendo que A é determinado por $(T_1 \cap T_2 \cap \ldots \cap T_{n-1})$. Também sabemos, por hipótese, que A é determinado por T_n.

Agora, aplicando a observação para $X := (T_1 \cap \ldots \cap T_{n-1}) \cup T_n$, podemos usar o lema anterior, obtendo que A é determinado por $T_1 \cap T_2 \cap \ldots \cap T_n = (T_1 \cap T_2 \cap \ldots \cap T_{n-1}) \cap T_n$. □

Definição 4.11. Sejam Ω um conjunto não-vazio e Ψ um conjunto finito (possivelmente vazio) de funções $\psi : \Omega \to \mathbb{R}$, tal que toda configuração de Ψ é factível. Seja $A \subseteq \Omega$ determinado por Ψ. Definimos $\mathrm{vbl}(A)$ como o menor conjunto (com respeito a $\subseteq$) $S \subseteq \Psi$ que determina A.

Demonstração. Temos de mostrar que tal S sempre existe (e é único, mas a unicidade é óbvia). Defina S como a intersecção de todos os conjuntos $T \subseteq \Psi$ que determinam A. Como Ψ é finito e, portanto, $\mathcal{P}(\Psi)$ é finito, podemos enumerar esses T como $T_1, T_2, \ldots, T_n$. Temos $n \geq 1$, pois Ψ é um desses T. Note que S está contido em todos os $T \subseteq \Psi$ que determinam A, de modo que, se S determina A, então S é o menor subconjunto de Ψ que determina A. Basta, portanto, mostrarmos que S determina A. Ora $S = T_1 \cap \ldots \cap T_n$, cada T_i determina A, $T_1 \cup \ldots \cup T_n = \Psi$ e toda configuração de Ψ é, por hipótese, factível: então, pelo lema anterior, S determina A. □

4.3 O algoritmo de Moser-Tardos

O **algoritmo de Moser-Tardos** é definido da seguinte forma:

Entrada:

- um espaço de probabilidade $(\Omega, \mathcal{F}, P)$
- um conjunto finito não-vazio Ψ de variáveis aleatórias discretas independentes $\psi : \Omega \to \mathbb{R}$ tais que para todo $v \in \mathrm{Im}(\psi)$ vale $P(\psi = v) > 0$
- uma família finita não-vazia $\mathcal{A}$ de eventos determinados por Ψ
- o valor $\mathrm{vbl}(A)$ para cada $A \in \mathcal{A}$
- um algoritmo sorteia$(\Omega', \mathcal{F}', P', \psi')$ que recebe um espaço de probabilidade $(\Omega', \mathcal{F}', P')$ e uma variável aleatória ψ', e retorna um valor sorteado para ψ' de acordo com sua distribuição
- um algoritmo ocorre$(\Omega', \Psi', (v'_{\psi'})_{\psi' \in \Psi'}, A')$ que recebe um conjunto $\Omega' \neq \emptyset$, um conjunto finito e não-vazio Ψ' de funções $\psi' : \Omega' \to \mathbb{R}$, uma configuração $(v'_{\psi'})_{\psi' \in \Psi'}$ de Ψ' e $A' \subseteq \Omega'$, e retorna 1 se $\{\omega' \in \Omega' : \psi'(\omega') = v'_{\psi'} \ \forall \psi' \in \Psi'\} \subseteq A'$ e 0 caso contrário

Instruções:

- para cada $\psi \in \Psi$, faça $v_\psi :=$ sorteia$(\Omega, \mathcal{F}, P, \psi)$
- enquanto existir $A \in \mathcal{A}$ tal que ocorre$(\Omega, \Psi, (v_\psi)_{\psi \in \Psi}, A) == 1$, faça:
  - escolha um tal A arbitrariamente (determinística ou aleatoriamente)
  - (reamostramos A) para cada $\psi \in \mathrm{vbl}(A)$, faça $v_\psi :=$ sorteia$(\Omega, \mathcal{F}, P, \psi)$
- retorne $(v_\psi)_{\psi \in \Psi}$

Demonstração. Façamos algumas observações que garantem que o algoritmo está bem definido.

Mostremos que vbl está bem definido para $A \in \mathcal{A}$: queremos verificar que toda configuração de Ψ é factível. De fato, para toda configuração $f : \Psi \to \mathbb{R}$, temos $P(\{\omega \in \Omega : \psi(\omega) = f(\psi) \ \forall \psi \in \Psi\}) = P(\bigcap_{\psi \in \Psi} \{\omega \in \Omega : \psi(\omega) = f(\psi)\}) = \prod_{\psi \in \Psi} P(\{\omega \in \Omega : \psi(\omega) = f(\psi)\}) > 0$ (pois cada multiplicando é positivo), donde $\{\omega \in \Omega : \psi(\omega) = f(\psi) \ \forall \psi \in \Psi\} \neq \emptyset$.

Como Ψ e $\mathrm{vbl}(A)$ são finitos (embora este último, em princípio, possa ser vazio), faz sentido executar instruções para cada $\psi \in \Psi$ e para cada $\psi \in \mathrm{vbl}(A)$.

□

Observação 4.12. A condição de que as variáveis aleatórias $\psi \in \Psi$ são discretas é redundante.

Demonstração. Usaremos, sem demonstração, o seguinte resultado de Análise na Reta:

Lema 4.13. *Seja* $\{x_v : v \in \Lambda\} \subseteq [0, +\infty)$. *Defina* $\sum^*_{v \in \Lambda} x_v := \sup_{F \subseteq \Lambda, |F| < +\infty} \sum_{v \in F} x_v \in [0, +\infty]$. *Se* $\sum^*_{v \in \Lambda} x_v < +\infty$, *então* $\{v \in \Lambda : x_v > 0\}$ *é enumerável.*

Note que

$$\sum_{v\in\mathrm{Im}(\psi)}^{*} P(\psi=v) = \sup_{F\subseteq\mathrm{Im}(\psi),|F|<+\infty}\sum_{v\in F} P(\psi=v) = \sup_{F\subseteq\mathrm{Im}(\psi),|F|<+\infty} P(\psi\in F)\leq 1<+\infty,$$

donde, pelo lema acima, $\{v\in\mathrm{Im}(\psi) : P(\psi=v)>0\}$ é enumerável. Como para todo $v\in\mathrm{Im}(\psi)$ vale $P(\psi=v)>0$, segue que $\mathrm{Im}(\psi)=\{v\in\mathrm{Im}(\psi) : P(\psi=v)>0\}$ é enumerável e, em particular, ψ é discreta.

