1 | Uma introdução à probabilidade e aos algoritmos aleatorizados

Intuitivamente, uma medida de probabilidade é uma forma quantitativa de expressar a chance com que um conjunto de resultados de um experimento que nos fornece resultados que não podem ser previamente determinados ocorra. Probabilidade é a disciplina dedicada à modelagem desses fenômenos com condições de incerteza.

Neste capítulo introduzimos o tratamento axiomático moderno da Probabilidade introduzido pelo matemático russo Andrei Nikolaevich Kolmogorov (1903-1987) por volta de 1930. Ao contrário das interpretações que estabelecem uma forma explícita de calcular probabilidades, o modelo axiomático estuda as propriedades que uma medida de probabilidade deve satisfazer. Veremos as noções importantes de probabilidade condicional e a independência de eventos.

Também introduzimos uma discussão informal sobre modelos de computação, análise de algoritmos e alguns exemplos de algoritmos que usam sorteios em suas computações. Uma definição precisa de algoritmo, que adiamos, é importante para entender os processos computacionais, conhecer seus limites e estabelecer sua eficiência na resolução de problemas.

1.1 ESPAÇO DE PROBABILIDADE DISCRETO

Um modelo probabilístico é um modelo matemático de um experimento aleatório. A modelagem probabilística tem sido importante em praticamente todas as áreas do conhecimento e o desenvolvimento da Teoria da Probabilidade tem sido estimulada pela ampla variedade de suas aplicações.

Exemplo 1.1. Monty Hall é o nome do apresentador de um concurso televisivo que era exibido na década de 1970, nos Estados Unidos, chamado *Let's Make a Deal*, agora é o nome de um problema clássico em probabilidade. O jogo consistia em o apresentador Monty Hall apresentar três portas a um espectador, esse concorre a um prêmio escondido atrás da porta. O processo de escolha será descrito a seguir.

O protocolo da brincadeira é: Monty Hall escolhe, ao acaso, uma das portas para esconder um carro; nas outras duas esconde um bode em cada. Na primeira etapa o concorrente escolhe uma porta ao acaso (que ainda não é aberta); em seguida Monty Hall abre uma das outras duas portas que o concorrente não escolheu, sabendo que ela esconde um bode e escolhendo ao acaso se houver mais de uma possibilidade. Com duas portas fechadas apenas, e sabendo que o carro está atrás de uma delas, o apresentador oferece ao concorrente a oportunidade de trocar de porta. O concorrente tem que decidir se permanece com a porta que escolheu no início do jogo ou se muda para a outra porta que ainda está fechada; feita a escolha, o apresentador abre

a porta escolhida e o concorrente leva o prêmio escondido pela porta. Assumindo que o objetivo do jogador é ganhar o carro, o problema é determinar uma estratégia de decisão que maximiza a chance de ganhar o carro. ◊

A resposta para esse problema será dada mais a frente no texto, no momento convidamos o leitor a refletir um pouco sobre o problema e identificar os experimentos aleatórios escondidos na descrição feita no parágrafo acima.

Um modelo probabilístico para um experimento aleatório é caracterizado por um *espaço amostral* — conjunto dos resultados possíveis — um *espaço de eventos* — família[1] dos subconjuntos de resultados que admitem uma probabilidade — e uma *(medida de) probabilidade* — uma função que associa um valor numérico a cada evento.

Espaço amostral O espaço amostral de um experimento aleatório, quase sempre denotado por Ω, é um conjunto não vazio que representa todos os resultados possíveis de um experimento de modo que resultados diferentes tenham representantes diferentes no conjunto. Um elemento de Ω é chamado de **ponto amostral** e a escolha de algum ponto amostral representa uma realização do experimento.

Exemplo 1.2. São experimentos com respectivos espaços amostrais

1. um dado é lançado e observamos a face para cima, $\Omega = \{1,2,3,4,5,6\}$;
2. uma moeda é lançada e observamos sua face para cima, $\Omega = \{\mathrm{Ca}, \mathrm{Co}\}$;
3. uma moeda é lançada até sair coroa, $\Omega = \{(\mathrm{Co}), (\mathrm{Ca},\mathrm{Co}), \ldots, (\mathrm{Ca},\mathrm{Ca},\ldots,\mathrm{Ca},\mathrm{Co}), \ldots, (\mathrm{Ca},\mathrm{Ca},\ldots)\}$, cada ponto amostral é representado por uma sequência de Ca que, eventualmente, termina com Co;
4. uma moeda é lançada sucessivamente e pergunta-se o que ocorre primeiro, uma sequência de três caras ou três coroas consecutivas. Um espaço amostral é considerar todas as sequências $(a_i : i \geqslant 0)$, com $a_i \in \{\mathrm{Ca}, \mathrm{Co}\}$ para todo i, de resultados possíveis, isto é, $\Omega = \{\mathrm{Ca}, \mathrm{Co}\}^{\mathbb{N}}$;
5. observamos tempo de vida de uma lâmpada (em alguma uniade de tempo), $\Omega = \{t \in \mathbb{R} : t \geqslant 0\}$;
6. um dardo é lançado num alvo circular de raio 1 e observamos o ponto atingido. Uum espaço amostral é obtido usando um sistema de coordenadas cartesianas com a origem no centro do alvo de modo que $\Omega = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 \leqslant 1\}$.

[1] Família é usado como sinônimo de conjunto.

O espaço amostral de um experimento aleatório reflete a observação do resultado de um experimento e não é único. No item 6 do exemplo acima, podemos escrever o espaço amostral com pontos dados em coordenadas polares $\{(r,\theta) \in \mathbb{R}^2 : 0 \leqslant r \leqslant 1 \text{ e } 0 \leqslant \theta < 2\pi\}$. ◊

Exercício 1.3. Identifique os experimentos aleatórios e descreva um espaço amostral para o problema de Monty Hall descrito no Exemplo 1.1.

Espaço de eventos Intuitivamente, um evento de um experimento aleatório é um acontecimento observável ao final da realização do experimento. Quando o experimento é realizado, deve ser sempre possível determinar se tal evento ocorreu ou não. Por exemplo, se um dado é lançado, os resultados "o número é par" e "o número é maior que 3" são eventos do experimento de lançar um dado.

Um evento é representado, no modelo probabilístico, por um subconjunto $A = A(\Omega)$ do espaço amostral Ω, o qual fica definido pela coleção de resultados possíveis do experimento que descrevem o evento. No exemplo do dado, se $\Omega = \{1,2,3,4,5,6\}$, então "é um número par" e "é um número maior que 3" são modelados por $\{2,4,6\}$ e $\{4,5,6\}$, respectivamente. Assim, um modelo de um evento aleatório é subconjunto do espaço amostral Ω e também é chamado de **evento aleatório**.

Na realização de um experimento, o evento $A \subseteq \Omega$ *ocorre* se o resultado observado é representado por um elemento de A, caso contrário o evento A *não ocorre*. Em especial, $\varnothing$ é o evento *impossível*; Ω é o evento *certo*; $\{\omega\}$ é um evento *elementar*, para cada elemento $\omega \in \Omega$; o *complemento* do evento A é o evento *não*-A dado por

$$\overline{A} = \Omega \setminus A := \{\omega \in \Omega : \omega \notin A\}.$$

Em um lançamento de dados temos $\Omega = \{1,2,3,4,5,6\}$ e são exemplos de eventos

- $A = \{2,4,6\}$, ou seja, A representa o evento "número par";
- $\overline{A} = \{1,3,5\}$, ou seja, $\overline{A}$ representa o evento "não é número par";
- $A \cap \overline{A} = \varnothing$, ou seja, $A \cap \overline{A}$ representa o evento "número par *e* número ímpar", que é o evento impossível;
- $A \cup \overline{A} = \Omega$, ou seja, $A \cup \overline{A}$ representa o evento "número par *ou* número ímpar", que é o evento certo;
- o evento "múltiplo de 2 *ou* múltiplo de 3 mas não múltiplo de ambos" é representado pela diferença simétrica $\{2,4,6\} \triangle \{3,6\} = (\{2,4,6\} \cup \{3,6\}) \setminus (\{2,4,6\} \cap \{3,6\}) = \{2,3,4\}$.

Dizemos que A e B são eventos **disjuntos** ou **eventos mutuamente exclusivos** quando eles não têm elementos em comum, isto é, $A \cap B = \varnothing$. Os eventos $A_1, A_2, \ldots, A_n$ são ditos **mutuamente exclusivos** se são disjuntos tomados dois-a-dois, isto é, $A_i \cap A_j = \varnothing$ sempre que $i \neq j$. Embora eventos sejam conjuntos e a Teoria dos Conjuntos tem uma linguagem tradicional e bem aceita a Probabilidade tem um linguagem particular para os eventos (veja a Tabela 1.1).

Notação	*Eventos*	*Conjunto*
Ω	espaço amostral, evento certo	universo
$\varnothing$	evento impossível	vazio
$\{\omega\}$	evento elementar	conjunto unitário
A	evento	subconjunto
A	ocorre A	$\omega \in A$
$\overline{A}$	não ocorre A	$\omega \notin A$ (complemento)
$A \cap B$	ocorre A e B	$\omega \in A \cap B$ (intersecção)
$A \cup B$	ocorre A ou B	$\omega \in A \cup B$ (união)
$A \setminus B$	ocorre A e não ocorre B	$\omega \in A$ e $\omega \notin B$ (diferença)
$A \triangle B$	ocorre A ou B, não ambos	$\omega \in A \cup B$ e $\omega \notin A \cap B$ (diferença simétrica)
$A \subseteq B$	se ocorre A, então ocorre B	$\omega \in A \Rightarrow \omega \in B$ (inclusão)

Tabela 1.1: "dicionário" de termos da Probabilidade.

Denotemos por $\mathcal{A}$ um conjunto de eventos aleatórios que podem ocorrer num experimento aleatório. Para ser consistente com a intuição, $\mathcal{A}$ deve conter os eventos $\varnothing$ e Ω entre seus elementos e ser fechado para as operações usuais de conjunto. Além disso, pedimos que satisfaça o seguinte: se $A_i \in \mathcal{A}$ para todo inteiro $i \geqslant 1$, então $\bigcup_{i \geqslant 1} A_i \in \mathcal{A}$ e uma justificativa para isso é dada adiante.

Um **espaço de eventos** é um conjunto $\mathcal{A}$ de eventos aleatórios de um experimento aleatório. Quais são as famílias de subconjuntos de Ω que podem ser tomadas como espaço de eventos é um assunto que não trataremos. Uma escolha óbvia é o conjunto 2^Ω das partes de Ω, mas em muitos casos é preciso restringir essa família a um subconjunto próprio de 2^Ω para que questões probabilísticas façam sentido. Por ora, apenas destacamos que é possível haver subconjuntos de um espaço amostral Ω que não são eventos aleatórios, como é o caso dado no Exemplo 1.12 adiante, na página 16. Esse fenômeno é importante quando Ω é muito grande (não enumerável).

Exercício 1.4. Descreva, de acordo com a solução dada no Exercício 1.3, o evento de interesse no problema de Monty Hall, isto é, o subconjunto que modela o evento "o

espectador concorrente ganha o carro".

Medida de probabilidade Uma medida de probabilidade sobre um espaço de eventos $\mathcal{A}$ de um espaço amostral Ω é uma função, genericamente denotada por $\mathbb{P}$, que atribui a cada evento aleatório um número real satisfazendo

Não-negatividade $\mathbb{P}(A) \geqslant 0$ para todo $A \in \mathcal{A}$;

Normalização $\mathbb{P}(\Omega) = 1$;

Aditividade enumerável $\mathbb{P}\big(\bigcup_{i\geqslant 1} A_i\big) = \sum_{i\geqslant 1} \mathbb{P}(A_i)$ sempre que $\{A_i : i \geqslant 1\}$ é um conjunto enumerável de eventos mutuamente exclusivos.[2]

As primeiras consequências importantes dessa definição são enunciadas na proposição a seguir.

Proposição 1.5 *Seja $\mathbb{P}$ é uma medida de probabilidade sobre um espaço de eventos de Ω.*

1. *A probabilidade do evento impossível é $\mathbb{P}(\varnothing) = 0$.*
2. *Aditividade finita: se $A_1, A_2, \ldots, A_n$ são eventos mutuamente exclusivos então*
$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \mathbb{P}(A_i).$$
3. *A probabilidade do complemento é $\mathbb{P}(\overline{A}) = 1 - \mathbb{P}(A)$, para todo evento A.*
4. *Monotonicidade: se $A \subseteq B$ então $\mathbb{P}(A) \leqslant \mathbb{P}(B)$. Também, $\mathbb{P}(B \setminus A) = \mathbb{P}(B) - \mathbb{P}(A)$.*
5. *Regra da Adição: $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$ para quaisquer eventos A e B.*

Demonstração. Fazendo $A_1 = \Omega$ e $A_i = \varnothing$ para todo $i \geqslant 2$ temos, pela aditividade enumerável, que

$$\mathbb{P}(\Omega) = \mathbb{P}(\Omega \cup \varnothing \cup \varnothing \cup \cdots \cup \varnothing \cup \cdots) = \mathbb{P}(\Omega) + \sum_{i\geqslant 2} \mathbb{P}(\varnothing)$$

portanto, pela não-negatividade, resta que $\mathbb{P}(\varnothing) = 0$. Agora, definindo $A_i = \varnothing$ para todo $i > n$

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \mathbb{P}\left(\bigcup_{i\geqslant 1} A_i\right) = \sum_{i\geqslant 1} \mathbb{P}(A_i) = \sum_{i=1}^{n} \mathbb{P}(A_i) + \sum_{i>n} \mathbb{P}(\varnothing) = \sum_{i=1}^{n} \mathbb{P}(A_i)$$

[2] O lado esquerdo da igualdade não depende de uma enumeração particular dos conjuntos A_i e, nesse caso, o mesmo vale para o lado direito, veja (s.1) do apêndice.

que é o resultado afirmado no item 2.

A probabilidade da complemento segue do item anterior e da normalização. Os detalhes ficam a cargo do leitor.

Para monotonicidade, consideremos A e B eventos tais que $A \subseteq B$. Usamos que B pode ser escrito como a união disjunta $A \cup (B \setminus A)$, donde $\mathbb{P}(B) = \mathbb{P}(A) + \mathbb{P}(B \setminus A)$ e como $\mathbb{P}(\overline{A} \cap B) \geqslant 0$ temos $\mathbb{P}(B) \geqslant \mathbb{P}(A)$. Notemos que, como consequência imediata, temos $\mathbb{P}(A) \leqslant 1$ para todo evento A.

Finalmente, a união $A \cup B$ pode ser escrita como duas uniões disjuntas $(A \setminus B) \cup (B \setminus A) \cup (A \cap B)$ donde concluímos que

$$\mathbb{P}(A \cup B) = \mathbb{P}(A \setminus B) + \mathbb{P}(B \setminus A) + \mathbb{P}(A \cap B). \tag{1.1}$$

Agora, A pode ser escrito como a união disjunta $(A \setminus B) \cup (A \cap B)$ e, analogamente, $B = (B \setminus A) \cup (A \cap B)$, portanto $\mathbb{P}(A) = \mathbb{P}(A \setminus B) + \mathbb{P}(A \cap B)$ assim como $\mathbb{P}(B) = \mathbb{P}(B \setminus A) + \mathbb{P}(A \cap B)$. Isolando $\mathbb{P}(A \setminus B)$ e $\mathbb{P}(B \setminus A)$ nessas duas igualdades e substituindo na equação (1.1) prova a Regra da Adição. □

O seguinte limitante é bastante útil e pode ser facilmente provado usando indução e a Regra da Adição.

Corolário 1.6 (Subaditividade) *Se $A_1, A_2, \ldots, A_n$ são eventos então*

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) \leqslant \sum_{i=1}^{n} \mathbb{P}(A_i).$$

Exemplo 1.7 (*lançamento de uma moeda equilibrada*). O modelo probabilístico para o lançamento de uma moeda equilibrada é $\Omega = \{\mathrm{Ca}, \mathrm{Co}\}$ e $\mathbb{P}(\varnothing) = 0$, $\mathbb{P}(\{\mathrm{Ca}\}) = \mathbb{P}(\{\mathrm{Co}\}) = 1/2$, $\mathbb{P}(\Omega) = 1$. ♢

Exemplo 1.8 (*lançamento de um dado equilibrado*). No caso do lançamento de um dado equilibrado atribuímos a probabilidade $1/6$ a cada uma das faces, o que é interpretado como todas as faces serem equiprováveis. A partir disso qualquer subconjunto $A \subseteq \Omega$ de faces do dado é um evento que tem probabilidade de ocorrência dada por

$$\mathbb{P}(A) = \frac{|A|}{6}$$

de modo que os axiomas de probabilidade ficam satisfeitos. ♢

Nesses dois exemplos vimos o modo clássico de interpretar probabilidade no caso finito, os eventos elementares são equiprováveis e nos referimos a eles como uma "escolha aleatória", ou uma "ocorrência ao acaso". Nos espaços amostrais infinitos esse não é o caso, pode não haver uma interpretação viável ou podem haver mais de um modo natural de definir probabilidade para o significado intuitivo clássico.

O próximo exemplo é conhecido como o paradoxo de Bertrand. A rigor não é um paradoxo, mas apresenta a possibilidade de mais de uma interpretação para "ao acaso", ou "aleatório", e que levam a resultados diferentes.

Exemplo 1.9 (*Paradoxo de Bertrand*). Qual é a probabilidade de que uma corda AB escolhida ao acaso numa circunferência de raio 1 tenha comprimento maior que $\sqrt{3}$? Numa circunferência de raio 1, um triângulo equilátero inscrito tem lado $\sqrt{3}$ (Figura 1.1). Na primeira interpretação a escolha da corda se dá ao tomarmos A e B

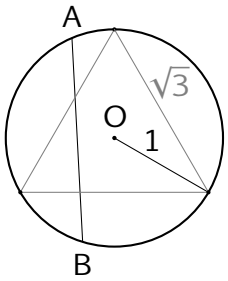


Figura 1.1: paradoxo de Bertrand.

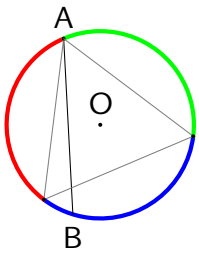
escolhidos ao acaso dentre os pontos da circunferência. Consideremos o triângulo rotacionado de modo que um de seus vértices coincida com o ponto A. A corda tem comprimento maior que o lado do triângulo se B está no arco da circunferência entre os outros dois vértices do triângulo, o que ocorre com probabilidade 1/3, pois os vértices dividem a circunferência em três arcos de mesmo comprimento (Figura 1.2(a)).

Na segunda interpretação, a corda é obtida por uma escolha de P no interior da circunferência e AB é a corda cujo ponto médio é P (Figura 1.2(b)). A corda é maior que o lado do triângulo se P está no interior da circunferência de centro O e raio 1/2, o que ocorre com probabilidade 1/4.

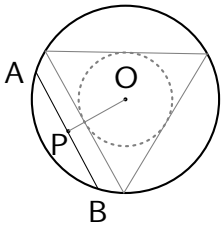
Na terceira é última interpretação para uma corda aleatória, fixamos um raio. A corda é obtida escolhendo um ponto P no raio e tomando a corda que passa por P e perpendicular ao raio (Figura 1.2(c)). A corda é maior do que um lado do triângulo, se o ponto escolhido está mais próximo do centro do círculo, que o ponto onde o lado do triângulo intersecta o raio, logo se $|OP| \in (0, 1/2)$ o que ocorre com probabilidade 1/2. ◊

Exemplo 1.10. Quando escolhemos um inteiro positivo, com a probabilidade de escolher i dada por $(1/2)^i$, e estendemos a probabilidade a qualquer subconjunto A de inteiros positivos pondo

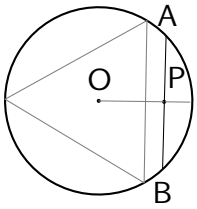
$$\mathbb{P}(A) := \sum_{a \in A} \mathbb{P}(\{a\}) \tag{1.2}$$



(a) a corda é dada por uma escolha aleatória de A e de B na circunferência. A probabilidade procurada é 1/3.



(b) a corda AB é definida por P, seu ponto médio. A probabilidade procurada é 1/4.



(c) a corda é dada pela escolha de um raio e pela escolha de um ponto P nesse raio. A probabilidade procurada é 1/2.

Figura 1.2: as três interpretações do paradoxo de Bertrand.

temos um modelo probabilístico. De fato, temos (veja (s.6a) do apêndice)

$$\mathbb{P}(\Omega) = \sum_{i\geqslant 1}\left(\frac{1}{2}\right)^i = 1$$

e a convergência absoluta dessa série implica que toda subsérie dela é convergente (veja (s.4) do apêndice), assim temos que a probabilidade dada na equação (1.2) está bem definida, isto é, $\mathbb{P}(A)$ como definido acima é um número real não negativo menor ou igual a 1. Também segue da convergência absoluta que um rearranjo da série resulta noutra série que converge para o mesmo resultado, donde obtemos a aditividade enumerável da medida de probabilidade,

$$\mathbb{P}\left(\bigcup_{i\geq 1} A_i\right) = \sum_{a\in\bigcup_i A_i}\mathbb{P}(\{a\}) = \sum_{i\geq 1}\sum_{a\in A_i}\mathbb{P}(\{a\}) = \sum_{i\geq 1}\mathbb{P}(A_i) \tag{1.3}$$

para qualquer conjunto enumerável $\{A_i : i \geqslant 1\}$ de eventos mutuamente exclusivos.

Nesse espaço, a probabilidade de escolher um número par é

$$\sum_{a \text{ par}}\mathbb{P}(\{a\}) = \sum_{k\geqslant 1}\left(\frac{1}{2}\right)^{2k} = \sum_{k\geqslant 1}\left(\frac{1}{4}\right)^k = \frac{1}{3}$$

portanto, calculando a probabilidade do complemento, a probabilidade de escolher um número ímpar é 2/3. A probabilidade de escolha de um múltiplo de 3 é 1/7 e a probabilidade da escolha de um múltiplo de 6 é 1/63 (verifique). Usando a Regra da Adição e o fato de que ser múltiplo de 6 equivale a ser múltiplo de 2 *e* múltiplo de 3, temos que a probabilidade de escolha de um múltiplo de 2 *ou* um múltiplo de 3 é $1/3 + 1/7 - 1/63 = 29/63 \approx 0{,}46$. ◊

Exemplo 1.11. No intervalo $\Omega = [0,1]$ da reta real podemos definir uma medida de probabilidade $\mathbb{P}$ de modo que os intervalos (a,b), $(a,b]$, $[a,b)$, $[a,b]$ tenham probabilidade $|b-a|$, entretanto não há tal medida de modo que $\mathbb{P}(A)$ esteja definida para

todo $A \subseteq \Omega$, ou seja, nem todo subconjunto do espaço amostral é evento (veja, e.g., Rosenthal, 2006, proposição 1.2.6). O conjunto dos eventos aleatórios é subconjunto próprio do conjunto das partes do intervalo. ◊

Exemplo 1.12 (*probabilidade geométrica clássica*). Consideremos o experimento 6 do Exemplo 1.2. É possível definir uma medida de probabilidade para $A \subseteq \Omega$ como a área de A proporcionalmente a de Ω, i.e.,

$$\mathbb{P}(A) = \frac{\text{Área}(A)}{\pi}$$

Assim, a probabilidade de um lançamento aleatório acertar o círculo de mesmo centro do alvo e raio 1/2 é 1/4. Ademais, há subconjuntos de Ω que não têm uma probabilidade associada pois não é possível definir área para todo subconjunto do plano (veja, e.g., Gelbaum e Olmsted, 1964, capítulo 11). ◊

Há uma diferença fundamental entre os modelos probabilísticos dos exemplos 1.8 e 1.10 e os modelos dos exemplos 1.11 e 1.12. Nos dois primeiros é possível atribuir probabilidade a todo subconjunto do espaço amostral, o que não é possível nos outros dois. O primeiro caso ($\mathcal{A} = 2^{\Omega}$) sempre vale em um espaço amostral enumerável (finito ou infinito) é chamado de **espaço amostral discreto**. Um espaço que têm a mesma cardinalidade dos reais, que também é o caso do item 4 do Exemplo 1.2, é chamado de **espaço amostral contínuo**. Para o espaço de eventos de um espaço amostral contínuo qualquer vale que *não há medida de probabilidade que possa ser definida para todo subconjunto* desses espaços. A explicação desse fenômeno é muito técnica para ser dada aqui, as ferramentas necessárias vão além do escopo deste texto. Em resumo, o espaço de eventos $\mathcal{A}$ é uma necessidade técnica e sua compreensão vai muito além do que precisamos neste texto que é dedicado ao caso discreto.

Exercício 1.13. Determine uma medida de probabilidade para os eventos do problema de Monty Hall.

1.1.1 ESPAÇO DE PROBABILIDADE

Probabilidade pode ser estudada do ponto de vista abstrato sem se referir a experimentos aleatórios e sem que os números associados aos eventos tenham qualquer interpretação. Formalmente, exigimos que qualquer medida de probabilidade $\mathbb{P}$ esteja definida sobre uma família $\mathcal{A}$ de subconjuntos de Ω que deve satisfazer: (i) $\Omega \in \mathcal{A}$; (ii) se $A \in \mathcal{A}$ então $\overline{A} \in \mathcal{A}$; (iii) se $A_i \in \mathcal{A}$ para todo $i \geqslant 1$, então $\bigcup_{i \geqslant 1} A_i \in \mathcal{A}$. Uma família de subconjuntos como acima é dita σ-**álgebra de subconjuntos de** Ω.

Um **espaço de probabilidade**, assim como um modelo probabilístico, é uma terna $(\Omega, \mathcal{A}, \mathbb{P})$ tal que Ω é um conjunto não vazio, chamado **espaço amostral**; $\mathcal{A}$

é uma σ-álgebra de subconjuntos de Ω; e $\mathbb{P}\colon \mathcal{A} \to [0,1]$ é uma **medida de probabilidade**.

Deixamos para a reflexão do leitor o fato de que todo modelo probabilístico de um experimento aleatório corresponde a um espaço de probabilidades e todo espaço de probabilidades corresponde ao modelo probabilístico de um experimento ideal e usaremos essas terminologias sem distinção.

1.1.2 O CASO DISCRETO

Um **modelo probabilístico discreto**, ou **espaço de probabilidade discreto**, é um espaço $(\Omega, \mathcal{A}, \mathbb{P})$ em que Ω é enumerável (finito ou infinito). Assumiremos que $\mathcal{A} = 2^\Omega$ sempre que não for dito outra coisa. Nesse caso, todo experimento tem seu modelo probabilístico especificado quando estabelecemos

1. um espaço amostral enumerável Ω;
2. uma função $p\colon \Omega \to [0,1]$ tal que $\sum_{\omega\in\Omega} p(\omega) = 1$, dita função de probabilidade.

De fato, dado (Ω, p) como acima podemos definir uma função $\mathbb{P}$ sobre 2^Ω tomando

$$\mathbb{P}(A) := \sum_{\omega\in A} p(\omega)$$

que é um número real positivo para qualquer $A \subseteq \Omega$, como já observamos no Exemplo 1.10. Claramente, $\mathbb{P}(A) \geqslant 0$ e $\mathbb{P}(\Omega) = \sum_\omega \mathbb{P}(\{\omega\}) = 1$. Ainda, se A_i para $i \geqslant 1$ são eventos mutuamente exclusivos então $\mathbb{P}\big(\bigcup_{i\geqslant 1} A_i\big) = \sum_{i\geqslant 1} \mathbb{P}(A_i)$ segue da convergência absoluta e da exclusão mútua, como deduzimos na equação (1.3).

Convencionamos a notação

$$\mathbb{P}(\omega) := \mathbb{P}(\{\omega\})$$

para os eventos elementares.

Para registro, enunciamos o seguinte resultado sem prova.

Teorema. *Se $\Omega \neq \varnothing$ é enumerável e $p\colon \Omega \to [0,1]$ é tal que $\sum_{\omega\in\Omega} p(\omega) = 1$ então $(\Omega, 2^\Omega, \mathbb{P})$ com $\mathbb{P}(A) = \sum_{\omega\in A} p(\omega)$ para todo $A \in 2^\Omega$ é um espaço de probabilidade. Reciprocamente, se $(\Omega, 2^\Omega, \mathbb{P})$ é um espaço de probabilidade sobre Ω enumerável então (Ω, p) com $p(\omega) := \mathbb{P}(\{\omega\})$ satisfaz as duas condições de um modelo probabilístico discreto.*

Exemplo 1.14. No item 3 do Exemplo 1.2 uma moeda equilibrada é lançada repetidamente até sair coroa. Esse experimento é modelado pelo espaço amostral $\Omega = \{(\text{Co}), (\text{Ca},\text{Co}), \ldots, (\text{Ca},\text{Ca},\ldots)\}$ munido da função de probabilidade

$$p((c_1, c_2, \ldots, c_i)) = \begin{cases} 2^{-i} & \text{para} c_j = \text{Co se } j = i \text{ e } c_j = \text{Ca caso contrário,} \\ 0 & \text{nos outros casos.} \end{cases}$$

Da argumentação feita no Exemplo 1.10, página 14, deduzimos igualmente que Ω e p definem um modelo probabilístico discreto para o experimento. $\Diamond$

Vejamos um modelo probabilístico para Monty Hall. O experimento consiste das seguintes três etapas

1. o apresentador esconde o carro atrás de uma das portas escolhida com probabilidade 1/3;

2. com probabilidade 1/3, uma porta é escolhida pelo jogador;

3. o apresentador revela, dentre as duas que o jogador não escolheu, aquela que não esconde o carro. Se houver duas possibilidades então o apresentador escolhe uma delas com probabilidade 1/2.

O espaço amostral é definido pelas ternas (e_1, e_2, e_3) em que e_i é a porta escolhida no passo i descrito acima e, se a portas estão numeradas por 1,2 e 3, então definimos um modelo probabilístico discreto com Ω dado por

$$\Big\{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1),$$
$$(1,1,2),(1,1,3),(2,2,1),(2,2,3),(3,3,1),(3,3,2)\Big\}.$$

e probabilidades de acordo com o diagrama de árvore mostrado na Figura 1.3 abaixo; um caminho seguido pelo jogador numa rodada do jogo corresponde a um caminho na árvore, a partir da raiz (o ponto mais alto) até uma folha (um dos pontos mais baixos). A primeira ramificação corresponde a escolha de porta para esconder o carro, as segundas ramificações correspondem a escolha do jogador e as terceiras ramificações correspondem a escolha de porta para abrir feita pelo apresentador. Os

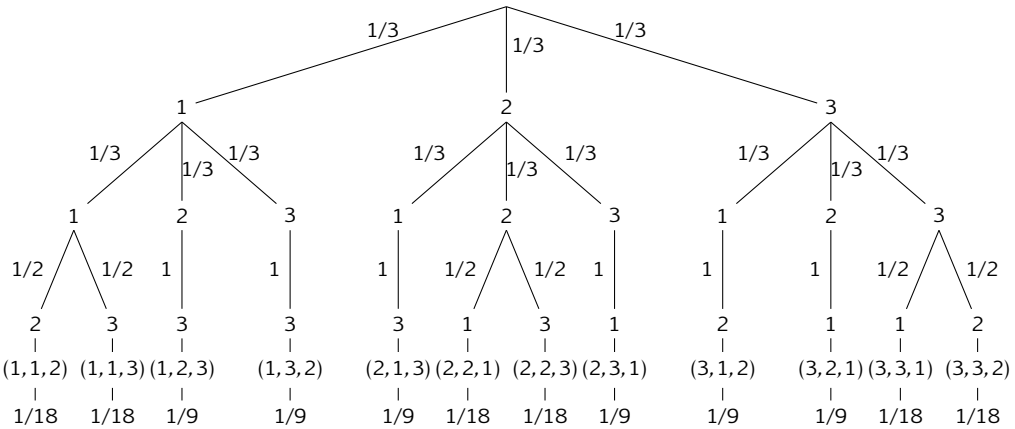


Figura 1.3: diagrama de árvore de um modelo para Monty Hall.

eventos que interessam, a saber "o jogador vence trocando de porta" e "o jogador

vence não trocando de porta", são complementares e denotados por A e $\overline{A}$ respectivamente, de modo que $A = \{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\}$ e $\overline{A}$ é dada pelas ternas restantes de Ω. O jogador ganha o carro trocando de porta com probabilidade

$$\mathbb{P}(A) = \mathbb{P}(\{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\}) = \frac{2}{3}$$

portanto, ganha sem trocar de porta com probabilidade $1 - 2/3 = 1/3$, que corresponde à probabilidade de ter escolhido a porta certa já na primeira oportunidade de escolha. Portanto, a melhor estratégia é trocar de porta quando é oferecida essa oportunidade.

1.1.3 CONTINUIDADE DE UMA MEDIDA DE PROBABILIDADE

Consideremos novamente o exemplo do lançamento de uma moeda equilibrada até sair coroa, Exemplo 1.14, com $\Omega = \{(\text{Co}), (\text{Ca},\text{Co}), \ldots, (\text{Ca},\text{Ca},\ldots)\}$ munido da função de probabilidade $p((c_1, c_2, \ldots, c_i)) = 2^{-i}$ e $p((\text{Ca},\text{Ca},\ldots)) = 0$. Como cada resultado de um lançamento é igualmente provável, ocorre com probabilidade $1/2$, e não depende dos resultados dos outros lançamentos deve ser intuitivamente válido[3] a interpretação de que $(\text{Ca},\text{Ca},\ldots,\text{Ca},\text{Co})$ ocorre com probabilidade

$$\frac{1}{2}\cdot\frac{1}{2}\cdots\frac{1}{2} = \left(\frac{1}{2}\right)^{\text{número de lançamentos}} \tag{1.4}$$

e como cada ponto amostral em $\Omega \setminus \{(\text{Ca},\text{Ca},\ldots)\}$ está associado a um único inteiro positivo $\mathbb{P}(\Omega \setminus \{(\text{Ca},\text{Ca},\ldots)\}) = \sum_{n\geqslant 1} 2^{-n} = 1$ o que nos obriga a tomar como 0 a probabilidade para o evento "nunca sair coroa".

Nessa seção veremos que essa obrigação respeita a proposta intuitiva para finitos lançamentos tomada em na equação (1.4) acima.

Consideremos o evento A_n definido por "não sai coroa até o n-ésimo lançamento". Esse evento ocorre com probabilidade

$$1 - \mathbb{P}(\overline{A_n}) = 1 - \sum_{i=1}^{n}\left(\frac{1}{2}\right)^i = 1 - \frac{1/2 - (1/2)^{n+1}}{1/2} = \left(\frac{1}{2}\right)^n.$$

Pensando ainda de modo intuitivo, queremos que o evento "nunca sair coroa", representado por $\lim_{n\to\infty} A_n$, tenha probabilidade $\lim_{n\to\infty}\mathbb{P}(A_n) = \lim_{n\to\infty} 2^{-n} = 0$. Essa "passagem ao limite", $\mathbb{P}(\lim_{n\to\infty} A_n) = \lim_{n\to\infty}\mathbb{P}(A_n)$, é garantida pela Aditividade enumerável da medida.

Uma sequência qualquer $(A_n : n \geqslant 1)$ de eventos em um espaço de probabilidade $(\Omega, \mathcal{A}, \mathbb{P})$ é dita **monótona** se vale um dos casos

[3] Essa noção intuitiva é formalizada na seção 1.3.4.

crescente: $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n \subseteq A_{n+1} \subseteq \cdots$ e definimos

$$\lim_{n\to\infty} A_n := \bigcup_{n\geqslant 1} A_n.$$

decrescente: $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_n \supseteq A_{n+1} \supseteq \cdots$ e definimos

$$\lim_{n\to\infty} A_n := \bigcap_{n\geqslant 1} A_n.$$

Se $(A_n\colon n \geqslant 1)$ é uma sequência crescente, então o limite pode ser escrito como uma união de eventos disjuntos $A_1 \cup (A_2 \setminus A_1) \cup (A_3 \setminus A_2) \cup \cdots$ de modo que se tomamos $A_0 := \varnothing$ então

$$\mathbb{P}\Big(\lim_{n\to\infty} A_n\Big) = \lim_{n\to\infty} \sum_{i=1}^{n} \mathbb{P}(A_i \setminus A_{i-1}) = \lim_{n\to\infty} \sum_{i=1}^{n} (\mathbb{P}(A_i) - \mathbb{P}(A_{i-1})) = \lim_{n\to\infty} \mathbb{P}(A_n).$$

No caso em que $(A_n\colon n \geqslant 1)$ é decrescente tomamos os complementos e temos que $(\overline{A_n}\colon n \geqslant 1)$ é crescente, portanto,

$$\mathbb{P}\Big(\lim_{n\to\infty} \overline{A_n}\Big) = \mathbb{P}\left(\bigcup_{n\geqslant 1} \overline{A_n}\right) = \mathbb{P}\left(\overline{\bigcap_{n\geqslant 1} A_n}\right) = \mathbb{P}\Big(\overline{\lim_{n\to\infty} A_n}\Big) = 1 - \mathbb{P}\Big(\lim_{n\to\infty} A_n\Big)$$

por outro lado

$$\mathbb{P}\Big(\lim_{n\to\infty} \overline{A_n}\Big) = \lim_{n\to\infty} \mathbb{P}\Big(\overline{A_n}\Big) = \lim_{n\to\infty} \Big(1 - \mathbb{P}\Big(\overline{A_n}\Big)\Big) = 1 - \lim_{n\to\infty} \mathbb{P}(A_n)$$

portanto

$$\lim_{n\to\infty} \mathbb{P}(A_n) = \mathbb{P}\Big(\lim_{n\to\infty} A_n\Big).$$

Em resumo, se $(A_n\colon n \geqslant 1)$, é uma sequência monótona de eventos então

$$\mathbb{P}\Big(\lim_{n\to\infty} A_n\Big) = \lim_{n\to\infty} \mathbb{P}(A_n). \tag{1.5}$$

De volta ao exemplo do início da seção, consideremos o evento A_n definido por "não sai coroa até o n-ésimo lançamento". Então a sequência $(A_n\colon n \geqslant 1)$ é monótona pois $A_n \supset A_{n-1}$ para todo $n > 1$, portanto,

$$\mathbb{P}((\mathrm{Ca}, \mathrm{Ca}, \ldots)) = \mathbb{P}\Big(\lim_{n\to\infty} A_n\Big) = \lim_{n\to\infty} \mathbb{P}(A_n) = 0.$$

Exemplo 1.15. Consideremos o lançamento de uma moeda equilibrada infinitas vezes, o que pode ser modelado pelo espaço amostral contínuo $\Omega = \{\mathrm{Ca}, \mathrm{Co}\}^{\mathbb{N}}$. Intuitivamente, parece ser claro que esperaríamos que a probabilidade de nunca sair cara deveria ser zero: se lançarmos n vezes, a probabilidade de nunca sair cara é 2^{-n}, então no limite a probabilidade é 0. A propriedade dada na equação (1.5) permite a passagem ao limite: A_n é o evento "nos primeiros n lançamentos ocorre pelo menos

uma cara"; para $n \geqslant 1$ temos uma sequência monótona de eventos. O limite é o evento "em algum momento, ocorre cara" cuja probabilidade é

$$\lim_{n\to\infty} \mathbb{P}(A_n) = \lim_{n\to\infty} 1 - 2^{-n} = 1.$$

Portanto, de fato, a probabilidade de nunca sair cara é zero.

Uma medida de probabilidade nesse caso pode ser definida do seguinte modo. Para cada natural n e cada sequência $(c_1, \ldots, c_n)$ de caras e coroas tomamos os conjuntos cilíndricos dados por essa sequência $\{(x_1, x_2, \ldots) \in \Omega : (x_1, \ldots, x_n) = (c_1, \ldots, c_n)\}$. O espaço de eventos $\mathcal{A}$ é a interseção de todas as σ-álgebras de Ω que contêm os cilindros. Se a probabilidade de um cilindro é a probabilidade da sequência de caras e coroas que o define, então um famoso teorema devido a Komolgorov garante que essa probabilidade pode ser estendida de modo único a todo $\mathcal{A}$. $\Diamond$

Vimos que a propriedade dada na equação (1.5) segue da Não-negatividade, Normalização e Aditividade enumerável de uma medida de probabilidade. Se tomarmos por princípios para $\mathbb{P}$ que valem Não-negatividade, Normalização, Aditividade finita (isto é, o item 2 da Proposição 1.5) e a propriedade dada na equação (1.5) para sequências monótonas de eventos, então vale a Aditividade enumerável. De fato, assumindo os axiomas Não-negatividade, Normalização, a equação (1.5) e o item 2 da Proposição 1.5, se $(A_n : n \geqslant 1)$ é qualquer sequência de eventos mutuamente exclusivos então

$$B_n := \bigcup_{i \geqslant n} A_i$$

é uma sequência monótona decrescente e $\lim_{n\to\infty} B_n = \varnothing$. Usando a aditividade finita de $\mathbb{P}$

$$\mathbb{P}\left(\bigcup_{i\geqslant 1} A_i\right) = \mathbb{P}\left(\bigcup_{i=1}^{n-1} A_i\right) + \mathbb{P}\left(\bigcup_{i\geqslant n} A_i\right) = \sum_{i=1}^{n-1} \mathbb{P}(A_i) + \mathbb{P}(B_n)$$

e se tomamos o limite quando n tende ao infinito

$$\mathbb{P}\left(\bigcup_{i\geqslant 1} A_i\right) = \sum_{i\geqslant 1} \mathbb{P}(A_i).$$

Diferente do que fizemos no Exemplo 1.10, página 14, a demonstração para espaços contínuos de que uma função candidata a medida de probabilidade é enumeravelmente aditiva é difícil. Usualmente, o que é feito é verificar que uma função é finitamente aditiva e contínua para toda sequência $(B_n : n \geqslant 1)$ tal que $\lim_{n\to\infty} B_n = \varnothing$, isso é suficiente para garantir que é enumeravelmente aditiva e, em geral, uma tarefa mais fácil de realizar.

1.2 CONVENÇÕES DE NOTAÇÃO

Sejam A um evento aleatório e A_Ω o subconjunto que o representa em $(\Omega, \mathcal{A}, \mathbb{P})$. Denotamos por $\mathbb{P}[A]$ a probabilidade do evento descrito por A, isto é, $\mathbb{P}[A] := \mathbb{P}(A_\Omega)$. Caso haja a necessidade de evidenciar o espaço amostral escreveremos

$$\mathop{\mathbb{P}}_{\Omega}[A] \quad \text{ou} \quad \mathop{\mathbb{P}}_{\omega\in\Omega}[\text{ocorre A}] \quad \text{ou} \quad \mathop{\mathbb{P}}_{\omega\in\Omega}[\omega \in A] \tag{1.6}$$

com o mesmo sentido, o de $\mathbb{P}(A_\Omega)$. Por exemplo, no caso em que uma moeda equilibrada é lançada até sair coroa, o evento caracterizado por "o número de lançamentos é par" é definido em $(\Omega, \mathcal{A}, \mathbb{P})$ por $\{(c_1, \ldots, c_i)\}$ e $\mathbb{P}[\text{número par de lançamentos}]$ denota o mesmo que $\mathbb{P}(\{(c_1, \ldots, c_i) : i \text{ é par}\})$.

Caso Ω seja finito e a menos que seja dada explicitamente outra medida, então a notação na equação (1.6) significa que estamos assumindo a medida de **probabilidade uniforme**: $\mathbb{P}(\omega) = 1/|\Omega|$ para todo $\omega \in \Omega$. Por exemplo, seja $p(x)$ um polinômio não nulo com coeficientes inteiros e Ω um conjunto finito de números inteiros. A probabilidade de que o sorteio de um elemento de Ω, todos com o mesma probabilidade de ocorrência, resulte numa raiz do polinômio é descrita por

$$\mathop{\mathbb{P}}_{x\in_R\Omega}[p(x) = 0]$$

que é a probabilidade do evento $R = \{\omega \in \Omega : p(\omega) = 0\}$ e que, caso não seja dito nada a respeito da medida, é dada por $\mathbb{P}(R) = |R|/|\Omega|$.

Nos algoritmos assumiremos a possibilidade de se fazer escolhas aleatórias, ou seja, assumiremos que os algoritmos dispõem de uma fonte de bits aleatórios e escrevemos a instrução

$$a \xleftarrow{R} \{0, 1\}$$

para denotar o fato de que a é uma variável do algoritmo e que após a execução da atribuição $\xleftarrow{R}$ o valor da variável a é um elemento qualquer de $\{0, 1\}$ com probabilidade 1/2. De um modo geral, se Ω é um conjunto finito, então escrevemos a instrução

$$a \xleftarrow{R} \Omega$$

chamada de atribuição por uma **escolha aleatória uniforme** em Ω, o que significa que a assume qualquer um dos elementos de Ω com igual probabilidade, a saber $1/|\Omega|$.

1.2.1 SIGILO PERFEITO

Vejamos, como aplicação dos conceitos elementares de probabilidade, uma das contribuições do grande matemático americano Claude Shannon (1916 – 2001) que

é considerado fundador da Teoria da Informação.

Um *sistema de codificação* é definido por uma quina $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ de conjuntos, onde $\mathcal{P}$ é o conjunto dos textos comuns (ou legíveis); $\mathcal{C}$ é o conjunto dos textos codificados (ou ilegíveis); $\mathcal{K}$ é o espaço das chaves que são usadas para codificar e decodificar um texto; $\mathcal{E}$ é o conjunto das funções de codificação $\mathsf{E}_k\colon \mathcal{P} \to \mathcal{C}$ com $k \in \mathcal{K}$; $\mathcal{D}$ é o conjunto das funções de decodificação $\mathsf{D}_k\colon \mathcal{C} \to \mathcal{P}$ com $k \in \mathcal{K}$. Essas funções são tais que para cada $e \in \mathcal{K}$ existe $d \in \mathcal{K}$ para as quais vale $\mathsf{D}_d(\mathsf{E}_e(p)) = p$.

Exemplo 1.16 (*cifra de César*). Essa técnica identifica o alfabeto $\{a, b, \ldots, z\}$ com o conjunto $\{0,\ldots,25\}$ dos restos da divisão inteira por 26 e $\mathcal{K} = \mathcal{P} = \mathcal{C} := \{0,\ldots,25\}^\ell$ em que ℓ é o comprimento da mensagem. Para uma chave $e \in \mathcal{K}$ a mensagem $\mathsf{x} = x_1x_2\ldots x_\ell$ é codificada como $\mathsf{E}_e(\mathsf{x}) = y_1y_2\ldots y_\ell$ com $y_i = (x_i + e) \bmod 26$, para todo i, e é decodificada como $\mathsf{D}_e(\mathsf{x}) = y_1y_2\ldots y_\ell$ com $y_i = (x_i - e) \bmod 26$ para todo i. Por exemplo, para a chave $e = 3$ o texto "essaauladasono" é codificado como "hvvddzodgdvrqr".

A cifra de César deve seu nome à descoberta de registros que indicam que o imperador romano Júlio César tenha usado esse método de cifragem. Não é conhecida a eficiência dessa prática na época de Júlio César mas, por ser facilmente decifrada, admite-se que tenha sido efetiva devido ao fato de que a maioria das pessoas eram analfabeta naquela época.

Para uma abordagem probabilística, tomemos $\ell = 1$ e $(\mathcal{K},\mathbb{P})$ com $\mathbb{P}$ a medida uniforme. Dadas quaisquer duas mensagens legíveis $m_1, m_2 \in \mathcal{P}$ e uma mensagem codificada $y \in \mathcal{C}$, temos

$$\mathbb{P}(\{k \in \mathcal{K}\colon \mathsf{E}_k(m_1) = y\}) = \frac{1}{26} = \mathbb{P}(\{k \in \mathcal{K}\colon \mathsf{E}_k(m_2) = y\})$$

ou seja, o conhecimento do texto codificado não dá nenhuma informação a respeito do texto legível. No caso $\ell = 2$ a situação é outra. Se $ab, az \in \mathcal{P}$ e $bc \in \mathcal{C}$ então $\mathbb{P}(\{k \in \mathcal{K}\colon \mathsf{E}_k(ab) = bc\}) = 1/26$ pois podemos tomar $k = 1$ e essa é a única chave que codifica ab em bc, por outro lado não existe chave que codifica az em bc de modo que $\mathbb{P}(\{k \in \mathcal{K}\colon \mathsf{E}_k(az) = bc\}) = 0$. Agora, o conhecimento do texto codificado dá alguma informação a respeito do texto legível. ◊

Exemplo 1.17 (*one-time pad*). O seguinte sistema de codificação, conhecido por *one-time pad*, foi descrito pela primeira vez em 1882 por Frank Miller, reinventado e patenteado por Gilbert Sandford Vernam em 1919 e, posteriormente, aperfeiçoado por Joseph Mauborgne, que reconheceu que se a chave fosse aleatória e usada uma única vez o sistema seria muito seguro.

Tomamos $\mathcal{P} = \mathcal{K} = \mathcal{C} := \{0,1\}^n$ e para uma chave k escolhida previamente definimos

$$\mathsf{E}_k(x) := x \oplus k \qquad \text{e} \qquad \mathsf{D}_k(y) := y \oplus k. \tag{1.7}$$

em que $x \oplus y$ é a soma módulo 2 (ou o *ou exclusivo*) coordenada-a-coordenada das sequências binárias x e y. Não é difícil verificar que em (1.7) vale $\mathsf{D}_k(\mathsf{E}_k(x)) = x$. ♢

Um sistema de codificação tem **sigilo perfeito** se para quaisquer $m_1,\ m_2 \in \mathcal{P}$ de mesmo comprimento e para todo $\mathsf{C} \in \mathcal{C}$ vale

$$\mathop{\mathbb{P}}_{k \in_R \mathcal{K}}[\mathsf{E}_k(m_1) = \mathsf{C}] = \mathop{\mathbb{P}}_{k \in_R \mathcal{K}}[\mathsf{E}_k(m_2) = \mathsf{C}].$$

Pelas convenções de notação, o espaço amostral é o conjunto das chaves, a descrição $[\mathsf{E}_k(m_1) = \mathsf{C}]$ corresponde ao evento $\{k \in \mathcal{K}\colon \mathsf{E}_k(m_1) = \mathsf{C}\}$ formado por todas as chaves que codificam m_1 como C e, também, está implícito que probabilidade de uma chave qualquer ser escolhida é $1/|\mathcal{K}|$.

Em outras palavras, conhecido um texto cifrado, o sigilo perfeito requer que qualquer texto legível tenha a mesma probabilidade de ser o texto legível subjacente ao texto cifrado.

Claude Shannon provou que o *one-time pad* é uma codificação "inviolável" no sentido de que o sistema tem sigilo perfeito. O *one-time pad* não é o único sistema que possui sigilo perfeito, mas foi o primeiro a ser descoberto.

A codificação do texto legível $m \in \mathcal{P}$ usando a chave $k \in \mathcal{K}$ é o texto cifrado $\mathsf{C} = m \oplus k$, logo $m \oplus \mathsf{C} = m \oplus (m \oplus k) = (m \oplus m) \oplus k = k$, portanto, dados m e C existe uma única chave $k \in \mathcal{K}$ tal que $\mathsf{E}_k(m) = \mathsf{C}$, de modo que

$$\mathop{\mathbb{P}}_{k \in_R \mathcal{K}}[m_1 \oplus k = \mathsf{C}] = \frac{|\{k \in \mathcal{K}\colon k \oplus m_1 = \mathsf{C}\}|}{|\mathcal{K}|} = \frac{1}{|\mathcal{K}|} = \mathop{\mathbb{P}}_{k \in_R \mathcal{K}}[m_2 \oplus k = \mathsf{C}]$$

para todos os textos legíveis $m_1,\ m_2 \in \mathcal{P}$ e todo texto cifrado $\mathsf{C} \in \mathcal{C}$. Isso demonstra o seguinte resultado.

Teorema 1.18 *O* one-time pad *tem sigilo perfeito.* □

1.2.2 TESTE DE IDENTIDADE POLINOMIAL

Nosso primeiro exemplo de um algoritmo aleatorizado é um teste de identidade entre polinômios: dados dois polinômios p e q sobre x, decidir de modo eficiente se eles são idênticos.

Há muitas questões a serem esclarecidas nessa formulação do problema: o que significam "eficiente", "dado um polinômio" e "idênticos"? Isso será tratado mais tarde, na seção 1.7.3, por ora basta saber que "dado um polinômio p" significa que p é um polinômio dado por uma caixa preta da qual a função polinomial $p(x)$ pode ser avaliada em qualquer número x. Além disso, vamos considerar o problema equivalente de decidir se um polinômio dado f é identicamente nulo. Uma algoritmo que resolve esse problema também resolve o problema original, basta tomarmos $f(x) = p(x) - q(x)$.

Um algoritmo para esse problema funciona do seguinte modo: dado $f(x)$ de grau no máximo d, escolhemos aleatoriamente $a \in \{1, 2, \ldots, 4d\}$ e avaliamos $f(a)$; se $f(a) = 0$, então respondemos *sim*, caso contrário respondemos *não*. A resposta *sim* significa que $f(x)$ é o polinômio nulo e a resposta *não* significa que $f(x)$ não é o polinômio nulo. Esse algoritmo pode responder errado dependendo de f e da escolha aleatória a e devemos tentar garantir que a probabilidade de ocorrer o erro seja pequena.

Se o polinômio f é nulo então a resposta está sempre certa. Suponhamos que f não é nulo. Nesse caso, se a escolha aleatória a for uma raiz do polinômio então $f(a) = 0$ e a resposta a resposta *sim* dada pelo algoritmo está errada e se a escolha aleatória a não for uma raiz do polinômio então $f(a) \neq 0$ e a resposta a resposta *sim* dada pelo algoritmo está correta. Em resumo, uma resposta *não* dada pelo algoritmo está correta e uma resposta *sim* pode estar errada. O algoritmo descrito abaixo resume essa estratégia.

Instância: d inteiro positivo e f polinômio de grau no máximo d.
Resposta: *não* se f não é nulo, senão *sim* com probabilidade de erro no máximo 1/4.

1 $a \xleftarrow{R} \{1, 2, \ldots, 4d\}$;
2 **se** $f(a) = 0$ **então responda** *sim*.
3 **senão responda** *não*.

Algoritmo 1: teste de identidade entre polinômios.

Para determinar um limitante para a probabilidade do algoritmo responder errado, seja f um polinômio não nulo e de grau no máximo d e consideremos o evento E formado pelas raízes de f que pertencem ao espaço amostral $\Omega = \{1, 2, \ldots, 4d\}$. Então $|E| \leqslant \text{grau}(f) \leqslant d$ pois f tem no máximo $\text{grau}(f)$ raízes distintas pelo teorema fundamental da álgebra. O algoritmo erra se a escolha aleatória resulta num elemento de E, portanto,

$$\mathbb{P}[\text{erro}] \leqslant \frac{1}{4}.$$

Proposição 1.19 *Sejam d um inteiro positivo, f um polinômio não nulo de grau no máximo d e $\Omega \subseteq \mathbb{Z}$ finito. Então a probabilidade com que uma escolha aleatória uniforme em Ω seja raiz de f é no máximo $d/|\Omega|$, portanto, o Algoritmo 1 erra com probabilidade no máximo 1/4.* □

Formalizaremos mais adiante a possibilidade de fazer essa probabilidade arbitrariamente pequena ao custo de mais computação. Intuitivamente, como no caso dos lançamentos de moedas, imagine tal algoritmo sendo executado em computadores

diferentes concomitantemente. Basta um deles responder *não* para que a resposta definitiva seja *não*. Se são dez computadores a probabilidade de erro é a probabilidade de 10 respostas *sim*, ou seja, no máximo $(1/4)^{10} < 10^{-6}$ (em metros é menos que o diâmetro do fio da teia de uma aranha), desprezível na prática.

O que é $p(x) = 0$? Existem duas respostas possíveis que levam a dois problemas computacionais diferentes:

- *Valor zero em todos os pontos*: (EZE – *Evaluates to Zero Everywhere*) dado o polinômios $p(x)$ sobre $\mathbb{F}$, decidir se para toda escolha de $y \in \mathbb{F}$ o valor de $p(y)$ é 0. Por exemplo, nos inteiros módulo 2 (o corpo finito $\mathbb{Z}_2$) o polinômio $x^2 - x$ vale zero em todos os pontos.
- *Polinômio identicamente nulo*: (PIT – *Polynomial Identity Testing*) o problema é decidir se um polinômio dado por uma expressão aritmética, após expandido como combinação linear de monômios tem todos os coeficientes iguais a zero. Por exemplo, a expressão $(x+5)^2 - x^2 - 3x - 4$ com aritmética módulo 7 (corpo finito $\mathbb{Z}_7$) resulta em coeficientes iguais a zero na expansão.

Exercício 1.20. Prove que para todo inteiro $n \geqslant 1$ vale que

$$(x+1)^n - (x^n + 1) \equiv 0 \pmod{n}$$

se e somente se n é primo.

1.3 PROBABILIDADE CONDICIONAL

Lançamos dois dados equilibrados, um deles é vermelho e tem doze faces numeradas de 1 a 12 e o outro preto com vinte faces numeradas de 1 a 20.

Se temos a informação de que a soma dos resultados é 15, e isso é tudo que sabemos, qual é a probabilidade do dado vermelho ter resultado 6?

Definimos um modelo discreto para o experimento tomando o espaço amostral Ω composto pelos $12 \cdot 20 = 240$ pontos amostrais, dados pelos pares ordenados de resultados de cada dado, com a medida uniforme de probabilidade. Sejam Q_Ω o subconjunto dos 12 eventos elementares de Ω que representa o evento "a soma é 15" e S_Ω o evento "o valor do dado vermelho é 6".

Se é certo que ocorre Q_Ω então vamos renormalizar a probabilidade de cada ponto amostral ω em Q_Ω de modo que Q_Ω tenha probabilidade 1, ou seja, tomamos $\mathbb{P}_Q(\omega) = \mathbb{P}_\Omega(\omega)/\mathbb{P}_\Omega(Q_\Omega) = (1/|\Omega|)/(|Q_\Omega|/|\Omega|) = 1/12$, para todo $\omega \in Q_\Omega$. Ademais, $S_Q = S_\Omega \cap Q_\Omega = \{(6,9)\}$ tem probabilidade $\mathbb{P}_Q(S_Q) = 1/12$. Essa é a probabilidade de ocorrer um 6 vermelho sob a condição de que a soma dos dados é 15.

Em um espaço de probabilidade discreto definido por Ω e $\mathbb{P}$, a probabilidade de ocorrência do evento A condicionada a ocorrência o evento E, ou como dizemos a **probabilidade condicional** de A dado E, em que $\mathbb{P}(E) \neq 0$, é definida por

$$\mathbb{P}(A \mid E) := \frac{\mathbb{P}(A \cap E)}{\mathbb{P}(E)} \tag{1.8}$$

e $\mathbb{P}(A \mid E)$ é lido como a **probabilidade de** A **dado** E. Por exemplo, se Ω é finito com medida de probabilidade uniforme e $E \neq \varnothing$ então

$$\mathbb{P}(A \mid E) = \frac{\mathbb{P}(A \cap E)}{\mathbb{P}(E)} = \frac{|A \cap E|}{|E|}$$

que é, essencialmente, a medida uniforme em E. No exemplo acima, do par de dados,

$$\mathbb{P}(S \mid Q) = \frac{\mathbb{P}(S \cap Q)}{\mathbb{P}(Q)} = \frac{|\{(6,9)\}|}{|\{(i,j)\colon i+j=15\}|} = \frac{1}{12}.$$

Exercício 1.21. Considere um espaço de probabilidade $(\Omega, \mathcal{A}, \mathbb{P})$ e $E \in \mathcal{A}$ um evento com probabilidade positiva. Verifique que $\mathbb{P}_E(A) := \mathbb{P}(A \mid E)$ é uma medida de probabilidade para os eventos em $\mathcal{A}$ (i.e, satisfaz os axiomas de probabilidade da página 12). Verifique, também, que $(E, \{A \cap E\colon A \in \mathcal{A}\}, \mathbb{P}_E)$ é um espaço de probabilidade.

Exemplo 1.22. Uma urna tem 20 bolas azuis e 10 bolas brancas. Das bolas azuis, 5 têm a letra X e 15 têm a letra Y gravada nelas; das bolas brancas, 1 têm a letra X e 9 tem a letra Y. Uma bola é escolhida ao acaso. Qual é a probabilidade dessa bola ser azul e com a letra X? Se A representa o evento "bola azul" e X o evento "letra X" então $\mathbb{P}(X \mid A) = 5/20 = 1/4$, que é a proporção de bolas azuis com a letra X. Usando a equação (1.8) podemos deduzir que a probabilidade de sortear uma bola azul e com a letra X é $\mathbb{P}(A \cap X) = \mathbb{P}(X \mid A) \cdot \mathbb{P}(A) = (1/4) \cdot (20/30) = 1/6$. ◊

Regra do Produto A igualdade, que decorre da definição,

$$\mathbb{P}(A \cap E) = \mathbb{P}(A \mid E) \cdot \mathbb{P}(E)$$

usada no Exemplo 1.22 é consequência direta da definição de probabilidade condicional e é conhecida como **teorema da multiplicação** ou **Regra do Produto**. Um caso geral desse teorema é dado no Exercício 1.24 abaixo.

Há três urnas e em cada urna um par de bolas. Na primeira urna há um par de bolas brancas, na segunda um par de bolas pretas e na terceira um par com uma

bola de cada cor. Uma urna é escolhida uniformemente e, sem olhar para o interior da urna, uma bola é escolhida uniformemente é retirada. A bola retirada é branca. Qual a probabilidade da bola que ficou sozinha na urna ser preta?

Vamos denotar por B o evento "retirou uma bola branca" e por T o evento "ficou uma bola preta", ambos eventos do experimento composto por dois experimentos realizados consecutivamente. Queremos determinar $\mathbb{P}(\,T \mid B\,)$. Notemos que $\mathbb{P}(\,B \mid T\,)$ corresponde a sortear no segundo experimento uma bola branca na terceira urna, o que ocorre com probabilidade 1/2.