□

Observe que o algoritmo de Moser-Tardos pode não parar. Porém, se ele para, então ele retorna uma configuração $(v_\psi)_{\psi\in\Psi}$ tal que ocorre$(\Omega,\Psi,(v_\psi)_{\psi\in\Psi},A)==0$ para todo $A\in\mathcal{A}$. Pela definição do algoritmo ocorre, temos que $\{\omega\in\Omega : \psi(\omega)=v_\psi\ \forall\psi\in\Psi\}\nsubseteq A$, donde, pelo fato de A ser determinado por Ψ, $\{\omega\in\Omega : \psi(\omega)=v_\psi\ \forall\psi\in\Psi\}\subseteq A^c$. Assim, $\bigcap_{A\in\mathcal{A}}A^c\supseteq\{\omega\in\Omega : \psi(\omega)=v_\psi\ \forall\psi\in\Psi\}$ e $P(\bigcap_{A\in\mathcal{A}}A^c)\geq P(\{\omega\in\Omega : \psi(\omega)=v_\psi\ \forall\psi\in\Psi\}) = P(\bigcap_{\psi\in\Psi}\{\omega\in\Omega : \psi(\omega)=v_\psi\}) = \prod_{\psi\in\Psi}P(\{\omega\in\Omega : \psi(\omega)=v_\psi\})>0$ (pois cada multiplicando é positivo).

Ou seja, o algoritmo, se para, encontra uma configuração das variáveis aleatórias que evita todos os eventos ruins, garantindo, consequentemente, que a probabilidade de nenhum evento ruim ocorrer é positiva.

De agora em diante, salvo menção em contrário, $\Omega,\mathcal{F},P,\Psi,\mathrm{vbl},\mathcal{A}$ são como na entrada do algoritmo de M-T.

Observação 4.14. Suponha que $\mathcal{A}=\{A_x : x\in X\}$.

Defina o grafo $G=(X,E)$ por $E=\{\{x,y\}\subseteq X : x\neq y \text{ e } \mathrm{vbl}(A_x)\cap\mathrm{vbl}(A_y)\neq\emptyset\}$. Então G é um grafo de dependência para a família de eventos (indexados por X) $(A_x)_{x\in X}$

Demonstração. Seja $x\in X$. Mostremos que A_x é mutuamente independente da família de eventos Z $:=\{A_z : z\in X\setminus\Gamma^*(x)\}$. Seja $Y\subseteq Z$. Queremos mostrar que $P(A_x\cap\bigcap_{y\in Y}A_y)=P(A_x)P(\bigcap_{y\in Y}A_y)$. (Note que o caso $Y=\emptyset$ é trivial, então podemos ignorá-lo.) Pela Proposição 4.7 (podemos usar tal proposição porque Ψ é finito e, como vimos na demonstração da Observação 4.12, a imagem de cada $\psi\in\Psi$ é enumerável), temos que cada A_y está em $\sigma(\mathrm{vbl}(A_y))\subseteq\sigma(\bigcup_{y\in Y}\mathrm{vbl}(A_y))$. Analogamente, $A_x\in\sigma(\mathrm{vbl}(A_x))$. Como $\sigma(\bigcup_{y\in y}\mathrm{vbl}(A_y))$ é fechado em relação a intersecções finitas, segue que $\bigcap_{y\in Y}A_y\in\sigma(\bigcup_{y\in y}\mathrm{vbl}(A_y))$, donde, pela Proposição 4.6, $\bigcap_{y\in Y}A_y$ é determinado por $\bigcup_{y\in y}\mathrm{vbl}(A_y)$.

Ponhamos $A:=\bigcap_{y\in Y}A_y$. Temos, pela Proposição 4.5, que

$$A=\bigcup_{j=1}^{N}\{\omega\in\Omega : \psi(\omega)=g_j(\psi)\ \forall\psi\in\bigcup_{y\in Y}\mathrm{vbl}(A_y)\}$$

para certas configurações g_j de $\bigcup_{y\in Y}\mathrm{vbl}(A_y)$ e certo $N\in\{0,1,2,\ldots\}\cup\{\infty\}$. (Note que sabemos que a união é enumerável, porque $\bigcup_{y\in Y}\mathrm{vbl}(A_y)\subseteq\Psi$ é finito e cada $\psi\in\Psi$ tem imagem enumerável). Podemos supor que os g_j são distintos e que, portanto, a união é disjunta. Analogamente, $A_x=\bigcup_{i=1}^{M}\{\omega\in\Omega : \psi(\omega)=f_i(\psi)\ \forall\psi\in\mathrm{vbl}(A_x)\}$, em que os f_i são configurações de $\mathrm{vbl}(A_x)$. Além disso, podemos, novamente, supor f_i distintos e a união, disjunta.

Vamos fazer a prova no caso $N=M=\infty$, já que se lida com os demais casos de maneira semelhante. Usando o seguinte

Lema 4.15. *Se $(x_n)_{n\in\mathbb{N}}, (y_m)_{m\in\mathbb{N}}$ são sequências em $[0,+\infty]$, então $\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} x_n y_m = (\sum_{n=1}^{\infty} x_n)(\sum_{m=1}^{\infty} y_m)$, onde se convenciona, como de costume em Teoria da Medida, que $0\cdot\infty = 0$*

temos que

$$
\begin{aligned}
&P(A_x \cap A)\\
=&P(\bigcup_{i=1}^{\infty}[\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x)] \cap \bigcup_{j=1}^{\infty}[\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y)])\\
=&P(\bigcup_{i=1}^{\infty}\bigcup_{j=1}^{\infty}[\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x)] \cap [\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y)])\\
=&\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} P([\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x)] \cap [\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y)])\\
=&\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\prod_{\psi\in\mathrm{vbl}(A_x)\cup\bigcup_{y\in Y}\mathrm{vbl}(A_y)} P(\psi = f_i(\psi) \text{ se } \psi\in\mathrm{vbl}(A_x), \psi = g_j(\psi) \text{ se } \psi\in\bigcup_{y\in Y}\mathrm{vbl}(A_y))\\
=&\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} P(\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x))P(\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y))\\
=^{4.15}&(\sum_{i=1}^{\infty} P(\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x))(\sum_{j=1}^{\infty} P(\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y)))\\
=&P(\bigcup_{i=1}^{\infty}[\psi = f_i(\psi)\ \forall\psi \in \mathrm{vbl}(A_x)])P(\bigcup_{j=1}^{\infty}[\psi = g_j(\psi)\ \forall\psi \in \bigcup_{y\in Y}\mathrm{vbl}(A_y))\\
=&P(A_x)P(A)
\end{aligned}
$$

□

Essa observação nos diz que o teorema de Moser-Tardos abaixo tem hipóteses no mínimo tão fortes quanto as do LLL com grafos de dependência. O que torna o teorema interessante é que ele nos dá um algoritmo para encontrar um objeto que evita todos os eventos ruins.