Usando a definição de condicional e a Regra do Produto escrevemos

$$\mathbb{P}(\,T \mid B\,) = \mathbb{P}(T \cap B)\mathbb{P}(B) = \mathbb{P}(\,B \mid T\,)\mathbb{P}(T)\mathbb{P}(B).$$

Ainda, $\mathbb{P}(T)$ corresponde a probabilidade de sortear a terceira urna no primeiro experimento, o que ocorre com probabilidade 1/3. Logo, $\mathbb{P}(\,T \mid B\,) = 1/3$.

Consideremos agora três urnas, digamos A, B e C, cada uma com a mesma probabilidade de ser escolhida, 1/3. Em cada uma das urnas há seis bolas, cada uma com a mesma probabilidade de ser escolhida, 1/6. Na urna A temos três bolas pretas e três bolas vermelhas; na urna B temos duas bolas pretas e quatro vermelhas; na urna C todas as bolas são pretas. Uma urna é escolhida aleatoriamente e, em seguida, uma bola é escolhida aleatoriamente e observamos a cor dessa bola. Vamos definir um modelo probabilístico discreto $(\Omega, \mathbb{P})$ para esse *experimento composto* de modo que $\mathbb{P}$ respeita as probabilidades em cada experimento num sentido que ficará claro abaixo.

Temos dois experimentos aleatórios, o primeiro consiste de sortear uma urna e o segundo e sortear uma bola da urna que foi escolhida. Para o primeiro experimento temos o modelo discreto $(\Omega_1, \mathbb{P}_1)$ dado pelo espaço amostral $\Omega_1 = \{A, B, C\}$ e a medida de probabilidade uniforme $\mathbb{P}_1$. Para o segundo experimento tomamos $(\Omega_2, \mathbb{P}_2)$ com o espaço amostral $\Omega_2 = \{V, P\}$ em que usamos os eventos atômicos (pontos amostrais) $V \in \Omega_2$ para representar "bola vermelha" e $P \in \Omega_2$ para representar "bola preta" e cujas probabilidades dependem da urna e são dadas na Tabela 1.2 abaixo, mas que por abuso de notação escrevemos $\mathbb{P}_2$ em todos os casos.

urna	$\mathbb{P}_2(V)$	$\mathbb{P}_2(P)$
A	1/2	1/2
B	2/3	1/3
C	0	1

Tabela 1.2: probabilidade dos eventos "bola vermelha" e "bola preta" em cada urna.

Um espaço amostral para o experimento composto pelos dois sorteios é $\Omega :=$

$\Omega_1 \times \Omega_2 = \{A, B, C\} \times \{V, P\}$. A probabilidade $\mathbb{P}$ é definida de tal modo que $\mathbb{P}(E \times \Omega_2) = \mathbb{P}_1(E)$ e $\mathbb{P}(\Omega_1 \times F) = \mathbb{P}_2(F)$.

Agora, por exemplo, o evento "urna A" é modelado no primeiro experimento e no experimento composto por, respectivamente

$$U_{\Omega_1} = \{A\} \quad \text{e} \quad U_\Omega = \{A\} \times \Omega_2 = \{(A, V), (A, P)\}$$

de modo que para a medida $\mathbb{P}$ em Ω temos para probabilidade de "urna A"

$$\mathbb{P}(U_\Omega) = \mathbb{P}(\{A\} \times \Omega_2) = \mathbb{P}_1(U_{\Omega_1}) = \frac{1}{3}.$$

O evento "bola preta" é modelado em Ω por $E_\Omega = \Omega_1 \times \{P\} = \{(A, P), (B, P), (C, P)\}$ de modo que temos para probabilidade de "bola preta"

$$\mathbb{P}(E_\Omega) = \mathbb{P}(\Omega_1 \times \{P\}) = \mathbb{P}((A, P)) + \mathbb{P}((B, P)) + \mathbb{P}((C, P))$$

entretanto, diferente do caso anterior, o resultado do segundo experimento depende do resultado do primeiro; o que conhecemos são as probabilidades condicionais de "cor" dado "urna". O ponto amostral (A, P) de Ω é dado por

$$(\Omega_1 \times \{P\}) \cap (\{A\} \times \Omega_2) = E_\Omega \cap U_\Omega \tag{1.9}$$

e tem probabilidade dada pela Regra do Produto da seguinte forma

$$\mathbb{P}(E_\Omega \cap U_\Omega) = \mathbb{P}(E_\Omega \mid U_\Omega)\mathbb{P}(U_\Omega) = \mathbb{P}_2(E_{\Omega_2})\mathbb{P}_1(U_{\Omega_1}) = \frac{1}{2} \cdot \frac{1}{3} \tag{1.10}$$

pois dado que ocorre "urna A", a probabilidade de "bola preta" é $\mathbb{P}(E_\Omega \mid U_\Omega) = \mathbb{P}_2(P_{\Omega_2}) = 1/2$.

Podemos determinar de maneira análoga a probabilidade de todo ponto amostral de Ω, cada um é dado por um interseção e pode ser escrito como na equação (1.9) e a probabilidade é calculada pela Regra do Produto como na equação (1.10).

Quando o espaço amostral é pequeno, como nesse exemplo, pode ser conveniente descrevermos o modelo probabilístico através de um diagrama de árvore como o da Figura 1.4 abaixo e como já fizemos para Monty Hall (Figura 1.3, pág. 18). O diagrama de árvore da Figura 1.4 representa cada etapa do experimento em um nível da árvore, com as respectivas probabilidades nas ramificações correspondentes aos resultados de cada etapa. A partir do segundo nível essas probabilidades são condicionadas ao que ocorreu na etapas anteriores. Uma maneira pragmática de calcular a probabilidade dada pela Regra do Produto é tomar o produto das probabilidades no caminho até ele nessa árvore, por exemplo, $\mathbb{P}((A, P)) = 1/3 \cdot 1/2$.

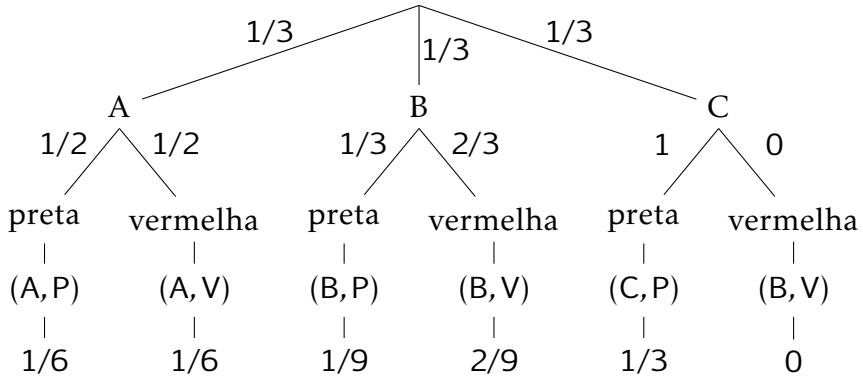


Figura 1.4: diagrama de árvore.

Como o modelo é discreto, estendemos a probabilidade para qualquer subconjunto de $\Omega_1 \times \Omega_2$ somando a probabilidade de seus elementos. Por exemplo, o evento dado por "a bola sorteada é preta" ocorre com probabilidade

$$\mathbb{P}((A,P)) + \mathbb{P}((B,P)) + \mathbb{P}((C,P)) = \frac{11}{18}.$$

Notemos que essa probabilidade não depende do número de bolas pretas na urna C, portanto, embora a medida de probabilidade em cada experimento seja uniforme a probabilidade de "a bola sorteada é preta" não é a quantidade de bolas pretas dividido pelo número total de bolas, que é um erro cometido frequentemente nesse caso. ◊

Exercício 1.23. Verifique a seguinte igualdade para o teorema da multiplicação com três eventos

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A) \cdot \mathbb{P}(B \mid A) \cdot \mathbb{P}(C \mid A \cap B) \tag{1.11}$$

e identifique seu uso no modelo probabilístico para o problema de Monty Hall, página 18.

Exercício 1.24 (teorema da multiplicação). Sejam $A_1, A_2, \ldots, A_n$ eventos de um modelo probabilístico. Prove que

$$\mathbb{P}(A_1 \cap \cdots \cap A_n) = \mathbb{P}(A_1)\mathbb{P}(A_2 \mid A_1)\mathbb{P}(A_3 \mid A_2 \cap A_1) \cdots \mathbb{P}(A_n \mid A_{n-1} \cap A_{n-2} \cap \cdots \cap A_1)$$

sempre que as probabilidades condicionais estão definidas (veja que é suficiente pedir que $\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{n-1}) > 0$).

Exercício 1.25. Sejam A, B e C eventos de um mesmo espaço amostral. Verifique que vale a seguinte igualdade

$$\mathbb{P}(C \cap A \mid B) = \mathbb{P}(C \mid A \cap B)\mathbb{P}(A \mid B)$$

sempre que as condicionais estão definidas.

1.3.1 O TEOREMA DA PROBABILIDADE TOTAL

Se E e A são eventos, com $0 < \mathbb{P}(E) < 1$, então o evento A ocorre se, e somente se, ocorre (A e E) ou (A e $\overline{E}$), e esses eventos entre parênteses são disjuntos; mais que isso $\{A \cap E, A \cap \overline{E}\}$ é uma partição de A, portanto,

$$\begin{aligned}\mathbb{P}(A) &= \mathbb{P}\Big((A \cap E) \cup (A \cap \overline{E})\Big) \\ &= \mathbb{P}(A \cap E) + \mathbb{P}(A \cap \overline{E}) \\ &= \mathbb{P}(A \mid E)\mathbb{P}(E) + \mathbb{P}(A \mid \overline{E})\mathbb{P}(\overline{E}). \end{aligned} \tag{1.12}$$

No problema de Monty Hall se o convidado fica com a porta que escolheu inicialmente, então a probabilidade de ganhar um carro é 1/3, que é a probabilidade dele ter escolhido a porta certa logo de início. Agora, vamos supor que o convidado troca de porta. Nesse caso, denotamos por A o evento "ganha o carro" e por E o evento "a porta escolhida na primeira etapa esconde o carro". Claramente, $\mathbb{P}(A \mid E) = 0$ e $\mathbb{P}(E) = 1/3$. Se a primeira escolha não era a correta então o convidado ganha o carro, ou seja, $\mathbb{P}(A \mid \overline{E}) = 1$. Com isso temos por (1.12)

$$\mathbb{P}(A) = \mathbb{P}(A \mid E)\mathbb{P}(E) + \mathbb{P}(A \mid \overline{E})\mathbb{P}(\overline{E}) = 0\frac{1}{3} + 1\frac{2}{3} = \frac{2}{3}$$

portanto, é melhor trocar de porta.

O caso geral dessa igualdade é conhecido como o Teorema da Probabilidade Total. Segue da dedução acima e usando indução em n que se $\{E_1, E_2, \ldots, E_n\}$ é um conjunto de eventos que particionam o espaço amostral Ω com $\mathbb{P}(E_i) > 0$ para todo $i \geqslant 1$, então vale

$$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \cap E_i) = \sum_{i=1}^{n} \mathbb{P}(A \mid E_i)\mathbb{P}(E_i) \tag{1.13}$$

para qualquer evento A. Deixamos a prova por conta do leitor do resultado abaixo considerando partições enumeráveis.

Teorema 1.26 (Teorema da Probabilidade Total) *Seja* $\{E_i : i \geqslant 1\}$ *uma partição (enumerável) do espaço amostral* Ω *com* $\mathbb{P}(E_i) > 0$ *para todo* i. *Então*

$$\mathbb{P}(A) = \sum_{i \geqslant 1} \mathbb{P}(A \cap E_i) = \sum_{i \geqslant 1} \mathbb{P}(A \mid E_i)\mathbb{P}(E_i)$$

para qualquer evento A.

Exemplo 1.27 (Ross, 2010). As seguradoras de automóveis classificam motoristas em *mais propensos a acidentes* e *menos propensos a acidentes*. Com isso estimam que os mais propensos são 30% da população e que esses se envolvem em acidente no período de um ano com probabilidade 0,4, enquanto que os menos propensos a

acidentes se envolvem em acidente no período de um ano com probabilidade 0,2. Denotemos por A^+ o evento definido pelos motoristas mais propensos a acidentes. Então a probabilidade de um novo segurado se envolver em acidente em um ano é

$$\mathbb{P}(A_1) = \mathbb{P}(A_1 \mid A^+)\mathbb{P}(A^+) + \mathbb{P}(A_1 \mid \overline{A^+})\mathbb{P}(\overline{A^+}) = 0{,}4 \cdot 0{,}3 + 0{,}2 \cdot 0{,}7 = 0{,}26$$

e se um novo segurado se envolve em acidente nesse prazo, a probabilidade dele ser propenso a acidentes é

$$\mathbb{P}(A^+ \mid A_1) = \frac{\mathbb{P}(A_1 \cap A^+)}{\mathbb{P}(A_1)} = \frac{\mathbb{P}(A_1 \mid A^+)\mathbb{P}(A^+)}{\mathbb{P}(A_1)} = \frac{0{,}4 \cdot 0{,}3}{0{,}26} = \frac{6}{13}.$$

Portanto, a probabilidade de ser propenso a acidente dado que se envolve em acidente em um ano é 0,46, aproximadamente. Dado que o motorista não se envolve em acidente em um ano, a probabilidade de ser propenso a acidente é $\mathbb{P}(A^+ \mid \overline{A}_1) \approx 0{,}24$. ◊

Urna de Pólya Uma urna contém duas bolas, uma branca e uma preta. Em cada instante $t \in \{1, 2, \ldots\}$ sorteamos uma bola da urna. A bola sorteada é devolvida para a urna junto com uma outra bola da mesma cor dessa sorteada. Assim, o t-ésimo sorteio ($t \geqslant 1$) ocorre com $t+1$ bolas na urna. Neste exemplo nós vamos calcular a a probabilidade com que uma bola preta é sorteada em cada instante.

Dado $t \geqslant 1$, seja P_t o evento "a t-ésima bola sorteada é preta"; se não é sorteada uma bola preta então é sorteada uma bola branca, tal evento denotamos por B_t. Certamente,

$$\mathbb{P}(P_1) = \frac{1}{2}.$$

Pelo Teorema da Probabilidade Total (eq. (1.12))

$$\mathbb{P}(P_2) = \mathbb{P}(P_2 \mid P_1)\mathbb{P}(P_1) + \mathbb{P}(P_2 \mid B_1)\mathbb{P}(B_1)$$

e se ocorre P_1, então para o segundo sorteio há 2 bolas pretas dentre 3 bolas, portanto, $\mathbb{P}(P_2 \mid P_1) = 2/3$ e, analogamente, $\mathbb{P}(P_2 \mid B_1) = 1/3$, de modo que

$$\mathbb{P}(P_2) = \frac{2}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{2}.$$

Para computar $\mathbb{P}(P_3)$ precisamos de um pouco mais de esforço. Pelo Teorema da Probabilidade Total, eq. (1.13), temos

$$\mathbb{P}(P_3) = \mathbb{P}((P_1 \cap P_2) \cap P_3) + \mathbb{P}((P_1 \cap B_2) \cap P_3) + \mathbb{P}((B_1 \cap P_2) \cap P_3) + \mathbb{P}((B_1 \cap B_2) \cap P_3)$$

e cada termo dessa soma pode ser computado pela Regra do Produto (especificamente, equação (1.11) na página 30). Por exemplo

$$\mathbb{P}(P_1 \cap P_2 \cap P_3) = \mathbb{P}(P_1)\mathbb{P}(P_2 \mid P_1)\mathbb{P}(P_3 \mid P_1 \cap P_2) = \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4}$$

de modo que temos $\mathbb{P}(P_3) =$

$$\begin{aligned}&\mathbb{P}((P_1 \cap P_2) \cap P_3) + \mathbb{P}((P_1 \cap B_2) \cap P_3) + \mathbb{P}((B_1 \cap P_2) \cap P_3) + \mathbb{P}((B_1 \cap B_2) \cap P_3) \quad (1.14)\\ &= \frac{1}{2}\cdot\frac{2}{3}\cdot\frac{3}{4} + \frac{1}{2}\cdot\frac{1}{3}\cdot\frac{2}{4} + \frac{1}{2}\cdot\frac{1}{3}\cdot\frac{2}{4} + \frac{1}{2}\cdot\frac{2}{3}\cdot\frac{1}{4} = \frac{1}{2}.\end{aligned}$$

Até aqui, $\mathbb{P}(P_1) = \mathbb{P}(P_2) = \mathbb{P}(P_3) = 1/2$.

Notemos que $\mathbb{P}(B_1 \cap B_2 \cap P_3) = \mathbb{P}(P_1 \cap P_2 \cap B_3)$ e que $\mathbb{P}(B_1 \cap P_2 \cap P_3) = \mathbb{P}(P_1 \cap B_2 \cap B_3)$ de modo que, substituindo na equação (1.14) acima

$$\mathbb{P}(P_3) = \mathbb{P}(P_1 \cap (P_2 \cap P_3)) + \mathbb{P}(P_1 \cap (B_2 \cap P_3)) + \mathbb{P}(P_1 \cap (P_2 \cap B_3)) + \mathbb{P}(P_1 \cap (B_2 \cap B_3))$$

que por (1.13) é $\mathbb{P}(P_1)$ e, então, $\mathbb{P}(P_1) = \mathbb{P}(P_3)$. Tal simetria vale para qualquer t de modo que $\mathbb{P}(P_t) = \mathbb{P}(P_1)$ para todo $t \geqslant 1$. Vamos demonstrar esse fato.

Consideremos uma sequência de eventos $E_1, E_2, \ldots, E_{t-1}, P_t$ em que para cada i, $1 \leqslant i < t$, temos $E_t \in \{P_t, B_t\}$. As 2^{t-1} possíveis sequências de eventos $E_1, \ldots, E_{t-1}$ particionam o espaço amostral de modo que, pelo Teorema da Probabilidade Total e o caso geral da Regra do Produto (Exercício 1.24 na página 30), a probabilidade de P_t é

$$\begin{aligned}&\sum_{(E_1,\ldots,E_{t-1})} \mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_{t-1} \cap P_t) =\\ &\qquad \sum_{(E_1,\ldots,E_{t-1})} \mathbb{P}(E_1) \cdot \left(\prod_{i=2}^{t-1} \mathbb{P}\left(E_i \;\middle|\; \bigcap_{j=1}^{i-1} E_j \right) \right) \cdot \mathbb{P}\left(P_t \;\middle|\; \bigcap_{j=1}^{t-1} E_j \right) \quad (1.15)\end{aligned}$$

em que a soma é sobre todas as 2^{t-1} sequências de eventos. Os somandos no lado direito da equação (1.15) são

$$\mathbb{P}(E_1)\mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_3 \mid E_1 \cap E_2)\cdots\mathbb{P}(P_t \mid E_1 \cap \cdots \cap E_{t-1}) = \prod_{i=1}^{t} \frac{n_i}{i+1} \qquad (1.16)$$

onde n_i é a quantidade de bolas da cor sorteada no i-ésimo sorteio e os denominadores são $i+1$ porque em cada sorteio o número total de bolas aumenta 1.

Fixado um instante t e supondo que até esse instante tenham sido sorteadas m bolas brancas e, portanto, $t - m$ bolas pretas, sejam $1 \leqslant t_1 < t_2 < \cdots < t_m \leqslant t$ os instantes em que ocorrem sorteio de bola branca. Quando foi realizado o primeiro sorteio de uma bola branca, havia uma bola branca de modo que $n_{t_1} = 1$, no segundo sorteio $n_{t_2} = 2$, e assim por diante, até o último sorteio de bola branca no instante t_m quando $n_{t_m} = m$. Nos momentos em que não foram sorteados bolas brancas, foram sorteados bolas pretas, sejam $1 \leqslant s_1 < s_2 < \cdots < s_{t-m} \leqslant t$ tais instantes em que ocorrem sorteio de bolas pretas. De modo análogo temos que $n_{s_1} = 1$, $n_{s_2} = 2$, ..., $n_{s_{t-m}} = t - m$.

Agora, notemos que nos numeradores no lado direito da equação (1.16) ocorrem os números $1,2,\ldots,m$ e $1,2,\ldots,t-m$ de modo que o fator determinante no cálculo é a probabilidade de ocorrer m sorteios de bolas brancas e $t-m$ sorteios de bolas pretas, a ordem não importa. Dessa observação concluímos que os somandos no lado direito da equação (1.15) são

$$\frac{1}{2}\cdot\frac{2}{3}\cdots\frac{m}{m+1}\cdot\frac{1}{m+2}\cdot\frac{2}{m+3}\cdots\frac{t-m}{t+1}=\frac{m!(t-m)!}{(t+1)!}=\frac{1}{(t+1)\binom{t}{m}}. \tag{1.17}$$

Há $\binom{t-1}{m}$ sequências $E_1,E_2,\ldots,E_{t-1}$ de eventos com m posições correspondentes ao sorteio de bola branca e cada uma tem probabilidade dada pela equação (1.17), portanto

$$\mathbb{P}(P_t)=\sum_{m=0}^{t-1}\binom{t-1}{m}\frac{1}{(t+1)\binom{t}{m}}=\sum_{m=0}^{t-1}\frac{1}{(t+1)}\frac{t-m}{t}=\frac{1}{t(t+1)}\sum_{m=1}^{t}m=\frac{1}{2}$$

ou seja, $\mathbb{P}(P_t)=1/2$ para todo $t\geqslant 1$. ◊

Um fato interessante que deduzimos do exemplo acima é que usando a equação (1.17) podemos concluir que a probabilidade de haver m bolas brancas após t-ésimo sorteio é

$$\mathbb{P}[\text{há } m \text{ bolas brancas após } t\text{-ésimo sorteio}]=\binom{t}{m}\frac{1}{(t+1)\binom{t}{m}}=\frac{1}{t+1}$$

que não depende de m.

1.3.2 O TEOREMA DE BAYES

Suponha que *probabilite* é um vírus que afeta 10% da população de estudantes universitários. Um professor de Probabilidade aplica um teste que detecta *probabilite* mas eventualmente se engana: 3% de falsos positivos e 1% de falsos negativos. Se for detectado *probabilite* em um indivíduo escolhido ao acaso, qual é a probabilidade que ele tenha o vírus? Queremos determinar $\mathbb{P}(B\mid A)$ onde A é o evento "foi detectado *probabilite*" e B o evento "tem *probabilite*" sendo conhecidos $\mathbb{P}(B)=0{,}1$, $\mathbb{P}(\overline{A}\mid B)=0{,}01$, logo $\mathbb{P}(A\mid B)=0{,}99$ e $\mathbb{P}(A\mid\overline{B})=0{,}03$.

Da definição de condicional e da Lei de Probabilidade Total

$$\mathbb{P}(B\mid A)=\frac{\mathbb{P}(A\cap B)}{\mathbb{P}(A)}=\frac{\mathbb{P}(A\mid B)\mathbb{P}(B)}{\mathbb{P}(A)}=\frac{\mathbb{P}(A\mid B)\mathbb{P}(B)}{\mathbb{P}(A\mid B)\mathbb{P}(B)+\mathbb{P}(A\mid\overline{B})\mathbb{P}(\overline{B})},$$

usando os fatos conhecidos

$$\mathbb{P}(B\mid A)=\frac{0{,}99\cdot 0{,}1}{0{,}99\cdot 0{,}1+0{,}03\cdot 0{,}9}\approx 0{,}78.$$

Agora, supondo que *probabilite* seja uma contaminação muito rara, que afeta só 1,05% da população universitária, e que o teste seja um pouco mais acurado, só há 1% de chance de falsos positivos e falsos negativos, então

$$\mathbb{P}(\,\mathsf{B} \mid \mathsf{A}\,) = \frac{0{,}99 \cdot 0{,}0105}{0{,}99 \cdot 0{,}0105 + 0{,}01 \cdot 0{,}9895} \approx 0{,}51$$

logo o teste é essencialmente tão efetivo quanto decidir lançando uma moeda. Se o vírus for ainda mais raro, digamos que apena 0,5% da população universitária tenha o vírus. Assim,

$$\mathbb{P}(\,\mathsf{B} \mid \mathsf{A}\,) = \frac{0{,}99 \cdot 0{,}005}{0{,}99 \cdot 0{,}005 + 0{,}01 \cdot 0{,}995} \approx 0{,}34$$

ou seja, se o teste do professor detectou *probabilite* em um estudante é duas vezes mais provável que o indivíduo não tenha *probabilite*. Esse resultado aparentemente paradoxal ocorre porque o número de indivíduos não contaminados pelo *probabilite* é muito grande em relação ao número de contaminados, de modo que a quantidade de falsos positivos supera a quantidade positivos verdadeiros. Se 100.000 indivíduos forem testados, esperamos que aproximadamente 99.500 não tenham *probabilite* mas que ocorram $0{,}01 \cdot 99.500 \approx 1.000$ falsos positivos; também, esperamos que 500 indivíduos tenham *probabilite* e deles $0{,}99 \cdot 500 \approx 500$ verdadeiros positivos.

Seja $\mathsf{E}_1, \ldots, \mathsf{E}_n$ uma partição do espaço amostral. Se soubermos que o evento A ocorre, então qual é a probabilidade (condicionada) com que E_j tenha ocorrido? Para todo j

$$\mathbb{P}(\,\mathsf{E}_j \mid \mathsf{A}\,) = \frac{\mathbb{P}(\,\mathsf{A} \mid \mathsf{E}_j\,)\mathbb{P}(\mathsf{E}_j)}{\mathbb{P}(\mathsf{A})}$$

usando o Teorema da Probabilidade Total obtemos

$$\mathbb{P}(\,\mathsf{E}_j \mid \mathsf{A}\,) = \frac{\mathbb{P}(\,\mathsf{A} \mid \mathsf{E}_j\,)\mathbb{P}(\mathsf{E}_j)}{\sum_{i=1}^{n} \mathbb{P}(\,\mathsf{A} \mid \mathsf{E}_i\,)\mathbb{P}(\mathsf{E}_i)}$$

para todo evento com probabilidade positiva A. Esse resultado é conhecido como Teorema de Bayes.

Teorema 1.28 (Teorema de Bayes) *Se* $\{\mathsf{E}_i : i \geqslant 1\}$ *é uma partição do espaço amostral com* $\mathbb{P}(\mathsf{E}_i) > 0$ *para todo* i *e* $\mathbb{P}(\mathsf{A}) > 0$*, então*

$$\mathbb{P}(\,\mathsf{E}_j \mid \mathsf{A}\,) = \frac{\mathbb{P}(\,\mathsf{A} \mid \mathsf{E}_j\,)\mathbb{P}(\mathsf{E}_j)}{\sum_{i \geqslant 1} \mathbb{P}(\,\mathsf{A} \mid \mathsf{E}_i\,)\mathbb{P}(\mathsf{E}_i)}.$$

1.3.3 FILTRO BAYESIANO PARA *SPAM*

O uso de técnicas baseadas no Teorema de Bayes para classificar mensagens eletrônicas (*emails*) surgiu em 1996 num trabalho de Jason Rennie chamado *Ifile* e ganhou impulso com o ensaio de Graham (2002).

Previamente identificamos algumas características das mensagens que estão classificadas em dois conjuntos, as que são *spam* e as que não são *spam*, da seguinte forma. A frequência com que ocorre uma palavra no conjunto das mensagens que são *spam* define uma probabilidade da palavra condicionada à mensagem ser um *spam* e as palavras características de *spam* são as mais relevantes de acordo com essas probabilidades. Por exemplo, nas minhas mensagens muitos dos *spam*s têm a palavra *watch* enquanto que muitos dos não *spam*s têm a palavra "reunião"; a maioria das mensagens têm a palavra "a", tantos *spam*s quanto não *spam*s, logo "a" não deve ser uma característica classificatória; separamos as palavras tais que $\mathbb{P}[\text{ palavra} \mid \texttt{spam}]$ seja bem maior que $1/2$ e $\mathbb{P}[\text{ palavra} \mid \text{não } \texttt{spam}]$ seja bem menor que $1/2$. Ao final temos algumas características classificatórias, digamos n características, que podem estar ou não estar presentes nas mensagens futuras e que vão ajudar a classificá-las.

Dadas as n características, cada mensagem fica associada uma sequência binária de $\Omega := \{0,1\}^n$ em que cada coordenada da sequência correspondente indica se a mensagem tem ou não tem uma determinada característica. A primeira coordenada, especificamente, é 1 se a mensagem é *spam* e 0 caso, contrário. Assim, $(1,0,0,1,1)$ corresponde a uma mensagem que é *spam* não tem as características 2 e 3, mas tem as características 4 e 5. Denotemos por S o evento "*spam*" e por C_i o evento "tem a característica i". Na classificação prévia contamos a quantidade k_i de mensagens *spam* que têm a característica i dentre as K mensagens classificadas. Também, determinamos a quantidade ℓ_i de mensagens não *spam* que têm a característica i dentre as L mensagens classificadas. Com essa informação determinamos

$$\mathbb{P}(\,\mathsf{C}_i \mid \mathsf{S}\,) = \frac{k_i}{\mathsf{K}} \quad \text{e} \quad \mathbb{P}(\,\mathsf{C}_i \mid \overline{\mathsf{S}}\,) = \frac{\ell_i}{\mathsf{L}}$$

para cada característica $i > 1$. Pelo Teorema de Bayes, a probabilidade de uma mensagem que apresenta a característica i ser *spam* é

$$p_i := \mathbb{P}(\,\mathsf{S} \mid \mathsf{C}_i\,) = \frac{\mathbb{P}(\,\mathsf{C}_i \mid \mathsf{S}\,)\mathbb{P}(\mathsf{S})}{\mathbb{P}(\,\mathsf{C}_i \mid \mathsf{S}\,)\mathsf{P}(\mathsf{S}) + \mathbb{P}(\,\mathsf{C}_i \mid \overline{\mathsf{S}}\,)\mathbb{P}(\overline{\mathsf{S}})}.$$

Se assumimos que, a priori, temos a mesma chance de receber um *spam* quanto um não *spam* então $\mathbb{P}(\mathsf{S}) = \mathbb{P}(\overline{\mathsf{S}}) = 1/2$ e assumindo $\mathsf{K} = \mathsf{L}$, para simplificar, a equação acima se resume a

$$p_i = \frac{(k_i/\mathsf{K})\mathbb{P}(\mathsf{S})}{(k_i/\mathsf{K})\mathbb{P}(\mathsf{S}) + (\ell_i/\mathsf{L})\mathbb{P}(\overline{\mathsf{S}})} = \frac{k_i}{k_i + \ell_i}.$$

Recebida uma mensagem como classificá-la? Determinamos quais das n características estão presentes na mensagem, digamos que para algum subconjunto de índices I a mensagem tem C_i para todo $i \in \mathsf{I}$ e com essa informação calculamos $\mathbb{P}\big(\,\mathsf{S} \,\big|\, \bigcap_{i\in\mathsf{I}} \mathsf{C}_i\,\big)$. Se essa probabilidade for maior que um limiar $\varepsilon \in (0,1)$ estabelecido, então a mensagem recebida é classificada como *spam*, senão é classificada como não

spam. No que segue vamos provar que a probabilidade dessa mensagem ser *spam* é dada por

$$\mathbb{P}\left(S \,\middle|\, \bigcap_{i\in I} C_i \right) = \frac{\prod_{i\in I} p_i}{\prod_{i\in I} p_i + \prod_{i\in I}(1-p_i)}. \tag{1.18}$$

Para isso vamos assumir que valem

$$\mathbb{P}\left(\bigcap_{i\in I} C_i \,\middle|\, S \right) = \prod_{i\in I} \mathbb{P}(\, C_i \mid S\,) \tag{1.19}$$

$$\mathbb{P}\left(\bigcap_{i\in I} C_i \,\middle|\, \overline{S} \right) = \prod_{i\in I} \mathbb{P}(\, C_i \mid \overline{S}\,), \tag{1.20}$$

para todo $I \subset \{2,3,\ldots,n\}$, o que pode não ser uma hipótese muito realista. Usando a definição de probabilidade condicional

$$\mathbb{P}\left(S \,\middle|\, \bigcap_{i\in I} C_i \right) = \frac{\mathbb{P}(\,\bigcap_{i\in I} C_i \mid S\,)\mathbb{P}(S)}{\mathbb{P}(\bigcap_{i\in I} C_i)}$$

e da lei da probabilidade total

$$\frac{\mathbb{P}(\,\bigcap_{i\in I} C_i \,\big|\, S\,)\mathbb{P}(S)}{\mathbb{P}(\bigcap_{i\in I} C_i)} = \frac{\mathbb{P}(\,\bigcap_{i\in I} C_i \,\big|\, S\,)\mathbb{P}(S)}{\mathbb{P}(\,\bigcap_{i\in I} C_i \,\big|\, S\,)\mathbb{P}(S) + \mathbb{P}(\,\bigcap_{i\in I} C_i \mid \overline{S}\,)\mathbb{P}(\overline{S})}$$

e pelas equações (1.19) e (1.20)

$$\frac{\mathbb{P}(\,\bigcap_{i\in I} C_i \,\big|\, S\,)\mathbb{P}(S)}{\mathbb{P}(\,\bigcap_{i\in I} C_i \,\big|\, S\,)\mathbb{P}(S) + \mathbb{P}(\,\bigcap_{i\in I} C_i \mid \overline{S}\,)\mathbb{P}(\overline{S})} = \frac{\prod_{i\in I}\mathbb{P}(\, C_i \mid S\,)\mathbb{P}(S)}{\prod_{i\in I}\mathbb{P}(\, C_i \mid S\,)\mathbb{P}(S) + \prod_{i\in I}\mathbb{P}(\, C_i \mid \overline{S}\,)\mathbb{P}(\overline{S})}$$

e, usando que $\mathbb{P}(\, C_i \mid S\,) = \mathbb{P}(\, S \mid C_i\,)\mathbb{P}(C_i)/\mathbb{P}(S)$ e a igualdade análoga para $\overline{S}$

$$\frac{\prod_{i\in I}\mathbb{P}(\, C_i \mid S\,)\mathbb{P}(S)}{\prod_{i\in I}\mathbb{P}(\, C_i \mid S\,)\mathbb{P}(S) + \prod_{i\in I}\mathbb{P}(\, C_i \mid \overline{S}\,)\mathbb{P}(\overline{S})} = \frac{\prod_{i\in I}\mathbb{P}(\, S \mid C_i\,)}{\prod_{i\in I}\mathbb{P}(\, S \mid C_i\,) + \prod_{i\in I}\mathbb{P}(\, \overline{S} \mid C_i\,)}$$

donde seque a equação (1.18).

As hipóteses assumidas nas equações (1.19) e (1.20) significam, grosso modo, que o conhecimento de algumas das características não dá nenhuma pista sobre a presença ou não das outras características; estamos assumindo independência dos eventos e o significado preciso disso é o assunto da próxima seção.

1.3.4 INDEPENDÊNCIA DE EVENTOS

Se um dado é lançado duas vezes, então temos

$$\mathbb{P}[\text{ a soma é 7 } \mid \text{ o primeiro resultado é 4 }] = \frac{1}{6} = \mathbb{P}[\text{a soma é 7}]$$

entretanto

$$\mathbb{P}[\text{ a soma é } 12 \mid \text{o primeiro resultado é } 4\ ] = 0 \neq \mathbb{P}[\text{a soma é } 12\,].$$

O condicionamento de ocorrência de um evento A à ocorrência de B pode afetar ou não a probabilidade de ocorrência de A. Definimos que o evento A é independente do evento B se

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$$

Notemos que se A é independente de B então B é independente de A de modo que dizemos A e B são **eventos independentes**.

Se os eventos têm probabilidade não nula então decorre da definição acima que a independência de A e B equivale a $\mathbb{P}(A \mid B) = \mathbb{P}(A)$ e $\mathbb{P}(B \mid A) = \mathbb{P}(B)$. É imediato da definição o seguinte fato.

Proposição 1.29 *Todo evento* A *de um modelo probabilístico é independente do evento certo* Ω *e do evento impossível* $\varnothing$. □

A independência dos eventos A e B resulta na independência entre seus complementos, como enunciado a seguir.

Proposição 1.30 *Se* A *e* B *são eventos independentes de um modelo probabilístico, então* A *e* $\overline{B}$ *são eventos independentes,* $\overline{A}$ *e* B *são eventos independentes e* $\overline{A}$ *e* $\overline{B}$ *são eventos independentes.*

Demonstração. Deduzimos de $\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap \overline{B})$, usando a independência de A e B, que

$$\mathbb{P}(A \cap \overline{B}) = \mathbb{P}(A) - \mathbb{P}(A \cap B) = \mathbb{P}(A) - \mathbb{P}(A)\mathbb{P}(B) = \mathbb{P}(A)(1 - \mathbb{P}(B)) = \mathbb{P}(A)\mathbb{P}(\overline{B})$$

portanto são eventos independentes. Os outros casos são demonstrados de modo análogo. □

Para investigar o caso de três eventos, voltemos ao experimento de um dado lançado duas vezes. Sejam A o evento "a soma do dois lançamentos é 7", B o evento "o primeiro lançamento resulta 4" e C o evento "o segundo lançamento resulta 2". Como vimos, A e B são eventos independentes. Por razão análoga A e C são independentes. Porém $\mathbb{P}(A \mid B \cup C) = 2/11 \neq \mathbb{P}(A)$ e $\mathbb{P}(A \mid B \cap C) = 0 \neq \mathbb{P}(A)$, ou seja, A não é independente de [B *ou* C] e não é independente de [B *e* C].

Para três eventos, digamos A, B e C, queremos que A seja independente do par de eventos $\{B, C\}$ quando o conhecimento de qualquer informação a respeito da ocorrência de B, de C, ou de uma combinação deles pelas operações elementares de conjuntos não altere a probabilidade de ocorrer A.

Exercício 1.31. Assuma, como definição de "A *é independente de* $\{B, C\}$" se vale a equação (1.21) a seguir

$$\mathbb{P}(A \mid B \cap C) = \mathbb{P}(A \mid B) = \mathbb{P}(A \mid C) = \mathbb{P}(A). \tag{1.21}$$

Prove que se A é independente de $\{B, C\}$ então A é independente de cada um dos eventos da família

$$\Big\{\varnothing, B, \overline{B}, C, \overline{C}, B \cup C, B \cup \overline{C}, \overline{B} \cup C, \overline{B} \cup \overline{C}, B \cap C, B \cap \overline{C},$$
$$\overline{B} \cap C, \overline{B} \cap \overline{C}, \big(B \cap \overline{C}\big) \cup \big(\overline{B} \cap C\big), \big(\overline{B} \cup C\big) \cap \big(B \cap \overline{C}\big), \Omega\Big\}$$

que chamamos de **espaço de eventos gerado** (σ-álgebra gerada) por $\{B, C\}$.

Em vista disso, definimos que A **é independente de** $\{B, C\}$ se for independente de todo evento do espaço de eventos gerado por $\{B, C\}$. Tal definição é equivalente a (veja a equação (1.21)): A *é independente de* $\{B, C\}$ *se, e somente se, é independente de* B, *é independente de* C, *e é independente de* $B \cap C$, isto é,

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B),\ \mathbb{P}(A \cap C) = \mathbb{P}(A)\mathbb{P}(C) \text{ e } \mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B \cap C).$$

Ademais, notemos que essa definição é compatível com a definição de "A independente de B" dada anteriormente pois, pelas proposições 1.29 e 1.30, o evento A é independente de todo evento do espaço de eventos gerado por $\{B\}$, o qual é $\{\varnothing, B, \overline{B}, \Omega\}$.

Em geral, estamos interessados no caso em que cada evento é independente dos outros dois eventos restantes e, nesse caso, dizemos que os eventos A, B e C são **mutuamente independentes** o que é equivalente a

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B),\ \mathbb{P}(A \cap C) = \mathbb{P}(A)\mathbb{P}(C),\ \mathbb{P}(B \cap C) = \mathbb{P}(B)\mathbb{P}(C) \text{ e}$$
$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C).$$

Consideremos três lançamentos de uma moeda equilibrada e os eventos E_{12} dado por "o resultado do primeiro e do segundo lançamentos coincidem", E_{13} dado por "o resultado do primeiro e do terceiro coincidem" e E_{23} dado por "o resultado do segundo e do terceiro coincidem". Cada um desses eventos tem probabilidade 1/2. Os eventos são independentes quando tomados dois-a-dois: $\mathbb{P}(E_{12} \cap E_{13}) = \mathbb{P}(\{(Ca, Ca, Ca), (Co, Co, Co)\}) = 1/4$ e, analogamente, os eventos $E_{12} \cap E_{23}$ e $E_{13} \cap E_{23}$ têm probabilidade 1/4. Entretanto esses eventos não são mutuamente independentes pois $\mathbb{P}(E_{12} \cap E_{13} \cap E_{23}) = 1/4$ enquanto que $\mathbb{P}(E_{12})\mathbb{P}(E_{13})\mathbb{P}(E_{23}) = 1/8$.

Agora, num lançamento de dados tomamos os eventos $A = \{1, 2, 3, 4\}$ e $B = C = \{4, 5, 6\}$. Os eventos B e C não são independentes. Também A e B não são independentes pois $\mathbb{P}(A \cap B) = 1/6$ enquanto que $\mathbb{P}(A)\mathbb{P}(B) = 1/3$, logo A e C não são eventos independentes. Porém $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C)$.

Uma família enumerável de eventos $\mathcal{E} = \{E_n : n \in I\}$ é dita **mutuamente independente** se para todo subconjunto finito $J \subseteq I$ vale que

$$\mathbb{P}\left(\bigcap_{\ell \in J} E_\ell\right) = \prod_{\ell \in J} \mathbb{P}(E_\ell).$$

Para $k \in \{2, \ldots, n\}$ fixo, dizemos que a coleção $\mathcal{E}$ é k-**a**-k **independente** se todo subconjunto de índices $J \subseteq I$ com $|J| \leqslant k$ define uma subcoleção de eventos mutuamente independentes.

Independência condicional Dizemos que A_1 e A_2 são condicionalmente independentes dado B se

$$\mathbb{P}(A_2 \cap A_1 \mid B) = \mathbb{P}(A_1 \mid B)\mathbb{P}(A_2 \mid B).$$

Essa definição estende-se naturalmente, como acima, para um coleção com mais que dois eventos.

As equações (1.19) e (1.20) no exemplo para filtros anti-*spam* pede que as características $C_1, \ldots, C_n$, que são usadas para classificar as mensagens, sejam independentes quando condicionamos à ocorrência de *spam* e quando condicionamos à ocorrência de não *spam*, respectivamente.

No contexto do Exemplo 1.27, página 31, qual a probabilidade de um motorista se envolver num acidente no segundo ano dado que tenha se envolvido em acidente no primeiro ano de contrato? Denotemos por A_2 o evento "acidente no 2º ano de contrato". Assumiremos que A_1 e A_2 são condicionalmente independentes dado A^+, ou seja, $\mathbb{P}(A_2 \cap A_1 \mid A^+) = \mathbb{P}(A_1 \mid A^+)\mathbb{P}(A_2 \mid A^+)$. Se definirmos a medida de probabilidade $\mathbb{Q}(X) := \mathbb{P}(X \mid A_1)$, queremos determinar $\mathbb{Q}(A_2)$. Pelo Teorema da Probabilidade Total $\mathbb{Q}(A_2) = \mathbb{Q}(A_2 \mid A^+)\mathbb{Q}(A^+) + \mathbb{Q}(A_2 \mid \overline{A^+})\mathbb{Q}(\overline{A^+})$. Mas

$$\mathbb{Q}(A_2 \,\big|\, A^+) = \frac{\mathbb{Q}(A_2 \cap A^+)}{\mathbb{Q}(A^+)} = \frac{\mathbb{P}(A_2 \cap A^+ \mid A_1)}{\mathbb{P}(A^+ \mid A_1)} = \mathbb{P}(A_2 \,\big|\, A^+ \cap A_1) = \mathbb{P}(A_2 \,\big|\, A^+)$$

em que a última igualdade segue da independência condicional assumida (verifique). Lembremos que os motoristas propensos a acidentes se envolvem em acidente no período de um ano com probabilidade 0,4, logo $\mathbb{P}(A_2 \mid A^+) = 0{,}4$. Ainda, calculamos no Exemplo 1.27 que $\mathbb{Q}(A^+) = 6/13$. Desse modo temos que

$$\mathbb{Q}(A_2) = \mathbb{Q}(A_2 \,\big|\, A^+)\mathbb{Q}(A^+) + \mathbb{Q}(A_2 \mid \overline{A^+})\mathbb{Q}(\overline{A^+}) = 0{,}4 \cdot \frac{6}{13} + 0{,}2 \cdot \frac{7}{13} \approx 0{,}29.$$

Exemplo 1.32. Suponhamos que numa caixa há duas moedas, uma delas com duas caras e a outra é uma moeda comum. Uma moeda é sorteada e lançada duas vezes. Sejam A e B os eventos "o primeiro lançamento é cara" e "o segundo lançamento é

cara", respectivamente. Condicionados ao evento C definido por "a moeda normal foi a escolhida" os eventos A e B são independentes:

$$\mathbb{P}(A \cap B \mid C) = \frac{1}{4} = \frac{1}{2} \cdot \frac{1}{2} = \mathbb{P}(A \mid C) \cdot \mathbb{P}(B \mid C).$$

Entretanto os eventos A e B não são independentes pois

$$\mathbb{P}(A) = \mathbb{P}(A \mid C)\mathbb{P}(C) + \mathbb{P}(A \mid \overline{C})\mathbb{P}(\overline{C}) = \frac{1}{2} \cdot \frac{1}{2} + 1\frac{1}{2} = \frac{3}{4}$$
$$\mathbb{P}(B) = \mathbb{P}(B \mid C)\mathbb{P}(C) + \mathbb{P}(B \mid \overline{C})\mathbb{P}(\overline{C}) = \frac{1}{2} \cdot \frac{1}{2} + 1\frac{1}{2} = \frac{3}{4}$$

porém

$$\begin{aligned}\mathbb{P}(A \cap B) &= \mathbb{P}(A \cap B \mid C)\mathbb{P}(C) + \mathbb{P}(A \cap B \mid \overline{C})\mathbb{P}(\overline{C}) \\ &= \mathbb{P}(A \mid C)\mathbb{P}(B \mid C)\mathbb{P}(C) + \mathbb{P}(A \mid \overline{C})\mathbb{P}(B \mid \overline{C})\mathbb{P}(\overline{C}) = \frac{5}{8}\end{aligned}$$

por causa da independência condicional, usada para deduzir a segunda linha da equação acima. ◊

Agora, retomemos o exemplo do vírus da *probabilite*, dado na página 35, que é um vírus que contamina 0,5% da população universitária e que o teste para detectar o vírus tem 1% de chance de acusar falsos positivos e falsos negativos. Vimos que $\mathbb{P}(B \mid A) \approx 0{,}34$ onde A é o evento "foi detectado *probabilite*" e B o evento "tem *probabilite*". Agora, suponha que o estudante estava com dor de cabeça (o teste foi feito em véspera de prova). É sabido que 95% dos indivíduos que tem *probabilite* apresentam dor de cabeça, enquanto que 10% da população não contaminada apresenta dor de cabeça; também é sabido que o evento "ter dor de cabeça", que denominamos C, não afeta a precisão do teste no sentido de que A e C são condicionalmente independentes $\mathbb{P}(A \cap C \mid B) = \mathbb{P}(A \mid B)\mathbb{P}(C \mid B)$. Com esse fato, a probabilidade de ter *probabilite* dado que o teste deu positivo e o estudante tem dor de cabeça é

$$\begin{aligned}\mathbb{P}(B \mid A \cap C) &= \frac{\mathbb{P}(A \cap C \mid B)\mathbb{P}(B)}{\mathbb{P}(A \cap C \mid B)\mathbb{P}(B) + \mathbb{P}(A \cap C \mid \overline{B})\mathbb{P}(\overline{B})} \\ &= \frac{\mathbb{P}(A \mid B)\mathbb{P}(C \mid B)\mathbb{P}(B)}{\mathbb{P}(A \mid B)\mathbb{P}(C \mid B)\mathbb{P}(B) + \mathbb{P}(A \mid \overline{B})\mathbb{P}(C \mid \overline{B})\mathbb{P}(\overline{B})} \\ &= \frac{0{,}99 \cdot 0{,}95 \cdot 0{,}005}{0{,}99 \cdot 0{,}95 \cdot 0{,}005 + 0{,}01 \cdot 0{,}1 \cdot 0{,}995} \approx 0{,}82.\end{aligned}$$

1.3.5 REPETIÇÕES INDEPENDENTES DE UM EXPERIMENTO

Dados os espaços de probabilidade discretos $(\Omega_i, \mathbb{P}_i)$, para $1 \leqslant i \leqslant n$, podemos definir um espaço de probabilidade discreto cujo espaço amostral é dado pelas

sequências $(\omega_1, \omega_2, \ldots, \omega_n)$ do produto cartesiano $\Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ e a medida de probabilidade em cada ponto amostral é

$$\mathbb{P}(\omega) := \mathbb{P}_1(\omega_1)\mathbb{P}_2(\omega_2)\cdots\mathbb{P}_n(\omega_n)$$

para todo $\omega = (\omega_1, \omega_2, \ldots, \omega_n) \in \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$, a qual se estende do modo usual para todo $A \subseteq \Omega$. Esse espaço de probabilidade é chamado **espaço produto** e é o modelo probabilístico de um experimento sendo repetido n vezes sob condições idênticas. Não é difícil verificar que

$$\sum_{\omega\in\Omega} \mathbb{P}(\omega) = \sum_{\omega_1\in\Omega_1} \sum_{\omega_2\in\Omega_2} \cdots \sum_{\omega_n\in\Omega_n} \mathbb{P}_1(\omega_1)\mathbb{P}_2(\omega_2)\cdots\mathbb{P}_n(\omega_n) = 1$$

o que garante que o espaço produto de espaços discretos é um espaço de probabilidade discreto. Além disso, para $A_i \subseteq \Omega_i$, $1 \leqslant i \leqslant n$, temos $\mathbb{P}(A_1 \times A_2 \times \cdots \times A_n) = \mathbb{P}_1(A_1)\mathbb{P}_2(A_2)\cdots\mathbb{P}_n(A_n)$ (verifique).

Um modelo probabilístico para n lançamentos de uma moeda equilibrada em que os resultados dos lançamentos são mutuamente independentes é dado pelo espaço produto $(\Omega^n, \mathbb{P}^n)$, em que $(\Omega, \mathbb{P})$ é o modelo para um lançamento dado no Exemplo 1.7.

Exemplo 1.33. Sortear um número inteiro entre 0 e 999 é um experimento aleatório cujo espaço amostral é o conjunto dos números naturais até 999 e cuja medida de probabilidade é a uniforme, isto é, cada número ocorre com probabilidade 1/1.000.

De outro modo, se temos disponível os algarismos $0, 1, \ldots, 9$ podemos gerar uniformemente um número entre 0 e 999 se sortearmos $d_1, d_2, d_3 \in \{0, 1, \ldots, 9\}$, com os resultados mutuamente independentes, e tomarmos $n := d_1 \times 10^2 + d_2 \times 10^1 + d_3 \times 10^0$, então o espaço amostral é dado pelas ternas de algarismos (d_1, d_2, d_3) e a probabilidade de n é $(1/10)^3 = 1/1.000$.

Há uma correspondência biunívoca (dada por $n = n(d_1, d_2, d_3)$) entre esses dois espaços amostrais e que preserva a probabilidade dos pontos amostrais, com isso os eventos aleatórios têm a mesma probabilidade no dois modelos e sortear uniformemente um número de três algarismos e sortear uniformemente cada um de três algarismos são experimentos aleatórios equivalentes. ◊

Exercício 1.34. Considere um experimento aleatório modelado por $(\Omega, \mathbb{P})$. Em n repetições desse experimento, sejam $A_1, A_2, \ldots, A_n$ eventos de Ω^n tais que a j-ésima rodada sozinha determina se A_j ocorre, ou seja, existe um $E_j \subseteq \Omega$ tal que $A_j = \Omega^{j-1} \times E_j \times \Omega^{n-j}$. Se em Ω^n tomarmos a medida produto, então os eventos $A_1, A_2, \ldots, A_n$ são mutuamente independentes (dica: comece com a prova de que os eventos são dois a dois independentes.)

Repetições independentes de um algoritmo Uma técnica importante na utilidade de certos tipos de algoritmos probabilísticos é a possibilidade de reduzir o erro das respostas executando o algoritmo com a mesma entrada várias vezes: se um algoritmo erra com probabilidade ε, então em duas execuções errará com probabilidade ε^2, em $r \in \mathbb{N}$ execuções a probabilidade de erro é ε^r.

Nos algoritmos probabilísticos assumimos independência dos resultados nos sorteios. Primeiro, assumimos que todos os sorteios feitos durante uma rodada do algoritmo são independentes, isto é, o resultado de um ou mais sorteios não altera a probabilidade do resultado de um outro sorteio. Também, assumimos que os sorteios feitos durante uma execução não altera a probabilidade dos sorteios nas outras execuções de modo que se justifica e decaimento exponencial no erro descrito no parágrafo anterior.

Lembremos que o Algoritmo 1 erra quando declara um polinômio não nulo como nulo, o que pode ter ocorrido pela escolha de uma raiz do polinômio pelo algoritmo. Fixada uma instância do problema, suponhamos r rodadas independentes desse algoritmo (com a mesma instância). Se em alguma dessas rodadas o o Algoritmo 1 responde *não*, então essa é a resposta definitiva para o problema com essa instância. Se todas as r respostas forem *sim* então a resposta definitiva é *sim* e essa resposta estará errada se todas as r respostas de cada rodada estiverem erradas, o que ocorre com probabilidade 4^{-r}, pela independência dos eventos "resposta errada na i-ésima execução". Assim, se precisamos de uma garantia na resposta para o problema, por exemplo com probabilidade de erro menor que ε, para algum $\varepsilon > 0$ fixo, então basta escolher r de modo que $4^{-r} < \varepsilon$, ou seja, $r > \log_2 \sqrt{1/\varepsilon}$ rodadas.

Proposição 1.35 *Dado um real positivo* ε*, o problema* teste de identidade de polinômios em uma variável *pode ser resolvido por um algoritmo aleatorizado com probabilidade de erro menor que* ε*.* □

1.3.6 GERADOR DE NÚMEROS ALEATÓRIOS

Suponhamos que temos disponível uma fonte que gera bits aleatórios de modo uniforme e independente e queremos projetar um algoritmo que recebe um inteiro positivo M e nos devolve uma escolha aleatória em $\{0, 1, \ldots, M-1\}$. Se M é uma potência de 2, digamos que $M = 2^k$, então a resposta é simples: basta sortearmos k bits aleatórios $d_0, d_1, \ldots d_{k-1} \in \{0, 1\}$ que o resultado é o número $\sum_{i=0}^{k-1} d_i 2^i$ no domínio desejado com probabilidade $1/M$. No caso em que M não é potência de 2, digamos que $2^{k-1} < M < 2^k$ (o que significa que precisamos de $k = \lfloor \log_2 M \rfloor + 1$ bits aleatório) usamos o mesmo processo descrito no parágrafo anterior com a exceção de que se o resultado for maior ou igual a M, o processo é reiniciado e é repetido até que um

número entre 0 e $M-1$ seja obtido.

Instância: inteiro positivo $M \geqslant 2$.
Resposta: uma escolha aleatória uniforme em $\{0,1,\ldots,M-1\}$.

1 **repita**
2 | **para cada** $i \in \{0,\ldots,\lfloor \log_2 M \rfloor\}$ **faça** $d_i \xleftarrow{R} \{0,1\}$;
3 | $N \leftarrow \sum_i d_i 2^i$;
4 **até que** $N < M$;
5 **responda** N.

Algoritmo 2: gerador de números aleatórios.

O resultado das escolhas aleatórias na linha 2 do Algoritmo 2, digamos que seja a sequência $d_{k-1}d_{k-2}\ldots d_0$, é um evento elementar do espaço produto $(\{0,1\}^k, \mathbb{P}^k)$, em que $\mathbb{P}(0) = \mathbb{P}(1) = 1/2$, que tem probabilidade $\mathbb{P}^k(d_{k-1}d_{k-2}\ldots d_0) = (1/2)^k$. Essa sequência é a representação binária do número $\sum_{i=0}^{k-1} d_i 2^i$ que pertence ao conjunto $\{0,1,\ldots,2^k-1\}$.

Por exemplo, se $M = 7$ então $k = 3$. Com três bits $d_2d_1d_0$ temos as representações binárias dos naturais de 0 a 7. O laço da linha 1 gera qualquer um desses números com a mesma probabilidade, a saber $1/2^3 = 1/8$. Porém o algoritmo só termina se o sorteio for diferente de 7, isto é, não ocorre o evento $d_2d_1d_0 = 111$. Dado que esse evento não ocorre, qual a probabilidade do algoritmo responder 4? Usando probabilidade condicional $\mathbb{P}[\, N = 4 \mid N \neq 7 \,] = (1/8)/(7/8) = 1/7$ e, de fato, o algoritmo escolhe qualquer número em $\{0,\ldots,6\}$ com probabilidade 1/7.

No caso geral, definimos o evento $A = \{0,1,\ldots,M-1\}$ e para qualquer $t \in A$ a probabilidade do algoritmo responder t é dada por

$$\mathop{\mathbb{P}}_{N \in_R \{0,\ldots,2^k-1\}} [\, N = t \mid N \in A \,] = \frac{\mathbb{P}(\{t\} \cap A)}{\mathbb{P}(A)} = \frac{(1/2)^k}{M/2^k} = \frac{1}{M}.$$

Portanto, se o algoritmo termina, ou seja, dado que o sorteio N satisfaz $N < M$, então ele responde com um número entre 0 e $M-1$ de modo uniforme. Resta provarmos que o algoritmo termina, isto é, eventualmente a condição $N < M$ na linha 4 é satisfeita.

Fixamos uma instância M com $2^{k-1} < M < 2^k$ e $k = \lfloor \log_2 M \rfloor + 1$ é o número de bits sorteados. A probabilidade com que uma rodada do laço da linha 1 resulte em um inteiro N que pertença ao conjunto $\overline{A} = \{M,\ldots,2^k-1\}$ é

$$\mathbb{P}(\overline{A}) = \frac{2^k - M}{2^k} = 1 - \frac{M}{2^k}.$$

Definimos para todo $n \geqslant 1$ o evento A_n por "o algoritmo leva mais que n rodadas para terminar". Tal evento ocorre se nas n primeiras tentativas do laço na linha 1 ocorre um sorteio em $\overline{A}$ e daí pra diante pode ocorrer qualquer um dos dois casos, A

ou $\overline{A}$, logo $\mathbb{P}(A_n) = (1 - M/2^k)^n$ pela independência da ocorrência dos eventos $\overline{A}$ em cada rodada do laço.

Os eventos A_n formam uma sequência decrescente $A_n \supset A_{n+1}$ e $\lim_{n\to\infty} A_n = \bigcap_{n\geqslant 1} A_n$ é o evento "o algoritmo não termina", cuja probabilidade é, por continuidade (equação (1.5) na página 20),

$$\mathbb{P}[\text{o algoritmo não termina}] = \mathbb{P}\left(\lim_{n\to\infty} A_n\right) = \lim_{n\to\infty} \mathbb{P}(A_n) = \lim_{n\to\infty}\left(1 - \frac{M}{2^k}\right)^n = 0$$

pois $M < 2^k$, portanto, o algoritmo termina com probabilidade 1.

Notemos que, diferente do exemplo do Algoritmo 1, nesse caso a resposta está sempre correta e a aleatoriedade influencia na duração das rodadas, isto é, no tempo que leva para o algoritmo terminar. Esse algoritmo não só termina como, de fato, termina rápido, em poucas rodadas do laço da linha 1 com alta probabilidade. De $M > 2^{k-1}$ temos $\mathbb{P}(\overline{A}) < 1/2$, portanto, em n rodadas do laço todos os inteiros sorteados pertencem a $\overline{A}$ com probabilidade menor que 2^{-n}. A probabilidade de não terminar em, por exemplo, 4 rodadas é menor que 0,07, em 10 rodadas é menor que 0,00098.

1.4 EXERCÍCIOS

Exercício 1.36. Sejam A, B e C eventos aleatórios. Determine expressões que envolvem somente conjuntos e operações sobre conjuntos para

1. somente A ocorre;
2. A e B mas não C ocorrem;
3. os três eventos ocorrem;
4. pelo menos um evento ocorre;
5. pelo menos dois eventos ocorrem;
6. exatamente um evento ocorre;
7. exatamente dois eventos ocorrem;
8. nenhum evento ocorre;
9. não mais que dois eventos ocorrem.

Exercício 1.37 (desigualdade triangular). Prove que para eventos F e G de um espaço de probabilidade vale que

$$\mathbb{P}(F \triangle G) = \mathbb{P}(F \triangle H) + \mathbb{P}(H \triangle G)$$

para todo evento H.

Exercício 1.38. Considere o lançamento repetido de uma moeda equilibrada até sair coroa, como descrito no Exemplo 1.14. Com que probabilidade o número de lançamentos é par?

Exercício 1.39 (Princípio da inclusão–exclusão). Prove que para eventos A_1, A_2, ..., A_n vale

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \mathbb{P}(A_i) - \sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathbb{P}\big(A_i \cap A_j\big) + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=j+1}^{n} \mathbb{P}\big(A_i \cap A_j \cap A_k\big) + \cdots + (-1)^n \mathbb{P}\left(\bigcap_{i=1}^{n} A_i\right).$$

Exercício 1.40. Prove que o Corolário 1.6, página 13, admite a seguinte extensão: para qualquer conjunto enumerável $\{E_i : i \geqslant 1\}$ de eventos num espaço discreto vale

$$\mathbb{P}\left(\bigcup_{i\geqslant 1} E_i\right) \leqslant \sum_{i\geqslant 1} \mathbb{P}(E_i).$$

Exercício 1.41. Seja p primo e tome $\Omega := \{1, 2, \ldots, p\}$ com medida uniforme $\mathbb{P}(A) = |A|/p$, para todo $A \subseteq \Omega$. Prove que se A e B são independentes então pelo menos um desses eventos deve ser $\varnothing$ ou Ω.