Teorema 4.16 (Moser-Tardos)**.** *Suponha que $\mathcal{A} = \{A_x : x \in X\}$ (onde, para evitarmos trivialidades, os A_x são supostos distintos) e que o grafo G é definido como na observação anterior. Se existem $(r_x)_{x\in X}$ em $[0,1)$ tais que, para todo $x \in X$,*

$$P(A_x) \le r_x \prod_{z\in\Gamma(x)} (1 - r_z),$$

então o algoritmo de Moser-Tardos reamostra um evento A_x, em média, no máximo $\mu_x = r_x/(1 - r_x)$ vezes. Em particular, existe uma configuração das variáveis aleatórias $\psi \in \Psi$ tal que nenhum dos eventos ruins A_x ocorre.

Devido à aleatoriedade do algoritmo de Moser-Tardos, vê-se facilmente que o algoritmo pode não parar. Contudo, a tese do Teorema M-T nos diz que a esperança do

número total de reamostragens não ultrapassa $\sum_{x\in X}\mu_x < +\infty$. Isso implica que a probabilidade do número total de reamostragens ser $+\infty$, isto é, de o algoritmo não parar, é zero. Ademais, como a probabilidade do algoritmo parar é 1, temos que o evento "O algoritmo encontra uma configuração de Ψ tal que nenhum evento ruim ocorre" é não-vazio. Daí segue a afirmação que segue a expressão "em particular" no enunciado do Teorema M-T.

Observação 4.17. Nenhum evento ruim é igual a Ω, pois $P(A_x) \leq r_x \prod_{z\in\Gamma(x)}(1-r_z) \leq r_x < 1$, para todo $x \in X$. Além disso, se algum A_x for o evento vazio, então ele nunca será reamostrado, pois nunca ocorrerá.

Observação 4.18. Podemos supor que $r_x \neq 0$ para todo $x \in X$, porque, se existir $y \in X$ tal que $r_y = 0$, então $P(A_y) = 0$, donde, pelo fato de A_y ser determinado por Ψ (e, portanto, ser uma união de eventos-configurações de Ψ) e, por toda configuração de Ψ ter probabilidade positiva de ocorrer, teríamos $A_y = \emptyset$. Observando que o comportamento do algoritmo de M-T em $\{A_x : x \in X\}$ é o mesmo que em $\{A_x : x \in X, x \neq y\}$, exceto pela inicialização das variáveis aleatórias, concluímos que podemos supor que nenhum A_x é vazio e que, portanto, nenhum r_x vale zero.

4.4 Demonstração do Teorema M-T

Observemos, primeiramente, que o que chamamos de "algoritmo de Moser-Tardos" é, na verdade, uma coleção de algoritmos. Isso porque não especificamos o método de escolha do evento a ser reamostrado. Assim, vamos provar, na verdade, que qualquer algoritmo nessa coleção satisfaz a tese do Teorema Moser-Tardos. Doravante, supomos fixado o método de escolha.

Precisaremos de várias definições. Seguiremos [21] e [11].

4.4.1 Definições

Denotemos por N_x a variável aleatória que nos dá o número, em $\mathbb{N}\cup\{0,+\infty\}$, de reamostragens do evento A_x. Ponha $N := \sum_{x\in X} N_x$.

Definimos o registro de uma execução do algoritmo de M-T como a função $C : \{n \in \mathbb{N} : n \leq N\} \to X$ que, para cada número n em seu domínio, nos dá o único x em X tal que o evento A_x foi reamostrado na n-ésima reamostragem.

Todos os grafos neste texto são supostos finitos, salvo menção em contrário. Denotamos por $V(G)$ o conjunto de vértices do grafo G e por $E(G)$ o conjunto de elos do grafo G.

Pela inconsistência da literatura nas definições de caminhos, estabelecemos, aqui, que um *caminho num grafo* G é uma sequência de vértices $(u_1, u_2, \ldots, u_n), n \in \mathbb{N}$, tal que, para todo $i \in \{1, 2, \ldots, n-1\}$, vale que $\{u_i, u_{i+1}\}$ é um elo de G. Dizemos que o caminho é de u_1 a u_n. O caminho é dito *simples* se todos os u_i são distintos (inclusive o primeiro e o último).

Definimos, para cada $n \in \{3, 4, \ldots\}$, C_n como o grafo com conjunto de vértices $\{1, 2, \ldots, n\}$ e com conjunto de elos $\{\{i, i+1\} : i = 1, 2, \ldots, n-1\} \cup \{\{n, 1\}\}$.

Um grafo é *acíclico* se não é verdade que ele possui um subgrafo isomorfo a algum C_n.

Uma *árvore enraizada* é um par $t = (t_0, r)$ em que t_0 é uma árvore (grafo conexo e acíclico) e r é um vértice de t_0 chamado de *raiz de* t. Chamamos de vértices de t os vértices de t_0, pondo $V(t) := V(t_0)$.

Como se sabe, numa árvore, dados quaisquer vértices u, v, existe um único caminho simples de u a v. Em particular, numa árvore enraizada, existe um único caminho de cada vértice u até a raiz r. Se u não é a raiz, então tal caminho não é trivial (isto é, envolve mais de um vértice) e seu segundo vértice é chamado de *pai* de u. Dizemos que u é filho de v se v é pai de u. Dizemos que u é *descendente de* v (e que v *é ancestral de* u) se v é diferente de u e está no caminho simples que liga u à raiz.

Uma *árvore rotulada* é um par $\tau = (t, \sigma)$ em que t é uma árvore enraizada e $\sigma : V(t) \to X$ é uma função. Os elementos de X são denominados rótulos de τ. (Note que o X acima é o conjunto que indexa os eventos ruins. Portanto, a definição de árvore rotulada está amarrada ao contexto dado pelas hipóteses do Teorema M-T.) Pomos $V(\tau) := V(t)$, de modo que os vértices de τ são precisamente os vértices de t.

Uma *árvore testemunha* é uma árvore rotulada (t, σ) tal que, para todo vértice u de t, temos que os filhos de u têm rótulos em $\Gamma^*(\sigma(u))$. A árvore testemunha é dita *própria* se, para todo vértice u, os rótulos dos filhos de u são distintos.

Finalmente, se u é um vértice de uma árvore enraizada ou árvore rotulada, então definimos a *profundidade* de u como $d(u) :=$ "distância de u até a raiz de t". Assim, a profundidade da raiz é zero, a profundidade dos filhos da raiz é 1, etc.

4.4.2 Construindo árvores testemunha a partir do registro C

Para cada $n \in \mathbb{N}$, definiremos uma árvore testemunha $\tau(n)$, associada às n primeiras reamostragens feitas pelo algoritmo.

Definimos, primeiro, $\tau_n(n)$ como a árvore com apenas um vértice (portanto, a raiz) rotulado $C(n)$. Claramente, $\tau_n(n)$ é uma árvore testemunha. Note que, na verdade, tal árvore não é única, pois podemos escolher os nomes de cada vértice arbitrariamente. Para evitar tal problema, escolhemos, para cada classe de equivalência de árvores (dada pela relação de equivalência do isomorfismo de grafos) (note que a classe de equivalência não é um conjunto, isto é, é uma classe própria), uma árvore na classe e supomos que sempre que consideramos uma árvore dessa classe, estamos falando da árvore escolhida.

Agora, para cada $i = n-1, n-2, \ldots, 2, 1$ construímos $\tau_i(n)$ a partir de $\tau_{i+1}(n)$ assim: se existe vértice v de $\tau_{i+1}(n)$ tal que $C(i) \in \Gamma^*(\sigma(v))$, então escolhemos um tal v que maximize $d(v)$ e chamamos de $\tau_i(n)$ a árvore $\tau_{i+1}(n)$ com um vértice filho a mais para v rotulado $C(i)$. Se houver mais de um v maximizando $d(v)$, a escolha é feita de maneira arbitrária (por exemplo, escolhendo o v com o menor valor (podemos fazer isso porque estamos supondo que as árvores tem números naturais como vértices)). Se não existir nenhum v com tal propriedade, pomos $\tau_i(n) := \tau_{i+1}(n)$.

Finalmente, definimos $\tau(n) := \tau_1(n)$. Note que $\tau(n)$ é, de fato, uma árvore testemunha, pois $\tau_n(n)$ é uma árvore testemunha e, pela construção acima, para cada $i = 1, 2, \ldots, n-1$, se $\tau_{i+1}(n)$ é uma árvore testemunha, então $\tau_i(n)$ também é.

Dizemos que uma árvore testemunha τ ocorre no registro C se existe $n \in \mathbb{N}$ tal que $\tau = \tau(n)$.

4.4.3 Propriedades da construção

Lema 4.19. *Se τ é uma árvore testemunha e C é um registro, e se τ ocorre em C, então, para quaisquer vértices u, v com $d(u) = d(v)$, temos que $vbl(A_{\sigma(u)})$ é disjunto de $vbl(A_{\sigma(v)})$. Em particular, os filhos de cada vértice são independentes. Em particular, τ é própria.*

Demonstração. Existe $n \in \mathbb{N}$ tal que $\tau = \tau(n)$. Definamos, para cada vértice w de τ, $i(w)$ como o maior número em $\{1, 2, \ldots, n\}$ tal que w é vértice de $\tau_{i(w)}(n)$. Sejam u, v vértices de τ com $d(u) = d(v)$. Suponhamos, sem perda de generalidade, que $i(u) < i(v)$. Suponhamos, por absurdo, que $\text{vbl}(A_{\sigma(u)}) \cap \text{vbl}(A_{\sigma(v)}) \neq \emptyset$. Então, ao acrescentarmos o vértice u a $\tau_{i(u)+1}$ para construir $\tau_{i(u)}$, nós, conforme a construção descrita anteriormente, colocaríamos u como filho de um vértice w tal que $C(i(u)) \in \Gamma^*(\sigma(w))$ (de modo a maximizar $d(w)$). Ora, o vértice v é tal que $C(i(u)) = \sigma(u) \in \Gamma^*(\sigma(v))$, então temos que $d(u) = d_{\tau_{i(u)}}(u) \geq d_{\tau_{i(u)}}(v) + 1 = d(v) + 1$, uma contradição.

Para ver que os filhos de cada vértice são independentes, observe o seguinte: todos os filhos têm a mesma profundidade em τ, donde, pelo que acabamos de provar, têm vbl's disjuntos. Ora, isso significa que o conjunto $\{\sigma(f) : f \text{ é filho de } u\}$ não tem elos entre seus elementos no grafo de dependência G, o que implica que os eventos correspondentes são independentes (se houver apenas um ou menos eventos, é trivial; caso contrário, escolhemos um desses eventos e usamos o fato de que ele é mutuamente independente da família formada pelo restante dos eventos).

Agora, τ é própria porque dois filhos com mesmo rótulo seriam dependentes: lembre que um evento é independente de si mesmo se, e somente se, tem probabilidade zero ou um. Pela observação 4.17, os rótulos das árvores testemunha construídas nunca correspondem a tais eventos (estamos usando que, como Ψ é um conjunto de variáveis aleatórias independentes com $P(\psi = v) > 0$ para todo $v \in Im(\psi)$ e toda $\psi \in \Psi$, temos que toda configuração de Ψ tem probabilidade positiva e, portanto, pelo fato de todos os eventos ruins serem determinados por Ψ, segue que todos os eventos ruins de probabilidade zero ou um são vazios ou iguais a Ω). □

4.4.4 Controlando a probabilidade de ocorrência de uma árvore testemunha

Lema 4.20. *Se τ é uma árvore testemunha e C é um registro (note que C é aleatório, enquanto τ está fixa), então*

$$P(\tau \text{ ocorrer em } C) \leq \prod_{v \in V(t)} P(A_{\sigma(v)})$$

Demonstração. Defina o seguinte procedimento, chamado τ-check: em ordem decrescente de profundidade, percorra todos os vértices v de τ e gere uma configuração para as variáveis em $\text{vbl}(A_{\sigma(v)})$, de acordo com suas distribuições e de maneira independente de quaisquer outros sorteios, e verifique se o evento $A_{\sigma(v)}$ ocorre. Dizemos que o τ-check passa se $A_{\sigma(v)}$ ocorreu para todos os v.

Note que a probabilidade do τ-check passar é exatamente $\prod_{v \in V(t)} P(A_{\sigma(v)})$.

De maneira intuitiva, o que faremos agora é provar que, em certo sentido, sempre que τ ocorre em C, o τ-check passa. Para tornar isso preciso, primeiro supomos que, antes da execução do algoritmo de M-T e do τ-check, foi gerada, para cada variável $\psi \in \Psi$,

uma sequência infinita $\psi_0, \psi_1, \psi_2, \ldots$ de valores para ψ, de acordo com sua distribuição e de maneira independente.

Agora, supomos que τ-check e o algoritmo de M-T usam essas sequências como fontes para a geração de valores para as variáveis aleatórias: isto é, toda vez que esses algoritmos precisam de novos valores para uma variável aleatória ψ, eles tomam o próximo valor ψ_i não usado.

Vamos mostrar que, sob essas condições (que, como se vê, não alteram o comportamento dos procedimentos), se τ ocorre em C, então o τ-check passa, o que implicará na conclusão do lema.

Suponha que $\tau = \tau(n)$, para certo $n \in \mathbb{N}$. Seja v um vértice de τ. Queremos mostrar que, durante o τ-check, a configuração de $\mathrm{vbl}(A_{\sigma(v)})$ faz o evento $A_{\sigma(v)}$ ocorrer.