Exercício 1.42. Defina um sistema de codificação com $\mathcal{P} = \{\alpha, \beta\}$, $\mathcal{C} = \{a, b\}$ e $\mathcal{K} = \{0, 1\}$ tais que $\mathbb{P}_{\mathcal{K}}(0) = 1/10$ e $\mathbb{P}_{\mathcal{K}}(1) = 9/10$. A codificação $E_k(m)$, para cada $m \in \{\alpha, \beta\}$ é dada na Tabela 1.3 abaixo. Prove que o sistema não tem sigilo perfeito.

	$E_0(m)$	$E_1(m)$
α	a	b
β	b	a

Tabela 1.3: função de codificação.

Exercício 1.43 (Teorema de Shannon). Prove o seguinte resultado: dados um sistema de codificação com $\mathcal{P}, \mathcal{K}, \mathcal{C}$ finitos e $|P| = |K| = |C|$ e dada uma medida de probabilidade $\mathbb{P}_{\mathcal{P}}$ sobre $\mathcal{P}$ tal que $\mathbb{P}_{\mathcal{P}}(P) > 0$, para todo $P \in \mathcal{P}$, esse sistema tem sigilo perfeito se e somente se as chaves são equiprováveis, com probabilidade $1/|K|$, e existe um único $K \in \mathcal{K}$ tal que $E_K(P) = C$, para todo $P \in \mathcal{P}$ e todo $C \in \mathcal{C}$.

Exercício 1.44. Considere $\Omega = \{0, 1\}^n$ e suponha que x é o resultado de um sorteio uniforme em Ω e y é o resultado de um sorteio em Ω em que os resultados ocorrem de acordo com alguma medida de probabilidade, possivelmente diferente da uniforme. Os sorteios são independentes. Mostre que $y \oplus x$ (o símbolo $\oplus$ denota operação ou-exclusivo coordenada-a-coordenada) é qualquer elemento de Ω com probabilidade $(1/2)^n$.

Exercício 1.45. Vamos provar uma generalização do Exercício 1.44. Seja $(G, \otimes)$ um grupo abeliano finito, T um conjunto finito e $f\colon T \to G$ uma função qualquer. Suponha que x é o resultado de um sorteio uniforme em G e y é o resultado de um sorteio em T em que os resultados ocorrem de acordo com alguma medida de probabilidade, possivelmente diferente da uniforme, e os sorteios são independentes. Prove que $x \otimes f(y)$ é qualquer elemento do grupo G com probabilidade uniforme. Prove também que para quaisquer $i \in T, j \in G$ os eventos definidos por "$y = i$" e "$x \otimes f(y)$=j" são independentes.

Exercício 1.46. No exercício anterior, suponha que f seja invertível e T um conjunto de textos legíveis. A codificação de um texto legível $m \in T$ é feita transformando-o num elemento do grupo com $f(t)$, sorteando uma chave $k \in G$ uniformemente e calculando $c = k \otimes f(t)$. A decodificação é feita conhecendo-se a chave k e calculando $f^{-1}(c \otimes (-k))$, em que $-k$ é o elemento inverso de k em G. Verifique se tal sistema de codificação tem sigilo perfeito.

Exercício 1.47. Suponha que você tem três moedas e uma delas é viciada de modo que $\mathbb{P}(\text{cara}) = 2/3$. Escolhendo uma delas ao acaso a probabilidade de acertar qual é a viciada é um terço. Agora, suponha que o resultado do lançamento de cada uma delas, sem conhecer qual é a viciada, resulta em (cara, cara, coroa). Mostre, usando o Teorema de Bayes, que a probabilidade da primeira moeda ser a viciada é $2/5$.

Exercício 1.48. Três convidados chegaram numa festa vestindo chapéu e os entregaram na recepção. O funcionário, pouco cuidadoso, não identificou os chapéus e no final da festa os entregou aleatoriamente para as mesmas três pessoas. Use o princípio da inclusão–exclusão para mostrar que ninguém recebe o próprio chapéu com probabilidade 1/3. Generalize o resultado para n convidados e use a série de potências para a função exponencial (veja (s.8) do apêndice) para mostrar que a probabilidade de ninguém pegar o próprio chapéu converge, quando $n \to \infty$, para 1/e.

Exercício 1.49. Em um treino de paraquedistas um grupo de n paraquedistas estão enfileirados e um paraquedista é escolhido ao acaso no seu grupo. O paraquedista escolhido cumprimenta todos os paraquedistas do seu grupo e salta da avião; o grupo fica então dividido em dois: um grupo formado pelos paraquedistas que se encontravam a esquerda daquele que pulou e o outro grupo formado pelos paraquedistas a direita. O procedimento é repetido nos grupos restantes até sobrarem grupos de um único paraquedista, que pulam um a um. Note que paraquedistas que em algum momento ficam em grupos diferentes não se cumprimentaram e não se

cumprimentarão desse momento em diante. A ordem da fila dentro de cada grupo é sempre mantida. Prove que os paraquedistas das posições i e j, sem perda de generalidade $j > i$, se cumprimentam com probabilidade $2/(j-i+1)$.

Exercício 1.50. Considere uma moeda que resulta em cara com probabilidade $p \in (0,1)$. Prove que a probabilidade de que em n lançamentos (independentes) temos mais que k caras é no máximo

$$\binom{n}{k} p^k \leqslant \left(\frac{enp}{k}\right)^k.$$

Exercício 1.51. Considere uma moeda que resulta em cara com probabilidade $1/5$ e coroa com probabilidade $4/5$. Justifique que a probabilidade de sair menos que k coroas em $2k$ lançamentos (independentes) é

$$\sum_{i=0}^{k-1} \binom{2k}{i} (4/5)^i (1/5)^{2k-i} < (1/5)^{2k} 4^k \sum_{i=0}^{k-1} \binom{2k}{i}.$$

Prove que a probabilidade de sair menos que k coroas em $2k$ lançamentos dessa moeda é menor que $(4/5)^{2k}$ (dica: teorema do binômio de Newton para o somatório na equação acima).

Exercício 1.52. Considere uma moeda que resulta em cara com probabilidade $1/2-\varepsilon$ para algum $\varepsilon \in (0, 1/2)$ fixo. Prove que em k lançamentos a probabilidade de sair no máximo $k/2$ coroas é $\leqslant (1-4\varepsilon^2)^{t/2}$ (dica: verifique que $f(x) = (1/2+\varepsilon)^x (1/2-\varepsilon)^{t-x}$ é estritamente crescente no intervalo $(0, t/2)$).

Exercício 1.53. Considere o seguinte procedimento para gerar uma permutação da sequência $a = (1, 2, \ldots, n)$, para qualquer inteiro $n \geqslant 1$:

- para cada coordenada $i = 1, 2, \ldots, n-1$ do vetor a
  - sorteie uniformemente uma coordenada $j \in \{i, \ldots, n\}$
  - troque os componentes das coordenadas: coloque o número da coordenada j na coordenada i e o da coordenada i na coordenada j.

O vetor resultante é uma permutação do vetor inicial com probabilidade uniforme?

Exercício 1.54. Considere a seguinte proposta de algoritmo que recebe um inteiro positivo $M > 0$ e devolve um inteiro escolhido aleatoriamente em $\{0, 1, \ldots, M-1\}$:

Instância: inteiro positivo M.

Resposta: uma escolha aleatória em $\{0, 1, \ldots, M-1\}$.

1 seja k o número de bits de M;
2 $(d_0, d_1, \ldots, d_{k-1}) \xleftarrow{R} \{0,1\}^k$;
3 $N \leftarrow \sum_i d_i 2^i$;
4 **responda** N mod M.

Prove que as possíveis respostas não são equiprováveis.

Exercício 1.55. O seguinte gerador de números aleatórios é adaptado do exercício anterior.

Instância: inteiros positivos M e t.

Resposta: uma escolha aleatória em $\{0, 1, \ldots, M-1\}$.

1 seja k o número de bits de M;
2 $(d_0, d_1, \ldots, d_{k+t-1}) \xleftarrow{R} \{0,1\}^{k+t}$;
3 $N \leftarrow \sum_i d_i 2^i$;
4 **responda** N mod M.

No caso $t = 0$ a probabilidade da resposta não é uniforme (Exercício 1.54 acima). Prove que quanto maior é t mais próximo a probabilidade de N está da uniforme, no seguinte sentido

$$\sum_{n=0}^{M-1} \left| \mathbb{P}[N = n] - \frac{1}{M} \right| \leqslant \frac{1}{2^{t-1}}.$$

Exercício 1.56. Distribuímos uniformemente e independentemente n bolas em m caixas. Qual é a probabilidade com que a i-ésima caixa fica vazia? Qual é a probabilidade com que a j-ésima e a i-ésima caixas ficam vazias? Qual é a probabilidade com que nenhuma fica vazia? E de exatamente uma ficar vazia?

Prove que no caso $n = m$ o maior número de bolas em qualquer caixa é no máximo $2e \log_2 n$ com probabilidade $1 - n^{-4}$ (dica: estime a probabilidade de uma caixa ter muitas bolas, a fórmula de Stirling (d.3) pode ser útil nos cálculos, e use a subaditividade, Corolário 1.6 na página 13).

Exercício 1.57. Prove que no seguinte algoritmo a probabilidade p_n do laço executar pelo menos n vezes, para todo $n \geqslant 2$, é maior que 0 e, mais que isso, essa probabilidade é maior que 0 no limite, ou seja, $\lim p_n > 0$ quando $n \to \infty$ (pode ser útil a desigualdade $1 - x \geqslant \exp(-2x)$ para $x \in [0, 1/2]$).

```
1 j ← 0;
2 repita
3 |   j ← j + 1;
4 |   para cada i ∈ {1, 2, ..., j} faça d_i ←R {0, 1};
5 até que d_i = 1 para todo i.
```

*Exercício 1.58 (**Lema do isolamento**, (Mulmuley, Vazirani e Vazirani, 1987)).* O seguinte resultado diz que, independentemente da natureza de uma família $\mathcal{F}$ de conjuntos, uma atribuição aleatória de pesos aos elementos de $\bigcup_{F\in\mathcal{F}} F$ isola o elemento da família menos pesado com grande probabilidade. Este lema tem muitas aplicações na teoria da computação, em particular, Mulmuley e seus coautores o usaram para projetar um algoritmo aleatorizado paralelizável para encontrar emparelhamento de peso máximo em um grafo (exercícios 1.102 e 1.104).

Sejam E um conjunto finito e $\mathcal{F}$ uma família de subconjuntos de E. Uma *m-ponderação* de E é uma função $p\colon E \to \{1,\dots,m\}$ que atribui pesos inteiros para os elementos de E. O peso de um subconjunto não vazio $S \subseteq E$ é $p(S) = \sum_{e\in S} p(e)$. A ponderação p é *isolante para* $\mathcal{F}$ se o peso mínimo, $\min_{S\in\mathcal{F}} p(S)$, for alcançado em um único elemento de $\mathcal{F}$. O lema do isolamento é o seguinte resultado

Teorema *Dado $m \in \mathbb{N}$, para todo conjunto finito E e toda família $\mathcal{F} \subseteq 2^E$ temos*

$$\mathbb{P}\Big[\, p\in_R \{1,\dots,m\}^E \text{ é isolante para } \mathcal{F} \,\Big] \geqslant \left(1 - \frac{1}{m}\right)^{|E|}.$$

Para provar esse resultado, suponha que nenhum elemento de $\mathcal{F}$ é um superconjunto de outro elemento de $\mathcal{F}$ (os superconjuntos podem ser removidos sem afetar o min).

Considere P o conjunto de todas as m-ponderações de E e $P^{>1}$ o conjunto de todas as m-ponderações $E \to \{2,3,\dots,m\}$ de E que não atribuem o peso 1 a qualquer elemento de E.

Para cada ponderação $p \in P$ fixe $S_p \in \mathcal{F}$ de peso mínimo de acordo com p e defina a função $\phi\colon P^{>1} \to P$ da seguinte forma: $p' = \phi(p)$ é dada por

$$p'(i) = \begin{cases} p(i) - 1 & \text{se } i \in S_p \\ p(i) & \text{se } i \notin S_p. \end{cases}.$$

1. Prove que se $p \in P^{>1}$ então p' é isolante em $\mathcal{F}$.

2. Prove que ϕ é injetiva.

3. Prove que $\mathbb{P}_p[p \text{ é isolante em } \mathcal{F}] \geqslant |\phi(P^{>1})|/|P|$.

4. Conclua a demonstração do lema do isolamento.

Exercício 1.59 (Borel–Cantelli). Seja $(A_n : n \geqslant 1)$ uma sequência de eventos. Fixado n, o evento "*algum* A_k *ocorre para* $k \geqslant n$" é $\bigcup_{k \geqslant n} A_k$ e o evento *infinitos* A_n *ocorrem* ou, A_n ocorre infinitas vezes é o evento

$$\limsup_{n\to\infty} A_n := \bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} A_k$$

de todos ω tais que, para todo $n \in \mathbb{N}$ existe $k > n$ para o qual $\omega \in A_k$, ou seja, ω pertence a infinitos eventos da sequência. Como estimar a probabilidade de uma sequência de eventos ocorrer infinitas vezes?

Teorema (Borel–Cantelli) *Seja* $(A_n : n \geqslant 1)$, *uma sequência de eventos aleatórios de um espaço de probabilidade.*

1. *Se* $\sum_n \mathbb{P}(A_n)$ *converge, então* $\mathbb{P}[\, A_n$ *infinitas vezes* $\,] = 0$.
2. *Se* $\sum_n \mathbb{P}(A_n) = \infty$ *converge e os eventos da sequência são independentes, então* $\mathbb{P}[\, A_n$ *infinitas vezes* $\,] = 1$.

Exercício 1.60. Considere o conjunto dos números naturais e defina para todo subconjunto E e cada $n \geqslant 1$ a densidade relativa de E

$$P_n(E) := \frac{|\{1, 2, \ldots, n\} \cap E|}{n}.$$

Defina $p(E) := \lim_{n\to\infty} \mathbb{P}_n(E)$ quando o limite existe e seja $\mathcal{A}$ a família de subconjunto E para os quais o limite existe. Prove que se A e B são elementos disjuntos de $\mathcal{A}$ então $A \cup B \in \mathcal{A}$ e $p(A \cup B) = p(A) + p(B)$ e que esse não é o caso se os eventos não são disjuntos. Prove que p não é enumeravelmente aditiva. Finalmente, prove que p é invariante por translação, ou seja, se $p(A)$ existe então $p(\{a + 1 : a \in A\}) = p(A)$.

1.5 ANÁLISE DE ALGORITMOS

Um algoritmo define sem ambiguidade uma sequência de passos para resolver um problema computacional. Um *problema computacional* é caracterizado por um conjunto de *instâncias* (ou entradas), um conjunto de *respostas* e uma *relação* que associa instâncias a respostas. Por exemplo, o problema "multiplicar dois inteiros positivos" tem como instâncias pares (n, m) de inteiros positivos e como respostas inteiros positivos. A relação que se quer computar é definida pelos pares $((n, m), z)$ de instâncias e respostas tais que $n \cdot m = z$.

Entre instâncias e respostas temos uma relação e não uma função pois é possível que uma instância esteja associada a mais de uma resposta. Por exemplo, se as

instância são fórmulas da lógica proposicional e as respostas são valorações das variáveis com verdadeiro ou falso e que tornam a fórmula verdadeira, então a instância $(x_1 \ ou \ x_2) \ e \ não(x_3)$ tem três respostas possíveis.

Os algoritmos são, geralmente, descritos no que costumamos chamar de *pseudo-código*, uma mistura de algumas palavras-chave em português com sentenças construídas como em uma linguagem de programação estruturada como as linguagens C, Python, Java e muitas outras.

Um algoritmo executado sobre qualquer instância do problema produz uma resposta que deve estar correta, isto é, os pares formados por instância e respostas dessa instância devem fazer parte da relação do problema. Além da correção do algoritmo, queremos conhecer o comportamento do algoritmo com respeito ao consumo de recursos para resolver o problema. Alguns recursos para os quais podemos querer estimar o consumo por um algoritmo são o *espaço*, o *tempo*, a *comunicação* e a *aleatoriedade*. Os dois primeiros são os mais comumente estudados, o primeiro está associado a quantidade de armazenamento ("memória") extra usada para resolver uma instância do problema e o segundo ao número de *instruções elementares* realizadas pelo algoritmo. Além desses, pode ser de interesse a quantidade de unidades de informação transmitidas e recebidas e, quando analisamos algoritmos probabilísticos, a *quantidade de sorteios* usados pelo algoritmo (veja um caso na página 79). A *Análise de Algoritmos* é a disciplina da Teoria da Computação que trata das técnicas de prova da *correção* de algoritmos e de avaliação de *eficiência* quanto ao consumo de recursos.

Tamanho da instância Expressamos o consumo de recursos pelos algoritmos em função do *tamanho da instância*. Uma instância do problema é dada por alguma *codificação* dela usando um conjunto de símbolos (em último nível uma cadeia de bits) donde definimos o tamanho de uma instância como o número de símbolos (bits) usados na representação da instância. Na prática adotamos algumas simplificações que dependem muito do problema que esta sendo estudado e da representação usada e, em geral, o tamanho da instância é um inteiro positivo que descreve a quantidade de componentes da instância. Por exemplo, se o problema é multiplicar dois números inteiros, o tamanho é a quantidade de algarismos desses números (em alguma base com pelo menos dois algarismos). Se o problema é multiplicar duas matrizes de números inteiros, o tamanho pode ser a dimensão da matriz; essa simplificação supõe que o tamanho da matriz é muito grande quando comparada ao tamanho dos números que a compõe, porém se esse não é o caso então o tamanho dos números deve ser levado em conta. Esse também é o caso no problema de ordenação de uma sequência numérica, o tamanho pode ser dado pelo número de elementos da sequência. Em

algoritmos sobre grafos, o tamanho de uma instância é dado em função do número de vértices, do número de arestas, ou de ambos. Há justificativas razoáveis para tais simplificações e, mesmo que façamos escolhas concretas de codificação de instâncias, tentamos manter a discussão abstrata o suficiente para que as estimativas sejam independentes da escolha da codificação.

Tempo de execução Como estimamos o *tempo de execução* de um algoritmo para resolver uma determinada instância? São duas as ideias preliminares.

Primeiro, contamos as instruções executadas, não usamos a noção comum de tempo pois isso dependeria do que (ou quem) executa o algoritmo. Como já dissemos, determinamos quanto tempo o algoritmo demanda em função do tamanho da instância escrevendo uma função que caracteriza como o tempo de execução varia com o tamanho da entrada[4], essa função expressa a ordem de grandeza do crescimento do tempo de execução quando as instâncias crescem. O consumo de tempo dos algoritmos, como medida de sua eficiência, expressa o número de *instruções básicas* executadas pelo algoritmo em função do tamanho da entrada descrita em notação assintótica. Isso nos permite algumas simplificações: por exemplo, há casos em que podemos assumir que cada "linha" consome tempo constante. As operações e relações aritméticas podem ser assumidas de tempo constante. Porém, isso não é regra e tem de ser feito com cuidado, como no problema de multiplicação de inteiros, por exemplo, onde as operações aritméticas tem custo proporcional ao tamanho dos operandos.

Segundo, uma estimativa para o número de instruções executadas é dada para a classe das instâncias que têm o mesmo tamanho e expressamos o desempenho de algoritmos em função do tamanho da representação das instâncias. É possível que entre instâncias de mesmo tamanho a quantidade de recursos usados por uma algoritmo varie e precisamos adotar alguma medida resumo. Por exemplo, ao ordenar uma lista de dez números a quantidade de instruções executadas pode variar de acordo com a disposição dos números nessa lista, eventualmente, pode ser mais barato se a lista já está quase ordenada. Por isso, adotamos alguma estratégia para resumir o tempo de execução do algoritmo na classe das instâncias de mesmo tamanho: tomamos o *pior caso* — aquele em que o algoritmo consome mais recursos — ou o *caso médio* — a média ponderada de consumo de acordo com alguma distribuição de pesos na classe das instâncias de mesmo tamanho.

Em Complexidade Computacional convenciona-se chamar um algoritmo de **eficiente** com respeito ao tempo de execução se o número de instruções executadas é,

[4] É natural esperarmos que instâncias maiores demandem mais recurso dos algoritmos. Não vamos lidar com casos em que isso não vale.

no pior caso, limitado superiormente por uma função polinomial no tamanho da instância. Na teoria isso é muito conveniente pois

- a classe das funções polinomiais é fechada para soma, multiplicação e composição de funções, assim a noção de eficiência é preservada por práticas comuns de programação;
- os modelos formais tradicionais de computação são polinomialmente equivalentes, o que torna a escolha do modelo irrelevante para essa definição de eficiência;
- com algum cuidado, as várias representações computacionais de objetos abstratos, como um grafo por exemplo, têm tamanhos polinomialmente relacionados, o que faz a codificação ser irrelevante para essa definição de eficiência.

Na prática isso pode não ser representativo de eficiência pois o polinômio pode ter um grau muito alto o que torna uma implementação de um caso assim inviável para o uso na prática.

Terminamos essa seção com dois exemplos. Para o problema cuja instância é uma lista $a_1, a_2, \cdots, a_n$ de inteiros mais um inteiro x e a resposta é "sim", x ocorre na lista, ou "não", x não ocorre na lista. Uma solução é a busca linear: percorra a lista e verifique se cada elemento dela é o item procurado. Essa estratégia é expressa como abaixo.

Instância: uma lista $a_1, \ldots, a_n$ de inteiros e um inteiro x.
Resposta: *sim* se x ocorre na lista e *não* caso contrário.

1 $i \leftarrow 1$;
2 **enquanto** $a_i \neq x$ *e* $i < n$ **faça** $i \leftarrow i + 1$;
3 **se** $a_i = x$ **então responda** sim.
4 **senão responda** não.

Algoritmo 3: busca sequencial.

No melhor caso o elemento x ocorre na primeira posição da sequência, o algoritmo executa a atribuição na linha 1, o teste na linha 2, a comparação na linha 3 e responde. Essencialmente, um número constante (não depende de nenhum parâmetro da instância) de instruções. Nesse caso escrevemos que o tempo de execução é $O(1)$.

O pior caso ocorre quando o valor que está sendo procurado não está na lista. O número de instruções executadas é: 1 atribuição na linha 1, mais $4(n-1)$ instruções na linhas 2 (são feitas duas comparações, uma adição e uma atribuição, repetidas $n-1$ vezes), mais 1 comparação na linha 3, mais 1 instrução na linha 4. No total são

$4(n-1)+3$ instruções. A função $4n-1$ é linear em n, o tamanho da entrada, de modo que dizemos que o seu crescimento é da ordem de n e, usando notação assintótica, dizemos que o tempo de execução de pior caso do algoritmo é $O(n)$.

Nesse problema, podemos estimar a ordem de grandeza do número de instruções executadas considerando apenas o número de comparações que são feitas, as outras instruções contribuem com uma constante multiplicativa desse termo. Assim, se tivéssemos contado apenas o número de comparações também chegaríamos a conclusão de que o tempo de execução de pior caso do algoritmo é $O(n)$.

Resumindo, no melhor caso a quantidade de comparações é constante, não depende de n e no pior caso cresce linearmente com n. Para estimar o caso médio, vamos assumir que o item procurado está na lista. Também, vamos assumir que cada elemento da lista tem a mesma probabilidade de ser o valor buscado. Com tais hipóteses o número médio de comparações é i se a busca termina na posição i, portanto, o número médio de comparações é

$$\frac{1}{n}(1+2+\cdots+n) = \frac{n+1}{2}.$$

Nesse caso, dizemos que o tempo de execução de caso médio do algoritmo é $O(n)$ pois, novamente, temos uma função de crescimento linear em n.

Agora, consideremos o problema de ordenar uma sequência de inteiros. Uma solução é o seguinte algoritmo conhecido como ordenação por inserção:

Instância: uma sequência $a_1,\ldots,a_n$ de inteiros.
Resposta: uma permutação da sequência com os elementos em ordem não decrescente.

```
1 para i de 2 até n faça
2     x ← a_i;
3     j ← i − 1;
4     enquanto (a_j > x e j ⩾ 1) faça
5         a_{j+1} ← a_j;
6         j ← j − 1;
7     a_{j+1} ← x.
```

Algoritmo 4: ordenação por inserção.

Notemos que o tempo do algoritmo é determinado pela condição no laço da linha 4, as linhas 2, 3 e 7 são executadas $n-1$ vezes pelo laço da linha 1. Para i fixo, uma rodada completa do laço executa 2 comparações, 2 atribuições e 2 operações; no pior caso[5] o laço é executado para todo j de i até 1, depois o laço é falso para a segunda

[5] O pior caso para ordenação por inserção ocorrerá quando a lista de entrada estiver em ordem decrescente.

condição ($j = 0$). Logo são $6i$ instruções mais as 2 comparações finais. No pior caso, o custo de ordenação por inserção de uma sequência com n elementos é

$$T(n) = 2 + \sum_{i=2}^{n} 6i = 2 + 6\frac{(n+2)(n-1)}{2} = 3n^2 + 3n + 5$$

e dizemos que $T(n)$ tem ordem de crescimento n^2. Notemos o seguinte, a ordem de grandeza de $T(n)$ é dada pelo fato de termos dois laços aninhados e dentro desses laços o número de instruções executadas em cada rodada é constante de tal forma que, para a ordem de grandeza, não importa se contamos $j \leftarrow j-1$ como 1 ou 2 instruções, é suficiente estabelecer que corresponde a uma constante.

Para estimar o caso médio observamos que o fato determinante para o número de instruções executadas é o "tipo de ordem" dos elementos da sequência e não quais são os elementos em si. Por exemplo, ordenar $(1,2,3,4)$ usa o mesmo número de instruções que $(4,7,8,9)$, assim como ordenar $(1,4,3,5,2)$ e $(11,15,14,20,13)$. Em outras palavras, o mesmo tipo de ordem significa que a mesma permutação ordena as duas instâncias. Dito isso, assumimos que as instâncias são formadas por sequências de n inteiros distintos fixos e que qualquer uma das $n!$ permutações são igualmente prováveis.

Fixado i, com $2 \leqslant i \leqslant n$, consideremos a subsequência $(a_1,\ldots,a_i)$ da entrada. Para cada i vale que no início do laço da linha 1 temos nas $i-1$ primeiras posições a sequência $(a_1,\ldots,a_{i-1})$ ordenada e o laço da linha 4 procura a posição correta de a_i em $(a_1,\ldots,a_{i-1})$ ordenado. Definimos o posto(a_i) como a posição do elemento a_i no subvetor $(a_1,\ldots,a_i)$ ordenado. Por exemplo, com entrada $(3,6,2,5,1,7,4)$ o posto de 5 (que é o a_4) é 3 pois em $(3,6,2,5)$, quando ordenado, o número 5 ocupa a terceira posição. Dado i e que o subvetor $(a_1,\ldots,a_{i-1})$ está ordenado, o teste no laço é executado $i - \text{posto}(a_i) + 1$ vezes.

Exercício 1.61. Verifique que, de acordo com as definições e hipóteses acima, o posto de a_i é igualmente provável ser qualquer $j \in \{1,2,\ldots,i\}$.

Assim, o número médio de comparações é

$$\sum_{i=2}^{n} \sum_{\text{posto}=1}^{i} \frac{i - \text{posto} + 1}{i} = \sum_{i=2}^{n} \frac{i+1}{2} = \frac{(n+4)(n-1)}{2}$$

que é da ordem de n^2.

1.5.1 NOTAÇÃO ASSINTÓTICA

As estimativas para o custo de um algoritmo são expressas usando notação assintótica. A análise assintótica foca no comportamento do algoritmo para entradas

"suficientemente grandes" e simplifica a expressão da complexidade de tempo ao ignorar constantes multiplicativas e termos de ordem inferior, uma vez que essas diferenças não alteram a ordem de grandeza do crescimento da função. Além disso, essa abordagem permite uma simplificação substancial ao contar o número de instruções executadas e também facilita a codificação de instâncias (como números, por exemplo, que podem ser expressos em qualquer base com pelo menos dois símbolos, já que, nesse caso, eles têm representação logarítmica na quantidade de dígitos). A análise assintótica permite uma avaliação de desempenho representativa, independente de tecnologia ou detalhes de implementação, tornando mais simples a comparação direta entre algoritmos, independentemente das implementações específicas ou do ambiente de execução.

Abaixo f e g são funções reais[6] com g *assintoticamente positiva*, ou seja $g(n) > 0$ para todo n suficientemente grande. Dizemos que f é **assintoticamente muito menor** que g e escrevemos

$$f(n) = o(g(n)) \text{ quando } n \to \infty \tag{1.22}$$

se, e só se,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Por exemplo,

1. $1 = o(\log(\log(n)))$.
2. $\log(\log(n)) = o(\log(n))$.
3. $\log(n) = o(n^\varepsilon)$ para todo $\varepsilon > 0$.
4. $n^\varepsilon = o(n^c)$ para quaisquer $0 < \varepsilon < 1 \leqslant c$.
5. $n^c = o(n^{\log n})$ para todo $1 \leqslant c$.
6. $n^{\log n} = o(\exp(n))$.
7. $\exp(n) = o(n^n)$.
8. $n^n = o(\exp(\exp(n)))$.

Também usamos a notação $f \ll g$ o que nos permite escrever, a partir do exemplo acima, a sequência monótona

$$1 \ll \log(\log(n)) \ll \log(n) \ll n^\varepsilon \ll n^c \ll n^{\log n} \ll e^n \ll n^n \ll e^{e^n}.$$

[6] Uma função $f(n)$ que expressa o consumo de um recurso por algum algoritmo é uma função de $\mathbb{N}$ em $\mathbb{N}$, mas aqui vamos tratar a notação assintótica de modo um pouco mais geral que nos permitirá usá-la em outras situações.

Dizemos que f **assintoticamente menor** que g e escrevemos

$$f(n) = O(g(n)) \text{ quando } n \to \infty \tag{1.23}$$

se existe $n_0 > 0$ e existe $c > 0$ tais que para todo $n \geqslant n_0$

$$|f(n)| \leqslant cg(n).$$

Na definições dadas nas equações (1.22) e (1.23) o símbolo "=" não é a igualdade no sentido usual, é um abuso da notação que permitimos em troca de algumas conveniências. Temos que $n = O(n^2)$ e $n^2 + 2n + 1 = O(n^2)$ mas $n \neq n^2 + 2n + 1$. Quando usamos essas definições, em geral, omitimos o "quando $n \to \infty$" e fica implícito que as instâncias são suficientemente grandes.

Proposição 1.62 *Se $f(n) = o(g(n))$ então $f(n) = O(g(n))$.*

A prova é imediata da definição de limite. A recíproca da Proposição 1.62 não vale, como pode ser visto tomando-se $f(n) = g(n) = n^2$.

Proposição 1.63 *Se $f_1(n) = O(g_1(n))$ e $f_2(n) = O(g_2(n))$ então*

1. $f_1(n) + f_2(n) = O(\max\{g_1(n), g_2(n)\})$.
2. $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$.
3. $a \cdot f_1(n) = O(g_1(n))$ *para toda constante* $a \in \mathbb{R}$.