Para cada $\psi \in \mathrm{vbl}(A_{\sigma(v)})$, defina $S(\psi) := \{w \in V(\tau) : d(w) > d(v), \psi \in \mathrm{vbl}(A_{\sigma(w)})\}$ Observe que o τ-check, para cada $\psi \in \Psi$, por funcionar em ordem decrescente de profundidade, irá usar um valor novo para ψ para cada vértice em $S(\psi)$ antes de v. Além disso, o τ-check não passará pelos vértices com profundidade menor do que v antes de v. Em princípio, seria possível que o τ-check usasse novos valores para ψ ao passar por vértices de mesma profundidade de v antes de v. Isso não ocorre, porque, pelo Lema 4.19, $\psi \notin \mathrm{vbl}(A_{\sigma(w)})$ para nenhum $w \in V(\tau)$ com $d(w) = d(v)$. Logo, para toda $\psi \in \Psi$, o conjunto de todos os vértices de w de τ tais que o τ-check passa por w antes de v e usa um novo valor para ψ é precisamente $S(\psi)$. Consequentemente, o valor usado pelo τ-check para ψ na checagem de v é $\psi_{|S(\psi)|}$.

Note que, como a $i(v)$-ésima reamostragem feita pelo algoritmo de M-T é do evento $A_{\sigma(v)}$, segue que a configuração das variáveis em $\mathrm{vbl}(A_{\sigma(v)})$, após a amostragem inicial e as $i(v) - 1$ primeiras reamostragens, faz $A_{\sigma(v)}$ ocorrer. Para concluirmos a prova, basta, portanto, mostrarmos que em tais (re)amostragens, para cada $\psi \in \mathrm{vbl}(A_{\sigma(v)})$, o algoritmo de M-T pega valores da fonte para ψ exatamente $|S(\psi)|$ vezes.

Mais precisamente, queremos mostrar que, para toda $\psi \in \mathrm{vbl}(A_{\sigma(v)})$, vale que $|S(\psi)| = 1 + |\{k = 1, 2, \ldots, i(v) - 1 : \psi \in \mathrm{vbl}(A_{C(k)})\}|$. De fato, temos que $|S(\psi)| = |\{w \in V(\tau) : d(w) > d(v), \psi \in \mathrm{vbl}(A_{\sigma(w)})\}| = |\{i(w) : w \in V(\tau), d(w) > d(v), \psi \in \mathrm{vbl}(A_{\sigma(w)})\}| =^* |\{k = 1, 2, \ldots, i(v) - 1, i(v) : \psi \in \mathrm{vbl}(A_{C(k)})\}| = 1 + |\{k = 1, 2, \ldots, i(v) - 1 : \psi \in \mathrm{vbl}(A_{C(k)})\}$

(*) Usamos que os conjuntos de que se toma a cardinalidade são iguais. O da esquerda está contido no da direita: $i(w) < i(v)$, porque não podemos ter $i(w) = i(v)$ pela injetividade de i e não podemos ter $i(w) > i(v)$, porque daí w teria sido inserido antes na construção de τ e, como $\mathrm{vbl}(A_{C(i(w))})$ tem pelo menos um elemento (a saber, ψ) em comum com $\mathrm{vbl}(A_{C(i(v))})$, v seria inserido como filho de w ou de algum vértice com profundidade maior do que w, contradizendo $d(w) > d(v)$. O da direita está contido no da esquerda: seja k elemento do conjunto da direita; note que, na construção de τ, v já havia sido inserido antes de ser considerada a inserção de um vértice correspondente à k-ésima reamostragem do algoritmo de M-T. Como $\psi \in \mathrm{vbl}(A_{C(k)})$, segue que, na construção de $\tau_k(n)$, nós inserimos um vértice w tal que $i(w) = k$, w é filho de v ou de algum vértice com profundidade maior do que v (portanto $d(w) > d(v)$) e o rótulo de w é $\sigma(w) = C(k)$, o que implica que $\psi \in \mathrm{vbl}(A_{\sigma(w)})$.

□

Para cada $x \in X$, denotaremos por τ_x o conjunto de todas as árvores testemunha próprias com raiz rotulada x. Note que τ_x é enumerável, pois $\tau_x = \bigcup_{n \in \mathbb{N}} \tau_x^n$, onde τ_x^n é o conjunto de todas as árvores testemunha próprias com n vértices, e cada τ_x^n é finito.

Assim, a soma que consideramos a seguir é uma série, cuja ordem da soma dos seus termos não importa, já que os somandos são todos não-negativos.

Observação 4.21.

$$N_x = \sum_{\tau \in \tau_x} \mathbb{1}_{\tau \text{ ocorre em } C}$$

Demonstração. ($\leq$) Queremos mostrar que se $n_1 < n_2 \leq N$ ($n_1, n_2 \in \mathbb{N}$) são tais que $C(n_1) = C(n_2)$, então $\tau(n_1) \neq \tau(n_2)$. Ora, basta notar que, para cada $i = 1, 2$, temos que $\tau(n_i)$ possui $|\{n \in \mathbb{N} : n \leq n_i, C(n) = C(n_1)(= C(n_2))\}|$ vértices com rótulo $C(n_1)(= C(n_2))$, sendo que esse número é maior para n_2 do que para n_1.

($\geq$) Claro. □

A observação e o lema acima, juntos, nos permitem controlar a esperança de N_x, conforme veremos a seguir.

4.4.5 Controlando $E(N_x)$ e concluindo a prova do Teorema M-T

Ponhamos, para encurtar a escrita, $r'_x := r_x \prod_{z \in \Gamma(x)}(1 - r_z)$ para cada $x \in X$. Note que, pela última observação, temos que

$$E(N_x) = \sum_{\tau \in \tau_x} P(\tau \text{ ocorrer em } C) \leq^{4.20} \sum_{\tau \in \tau_x} \prod_{v \in V(\tau)} P(A_{\sigma(v)}) \leq^{\text{hipótese}} \sum_{\tau \in \tau_x} \prod_{v \in V(\tau)} r'_{\sigma(v)}$$

Nosso objetivo, agora, é mostrar que a expressão da direita é menor ou igual a $\mu_x = r_x/(1 - r_x)$, o que concluirá a demonstração do teorema de M-T.

Para isso, vamos definir, para cada $x \in X$, um processo de ramificação Galton-Watson para gerar um elemento de τ_x: na primeira etapa, produzimos uma árvore que possui apenas um nó (a raiz) com rótulo x. Na n-ésima etapa ($n \in \{2, 3, \ldots\}$), para cada vértice v gerado na $n-1$-ésima etapa e para cada $y \in \Gamma^*(\sigma(v))$, inserimos um vértice rotulado y como filho de v com probabilidade r_x (e, portanto, com probabilidade $1 - r_x$, nós não inserimos nada). Todos os sorteios são feitos de maneira independente. Note que o processo para se, e somente se, existe $n \in \mathbb{N}$ tal que todos os sorteios nos fazem não inserir os vértices. Se o processo para, ele constrói um elemento de τ_x, como se prova facilmente por indução.

Para relacionar a cota que queremos obter com o processo de ramificação G-W, vamos usar o seguinte

Lema 4.22. *A probabilidade p_τ de que o processo G-W gera $\tau \in \tau_x$ é*

$$p_\tau = \frac{1 - r_x}{r_x} \prod_{v \in V(\tau)} r'_{\sigma(v)}$$

Demonstração. (Note que, pela observação 4.18, sabemos que os denominadores acima não são zero.)

Defina, para cada $v \in V(\tau)$, $W_v := \{y \in \Gamma^*(\sigma(v)) : \text{nenhum filho de } v \text{ tem rótulo } y\}$

Observe que o processo G-W gera τ se, e somente se: para todo $n \in \mathbb{N}$ com $n \geq 2, n \leq$ "número total de etapas no processo (possivelmente infinito)", na n-ésima etapa do processo, para todo v gerado na $n-1$-ésima etapa, geramos todos os elementos de $\Gamma^*(\sigma(v)) \setminus W_v$ e não geramos nenhum elemento de W_v. O "se" é claro; vamos justificar o "somente se": suponha por absurdo que o "somente se" não valha e tome como n o

menor número natural tal que existe um vértice v gerado na $n-1$-ésima etapa tal que algum elemento de $\Gamma^*(\sigma(v)) \setminus W_v$ não é gerado ou algum elemento de W_v é. Note que, para todo $k \geq 2$, os vértices gerados na etapa k são todos de profundidade $k-1$ na árvore gerada final (e também nas árvores intermediárias, claro). Logo, o conjunto de vértices da árvore gerada por G-W com profundidade $n-1$ é diferente do conjunto de vértices de τ com profundidade $n-1$, uma contradição.

Do que acabamos de observar e da independência nos sorteios feitos no processo G-W, temos (onde r é a raiz de τ):

$$\begin{aligned} p_\tau &= \prod_{u \in W_r} (1 - r_{\sigma(u)}) \prod_{v \in V(\tau), v \neq r} (r_{\sigma(v)} \prod_{u \in W_v} (1 - r_{\sigma(u)})) \\ &= \frac{1}{r_x} \prod_{v \in V(\tau)} (r_{\sigma(v)} \prod_{u \in W_v} (1 - r_{\sigma(u)})) \\ &= \frac{1 - r_x}{r_x} \prod_{v \in V(\tau)} (\frac{r_{\sigma(v)}}{1 - r_{\sigma(v)}} \prod_{u \in \Gamma^*(\sigma(v))} (1 - r_{\sigma(u)})) \\ &= \frac{1 - r_x}{r_x} \prod_{v \in V(\tau)} (r_{\sigma(v)} \prod_{u \in \Gamma(\sigma(v))} (1 - r_{\sigma(u)})) \\ &= \frac{1 - r_x}{r_x} \prod_{v \in V(\tau)} r'_{\sigma(v)} \end{aligned}$$

□

Continuando a sequência de cotas superiores para $E(N_x)$ que fizemos no começo desta seção:

$$\begin{aligned} E(N_x) &\leq \sum_{\tau \in \tau_x} \prod_{v \in V(\tau)} r'_{\sigma(v)} = \frac{r_x}{1 - r_x} \sum_{\tau \in \tau_x} p_\tau \\ &= \frac{r_x}{1 - r_x} P(\text{o processo G-W gerar alguma árvore } \tau \in \tau_x) \\ &\leq \frac{r_x}{1 - r_x} = \mu_x \end{aligned}$$

Isso conclui a prova do teorema de M-T (4.16)

Capítulo 5

A região de Shearer

Vimos nos dois capítulos anteriores o LLL clássico e sua versão algorítmica, juntamente com suas respectivas demonstrações. O lema supõe que os eventos ruins respeitam um grafo de dependência e nos fornece as condições $P(A_x) \leq r_x \prod_{z\in\Gamma(x)}(1-r_z)\ \forall x \in X$, suficientes para que evitemos aqueles eventos. Somos naturalmente levados a tentar trocar os números $r_x \prod_{z\in\Gamma(x)}(1-r_z)$ por números maiores, de modo a tornar o lema eficaz para mais eventos.

Definiremos neste capítulo a região de Shearer, que é o limite teórico da região de eficácia do LLL. No próximo capítulo, veremos critérios melhores que o original e suas respectivas versões algorítmicas. Será conveniente fazer uma mudança de variáveis no critério do LLL visto anteriormente para compará-lo com os novos critérios.

5.1 A mudança de variáveis $\mu = r/(1-r)$

O LLL supõe que, para todo $x \in X$, temos a desigualdade

$$P(A_x) \leq r_x \prod_{z\in\Gamma(x)} (1-r_z),$$

para números $(r_x)_{x\in X}$ em $[0,1)$.

Verifica-se facilmente que a função $\mu : [0,1) \to [0,+\infty)$ dada por $\mu(r) := r/(1-r)$ é uma bijeção com inversa $r : [0,+\infty) \to [0,1)$ onde $r(\mu) = \mu/(1+\mu)$. Note que abusamos da notação, usando o mesmo símbolo para as funções e para as variáveis.

Usando as novas variáveis , obtemos o seguinte critério equivalente: "para todo $x \in X$ vale

$$P(A_x) \leq \frac{\mu_x}{\sum_{S\subseteq\Gamma^*(x)} \prod_{z\in S} \mu_z}$$

para números $(\mu_x)_{x\in X}$ em $[0,+\infty)$".

5.2 Reformulando o LLL

A única informação que o LLL precisa é a que está contida no grafo de dependência. Os eventos em si não são importantes. Para tornar isto mais preciso, fazemos a seguinte definição:

Definição 5.1. A *região de Shearer* de uma grafo $G = (X, E)$ é

$$\mathcal{R}(G) := \{p \in [0,1)^X : \text{para qualquer família de eventos } (A_x)_{x\in X} \text{ com } P(A_x) = p_x \ \forall x \in X \\ \text{e para a qual } G \text{ é um grafo de dependência, vale } P(\cap_{x\in X} A_x^c) > 0\}$$

Temos, então, a seguinte reformulação do LLL tradicional (onde já fizemos a mudança de variáveis):

Teorema 5.2 (LLL via região de Shearer)**.** *Seja $G = (X, E)$ um grafo finito. Então*

$$\bigcup_{\mu: X\to[0,+\infty)} \prod_{x\in X} \left[0, \frac{\mu_x}{\sum_{S\subseteq\Gamma^*(x)} \prod_{z\in S} \mu_z}\right] \subseteq \mathcal{R}(G)$$

É natural, daí, definir informalmente que *um* Lema de Lovász deve, dado um grafo G, fornecer uma família de vetores $R = (R_x)_{x\in X}$ de números reais tais que $\prod_{x\in X}[0, R_x) = [0, R) \subseteq \mathcal{R}(G)$. No caso do lema acima, temos, para cada $\mu : X \to [0, +\infty)$, um vetor $R = (R_x)_{x\in X}$ dado por $R_x = \dfrac{\mu_x}{\sum_{S\subseteq\Gamma^*(x)} \prod_{z\in S} \mu_z}$.