Demonstração. Vamos provar o item 1. Digamos que $|f_1(n)| \leqslant c_1 g_1(n)$ para todo $n \geqslant n_1$ e que $|f_2(n)| \leqslant c_2 g_2(n)$ para todo $n \geqslant n_2$, onde n_1, n_2, c_1, c_2 são as constantes dadas pela definição de notação O. Então

$$\begin{aligned} |f_1(n) + f_2(n)| \leqslant |f_1(n)| + |f_2(n)| &\leqslant c(g_1(n) + g_2(n)) \text{ com } c = \max\{c_1, c_2\} \\ &\leqslant 2c(\max\{g_1(n), g_2(n)\}) \end{aligned}$$

para todo $n \geqslant \max\{n_1, n_2\}$, o que prova a afirmação.

As outras propriedades são deduzidas de modo análogo e são deixadas como exercício. □

É preciso observamos alguns cuidados com o teorema acima pois, por exemplo, para inteiros positivos n e k não vale que $1^k + 2^k + \cdots + (n-1)^k + n^k = O(\max\{1^k, 2^k, \ldots, (n-1)^k, n^k\}) = O(n^k)$. O problema aqui é que o máximo só pode ser tomado sobre um número de termos que não dependa de n. De fato, temos que $1^k + \cdots + n^k = O(n^{k+1})$ e que $1^k + \cdots + n^k \neq O(n^k)$.

Fica como exercício a verificação do seguinte resultado.

Proposição 1.64 *Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$ então $f(n) = O(h(n))$.*

Alguns exemplos são dados a seguir.

1. $an^2 + bn + c = O(n^2)$ para toda constante $a > 0$.

   Primeiro, observamos que, para n suficientemente grande, $|an^2 + bn + c| \leqslant |a|n^2 + |b|n + |c|$ e agora usamos a Proposição 1.63 em cada operando das somas, de $an^2 = O(n^2)$, $|b|n = O(n)$ e $|c| = O(1)$ temos $an^2 + |b|n + |c| = O(\max\{n^2, n, 1\}) = O(n^2)$. Analogamente, para todo $k \in \mathbb{N}$

   $$\sum_{i=0}^{k} a_i n^i = O(n^k).$$

2. $n \log(n!) = O(n^2 \log n)$.

   Primeiro, temos $n = O(n)$. Depois, $n! = \prod_{i=1}^{n} i < \prod_{i=1}^{n} n = n^n$. Como log é crescente $\log(n!) < \log(n^n) = n \log(n)$, portanto $\log(n!) = O(n \log(n))$. Pela Proposição 1.63 $n \log(n!) = O(n^2 \log n)$.

3. Para toda constante $a > 1$, $\log_a(n) = O(\log(n))$. De fato,

   $$\log_a(n) = \frac{1}{\log a} \log n$$

   porém $\frac{1}{\log a} = O(1)$ e $\log n = O(\log n)$ e pela Proposição 1.63 $\log_a(n) = O(\log(n))$.

Um algoritmo eficiente com respeito ao tempo de execução é um algoritmo que nas instâncias de tamanho n tem tempo de execução $O(n^k)$ para algum inteiro positivo k fixo.

Convenções de uso da notação assintótica Ao usar notação assintótica nós desconsideramos os coeficientes, por exemplo, usamos $O(n^2)$ ao invés de $O(3n^2)$ e $O(1)$ ao invés de $O(1024)$ ainda que, como classes de funções, $O(n^2) = O(3n^2)$ e $O(1) = O(1024)$.

Escrevemos no argumento de $O(\cdot)$ somente o termo mais significativo, por exemplo, usamos $O(n^2)$ ao invés de $O(2n^2 + 5n \log n + 4)$. Nesse caso, da Proposição 1.63 vale que $2n^2 + 5n \log n + 4 = O(\max\{n^2, n \log n, 1\}) = O(n^2)$.

Quando notação assintótica aparece em equações, na forma "expressão 1 = expressão 2" onde "expressão" são expressões algébricas que envolvem notação assintótica, os termos assintóticos em "expressão 1" são quantificados universalmente, enquanto que os termos assintóticos em "expressão 2" são quantificados existencialmente. Por exemplo, em $n^3 + O(n^2) = O(n^3) + n^2 + n$ entendemos como: existe um $n_0 > 0$ tal que

para todo $f(n)$ em $O(n^2)$, existe $g(n)$ em $O(n^3)$ tal que $n^3 + f(n) = g(n) + n^2 + n$

para todo $n \geqslant n_0$.

Notação Ω e Θ A notação Ω adaptada por Donald Knuth da notação introduzida por outros matemáticos é definida por

$$f(n) = \Omega(g(n)) \text{ se, e somente se, } g(n) = O(f(n)).$$

Escrevemos $f(n) = \Theta(g(n))$ se, e somente se, $f(n) = O(g(n))$ e $g(n) = O(f(n))$.

Exercício 1.65. Verifique se valem as firmações

(a) $n^{1,5} = O(n^2)$.

(b) $\frac{n^2}{10} = O(n)$.

(c) $n^2 - 100n = O(n^2)$.

(d) $n \log(n) = O(n^2)$.

(e) $n = O(n \log n)$.

(f) $2^n = O(n)$.

(g) $2^n = O(2^{n-1})$.

(h) Se $a_k > 0$, então $\sum_{i=0}^{k} a_i n^i = \Theta(n^k)$.

Exercício 1.66. Suponha que $\lim_{n\to\infty} \frac{f(n)}{g(n)} = L$. Verifique se valem as firmações

(a) Se $L > 0$, $f(n) = \Theta(g(n))$.

(b) Se $L = 0$, $f(n) = O(g(n))$ mas $f(n) \neq \Theta(g(n))$.

(c) Se $L = \infty$, $f(n) = \Omega(g(n))$ mas

Análise do gerador de números aleatórios Uma execução do Algoritmo 2 com entrada M e com uma única rodada do laço tem tempo de execução $O(\log M)$ para sortear os bits e realizar a soma, outra execução com a mesma entrada M pode mais azarada e precisar de 2 rodadas, mas o tempo de execução continua $O(\log M)$ e uma outra execução com a mesma entrada M pode ser ainda muito azarada e precisar de M rodadas e o tempo de execução será $O(M \log M)$, que é exponencial no tamanho da entrada! Sabemos da seção 1.3.6 que isso é muito pouco provável.

Tipicamente, o algoritmo executa em média 2 rodadas do laço, onde a média do número de rodadas é a média ponderada pela probabilidade. De fato, se $p = M/2^k$ é a probabilidade com que o algoritmo executa exatamente uma rodada, $(1 - p)p$ é

a probabilidade com que o algoritmo executa exatamente duas rodadas, $(1-p)^2 p$ para três rodadas e assim por diante, o algoritmo executa exatamente k rodadas com probabilidade $(1-p)^{k-1}p$, portanto, o número médio de rodadas é (veja (s.6) do apêndice)

$$\sum_{k \geqslant 1} k(1-p)^{k-1}p = \frac{1}{p} = \frac{2^k}{M} \leqslant 2 \tag{1.24}$$

rodadas, onde $k = \lfloor \log_2 M \rfloor + 1$. Portanto, o *tempo médio de execução* é $2 \cdot O(\log M)$, ou seja, $O(\log M)$.

Um fato importante a ser ressaltado neste momento é que esse tempo médio que calculamos é sobre os sorteios do algoritmo, diferente do que fizemos com os algoritmos de busca sequencial e de ordenação por inserção, no início deste capítulo, onde a média foi feita sobre o tempo de execução nas diferentes entradas para o algoritmo.

Outros usos da notação O A notação assintótica também pode ser usada para descrever o termo de erro em uma aproximação de uma função matemática. Como antes, os termos mais significativos são escritos explicitamente e os termos menos significativos são resumidos em um único termo O.

Considere, por exemplo, a série de Taylor em torno do $x = 0$ para a exponencial e para o logaritmo:

$$\exp(x) = \sum_{i=0}^{\infty} \frac{x^n}{n!} \quad \text{e} \quad \ln(1+x) = \sum_{i=1}^{\infty} (-1)^{n+1} \frac{x^n}{n}$$

de modo que, quando $x \to 0$,

$$\exp(x) = 1 + x + \frac{x^2}{2} + O(x^3) \quad \text{e} \quad \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + O(x^4).$$

O Teorema dos Números Primos O Teorema dos Números Primos descreve a distribuição assintótica dos números primos entre os inteiros positivos. Ele foi provado independentemente por Jacques Hadamard e Charles Jean de la Vallée Poussin em 1896.

Seja $\pi(x)$ a quantidade de números primos menores ou iguais a $x \in \mathbb{R}^{>0}$. O Teorema dos Números Primos afirma que $\pi(x)$ é $x/\ln(x)$ assintoticamente, ou seja,

$$\pi(x) = |\{p \text{ primo} : p \leqslant x\}| = (1 + o(1))\frac{x}{\ln x}$$

quando $x \to \infty$ ou, equivalentemente,

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln(x)} = 1.$$

Antes desse teorema, foi provado por Chebyshev que

$$\pi(x) = \Theta\left(\frac{x}{\ln x}\right).$$

Uma estimativa para as constantes envolvidas na notação Θ é útil em algoritmos probabilísticos que sorteiam primos. Por exemplo, temos para todo $x \geqslant 17$ (Bach e Shallit, 1996, teo. 8.8.1)

$$\frac{x}{\ln x} < \pi(x) < 1{,}25506\frac{x}{\ln x}, \tag{1.25}$$

de fato, a primeira desigualdade vale para todo $x \geqslant 17$ e a segunda para todo $x > 1$; tratando todos os valores para $x < 17$ podemos afirmar que

$$0{,}3\frac{x}{\ln x} < \pi(x) < 1{,}3\frac{x}{\ln x}$$

para todo $x \geqslant 2$.

Exercício 1.67. Prove que se $n > 0$ é inteiro e m é sorteado em $\{1, 2, \ldots, (\log_2 n)^2\}$ uniformemente, então

$$\mathbb{P}[\, n \not\equiv 0 \pmod{m}\,] = \Omega\left(\frac{1}{\log\log n}\right).$$

1.5.2 ARITMÉTICA COM INTEIROS

O Algoritmo de Euclides O seguinte algoritmo é conhecido como Algoritmo de Euclides (de 300 aC). Ele computa o maior divisor comum de dois inteiros quaisquer baseado no fato de que $\mathrm{mdc}(a, b) = \mathrm{mdc}(b, a \bmod b)$ onde $a \bmod b$ é o resto na divisão de a por b.

Instância: um par de inteiros (a, b).
Resposta: $\mathrm{mdc}(a, b)$.

1 $a \leftarrow |a|$
2 $b \leftarrow |b|$;
3 **se** $b = 0$ **então responda** a.
4 **senão responda** $\mathrm{mdc}(b, a \bmod b)$.

Algoritmo 5: $\mathrm{mdc}(a, b)$.

O Algoritmo de Euclides está correto: podemos assumir que $a > b > 0$ pois $\mathrm{mdc}(a, b) = \mathrm{mdc}(|a|, |b|)$ e $\mathrm{mdc}(a, 0) = |a|$. Também, notemos que o algoritmo termina pois $0 \leqslant a \bmod b < b$, pelo Teorema da Divisão Euclidiana, logo o valor da variável b decresce estritamente a cada iteração. Para concluir, observamos que se a e $b > 0$ são inteiros então

$$d|a \text{ e } d|b \Leftrightarrow d|b \text{ e } d|a \bmod b$$

donde deduzimos que se $a, b > 0$ são inteiros, então $\mathrm{mdc}(a, b) = \mathrm{mdc}(b, a \bmod b)$.

O Algoritmo de Euclides computa $\mathrm{mdc}(a, b)$ em tempo $O(\log(|a|)\log(|b|))$. Consideremos a seguinte sequência dos parâmetros das chamadas recursiva do Algoritmo de Euclides, começando com $(a, b) = (r_0, r_1)$

$$\begin{aligned}
(r_1, r_0 \bmod r_1) &= (r_1, r_2) \\
(r_2, r_1 \bmod r_2) &= (r_2, r_3) \\
&\vdots \\
(r_{\ell-2}, r_{\ell-3} \bmod r_{\ell-2}) &= (r_{\ell-2}, r_{\ell-1}) \\
(r_{\ell-1}, r_{\ell-2} \bmod r_{\ell-1}) &= (r_{\ell-1}, r_\ell)
\end{aligned}$$

e $r_{\ell+1} = 0$. O custo em cada linha é o de uma divisão, ou seja, na linha i, $1 \leqslant i < \ell$, o custo é $O(\log(r_i)\log(q_i))$, onde $r_{i-1} = r_i q_i + r_{i+1}$. O custo total é

$$\sum_{i=1}^{\ell} \log(r_i)\log(q_i) \leqslant \log(b) \sum_{i=1}^{\ell} \log(q_i) = \log(b)\log(q_1 q_2 \cdots q_\ell) \leqslant \log(b)\log(a)$$

pois $a = r_0 \geqslant r_1 q_1 \geqslant r_2 q_2 q_1 \geqslant \cdots \geqslant r_\ell q_\ell \cdots q_2 q_1 \geqslant q_\ell \cdots q_2 q_1$.

Teorema 1.68 *O algoritmo* Euclides *com entrada* (a, b) *tem tempo de execução de pior caso* $O(\log(|a|)\log(|b|))$. □

Exercício 1.69 (o pior caso do Algoritmo de Euclides). No que segue, f_k é o k-ésimo número de Fibonacci, definido recursivamente por $f_1 = 1$, $f_2 = 1$ e $f_k = f_{k-1} + f_{k-2}$ para todo $k \geqslant 3$. Mostre que o Algoritmo de Euclides com entradas f_{k+2} e f_{k+1} executa k chamadas recursivas. Prove o seguinte resultado: *se* (a, b) *é o menor par de inteiros positivos que faz o Algoritmo de Euclides executar* k *chamadas recursivas então* $(a, b) = (f_{k+1}, f_{k+2})$.

O Algoritmo de Euclides estendido O Algoritmo de Euclides Estendido determina uma solução (x, y) inteira da equação

$$ax + by = \mathrm{mdc}(a, b)$$

para a, $b \in \mathbb{Z}$ quaisquer. Esse algoritmo é bastante útil na seguinte situação. Se $\mathrm{mdc}(a, n) = 1$ para inteiros a e $n > 1$, então a equação $ax \equiv c \pmod{n}$ tem solução, ou seja, existem x e y inteiros tais que $ax + ny = c$ e o algoritmo estendido os encontra. No caso $c = 1$ a solução é um **inverso multiplicativo de** a **módulo** n.

Instância: (a, b) par de inteiros não negativos.
Resposta: uma terna (d, x, y) tal que $d = \text{mdc}(a, b) = ax + by$.

1 **se** $b = 0$ **então responda** $(a, 1, 0)$.
2 $(d, x, y) \leftarrow$ Euclides_estendido$(b, a \bmod b)$
3 **responda** $(d, y, x - \lfloor a/b \rfloor y)$.

Algoritmo 6: Euclides_estendido(a, b).

Uma execução do algoritmo acima com entradas 86 e 64 resulta nos seguintes valores

a	b	$\lfloor a/b \rfloor$	x	y	d
86	64	1	3	−4	2
64	22	2	−1	3	2
22	20	1	1	−1	2
20	2	10	0	1	2
2	0	—	1	0	2

Exercício 1.70 (tempo de execução do algoritmo euclidiano estendido). Verifique que o tempo de execução do Algoritmo 6 é $O(\log(|a|)\log(|b|))$.

Exercício 1.71 (correção do algoritmo euclidiano estendido). Prove que o Algoritmo 6 responde corretamente. Para isso, suponha que (d', x', y') são os valores atribuídos na linha 2

$$d' = bx' + (a \bmod b)y' = bx' + (a - \lfloor a/b \rfloor b)y' = ay' + b(x' - \lfloor a/b \rfloor y').$$

A prova segue por indução.

Exponenciação modular Computar 2^n no modo tradicional é extremamente custoso quando n é grande; com 100 dígitos isso daria cerca de 10^{100} passos o que é impossível de realizar manualmente sem atalhos. Em geral a^b avaliado por multiplicações repetidas tem tempo de execução $\Omega(b(\log a)^2)$. Um jeito mais esperto é "elevar ao quadrado" repetidas vezes, por exemplo, para calcular 2^{24} podemos começar com $2^3 = 8$, elevá-lo ao quadrado, o que resulta $2^6 = 62$, elevá-lo ao quadrado, o que resulta $2^{12} = 4.096$, e elevá-lo ao quadrado, o que resulta $2^{24} = 16.777.216$. Para calcular 2^{29} com esse método, recursivamente, $2^{29} = 2 \cdot 2^{28}$, a raiz de 2^{28} é 2^{14} cuja raiz é 2^7 que é $2 \cdot 2^6$ que por sua vez é $2^2 \cdot 2^3$. Em resumo, a^b é avaliado com base na observação de que

$$a^b = \begin{cases} (a^{b/2})^2 & \text{se } b \text{ é par,} \\ a \cdot a^{b-1} & \text{se } b \text{ é ímpar.} \end{cases}$$

O seguinte algoritmo é uma versão iterativa da recursão acima para calcular $a^b \pmod n$. Seja $b_k b_{k-1} \dots b_1 b_0$ a representação binária de b e definimos

$$c_i := b_k 2^{i-1} + b_{k-1} 2^{i-2} + \cdots + b_{k-i+1} 2^0$$
$$d_i := a^{c_i} \bmod n$$

para $i \geqslant 1$ com $c_0 = 0$ e $d_0 = 1$. Computamos d_{i+1} a partir de d_i da seguinte forma

$$c_{i+1} = \begin{cases} 2c_i, & \text{se } b_{k-i} = 0 \\ 2c_i + 1, & \text{se } b_{k-i} \neq 0 \end{cases};$$

e

$$d_{i+1} = \begin{cases} {d_i}^2 \bmod n = a^{c_{i+1}} \bmod n = (a^{c_i})^2 \bmod n & \text{se } b_{k-i} = 0 \\ a \cdot {d_i}^2 \bmod n = a^{c_{i+1}} \bmod n = a \cdot (a^{c_i})^2 \bmod n, & \text{se } b_{k-i} \neq 0. \end{cases}$$

Portanto, $c_{k+1} = b_k 2^k + \cdots + b_1 2 + b_0 = b$ e $d_{k+1} = a^b \bmod n$.

Exemplo 1.72. Vamos usar essa estratégia para calcular $2^{24} \bmod 25$. Primeiro, 24 em base 2 fica $b = 11000$.

i	0	1	2	3	4	5
b_{4-i}	1	1	0	0	0	
c_i	0	1	3	6	12	24
d_i	1	2	8	14	21	16

Portanto $2^{24} \equiv 16 \pmod{25}$. ◊

Instância: inteiros não negativos a, b e $n > 1$.
Resposta: $a^b \bmod n$.

```
1  c ← 0;
2  d ← 1;
3  Seja b_k b_{k-1} ... b_1 b_0 a representação binária de b;
4  para i de k até 0 faça
5  |  c ← 2 · c;
6  |  d ← d · d mod n;
7  |  se b_i = 1 então
8  |  |  c ← c + 1;
9  |  |  d ← d · a  mod n;
10 responda d.
```

Algoritmo 7: exponenciação modular.

Notemos que em toda iteração os valores de d têm no máximo tantos dígitos quanto n, ou seja, têm $O(\log n)$ dígitos, portanto, a multiplicação e o resto têm tempo de execução $O(\log^2 n)$. O número de iterações é a quantidade de bits na representação de b, logo $O(\log b)$. Isso prova o seguinte resultado.

Teorema 1.73 *Se* $a = O(\log n)$ *então o Algoritmo 7 determina* $a^b \bmod n$ *em tempo* $O(\log(b)\log^2(n))$. □

Observação 1.74 (*sobre o custo computacional das operações aritméticas*). O custo das operações aritméticas elementares usados acima são os custos dos algoritmos escolares, não os dos mais eficientes. Se $M(n)$ é o custo para multiplicar dois números de até n bits, então temos os seguintes tempos de execução

Multiplicação	$M(n)$
Divisão	$O(M(n))$ (Newton–Raphson)
MDC	$O(M(n)\log n)$ (Stehlé–Zimmermann)
Exponenciação modular	$O(kM(n))$, k é o tamanho do expoente

Tabela 1.4: custo das operações aritméticas.

O tempo de uma multiplicação do algoritmo escolar é $M(n) = O(n^2)$. talvez o algoritmo mais usado no momento que este texto foi escrito é o algoritmo de Schönhage–Strassen de 1971 cujo tempo de execução de pior caso é $O(n\log n\log\log n)$ para dois números de n dígitos. Esse algoritmo foi o método de multiplicação mais rápido até 2007, quando o algoritmo de Fürer foi anunciado. Entretanto, o algoritmo de Fürer só alcança uma vantagem para valores astronomicamente grandes e não é usado na prática.

Harvey e Van Der Hoeven (2019) publicaram um algoritmo de tempo $O(n\log n)$ para a multiplicação de inteiros. Como Schönhage e Strassen conjecturam que $n\log(n)$ é mínimo necessário para a multiplicação esse pode ser o "melhor possível", porém, até o momento esse algoritmo também não é útil na prática pois os autores observam que as estimativas de custo valem para números com pelo menos $2^{4.096}$ bits. O número estimado de átomos no universo é $10^{80} \approx 2^{266}$.

1.6 TESTE DE IGUALDADE DE CADEIAS DE BITS

Nessa seção ilustramos como a aleatorização pode nos ajudar a resolver problemas computacionais de modo mais eficiente que as soluções determinísticas. Eficiência aqui é com respeito a troca de informação entre partes, isto é, estamos falando de complexidade de comunicação: Alice e Bob, desejam computar colaborativamente uma função $f(x,y)$ conhecida por ambos, onde $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Alice só conhece x, Bob só conhece y, a comunicação é cara e eles já tinham acordado um *protocolo* para comunicação. O custo desse protocolo é o número de bits comunicados entre partes para a pior escolha de entrada (x,y).

Em um protocolo Alice envia uma mensagem ma_1 para Bob; Bob, de posse dessa mensagem e de y, responde com alguma mensagem ba_1 (que depende apenas de ma_1 e de y); continuam dessa forma até que um deles seja capaz de calcular o valor de $f(x,y)$ e comunicá-lo para a outra parte. A complexidade, ou custo, do protocolo é o total de bits $|ma_1|+|ba_1|+\cdots+|ma_t|+|ba_t|$ (a mensagem x tem $|x|$ bits) trocados no pior caso, isto é, com a instância que maximiza o total de bits trocados entre as partes.

A complexidade de comunicação da função f é definida como o custo mínimo assumido dentre todos os protocolos legítimos para o cálculo de f. Por exemplo a função $f(x,y)=1$ se, e só se, $x=y$, tem complexidade de computação pelo menos n e testa "$x=y$?" (Arora e Barak, 2009, teo. 13.4).

Alice e Bob têm um enorme banco de dados, digamos que $\mathsf{A}=a_1a_2\ldots a_n$ e $\mathsf{B}=b_1b_2\ldots b_n$, respectivamente, que eles querem saber se são iguais.

Uma saída trivial é: Alice envia os n bits para Bob, então Bob verifica se os dois vetores são os mesmos e envia o resultado (sim ou não) para Alice. Toda a comunicação usa $n+1$ mensagens de um bit e é o melhor que podemos fazer.

O seguinte protocolo aleatorizado é mais econômico: encare as sequências binárias como a representação de dois número na base 2, sorteie um primo não muito grande e verifique se esses dois números deixam o mesmo resto quando divididos pelo número sorteado; os restos têm o mesmo tamanho do primo.

1. Alice sorteia um primo $p\leqslant n^2$ e manda os $\lfloor\log_2(p)\rfloor+1$ bits para Bob;

   Alice constrói o polinômio $\mathsf{A}(x)=a_1+a_2x+a_3x^2+\cdots+a_nx^{n-1}$, calcula $\mathsf{A}(2)$ e envia os $\lfloor\log_2(p)\rfloor+1$ bits de $\mathsf{A}(2)\bmod p$ para Bob;

2. Bob constrói o polinômio $\mathsf{B}(x)=b_1+b_2x+b_3x^2+\cdots+b_nx^{n-1}$, calcula $\mathsf{B}(2)\bmod p$ e compara com os $\lfloor\log_2(p)\rfloor+1$ bits de $\mathsf{A}(2)\bmod p$;

   Bob manda 1 para Alice se $\mathsf{A}(2)\bmod p=\mathsf{B}(2)\bmod p$ ou 0, caso contrário.

O custo desse protocolo é

$$2(\lfloor\log_2(p)\rfloor+1)+1\leqslant 2\log_2(n^2)+2+1<4(\log_2(n)+1)=4\log_2(2n)$$

pois se $p\leqslant n^2$, então $\log_2(p)\leqslant\log_2(n^2)$.

Tal protocolo erra se $\mathsf{A}\neq\mathsf{B}$, porém $\mathsf{A}(2)\equiv\mathsf{B}(2)\pmod p$. Por exemplo $1000000_2=64$ e $1001011_2=75$ e $64\bmod 11=9=75\bmod 11$. Notemos que dentre todos[7] os números primos entre 2 e 49, o 11 é o único com a propriedade de que $64\bmod p=75\bmod p$. Se $\mathsf{A}=\mathsf{B}$ o protocolo não erra.

[7] 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 e 47.

A probabilidade de erro é, então, a quantidade de primos $\leqslant n^2$ que causam erro dividido pela quantidade de primos $\leqslant n^2$.

(i) A quantidade de primos $\leqslant n^2$ que causam erro: Um primo p causa erro quando $A(2) \equiv B(2) \pmod{p}$, ou seja, p divide $m := A(2) - B(2)$. Se m é um inteiro positivo, temos do Teorema Fundamental da Aritmética que $m = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, onde $p_1, \ldots, p_k$ são os primos que dividem m. Como $p_i \geqslant 2$, temos que $m \geqslant 2^k$ de modo que $k \leqslant \log_2(m)$, isto é, m tem no máximo $\log_2(m)$ divisores primos. Ademais, $m < 2^n$, portanto, concluímos que são $\leqslant n$ divisores de $m = A(2) - B(2)$.

(ii) A quantidade de primos $\leqslant n^2$: Aqui usamos uma estimativa de Chebyshev para $\pi(n^2)$, equação (1.25) na página 62,

$$\pi(n^2) > \frac{n^2}{\ln(n^2)}$$

para todo $n \geqslant 17$.

Usando (i) e (ii)

$$\mathbb{P}[\,\text{erro}\,] \leqslant \frac{n}{n^2/\ln(n^2)} = \frac{\ln(n^2)}{n}.$$

Digamos que Alice está em alum lugar da Europa, Bob no Brasil e a base de dados deles tem 1,25 petabytes (PB), ou $n = 10^{16}$ bits. O tempo necessário para transmitir 1,25 PB de dados da Europa para o Brasil com uma largura de banda de 10 Gbps (10^{10} bits por segundo) é 1.000.000 segundos ou 11 dias, aproximadamente.

Por outro lado, o protocolo aleatorizado transmite apenas $2(\lfloor \log_2(n^2) \rfloor + 1) + 1 = 215$ bits, o que é muito rápido. Entretanto, há uma chance de erro medida pela probabilidade, que nesse caso é

$$\leqslant \frac{32\ln(10)}{10^{16}} = 2{,}31 \times 10^{-16}.$$

Repetido esse protocolo sorteando 5 primos, a probabilidade de erro cai para, aproximadamente, 10^{-81}.

Cabe destacar que canais de transmissão estão sujeitos a erros. As especificações dos hardwares de comunicação trazem uma taxa de erro de bits. A comunidade dessa área tem um consenso sobre o que é uma taxa aceitável, digamos que por volta 10^{-12}, o que significa que para cada 10^{12} bits transmitidos é esperado que 1 bit seja transmitido incorretamente. O protocolo determinístico, por transmitir mais dados, está mais sujeito aos erros de transmissão. Na prática, para garantir a integridade dos dados transmitidos, os protocolos de comunicação usam técnicas de redundância e detecção de erros, como códigos de correção de erros (ECC) e *checksums*, o que torna o processo ainda mais custoso.

1.7 ALGORITMOS ALEATORIZADOS

Nos algoritmos aleatorizados o tempo de execução depende do tempo para os sorteios realizados. Formalmente (seção **??**), consideramos que os algoritmos sorteiam bits de modo uniforme e independente, com cada sorteio em tempo constante, de modo que o sorteio de $a \in \Omega$ consome tempo proporcional ao tamanho da representação binária dos elementos de Ω, isto é, $O(\log|\Omega|)$. Em geral, se $\log|\Omega|$ for polinomial no tamanho da entrada então um sorteio não afeta a ordem do tempo de execução de um algoritmo eficiente e podemos considerar o tempo de um sorteio como sendo constante.

No teste de identidade polinomial, Algoritmo 1 na página 25, por exemplo, o tempo de execução depende do tempo para o sorteio do número a e do tempo para computar $f(a)$. O tempo para computar $f(a)$ depende da representação de f e será eficiente se for feito em tempo polinomial no tamanho da representação de f. Esse é o caso quando o polinômio é dado explicitamente e, assim, o Algoritmo 1 é um algoritmo probabilístico de tempo polinomial que erra com probabilidade limitada. Esse tipo de algoritmo, com tempo de execução determinístico e chance de errar limitada, é chamado na literatura de algoritmo *Monte Carlo*.

O algoritmo gerador de números aleatórios, Algoritmo 2, página 44, sempre responde certo. Em contrapartida o tempo de execução pode ser diferente em execuções distintas com a mesma instância. Esse tipo de algoritmo é chamado de *Las Vegas*.

Dado um algoritmo A e uma instância x para A, podemos representar as possíveis computações de A com x com uma árvore binária $T_{A,x}$ (Figura 1.5) em que cada

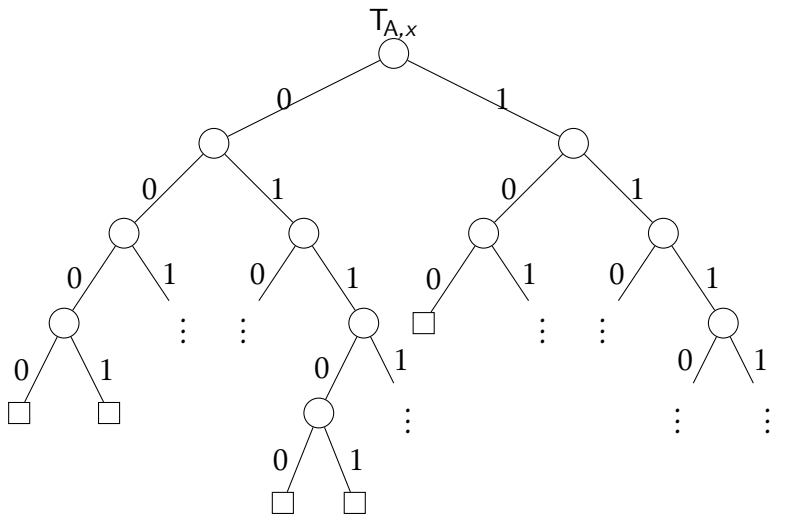


Figura 1.5: árvore de execução de A com instância x.

ramificação significa que um sorteio foi realizado, uma computação c específica está associada a um caminho da raiz até alguma folha ($\square$) e ela ocorre com probabilidade $\mathbb{P}(c) = 2^{-|c|}$ onde $|c|$ é o comprimento (número de arestas) no caminho.