Assim, queremos encontrar um critério que preencha a maior parte possível da região de Shearer.

Capítulo 6

Os novos critérios e suas versões algorítmicas

6.1 Os critérios BFPS e de Temmel

Explorando uma conexão profunda entre o LLL e o Gás de Rede, é possível traduzir resultados sobre este para resultados sobre aquele. (Para mais detalhes, ver [13].)

Por este caminho, em 2011, Bissacot, Fernández, Procacci e Scoppola ([6]) melhoraram o critério do LLL original, com o seguinte teorema:

Teorema 6.1 (LLL BFPS)**.** *Seja $G = (X, E)$ um grafo finito. Então*

$$\bigcup_{\mu: X \to [0,+\infty)} \prod_{x \in X} \left[0, \frac{\mu_x}{\sum_{S \subseteq \Gamma^*(x) \text{ independente}} \prod_{z \in S} \mu_z} \right] \subseteq \mathcal{R}(G)$$

Similarmente, em 2012, Temmel ([16]) obteve uma melhora do critério BFPS. Para escrever o critério de modo compacto, introduzimos a seguinte notação:

Notação 6.2. Se $\mu : X \to [0, +\infty)$ e $T \subseteq X$, então

$$\Xi_T(\mu) := \sum_{S \subseteq T \text{ independente}} \prod_{z \in S} \mu_z$$

onde entendemos que um conjunto $S \subseteq X$ é *independente* se não há elos $ij \in E$ com $i, j \in S$.

Teorema 6.3 (LLL de Temmel)**.** *Seja $G = (X, E)$ um grafo finito. Então*

$$\bigcup_{\mu: X \to [0,+\infty)} \prod_{x \in X} \left[0, \max\{ \frac{\mu_x}{\Xi_{\Gamma^*(x)}(\mu)}, \frac{\mu_x}{(1+\mu_x) \max_{k \in \Gamma(x)} \Xi_{\Gamma^*(x) \setminus \{k\}}(\mu)} \} \right) \subseteq \mathcal{R}(G)$$

Note que, de fato, cada critério é uma melhora do anterior (exceto pela troca de intervalos fechados por semi-abertos), pois

$$\begin{aligned} \frac{\mu_x}{\sum_{S \subseteq \Gamma^*(x)} \prod_{z \in S} \mu_z} &\leq \frac{\mu_x}{\sum_{S \subseteq \Gamma^*(x) \text{ independente}} \prod_{z \in S} \mu_z} = \frac{\mu_x}{\Xi_{\Gamma^*(x)}(\mu)} \\ &\leq \max\{ \frac{\mu_x}{\Xi_{\Gamma^*(x)}(\mu)}, \frac{\mu_x}{(1+\mu_x) \max_{k \in \Gamma(x)} \Xi_{\Gamma^*(x) \setminus \{k\}}(\mu)} \} \end{aligned}$$

vale para qualquer função $\mu : X \to [0, +\infty)$.

6.2 Algoritmizando os novos critérios

Uma pergunta natural é se o algoritmo de Moser-Tardos também irá parar (com probabilidade 1) sob as hipóteses, mais fracas, de BFPS e de Temmel. Para o primeiro critério, Pedgen ([12]) obteve uma resposta afirmativa, com uma demonstração muito parecida com a de Moser e Tardos que vimos anteriormente:

Teorema 6.4 (LLL BFPS algorítmico)**.** *Seja $\mathcal{A} = \{A_x : x \in X\}$ uma família finita de eventos distintos determinados por um conjunto finito Ψ de variáveis aleatórias (discretas) independentes, com $P(\psi = v) > 0 \ \forall v \in Im(\psi) \ \forall \psi \in \Psi$. Defina o grafo de dependência $G = (X, E)$ para $(A_x)_{x\in X}$ pondo, para $x \neq y$, $xy \in E \iff vbl(A_x) \cap vbl(A_y) \neq \emptyset$.*

Se uma função $\mu : X \to [0, +\infty)$ é tal que, para todo $x \in X$,

$$P(A_x) \leq \frac{\mu_x}{\Xi_{\Gamma^*(x)}(\mu)},$$

então o algoritmo de Moser-Tardos reamostra um evento A_x, em média, no máximo μ_x vezes. Em particular, existe uma configuração das variáveis aleatórias $\psi \in \Psi$ que evita os eventos ruins A_x.

Rogério Alves e Procacci([21], [22]) obtiveram um resultado mais geral, que nos permite dar uma resposta afirmativa para o questão relativa ao critério de Temmel:

Teorema 6.5 (Rogério Alves e Procacci)**.** *Seja $\mathcal{A} = \{A_x : x \in X\}$ uma família finita de eventos distintos determinados por um conjunto finito Ψ de variáveis aleatórias (discretas) independentes, com $P(\psi = v) > 0 \ \forall v \in Im(\psi) \ \forall \psi \in \Psi$. Defina o grafo de dependência $G = (X, E)$ para $(A_x)_{x\in X}$ pondo, para $x \neq y$, $xy \in E \iff vbl(A_x) \cap vbl(A_y) \neq \emptyset$.*

Se $(P(A_x))_{x\in X} \in \mathcal{R}(G)$, então o algoritmo de Moser-Tardos para com probabilidade 1. Em particular, existe uma configuração das variáveis aleatórias $\psi \in \Psi$ que evita os eventos ruins A_x.

Juntando o resultado de Rogério e Procacci com o de Temmel, obtemos o seguinte:

Teorema 6.6 (LLL de Temmel algorítmico)**.** *Seja $\mathcal{A} = \{A_x : x \in X\}$ uma família finita de eventos distintos determinados por um conjunto finito Ψ de variáveis aleatórias (discretas) independentes, com $P(\psi = v) > 0 \ \forall v \in Im(\psi) \ \forall \psi \in \Psi$. Defina o grafo de dependência $G = (X, E)$ para $(A_x)_{x\in X}$ pondo, para $x \neq y$, $xy \in E \iff vbl(A_x) \cap vbl(A_y) \neq \emptyset$.*

Se uma função $\mu : X \to [0, +\infty)$ é tal que, para todo $x \in X$,

$$P(A_x) < \max\{\frac{\mu_x}{\Xi_{\Gamma^*(x)}(\mu)}, \frac{\mu_x}{(1+\mu_x)\max_{k\in\Gamma(x)} \Xi_{\Gamma^*(x)\setminus\{k\}}(\mu)}\}$$

então o algoritmo de Moser-Tardos termina com probabilidade 1. Em particular, existe uma configuração das variáveis aleatórias $\psi \in \Psi$ que evita os eventos ruins A_x.

Capítulo 7

Conclusão

Apresentamos, neste texto, o Lema Local de Lovász, uma ferramenta do método probabilístico usada para provar a existência de objetos matemáticos que evitam um conjunto finito de eventos ruins. Vimos sua versão original, com o critério

$$P(A_x) \leq r_x \prod_{z \in \Gamma(x)} (1 - r_z) \; \forall x \in X,$$

onde os A_x são os eventos ruins e os r_x são números arbitrários no intervalo $[0, 1)$.

Vimos que, se supusermos que os eventos ruins são determinados por um conjunto finito de variáveis aleatórias discretas independentes, o algoritmo de Moser-Tardos encontra uma configuração que os evita com probabilidade 1. De fato, o algoritmo termina em tempo esperado no máximo $\sum_{x \in X} \mu_x < +\infty$, onde $\mu_x = r_x/(1 - r_x) \in [0, +\infty)$. O algoritmo é muito simples: ele sorteia um valor para cada v.a. de acordo com sua distribuição; depois, enquanto existir evento ruim ocorrendo, ele sorteia novos valores para as v.a.'s de que o evento depende. Os sorteios, naturalmente, são todos feitos de modo independente.

Nos dois últimos capítulos, estudamos o limite teórico do lema, dado pela região de Shearer, e dois critérios que melhoram o lema, BFPS e Temmel. Também enunciamos um resultado de Rogério Alves e Procacci que nos permitiu provar que o algoritmo de Moser-Tardos também para com probabilidade 1 sob as hipóteses dos novos critérios.

A prova fez uso dos resultados de Rogério Alves e Procacci e do próprio resultado de Temmel. Uma questão a se investigar é a possibilidade de se dar uma prova puramente combinatória, como a que vimos para o LLL original no capítulo 4.

Referências Bibliográficas

[1] D. Achlioptas and T. Gouleakis: *Algorithmic Improvements of the Lovász Local Lemma via Cluster Expansion*. Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS, 16-23, (2012).

[2] N. Alon and J. Spencer: The Probabilistic Method. Second Edition. Wiley-Interscience. New York (2003).

[3] Böttcher, J. ; Kohayakawa, Y. and Procacci, A. *Properly coloured copies and rainbow copies of large graphs with small maximum degree.* Random Structures & Algorithms, **40**, n. 4, 425-436, (2012).

[4] R. Bissacot.: *Técnicas para convergência da Expansão do Gás de Polímeros e uma aplicação ao Método Probabilístico*. PhD thesis, UFMG, (2009).

[5] R. Bissacot.: *O Lema Local de Lovász: do Método Mágico de Erdös à Teoria dos Gases de Rede*. Minicurso ministrado no II Colóquio de Matemática da região Sul. (2012)

[6] Bissacot, R., Fernández, R., Procacci, A. and Scoppola, B.: *An Improvement of the Lovász Local Lemma via Cluster Expansion.* Combinatorics, Probability and Computing. Vol **20** Issue 5, 709-719, (2011).

[7] Ndreca, S. ; Procacci, A. ; Scoppola, B. *Improved bounds on coloring of graphs.* European Journal of Combinatorics. Vol. **33** Issue 4, 592-609, (2012).

[8] Erdös, P. and Lovász, L.: *Problems and results on 3-chromatic hypergraphs and some related questions, in Infinite and finite sets*. Vol. II, Colloq. Math. Soc. Janos Bolyai, Vol. 10. (North-Holland, Amsterdam), (1975).

[9] Fernandez, R.; Procacci A.: *Cluster expansion for abstract polymer models.New bounds from an old approach*. Comm. Math. Phys. **274**, n.1, 123-140 (2007).

[10] T. G. Gouleakis: *Algorithmic aspects of the Lovasz Local Lemma*. Thesis. National Technical University of Athens, (2011).

[11] R. Moser and G. Tardos. *A constructive proof of the general Lovász Local Lemma*. Journal of the ACM, 57(2):1-15, (2010).

[12] Pegden, W.: *An improvement of the Moser-Tardos algorithmic local lemma*. SIAM J. Discrete Math, (2011).

[13] Scott, A.; Sokal, A.: *The repulsive lattice gas, the independent-set polynomial, and the Lovász local lemma*. J. Stat. Phys. **118**, no. 5-6, 1151–1261(2005).

[14] C. Temmel.: *Properties and applications of Bernoulli random Fields with strong dependency graphs*. PhD thesis, (2012).

[15] C. Temmel.: *Shearer's Measure and Stochastic Domination of Product Measures*. Journal of Theoretical Probability, (2012).

[16] C. Temmel.: *Sufficient conditions for uniform bounds in abstract polymer systems and explorative partition schemes*. Journal of Statistical Physics. (2014).

[17] R. Rajaraman (scribed by Eric Miles): *Lecture notes for CS 7880 Algorithmic Power Tools, Fall 2009* Disponível em `http://www.ccs.neu.edu/home/rraj/Courses/7880/F09/Lectures/GeneralLLL_Apps.pdf`

[18] J. Gao, X. Pérez-Giménez, T. Sauerwald: *Probabilistic Method: Lovasz Local Lemma* Disponível em `http://www.mpi-inf.mpg.de/departments/d1/teaching/ss11/ProbMethod/files/lll.pdf`

[19] A. Srinivasan (scribed by Ioana Bercea): *Lecture: New Constructive Aspects of the Lovász Local Lemma, and their Applications* Disponível em `http://www.cs.princeton.edu/~zdvir/apx11slides/Srinivasan_scribe.pdf`

[20] D. Knuth: *A (very incomplete) draft of section 7.2.2.2: Satisfiability.* The Art of Computer Programming, volume 4, pre-fascicle 6A

[21] Alves, R. G: *Duas aplicações da Mecânica Estatística: Percolação em Grafos Infinitos e Lema Local de Lovász Algorítmico*. Tese - Universidade Federal de Minas Gerais, 2013.

[22] Alves, R. G.; Procacci, A.: "Witness trees in the Moser-Tardos algorithmic Lovász Local Lemma and Penrose trees in the hard-core lattice gas", Journal of Statistical Physics, 156, p. 877-895 (2014)

[23] Rolla, Leonardo T.: *Introdução à Probabilidade*. Disponível em `http://mate.dm.uba.ar/~leorolla/papers/intro-probab.pdf`