Para termos um exemplo simples de uma árvore de execução, vamos modificar ligeiramente o Algoritmo 1 para que faça até dois sorteios: se no primeiro $f(a) \neq 0$ então pode parar e responder *não*, senão faça mais um sorteio. Além disso, as instâncias são polinômios de grau no máximo 2.

```
1 a ←R {1,2,3,4};
2 se f(a) ≠ 0 então responda não.
3 senão
4 |   a ←R {1,2,3,4};
5 |   se f(a) ≠ 0 então responda não.
6 |   senão responda sim.
```

As computações desse algoritmo com entrada $(x-1)(x-3)$ em função dos sorteios são caracterizadas pelas sequências: $(1,1,\textit{não})$, $(1,2,\textit{não})$, $(1,3,\textit{sim})$ e $(1,4,\textit{não})$, $(2,\textit{não})$, $(4,\textit{não})$, $(3,1,\textit{não})$, $(3,2,\textit{não})$, $(3,3,\textit{sim})$ e $(3,4,\textit{não})$, esquematizadas na Figura 1.6. Dos sorteios independentes decorre que a probabilidade de uma computação é o produto das probabilidades no ramo daquela computação, logo

$$\mathbb{P}[\,\text{erro}\,] = \mathbb{P}((1,3,\textit{sim})) + \mathbb{P}((3,3,\textit{sim})) = \frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}.$$

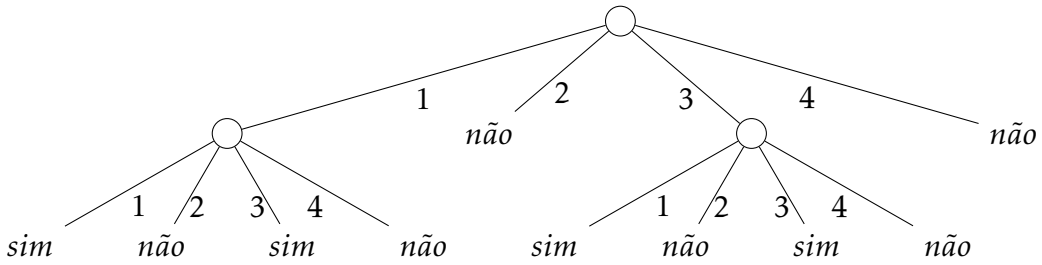


Figura 1.6: árvore de execução do algoritmo acima com entrada $(x-1)(x-3)$.

Definimos o modelo probabilístico discreto $(\Omega_{A,x}, \mathbb{P})$ com o espaço amostral formados pelos caminhos c na árvore de execução e a medida $2^{-|c|}$ em que $c \in \Omega_{A,x}$ é um ramo da árvore e $|c|$ o número de arestas em c.

Exercício 1.75. Verifique que para quaisquer dois ramos distintos de $T_{A,x}$ vale que a sequência binária de um ramo não pode ser prefixo da sequência de outro ramo. Prove que $\rho = \sum_c 2^{-|c|} \leqslant 1$. Nessa situação, ρ é a probabilidade com que A com instância x termina a computação?

Por fim, registramos que há, ainda, algoritmos aleatorizados que têm probabilidade de errar e têm probabilidade de demorar muito pra terminar e que em algumas referências são chamados de *Atlantic City*.

Amplificação Para problemas de decisão, aqueles que respondem "sim" ou "não" como no Algoritmo 1, a probabilidade de sucesso pode ser amplificada.

Os algoritmos Monte Carlo ainda são classificados em duas classes: de erro unilateral, que só erra uma das repostas, ou de erro bilateral, que erra nas duas respostas.

Para algoritmos Monte Carlo de erro unilateral, com probabilidade de erro $\leqslant \varepsilon$, podemos usar a estratégia de repetições independentes para atingir um limiar δ para o erro. Se A é um algoritmo Monte Carlo que erra nas respostas "sim", então em t rodadas do algoritmo com a mesma entrada o erro só se dá se todas as t respostas forem "sim", pois qualquer resposta "não" está correta, é definitiva. A probabilidade de erro será no máximo δ para todo t tal que $\varepsilon^t \leqslant \delta$, portanto,

$$t \geqslant \frac{\log(1/\delta)}{\log(1/\varepsilon)} = O(\log(1/\delta)).$$

Proposição 1.76 *Dados $\delta \in (0,1)$ e A um algoritmo Monte Carlo de erro unilateral, com $O(\log(1/\delta))$ repetições de A temos um algoritmo Monte Carlo de erro (unilateral) no máximo δ.* □

No caso de erro bilateral nenhuma das duas respostas é definitiva, a estratégia é repetir o algoritmo t vezes e devolver a resposta de maior ocorrência.

Proposição 1.77 *Dados $\delta \in (0,1)$ e A um algoritmo Monte Carlo de erro (bilateral) no máximo $1/5$, com $O(\log(1/\delta))$ repetições de A temos um algoritmo Monte Carlo de erro (bilateral) no máximo δ.* □

Demonstração. Em t repetições de A a resposta final está errada se menos que $t/2$ das repetições respondem corretamente. Usando o Exercício 1.51, a probabilidade de erro é no máximo $(1-1/5)^t$, portanto, com $O(\log(1/\delta))$ repetições de A temos um algoritmo Monte Carlo de erro no máximo δ. □

Exercício 1.78. Não há nada de especial na escolha de $1/5$ na probabilidade de erro, para conseguirmos amplificar o erro precisamos de probabilidade de erro menor que $1/2$. Prove a proposição acima para A algoritmo Monte Carlo de erro (bilateral) no máximo $1/2 - \varepsilon$ para $\varepsilon \in (0, 1/2)$ fixo (dica: Exercício 1.52).

Desaleatorização Os bits aleatórios que um algoritmo usa para resolver um problema é um recurso que queremos otimizar.

Todo algoritmo aleatorizado pode ser desalatorizado. Se um algoritmo aleatorizado resolve um problema computacional usando $O(r(n))$ bits aleatórios, então simulamos o algoritmo $2^{O(r(n))}$ vezes, uma para cada sequencia de bits aleatório e sem usar sorteio. Por exemplo, no caso ilustrado na Figura 1.6, simular todos os sorteios corresponde a percorre todos os ramos da árvore e responder "não", que é a resposta da maioria.

Assim, o objetivo da desaleatorização é reduzir ou eliminar o uso de sorteios porém sem penalizar muito os outros recursos. Um caso notório é a desaleatorização do Teste de Primalidade de Agrawal–Biswas (seção 3.1.6) que resultou no primeiro teste de primalidade determinístico de tempo polinomial.

Há várias técnicas de desaleatorização e veremos algumas neste texto. Não se conhece uma técnica que desaleatorize qualquer algoritmo aleatorizado sem penalizá-lo muito com relação ao tempo de execução. Esse é um problema importante da Complexidade Computacional.

1.7.1 CORTE-MÍNIMO EM GRAFOS

Um *grafo* G é dado por um par de conjuntos (V, E) em que V é finito, é o conjunto dos *vértices* de G, e $E \subseteq \binom{V}{2}$. O *grau* de um vértice $x \in V$ em G é a quantidade de arestas de E a que x pertence. Se contamos o número de pares $(v, e) \in V \times E$ tais que $v \in e$ temos, pela definição de grau, que a quantidade de pares é a soma dos graus dos vértices. Por outro lado, cada aresta é composta por dois vértices de modo que a quantidade de pares é $2|E|$. Esse resultado quase sempre é o primeiro teorema nos textos de Teoria dos Grafos: *em todo grafo, a soma dos graus dos vértices é duas vezes o número de arestas do grafo*. Nesta seção assumimos, sem perda de generalidade, que os grafos são sobre os vértices $V = \{1, 2, \ldots, n\}$ para algum n.

Um subconjunto de arestas de um grafo $G = (V, E)$ da forma

$$\nabla(A) := \left\{ \{u, v\} \in E \colon u \in A \text{ e } v \in \overline{A} \right\}$$

é chamado de **corte definido por** A em G.

Exemplo 1.79. A Figura 1.7 abaixo mostra um grafo G e um corte de arestas $\nabla(A)$ definido por $A = \{0, 1, 2, 7, 8\}$, o qual é formado pelas arestas (em azul na figura) $\{0,4\}, \{0,5\}, \{1,3\}, \{1,6\}, \{8,3\}, \{6,8\}, \{5,7\}, \{6,7\}, \{2,3\}, \{2,4\}$ que cruzam a reta vertical pontilhada. ◊

Um **corte mínimo** em G é um corte com

$$\text{mincut}(G) := \min\left\{ \left|\nabla(A)\right| \colon \varnothing \neq A \subsetneq V \right\}$$

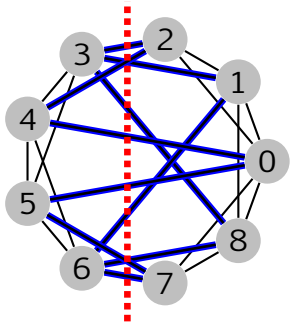


Figura 1.7: o corte definido por $\{0,1,2,7,8\}$ são as arestas em azul.

arestas.

No grafo do Exemplo 1.79, o corte definido por $\{2\}$ é mínimo, assim como o definido por $\{7\}$. O problema em que estamos interessados é enunciado como segue.

Problema computacional do corte mínimo em um grafo (MIN-CUT):
Instância: um grafo G.
Resposta: o tamanho de um corte mínimo.

A seguir responderemos uma versão de decisão desse problema: dados um grafo G e um inteiro positivo k, responda *sim* se $\text{mincut}(G) \leqslant k$, responda *não* caso contrário.

Para explicar um algoritmo probabilístico para esse problema, precisaremos de uma definição mais geral de grafo. Em um *multigrafo* as arestas formam um multiconjunto no qual entre um par de vértices pode haver mais de uma aresta[8].

Seja M um multigrafo. Para qualquer aresta e de M definimos por *contração da aresta* e a operação que resulta no multigrafo com os extremos da aresta e identificados e as arestas com esses extremos removidos, o multigrafo resultante é denotado por M/e (veja uma ilustração na Figura 1.9).

A ideia do algoritmo para decidir se $\text{mincut}(G) \leqslant k$ é repetir as operações

1. sortear uniformemente uma aresta,

2. contrair a aresta sorteada,

até que restem 2 vértices no multigrafo. As arestas múltiplas que ligam esses 2 vértices são arestas de um corte no grafo original. Os próximos parágrafos ilustram a ideia do algoritmo que será apresentado em seguida; para facilitar a compreensão mantemos nos rótulos dos vértices todas as identificações realizadas.

Considere o grafo G representado pelo diagrama da Figura 1.8. A Figura 1.9

[8] Formalmente, um multigrafo é definido por uma terna (V, E, ϕ) onde V e E são conjuntos e $\phi\colon E \to \binom{V}{2}$ associa a cada aresta um par de vértices.

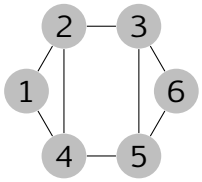


Figura 1.8: exemplo de um grafo.

abaixo representa uma sequência de três contrações de arestas, a aresta que sofre a contração está em vermelho. Se no último multigrafo da Figura 1.9 contraímos a

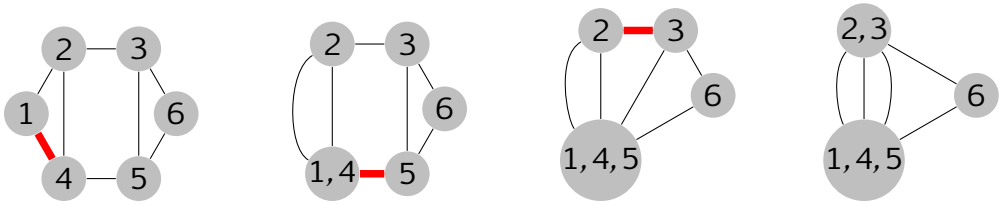
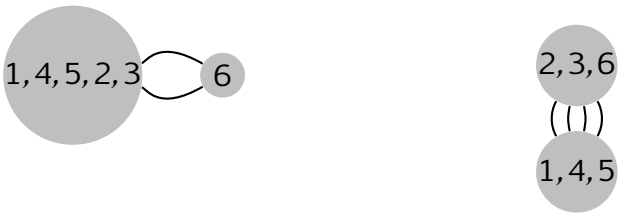


Figura 1.9: 4 multigrafos que representam uma sequência 3 três contrações de aresta. A contração da aresta vermelha resulta no multigrafo a direita. Para manter o registro das contrações acumulamos os rótulos nos vértices.

aresta dos vértices representados por $1,4,5$ e por $2,3$, então o multigrafo resultante dessa contração, mostrado na Figura 1.10(a), corresponde ao corte definido por $A = \{6\}$ no grafo original G. Esse corte em G tem duas arestas e é um corte mínimo. Por outro lado, se identificarmos os vértices representados por $2,3$ com 6, então o multigrafo obtido corresponde ao corte definido por $A = \{2,3,6\}$ em G e que tem 4 arestas, como na Figura 1.10(b).



(a) (b)

Figura 1.10: dois resultados possíveis para contração a partir do último multigrafo da Figura 1.9. Em (a) o resultado da contração da aresta de extremos $1,4,5$ e $2,3$. Em (b) o resultado da contração da aresta $2,3$ com 6. Ambos correspondem a um corte no grafo original G.

Exercício 1.80. Seja G um grafo. Prove que após uma sequência qualquer de contrações de arestas de G, um corte no multigrafo resultante corresponde a um corte no grafo original. Conclua que a sequência de operações realizadas, sortear aresta e contrair a aresta sorteada até que restem 2 vértices, determina um corte em G.

Sejam $G = (V, E)$ um grafo com n vértices e $C = \nabla(A)$ um corte mínimo em G. Vamos mostrar que a probabilidade do algoritmo que descrevemos encontrar o corte C é pelo menos $\binom{n}{2}^{-1} = \Omega(n^{-2})$. De $\text{mincut}(G) = |C|$ o grau mínimo de um vértice em G é pelo menos $|C|$, portanto, G tem pelo menos $|C|n/2$ arestas.

O espaço amostral nesse caso é dado pelas sequências de $n-2$ arestas distintas que correspondem as escolhas aleatórias do algoritmo. O algoritmo executado sobre G encontra o corte mínimo $C = \nabla(A)$ se nas $n-2$ rodadas somente contrai arestas com ambos os extremos em A, ou com ambos extremos em $\overline{A}$.

Denotemos por B_i o evento "a i-ésima escolha aleatória, a aresta e_i, não está em C". A probabilidade de escolher uma aresta de C na primeira escolha aleatória é

$$\frac{|C|}{|E(G)|} \leqslant \frac{|C|}{|C|n/2} = \frac{2}{n}$$

logo $\mathbb{P}(B_1) \geqslant 1 - \frac{2}{n}$. Agora, denotemos por G_1 o grafo resultante da primeira rodada de contrações. A probabilidade de escolher uma aresta de C na segunda escolha, dado que na primeira escolha não ocorreu uma aresta de C é

$$\frac{|C|}{|E(G_1)|} \leqslant \frac{|C|}{|C|(n-1)/2} = \frac{2}{n-1}$$

pois o multigrafo tem $n-1$ vértices e grau mínimo pelo menos $|C|$, logo

$$\mathbb{P}(B_2 \mid B_1) \geqslant 1 - \frac{2}{(n-1)}$$

e, genericamente, na i-ésima escolha a probabilidade de escolher uma aresta de C dado que até agora não foi escolhida uma aresta de C é

$$\mathbb{P}\left(B_i \;\middle|\; \bigcap_{j=1}^{i-1} B_j\right) \geqslant 1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}.$$

A probabilidade de nenhuma aresta escolhida ser de C é $\mathbb{P}(B_1 \cap B_2 \cap \cdots \cap B_{n-2})$ e pela Regra do Produto (Exercício 1.24, página 30), temos que

$$\mathbb{P}\left(\bigcap_{i=1}^{n-2} B_i\right) \geqslant \prod_{i=1}^{n-2}\left(\frac{n-i-1}{n-i+1}\right) = \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}.$$

Este algoritmo, devido a Karger (1993), recebe um grafo G com pelo menos 3 vértices e um inteiro positivo k, e responde *sim* ou *não*. Quando o algoritmo responde

sim é porque foi descoberto um corte com até k arestas, portanto, o corte mínimo tem tamanho no máximo k. Por outro lado, a resposta *não* significa que o algoritmo não achou um corte com até k arestas, o que não significa que o grafo não o tenha, portanto, a resposta *não* pode estar errada.

Instância: um grafo G com $n \geqslant 3$ vértices e $k \in \mathbb{N}$.
Resposta: *sim* caso $\text{mincut}(G) \leqslant k$, senão *não* com probabilidade de erro $< 1/2$.

1 **repita**
2 $i \leftarrow 0$;
3 $G_0 \leftarrow G$;
4 **repita**
5 $e \xleftarrow{R} E(G_i)$;
6 $G_{i+1} \leftarrow G_i/e$;
7 $i \leftarrow i+1$;
8 **até que** $i = n-2$;
9 **se** $|E(G_{n-2})| \leqslant k$ **então responda** *sim*.
10 **até que** *complete* $\binom{n}{2}$ *rodadas;*
11 **responda** *não*.

Algoritmo 9: corte mínimo.

Acima provamos o seguinte resultado.

Proposição 1.81 *Seja G um grafo com pelo menos três vértices. Fixado um corte mínimo C em G, a probabilidade do Algoritmo 9 determinar C no laço da linha 4 é pelo menos $1/\binom{n}{2}$.* □

Agora, vamos determinar a probabilidade de erro.

Teorema 1.82 *Dados G e k instância do Algoritmo 9, se $\text{mincut}(G) \leqslant k$, então*

$$\mathbb{P}[\,\text{erro}\,] = \mathbb{P}[\,\text{resposta } \textit{não}\,] < \frac{1}{e}.$$

Demonstração. Se G tem um corte com no máximo k arestas, então a probabilidade do algoritmo não encontrar um tal corte em nenhuma das iterações do laço na linha 1 é no máximo

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{\binom{n}{2}}.$$

Da sequência decrescente $(1-(1/n))^n$ convergir para e^{-1} (veja (s.8) no apêndice) temos que qualquer subsequência converge para o mesmo valor

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{\binom{n}{2}} \leqslant \frac{1}{e}$$

e isso prova o teorema. □

O número de instruções executadas no laço mais interno desse algoritmo é $O(|E|) = O(n^2)$. Contando as execuções do laço externo são $O(n^4)$ instruções executadas. A contração de uma aresta pode ser feita em $O(n)$. Notemos que se o número de rodadas dado pela condição na linha 10 for $\ell\binom{n}{2}$ então a probabilidade de erro é menor que $e^{-\ell}$. Nesse caso, pra $\ell = c \log n$ a probabilidade de erro é n^{-c} e o custo (tempo) é $O(n^5 \log n)$.

1.7.2 VERIFICAÇÃO DO PRODUTO DE MATRIZES

Nesta seção veremos um algoritmo que recebe as matrizes A, B e C e verifica o produto $A \cdot B = C$ realizando menos operações aritméticas que o o melhor algoritmo conhecido até hoje realiza para determinar o produto $A \cdot B$.

Problema computacional do teste de produto de matrizes:
Instância: matrizes A, B, C quadradas de ordem n sobre um corpo.
Resposta: *sim* se $AB = C$, caso contrário *não*.

Esse teste pode ser feito com custo $O(n^3)$ usando o algoritmo usual (escolar) para o produto de matrizes. Um dos algoritmos mais eficientes conhecidos é o de Coppersmith–Winograd (veja em Knuth, 1981), que realiza o produto de duas matrizes $n \times n$ perfazendo da ordem de $n^{2,376}$ operações aritméticas. O algoritmo aleatorizado devido a Freivalds (1977) apresentado a seguir decide se $AB = C$ com $O(n^2)$ operações aritméticas, mas pode responder errado caso $AB \neq C$.

A ideia do algoritmo de Freivalds para esse problema é que se $AB = C$ então $(vA)B = vC$ para todo vetor v e esse último teste tem custo da ordem de n^2 operações aritméticas. Porém, se $AB \neq C$ então é possível termos $(vA)B = vC$, por exemplo, caso v seja nulo. O que conseguimos garantir é que se o vetor v é aleatório então tal igualdade ocorre com probabilidade pequena.

Por exemplo, sejam

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \text{ e } B = \begin{pmatrix} 3 & 6 & 9 \\ 1 & 2 & 1 \\ 3 & 1 & 3 \end{pmatrix}, \; AB = \begin{pmatrix} 14 & 13 & 20 \\ 35 & 40 & 59 \\ 56 & 67 & 98 \end{pmatrix} \text{ e } C = \begin{pmatrix} 14 & 13 & 20 \\ 10 & 20 & 10 \\ 56 & 67 & 98 \end{pmatrix}$$

de modo que $AB \neq C$. Consideremos v um vetor binário. Para $v = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$ temos

$$\begin{aligned} vAB &= 0\begin{pmatrix} 14 & 13 & 20 \end{pmatrix} + 1\begin{pmatrix} 35 & 40 & 39 \end{pmatrix} + 0\begin{pmatrix} 56 & 67 & 98 \end{pmatrix} = \begin{pmatrix} 35 & 40 & 39 \end{pmatrix} \\ vC &= 0\begin{pmatrix} 14 & 15 & 13 \end{pmatrix} + 1\begin{pmatrix} 10 & 20 & 10 \end{pmatrix} + 0\begin{pmatrix} 56 & 67 & 98 \end{pmatrix} = \begin{pmatrix} 10 & 20 & 10 \end{pmatrix} \end{aligned}$$

portanto, $vABC \neq vC$, enquanto que para $v = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$ temos

$$vAB = 0\begin{pmatrix} 14 & 13 & 20 \end{pmatrix} + 0\begin{pmatrix} 35 & 40 & 39 \end{pmatrix} + 1\begin{pmatrix} 56 & 67 & 98 \end{pmatrix} = \begin{pmatrix} 56 & 67 & 98 \end{pmatrix}$$
$$vC = 0\begin{pmatrix} 14 & 13 & 20 \end{pmatrix} + 0\begin{pmatrix} 10 & 20 & 10 \end{pmatrix} + 1\begin{pmatrix} 56 & 67 & 98 \end{pmatrix} = \begin{pmatrix} 56 & 67 & 98 \end{pmatrix}$$

portanto, $vABC = vC$.

De fato, $vAB = vC$ para todo $v \in \{0,1\}^3$ cuja segunda coluna seja 0, isto é, para

$$v \in \left\{\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}\right\}$$

vale a igualdade, para qualquer outro vetor binário vale a diferença $vAB \neq vC$. Nesse exemplo a probabilidade de erro quando sorteamos v é $4/8 = 1/2$. Se escolhermos as coordenadas do vetor v dentre $\{0,1,2\}$ então 9 dos 27 vetores farão essa estratégia falhar, ou seja, a probabilidade de erro é $1/3$.

Instância: matrizes A, B, C quadradas de ordem n.
Resposta: *não* se $AB \neq C$, caso contrário *sim* com probabilidade de erro no máximo $1/2$.

1 $v \xleftarrow{R} \{0,1\}^n$;
2 **se** $(vA)B = vC$ **então responda** *sim*.
3 **senão responda** *não*.

Algoritmo 11: teste de produto de matrizes.

O produto $v(AB)$ é uma combinação linear das linhas de AB com coeficientes em v, assim como vC. Se $AB \neq C$, então há k linhas em que AB e C diferem, para algum $k \in \{1,\ldots,n\}$, e se as coordenadas do vetor v que são os coeficientes correspondentes a tais linhas forem 0, então teremos $v(AB) = vC$; isso ocorre com probabilidade $(1/2)^k \leqslant 1/2$, pois $k > 0$.

Proposição 1.83 *Sejam A, B, C matrizes $n \times n$ com entradas de um corpo. Se $AB \neq C$ então $\mathbb{P}_{v \in \{0,1\}^n}[(vA)B = vC] \leqslant 1/2$.* □

A técnica a seguir nos dá outra estratégia para o cálculo da probabilidade de erro desse algoritmo e é útil em outras situações (Mitzenmacher e Upfal, 2005).

Princípio da decisão adiada Muitas vezes um experimento probabilístico é modelado como uma sequência de escolhas aleatórias independentes. O princípio da decisão adiada diz que podemos optar pela ordem com que as escolhas são feitas, adiando as escolhas mais relevantes para efeito de cálculos. Aqui, ao invés de uma escolha uniforme $v \in \{0,1\}^n$ podemos considerar uma escolha de cada coordenada de v de modo uniforme e independente em $\{0,1\}$. Com a hipótese de que $AB \neq C$ a última escolha fica definida.

Outra prova da Proposição 1.83. Assumamos que cada coordenada de $\mathsf{v} \in \{0,1\}^n$ é sorteada com probabilidade 1/2 e independentemente uma das outras. Sejam A, B e C matrizes como acima e $\mathsf{D} := \mathsf{AB} - \mathsf{C}$ matriz não nula. Queremos estimar $\mathbb{P}[\,\mathsf{vD} = 0\,]$.

Se $\mathsf{D} \neq 0$ e $\mathsf{vD} = 0$, então existem ℓ e c tais que $d_{\ell,c} \neq 0$ com $\sum_{j=1}^{n} v_j d_{j,c} = 0$, assim podemos escrever

$$v_\ell = -\frac{1}{d_{\ell,c}} \sum_{\substack{j=1 \\ j\neq\ell}}^{n} v_j d_{j,c} \tag{1.26}$$

e se consideramos que cada coordenada de v foi sorteada independentemente, podemos assumir que v_i, para todo $i \neq \ell$, foi sorteado antes de v_ℓ de modo que o lado direito da igualdade (1.26) acima fica determinado e a probabilidade de sortear v_ℓ que satisfaça a igualdade é ou 0, caso o valor da direita não esteja em $\{0,1\}$, ou 1/2 caso contrário. Portanto, $\mathbb{P}[\,\mathsf{vD} = 0\,] = \mathbb{P}[\,\mathsf{vAB} = \mathsf{vC}\,] \leqslant \frac{1}{2}$. □

Desaleatorização Agora, vejamos nosso primeiro exemplo de desaleatorização. O algoritmo a seguir (devida a Kimbrel e Sinha, 1993) para o problema "AB = C?" tem custo $O(n^2)$, erra com probabilidade $\leqslant 1/2$ mas usa exponencialmente menos bits aleatórios.

O número de bits aleatórios utilizados no teste de Kimbrel–Sinha é $\lceil \log_2(2n) \rceil$, enquanto que o algoritmo de Freivalds usa n bits aleatórios.

Instância: matrizes A, B, C quadradas de ordem n.
Resposta: *não* se $\mathsf{AB} \neq \mathsf{C}$, caso contrário *sim* com probabilidade de erro no máximo 1/2.

1 $x \xleftarrow{R} \{1, 2, \ldots, 2n\}$;
2 $\mathsf{v} \leftarrow \begin{pmatrix} 1 & x & x^2 & \ldots & x^{n-1} \end{pmatrix}$;
3 **se** $(\mathsf{vA})\mathsf{B} = \mathsf{vC}$ **então responda** *sim*,
4 **senão responda** *não*.

Algoritmo 12: teste de Kimbrel–Sinha para produto de matrizes.

Vamos assumir que $\mathsf{AB} \neq \mathsf{C}$ e supor que existam n escolhas em $\{1, 2, \ldots, 2n\}$, denotadas $x_1, x_2, \ldots, x_n$, todas distintas entre si, para as quais $(\mathsf{vA})\mathsf{B} = \mathsf{vC}$. Com essas escolhas formamos a matriz de Vandermonde

$$\mathsf{V} = \mathsf{V}(x_1, x_2, \ldots, x_n) = \begin{pmatrix} 1 & x_1 & {x_1}^2 & \ldots & {x_1}^{n-1} \\ 1 & x_2 & {x_2}^2 & \ldots & {x_2}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & {x_n}^2 & \ldots & {x_n}^{n-1} \end{pmatrix}.$$

Se para cada linha

$$\mathsf{v}(x_i) = \begin{pmatrix} 1 & x_i & x_i^2 & \ldots & x_i^{n-1} \end{pmatrix}$$

dessa matriz vale que $(v(x_i) \cdot A) \cdot B = v(x_i) \cdot C$, então $VAB = VC$, o que implica $AB = C$ pois V é invertível, contrariando $AB \neq C$. Portanto, no caso em que $AB \neq C$, temos que o algoritmo responde errado em no máximo $n-1$ escolhas de $x \in \{1, 2, \ldots, 2n\}$. A probabilidade de erro é a probabilidade de escolher uma das no máximo $n-1$ escolhas ruins descritas no parágrafo acima e é menor que $n/m \leqslant 1/2$.

1.7.3 IDENTIDADE POLINOMIAL REVISITADA

Para descrevermos o caso geral do problema de testar a igualdade de polinômios começamos com algumas definições para evitar ambiguidades. As interpretações de um polinômio podem variar dependendo do contexto matemático em que ele é considerado. Na interpretação funcional, um polinômio é visto como uma função que associa a cada valor de uma variável x um valor correspondente, geralmente dentro dos números reais ou complexos. Por exemplo, o polinômio $p(x) = 2x^2 - 3x + 1$ é interpretado como uma função que toma um valor x e retorna $p(x)$, o que permite a análise de gráficos, zeros da função, e comportamento assintótico. Por outro lado, na interpretação algébrica, um polinômio é visto como um elemento de uma estrutura algébrica, como um anel ou um corpo. Nesse caso, o polinômio $p(x)$ é tratado como uma expressão formal, onde x é uma indeterminada e não assume valores numéricos, mas é manipulada simbolicamente dentro do anel dos polinômios. Esta interpretação é fundamental em álgebra abstrata, onde se estuda a fatoração, divisibilidade, e outras propriedades estruturais dos polinômios. Assim, enquanto a interpretação funcional foca no comportamento do polinômio como uma função, a interpretação algébrica trata o polinômio como um objeto simbólico em um contexto mais geral e abstrato.

Um *polinômio* nas variáveis $x_1, x_2, \ldots, x_n$ é uma expressão finita da forma, dita *canônica*,

$$\sum_{(i_1, i_2, \ldots, i_n)} c_{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}. \tag{1.27}$$

Os *coeficientes* $c_{i_1, i_2, \ldots, i_n}$ do polinômio são tomados de um corpo, por exemplo $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ e $\mathbb{Z}_p$, ou de um domínio de integridade, por exemplo $\mathbb{Z}$. Os *expoentes* $i_1, i_2, \ldots, i_n$ são inteiros não negativos. O conjunto de todos os polinômios nas variáveis $x_1, x_2, \ldots, x_n$ com coeficientes no corpo $\mathbb{F}$ é denotado por $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Uma expressão da forma $c x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ é um *monômio*.

O *grau total* do polinômio p dado pela equação (1.27), denotado por $\partial(p)$, é o maior valor de $i_1 + i_2 + \cdots + i_n$ dentre todos os índice para os quais $c_{i_1, i_2, \ldots, i_n} \neq 0$ e o *grau em* x_j, denotado $\partial_{x_j}(p)$, é o maior valor de i_j tal que $c_{i_1, i_2, \ldots, i_n} \neq 0$.

O que é "$p = q$?" Como vimos na seção 1.2.2, temos de fato dois problemas computacionais aqui:

(1) decidir se, como funções, $p(x_1, x_2, \ldots, x_n) = q(x_1, x_2, \ldots, x_n)$ e

(2) decidir se $p(x_1, x_2, \ldots, x_n)$ e $q(x_1, x_2, \ldots, x_n)$ escritos na forma canônica têm os mesmos coeficientes.

Essa distinção muitas vezes passa desapercebida porque se o polinômio é avaliado em um corpo infinito, como os números reais ou complexos (ou mesmo num domínio de integridade como $\mathbb{Z}$), então as duas noções coincidem. Nos corpos finitos pode acontecer de polinômios distintos determinarem a mesma função polinomial como, por exemplo, os polinômios 0 e $x^7 - x$ que são diferentes, mas como funções de $\mathbb{Z}_7$ em $\mathbb{Z}_7$ são iguais pois $x^7 \equiv x \pmod{n}$ pelo Pequeno Teorema de Fermat (Teorema 3.2, página 156).

Certamente, (2) implica (1). Em corpos infinitos como $\mathbb{Q}$, $\mathbb{R}$ e $\mathbb{C}$ e em corpos finitos que contenham uma quantidade suficientemente grande de elementos ($|\mathbb{F}| > \partial(p)$) vale que (1) implica (2) e isso segue do Teorema 1.84 abaixo. Portanto, sob a hipótese de que $|\mathbb{F}|$ é suficientemente grande os problemas EZE e PIT descritos acima são equivalentes.

$p = q$, para dois polinômios $p, q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, se os polinômios *são iguais quando escritos na forma canônica*. Por exemplo, a seguinte identidade entre expressões algébricas que correspondem a polinômios é verdadeira (ambas são o determinante da matriz de Vandermonde)

$$\sum_{\sigma \in \mathbb{S}_n} \operatorname{sinal}\left(\prod_{1 \leqslant i < j \leqslant n} (\sigma(j) - \sigma(i))\right) \prod_{i=1}^{n} {x_i}^{\sigma(i)-1} = \prod_{1 \leqslant i < j \leqslant n} (x_j - x_i)$$

onde $\mathbb{S}_n$ é o conjunto de todas as permutações $\sigma \colon \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$.

A solução trivial de obter o polinômio na forma canônica a partir de uma expressão algébrica está fora de questão pois é uma tarefa que pode ser muito custosa. Por exemplo, o polinômio $\prod_{1 \leqslant i < j \leqslant n}(x_i - x_j)$ escrito como combinação linear de monômios resulta numa expressão da forma

$$\sum z_1 z_2 \cdots z_{\binom{n-1}{2}} z_{\binom{n}{2}}$$

onde $z \in \{x_i, x_j\}$ para cada par $1 \leqslant i < j \leqslant \binom{n}{2}$, com a soma de 2^n parcelas. Cada parcela tem tamanho da ordem de n^2 termos, totalizado uma expressão com tamanho da ordem de $n^2 2^n$, ou seja, exponencialmente maior que a expressão original.

O que é "dado p"? Vamos descartar a possibilidade dos polinômio serem dados na forma canônica, isso torna o problema trivial. Quando nos referirmos a uma instância do problema computacional, "dado p", significa

(1) o polinômio é dado como uma *caixa-preta* para a qual fornecemos $a \in \mathbb{F}^n$ e recebemos $p(a)$, as únicas informações explícitas que temos são o número de variáveis n, um limitante superior para o grau do polinômio e uma descrição de $\mathbb{F}$;

(2) ou é dado explicitamente por uma expressão aritmética, como um determinante por exemplo, ou, mais concretamente, como um circuito aritmético. Nesse caso, deduzimos o número de variáveis n, um limitante superior para o grau e faz parte do algoritmo avaliar $p(a)$.

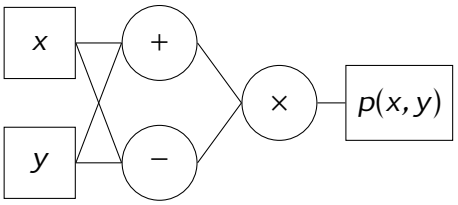


Figura 1.11: exemplo de um circuito aritmético que computa $p(x, y) = x^2 - y^2$

A definição precisa de circuito aritmético é dada no capítulo 4. Por ora, basta sabermos que é análogo aos circuitos lógicos, mas que nas portas "lógicas" são realizadas as operações do corpo (veja um exemplo na Figura 1.11). O tamanho do circuito é dado pela quantidade de portas aritméticas. O grau do polinômio dado por um circuito é no máximo 2^t, em que t é o tamanho do circuito, e esse polinômio é avaliado numa entrada x simulando-se o circuito em tempo polinomial em t.

A estratégia do algoritmo probabilístico para esse problema é idêntica ao caso de uma variável, sorteamos n números e avaliamos o polinômio nessa n-upla. Notemos que não há uma generalização imediata do Teorema Fundamental da Álgebra pois, por exemplo, um polinômio sobre um corpo como $\mathbb{Q}$ e com várias variáveis pode ter um número infinito de raízes, como é o caso de $x_1 x_2$. A estratégia é baseada no seguinte teorema.

Teorema 1.84 (teorema de DeMillo–Lipton–Schwartz–Zippel) *Sejam $\mathbb{F}$ um corpo (ou domínio de integridade), $p \in \mathbb{F}[x_1, \ldots, x_n]$ um polinômio não nulo com grau total $d \geqslant 0$ e $S \subseteq \mathbb{F}$ finito e não vazio. Então*

$$\mathop{\mathbb{P}}_{r_1,\ldots,r_n \in_R S}[p(r_1, \ldots, r_n) = 0] \leqslant \frac{d}{|S|}.$$

Esse resultado foi descoberto várias vezes e de modo independente (DeMillo e Lipton, 1978; Schwartz, 1979; Zippel, 1979, dentre outros).

Em uma variável ($n = 1$) o Teorema 1.84 segue do Teorema Fundamental da Álgebra, pois p tem no máximo $\partial(p) = d$ raízes. Se $n = 2$ e $k := \partial_{x_2}(p)$, então podemos reescrever p como

$$p(x_1, x_2) = \mathsf{A}(x_1)x_2^k + \mathsf{B}(x_1, x_2)$$

com polinômios $\mathsf{A} \in \mathbb{F}[x_1]$ de grau total $\partial(\mathsf{A}) \leqslant d-k$ e $\mathsf{B} \in \mathbb{F}[x_1, x_2]$ de grau $\partial_{x_2}(\mathsf{B}) < k$. Assim, se $r \in \mathsf{S}$ é uma raiz de A, então $p(r, x_2) = \mathsf{B}(r, x_2)$ tem menos que k raízes. Portanto, a quantidade de pares $(r_1, r_2) \in \mathsf{S}^2$ com r_1 raiz de A é no máximo $\partial(\mathsf{A}) \cdot |\mathsf{S}| \leqslant (d-k)|\mathsf{S}|$. Agora, se $r \in \mathsf{S}$ não é uma raiz de A, então $p(r, x_2) = \mathsf{A}(r)x_2^k + \mathsf{B}(r, x_2)$ é um polinômio em 1 variável e, portanto, tem no máximo k raízes. Se denotamos por R_p o conjunto das raízes de p em S^2, então temos

$$|\mathsf{R}_p| \leqslant \left|\{(r_1, r_2) \in \mathsf{S}^2 \colon \mathsf{A}(r_1) = 0\} \cup \{(r_1, r_2) \in \mathsf{S}^2 \colon \mathsf{A}(r_1) \neq 0\}\right| \leqslant (d-k)|\mathsf{S}| + k|\mathsf{S}| = \partial(p)|\mathsf{S}|.$$

Essa é uma estimativa justa, o polinômio $(x_1 - x_2)^2 - 1$ de grau total 2 no corpo $\mathbb{F}_3$ tem raízes $(0,1)$, $(1,0)$, $(2,1)$, $(1,2)$, $(0,2)$ e $(2,0)$, ou seja, $6 = 2 \cdot 3$ raízes.

O argumento que acabamos de descrever é indutivo, provamos o caso de duas variáveis usando o caso de uma variável. Podemos, sem muita dificuldade, generalizar o passo acima para provar Teorema 1.84, deixamos a demonstração para o final dessa seção.

Com o Teorema 1.84 nós podemos resolver probabilisticamente o problema de identidade polinomial. Suponha que nos seja dado um polinômio $p(x_1, \ldots, x_n)$ de grau total $d < |\mathbb{F}|$. O algoritmo sorteia $r_1, \ldots, r_n$ em $\mathsf{S} \subseteq \mathbb{F}$ (ou em $\mathbb{F}$) finito e suficientemente grande, computa $p(r_1, \ldots, r_n)$ e decide, de modo que a probabilidade de erro é no máximo $d/|\mathsf{S}|$.

O algoritmo para polinômios de várias variáveis é uma adaptação simples do Algoritmo 1 e é como segue.

Instância: $d > 0$ e $p(x_1, \ldots, x_n)$ de grau total no máximo d.
Resposta: *não* se $p \neq 0$, caso contrário *sim* com probabilidade de erro no máximo $1/2$.

1 **para cada** $i \in \{0, \ldots, n-1\}$ **faça** $x_i \xleftarrow{\mathrm{R}} \{1, 2, \ldots, 2d\}$;
2 **se** $p(x_1, x_2, \ldots, x_n) \neq 0$ **então responda** *não*.
3 **senão responda** *sim*.

Algoritmo 13: identidade entre polinômios.

A probabilidade de erro do Algoritmo 13 segue do teorema de Schwartz–Zippel. Assumindo p não nulo, um algoritmo que escolhe aleatoriamente $x_1, \ldots, x_n$ em $\mathsf{S} = \{1, 2, \ldots, 2d\}$ e decide baseado no teste $p(x_1, \ldots, x_n) = 0$ erra com probabilidade no

máximo $d/|S| = 1/2$. Repetindo k vezes o algoritmo, se $p \neq 0$ então o algoritmo responde *sim* somente se nas k iterações (independentes) foi sorteada uma raiz de p cuja probabilidade é

$$\mathbb{P}[\,\text{erro}\,] \leqslant \left(\frac{1}{2}\right)^k.$$

O problema da igualdade de polinômios, tanto na versão *caixa-preta*, aquele em que o polinômio é dado implicitamente, quanto no modelo não-caixa-preta é bastante estudado por causa do impacto desse problema tando em temas teóricos quanto em temas práticos da computação, é um problema central em Complexidade Computacional e para Projeto de Algoritmos.

Problema computacional do teste de identidade polinomial (PIT):
Instância: um polinômio p de n variáveis sobre $\mathbb{F}$.
Resposta: *sim* se p é identicamente nulo, *não* caso contrário.

A solução probabilística proposta para o PIT é de tempo polinomial desde que o polinômio tenha grau limitado e que seja possível avaliar o polinômio nos valores do conjunto S com cardinalidade igual ao dobro do grau. Então, se é dado p como um oráculo, o polinômio pode ser avaliado de forma eficiente por definição, se é p como uma fórmula aritmética ou circuito, então pode ser avaliado eficientemente percorrendo o circuito. Não se conhece algoritmo determinístico de tempo subexponencial para este problema.

> *Problema* 1 (PIT). Existe um algoritmo determinístico para PIT que executa um número de operações em $\mathbb{F}$ que é polinomial no tamanho do circuito aritmético?

O desafio é projetar algoritmos determinísticos para PIT, ou, menos ambiciosamente, reduzir a quantidade de aleatoriedade necessária para resolver o problema.

Prova do Teorema 1.84 A prova do teorema é uma consequência do seguinte resultado.

Lema 1.85 (Lema de Schwartz) *Sejam $\mathbb{F}$ um corpo, $S \subseteq \mathbb{F}$ finito e $p \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ um polinômio não nulo. Então, a quantidade de n-uplas $(r_1, \ldots, r_n) \in S^n$ tais que $p(r_1, \ldots, r_n) = 0$ é no máximo $\partial(p) \cdot |S|^{n-1}$.*

Demonstração. Por indução em $n \geqslant 1$. Pela dedução acima a base, $n = 1$, vale. Vamos provar o passo da indução. Assumimos que para $n \geqslant 2$ fixo, todo polinômio q nas variáveis $x_1, \ldots, x_{n-1}$ tem no máximo $\partial(q) \cdot |S|^{n-2}$ raízes em S^{n-1}. Vamos mostrar que $p \in \mathbb{F}[x_1, \ldots, x_n]$ tem no máximo $\partial(p) \cdot |S|^{n-1}$ raízes em S^n.

Podemos supor sem perda de generalidade que $\partial_{x_n}(p) = k > 0$ e com isso podemos escrever

$$p(x_1, \ldots, x_{n-1}, x_n) = \sum_{j=0}^{k} g_j(x_1, \ldots, x_{n-1}) \cdot x_n^j.$$

Definimos os conjuntos

$$R_p := \{(x_1, \ldots, x_{n-1}, x_n) \in S^n : p(x_1, \ldots, x_{n-1}, x_n) = 0\}$$

e

$$R_{g_k} := \{(a_1, \ldots, a_{n-1}, x_n) \in S^n : g_k(a_1, \ldots, a_{n-1}) = 0\}.$$

Pela hipótese indutiva g_k tem no máximo $\partial(g_k) \cdot |S|^{n-2}$ raízes em S^{n-1}, portanto, $|R_{g_k}| \leqslant \partial(g_k) \cdot |S|^{n-2}|S| = \partial(g_k) \cdot |S|^{n-1}$.

Em $R_p \setminus R_{g_k}$ temos os pontos $(a_1, \ldots, a_{n-1}, x_n) \in R_p$ tais que $g_k(a_1, \ldots, a_{n-1}) \neq 0$, portanto $p(a_1, \ldots, a_{n-1}, x_n)$ é um polinômio em x_n de grau k, logo tem $\leqslant k$ raízes. Daí, $|R_p \setminus R_{g_k}| \leqslant k|S|^{n-1}$. Portanto,

$$|R_p| = |R_{g_k}| + |R_p \setminus R_{g_k}| \leqslant (\partial(g_k) + k)|S|^{n-1} \leqslant \partial(p) \cdot |S|^{n-1}$$

é a estimativa procurada para o número de raízes de p em S^n. □

Exercício 1.86. Resolva o problema da verificação do produto de matrizes, apresentado na seção 1.7.2, usando teste de identidade de polinômios. Em particular, use o Teorema 1.84 para provar que se sorteamos $v \in S$ uniformemente, para um S adequado, então $\mathbb{P}[vAB = vC] \leqslant 1/|S|$.

Exercício 1.87. Considere a seguinte adaptação no protocolo da seção 1.6: Alice escolhe (sem sorteio) um primo $p \in [n^2, 2n^2]$ (tal primo sempre existe pelo Postulado de Bertrand[9]. Alice sorteia $\alpha \in \{0, 1, 2, \ldots, p-1\}$, calcula $A(\alpha)$ mod p e manda p, α e $A(\alpha)$ mod p para Bob. Bob calcula $B(\alpha)$ mod p e manda para Alice se $A(\alpha) = B(\alpha)$ ou não. Qual o custo de comunicação do protocolo? Demonstre, usando teste de identidade de polinômios, que tal protocolo erra com probabilidade $< 1/n$. Qual a estimativa para a probabilidade de erro quando $p \in [n^c, 2n^c]$ para uma constante $c > 2$.

Observação 1.88. No caso em que $\mathbb{F}$ é finito e o grau do polinômio é maior que $|\mathbb{F}|$ é possível resolver o problema numa extensão do corpo $\mathbb{F}$. No caso de domínios infinitos, como o $\mathbb{Z}$, podem ocorrem números muito grandes de modo a influenciar o custo para a avaliação do polinômio, o que pode ser resolvido, na versão com circuito aritmético, usando aritmética modular (através do resultado do Exercício 1.67).

[9] Existe pelo menos um número primo p com $n < p < 2n$ para todo $n > 3$.

1.8 O JANTAR DOS FILÓSOFOS, UM CASO NÃO-ENUMERÁVEL

Nessa seção vamos considerar um problema de computação distribuída cuja solução probabilística tem espaço amostral não enumerável. O jantar dos filósofos, originalmente proposto por Dijkstra em 1965, ilustra o problema de alocação de recursos em sistemas distribuídos.

> Cinco filósofos estão reunidos para um jantar em torno de uma mesa circular. A vida de um filósofo consiste basicamente em pensar. Enquanto está pensando, um filósofo não interage com os outros filósofos, entretanto, o filósofo acaba por sentir fome em algum momento. Para se alimentar, ele dispõe de um prato de macarrão que nunca se esvazia, um garfo a sua esquerda e um garfo a sua direita, mas o macarrão encontra-se de tal forma oleoso que é impossível comê-lo com apenas um garfo, sendo necessários dois. Um filósofo pode pegar apenas um garfo de cada vez e é impossível utilizar um garfo que esteja sendo utilizado por um vizinho. Uma vez que um filósofo tenha dois garfos ele se alimenta, devolve os dois garfos e volta a pensar. Um filósofo faminto e que não seja capaz de pegar seus dois garfos todas as vezes que tente pegá-lo entra em inanição.

Em suma, um filósofo opera indefinidamente no ciclo: pensar, tentar comer, comer. Para comer um filósofo necessita de acesso exclusivo a dois recursos, cada um deles é compartilhado com um vizinho.

O problema computacional consiste em projetar um protocolo que represente os filósofos e os garfos de maneira apropriada, para que se comportem como as entidades descritas no enunciado. O protocolo deve garantir que os filósofos comam. Esse problema não tem solução determinística, entretanto apresentaremos uma solução probabilística devida a Lehmann e Rabin (1981).

O tratamento que daremos a ambos os tópicos, computação distribuída e espaço não enumerável, será um tanto informal.

Espaço produto O modelo probabilístico para esse problema envolve um espaço amostral equivalente ao de lançar uma moeda infinitas vezes. Se temos dois espaços de probabilidade $(\Omega_1, \mathcal{A}_1, \mathbb{P}_1)$ e $(\Omega_2, \mathcal{A}_2, \mathbb{P}_2)$ o espaço produto tem espaço amostral $\Omega_1 \times \Omega_2$ mas o espaço de eventos não é simplesmente $\mathcal{A}_1 \times \mathcal{A}_2$, mas sim a menor[10] σ-álgebra que contém todos os produtos de eventos $A_1 \times A_2$, que denotamos por $\mathcal{A}_1 \otimes$

[10] É a intersecção de todas as σ-álgebras de $\Omega_1 \times \Omega_2$. Note-se que a intersecção de σ-álgebras de $\Omega_1 \times \Omega_2$ é uma σ-álgebra de $\Omega_1 \times \Omega_2$.

$\mathcal{A}_2$. No produto, a medida de probabilidade é tal que $\mathbb{P}(A_1 \times A_2) = \mathbb{P}_1(A_1)\mathbb{P}_2(A_2)$ para todos $A_1 \in \mathcal{A}_1$ e $A_2 \in \mathcal{A}_2$. O espaço de probabilidade $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \otimes \mathcal{A}_2, \mathbb{P})$ é o *espaço produto*.

Essa definição pode ser estendida para o produto de vários espaços, até uma quantia infinita enumerável deles, com $\Omega = \prod_n \Omega_n$ e $\mathcal{A} = \otimes_n \mathcal{A}_n$ é a menor σ-álgebra que contém os eventos

$$A_1 \times A_2 \times \cdots \times A_k \times \Omega_{k+1} \times \Omega_{k+2} \times \cdots$$

para todo inteiro positivo k, para todo $A_i \in \mathcal{A}_i$, para todo inteiro positivo i. É possível demonstrar que há uma única medida de probabilidade $\mathbb{P}$ que satisfaz

$$\mathbb{P}(A_1 \times A_2 \cdots A_k \times \Omega_{k+1} \times \Omega_{k+2} \times \cdots) = \mathbb{P}_1(A_1)\mathbb{P}_2(A_2)\cdots\mathbb{P}_k(A_k).$$

Exemplo 1.89. Consideremos o espaço amostral formado por todas as sequências binárias

$$\{0,1\}^{\mathbb{N}} := \{(b_0, b_1, b_2, \ldots) \colon b_i \in \{0,1\}\ (\forall i)\}.$$

Denotemos por $\mathcal{C}_k$ a família de todos os eventos de $\{0,1\}^{\mathbb{N}}$ cuja ocorrência é decidida pelos k primeiros bits das sequências. Por exemplo, as sequências tais que $b_1 \neq b_2$ e $b_3 = 0$ é um elemento de $\mathcal{C}_3$; o evento "dois zeros nos sete primeiros lançamentos" é um elemento de $\mathcal{C}_7$. Dado subconjunto $B \subseteq \{0,1\}^k$ definimos $B_\Omega \subseteq \{0,1\}^{\mathbb{N}}$ por

$$B_\Omega := \{(b_0, b_1, b_2, \ldots) \colon (b_1, b_2, \ldots, b_k) \in B\}.$$

O conjunto B_Ω pode ser identificado com $B \times \{0,1\}^{\mathbb{N}}$, temos $B_\Omega \in \mathcal{C}_k$ e todo elemento de $\mathcal{C}_k$ pode ser escrito dessa forma para algum $B \subseteq \{0,1\}^k$. A família $\mathcal{C}_k$ é uma σ-álgebra de subconjuntos de $\{0,1\}^{\mathbb{N}}$, para cada inteiro positivo k, e no jargão de Probabilidade, esses são chamados de *eventos cilíndricos*. Notemos que $\mathcal{C}_k \subseteq \mathcal{C}_{k+1}$ e que o evento "não ocorre 1" não pode ser expresso por nenhuma dessas famílias.

Agora, fazemos

$$\mathcal{C} := \bigcup_{k \geqslant 1} \mathcal{C}_k$$

a família dos eventos cuja ocorrência é decidida por um número fixo de bits iniciais. A família $\mathcal{C}$ não é uma σ-álgebra de subconjuntos de $\{0,1\}^{\mathbb{N}}$ pois se tomamos B_k o conjunto das sequências com $b_k = 1$ então temos $B_k \in \mathcal{C}_k$ mas $\overline{\bigcup_k B_k} \notin \mathcal{C}$. Entretanto, $\mathcal{C}$ é uma *álgebra* de subconjuntos de $\{0,1\}^{\mathbb{N}}$, isto é, satisfaz:

(*i*) $\varnothing$ é um elemento da família,

(*ii*) o complemento de qualquer elemento da família também pertence a família e

(*iii*) a união de quaisquer dois elementos da família pertence a família.

Um teorema famoso, conhecido como *Teorema de Extensão de Carathéodory* nos diz que toda função $\mathsf{P}\colon \mathcal{C} \to [0,1]$ que satisfaz

(i) $\mathsf{P}(\{0,1\}^{\mathbb{N}}) = 1$ e

(ii) $\mathsf{P}(\bigcup_n \mathsf{B}_n) = \sum_n \mathsf{P}(\mathsf{B}_n)$, para $\{\mathsf{B}_n\}_{n\in\mathbb{N}}$ elementos disjuntos da álgebra,

pode ser *estendida de maneira única* para uma medida de probabilidade sobre a menor σ-álgebra que contém a álgebra $\mathcal{C}$.

Definimos uma função P de acordo com as hipóteses do parágrafo anterior do seguinte modo. Todo $\mathsf{A} \in \mathcal{C}$ é da forma $\mathsf{B} \times \{0,1\}^{\mathbb{N}}$ para algum $\mathsf{B} \subseteq \{0,1\}^k$, para algum natural k. Nesse caso,

$$\mathsf{P}(\mathsf{A}) \coloneqq \frac{|\mathsf{B}|}{2^k}. \tag{1.28}$$

Essa definição é consistente (veja Exercício 1.105 no final desse capítulo) e ainda vale $\mathsf{P}(\{0,1\}^{\mathbb{N}}) = 1$ (por quê?). A função P é enumeravelmente aditiva (veja o Exercício 1.106 no final do capítulo), portanto, pode ser estendida para uma medida de probabilidade $\mathbb{P}$ sobre a menor σ-álgebra que contém $\mathcal{C}$.

Nesse espaço de probabilidade os pontos amostrais têm probabilidade zero. Dado $(b_0, b_1, b_2, \dots) \in \{0,1\}^{\mathbb{N}}$, definimos o evento

$$\mathsf{E}_k \coloneqq \{(\omega_0, \omega_1, \dots) \in \{0,1\}^{\mathbb{N}} \colon \omega_j = b_j \text{ para todo } j \leqslant k\}$$

para todo natural k, logo $(b_0, b_1, b_2, \dots) = \bigcap_{k\in\mathbb{N}} \mathsf{E}_k$ e a probabilidade do ponto amostral é o limite de $\mathbb{P}(\mathsf{E}_k)$ quando $k \to \infty$ pela continuidade de $\mathbb{P}$. Usando a equação (1.28) essa probabilidade é $\lim_{k\to\infty}(1/2)^k = 0$. Como consequência da aditividade, eventos enumeráveis têm probabilidade 0.

Esse espaço de probabilidade que acabamos de definir é equivalente a distribuição uniforme no intervalo $[0,1]$, descrito no Exemplo 1.11, página 15. Para os detalhes dessa construção e da equivalência convidamos o leitor a consultar o capítulo 1 de Billingsley (1979). ◊

Definições preliminares dos processos "filósofos" No modelo que usamos, computação distribuída é a computação é realizada pela execução de um conjunto de *processos* concorrentes, cada processo executa um algoritmo. Cada filósofo corresponde a um processo e seu algoritmo define as ações dos filósofos.

Uma *ação atômica* é qualquer conjunto de instruções de um algoritmo distribuído executadas de modo indissociável, nenhum outro processo executa instrução enquanto uma ação atômica não termina.

Variáveis representam os garfos e são compartilhadas, sendo cada garfo modelado por um espaço de memória acessível apenas aos processos que representam os

filósofos que o compartilham. Pegar e devolver um garfo são mudanças no valor de uma variável e o acesso às variáveis é uma ação atômica, um filósofo verifica se um garfo está disponível e, caso disponível, o pega sem que seja incomodado por algum de seus vizinhos nesse ínterim. Ademais,

é garantido que sempre que um processo requisita o conteúdo de uma variável compartilhada, ele acabará por recebê-lo em algum momento futuro.

Um *escalonamento* é uma função que define, a partir do comportamento passado de todos os processos, o próximo processo a efetuar uma ação atômica. Conhecer o passado dos processos inclui conhecer os resultados de sorteios aleatórios passados, as memórias compartilhadas e privadas dos processos. Não há nenhum tipo de hipótese em relação às taxas de atividade de cada processo. Não está excluída a possibilidade de que o escalonamento seja malicioso e trabalhe contra a solução, fazendo o máximo possível para impedir que os filósofos se alimentem. Um escalonamento é *justo* se

todos os processos são ativados um número infinito de vezes

quaisquer que sejam os resultados de sorteios aleatórios. Daqui em diante só consideramos escalonamentos justos.

Uma *solução* para o problema do jantar dos filósofos deve ser

- *distribuída*: não há um processo controlador ou uma memória central com a qual todos os outros processos possam se comunicar;
- *simétrica*: todos os processos devem executar o mesmo algoritmo e todas as variáveis têm a mesma inicialização. Além disso, os processos ignoram suas identidades.

O objetivo é encontrar um protocolo de ação que, respeitando as restrições acima, garanta que os filósofos se alimentem.

Não há solução determinística Suponha que exista uma solução distribuída e simétrica e vamos definir um escalonamento que impeça os filósofos de se alimentarem.

Sem perda de generalidade, podemos enumerar os processos de 1 a n. O escalonamento ativa cada um dos processos por uma ação atômica, ordenadamente, e repete essa ordem de ativação indefinidamente. Dado que os processos se encontram inicialmente no mesmo estado, a simetria é preservada a cada rodada de n ativações e dado que é impossível que todos os filósofos estejam se alimentando simultaneamente, esse escalonamento impede que todos os filósofos se alimentem.

Uma computação em *deadlock* é uma computação em que existe um instante t no qual um filósofo está tentando comer, mas a partir do qual nenhum filósofo come.

Teorema. *Não existe uma solução distribuída e simétrica para o problema do Jantar dos Filósofos que seja livre de* deadlock.

Solução probabilística O que impede uma solução determinística para o problema do jantar dos filósofos é a simetria entre os processos. Para quebrar a simetria, vamos equipar os filósofos com moedas, permitindo que escolham aleatoriamente qual dos dois garfos tentarão pegar. A cada instante t o processo ativo tem a sua disposição um bit aleatório b_t com probabilidade 1/2 de ser qualquer um dos dois valores e de modo que em instantes distintos os valores dos bits são independentes. O espaço amostral $\{0,1\}^{\mathbb{N}}$ é formado de todas as sequências binárias $\omega = (b_0, b_1, b_2, \dots)$. Esse espaço não é enumerável e usaremos o tratamento descrito acima.

Consideraremos o caso de $n \geqslant 3$ filósofos, denotados por P_i, para $1 \leqslant i \leqslant n$, mas sem que eles reconheçam qualquer identidade e dispostos na ordem (cíclica) $\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_3, \dots, \mathsf{P}_n$ no sentido anti-horário. Os filósofos se comportam da maneira descrita pelo Algoritmo 15, no qual representamos por 0 o garfo da esquerda, por 1 o garfo da direita e as linhas são instruções atômicas.

```
1 enquanto verdadeiro faça
2     pense;
3     ℓ ←R {0,1};
4     se garfo ℓ disponível então pegue o garfo ℓ, senão vá para linha 4;
5     se garfo 1 − ℓ disponível então pegue o garfo 1 − ℓ e vá para linha 7;
6     devolva o garfo ℓ e vá para linha 3;
7     coma;
8     devolva um garfo;
9     devolva o outro garfo.
```

Algoritmo 15: algoritmo dos Filósofos

Um escalonamento S e uma sequência infinita de bits $\omega = (b_i \in \{0,1\} \colon i \in \mathbb{N})$ definem uma, e só uma, computação, que é uma sequência infinita de ações atômicas do Algoritmo 15

$$\textsc{comp}(\mathsf{S}, \omega) := ((\alpha, \mathsf{P}, b)_t \colon t \in \mathbb{N})$$

em que α é a ação atômica efetuada pelo filósofo P no instante t para a qual há a disposição um bit aleatório $b = b_t \in \{0,1\}$ que pode ser usado ou não.

Fixamos um escalonamento (justo) S. A sequência aleatória ω induz uma distribuição de probabilidade no espaço de todas as computações. O objetivo é demonstrar que no sistema dos filósofos com algoritmos aleatorizados a probabilidade de ocorrência *deadlock* é zero. Em particular, fixado S temos que $\omega \in \{0,1\}^{\mathbb{N}}$ define se a computação está ou não em *deadlock* de modo que $\{\omega \in \{0,1\}^{\mathbb{N}} : \textsc{comp}(S,\omega)$ em *deadlock*$\}$ é um evento aleatório pois depende de uma quantidade finita de bits iniciais de ω.

Se um filósofo P pega garfo apenas um número finito de vezes, então a partir de algum instante t não pega mais os garfos e, além disso, seu vizinho a esquerda P_e sempre tem o garfo que compartilham disponível. Vamos supor que P_e pega garfo infinitas vezes e provar que com probabilidade 1 ele se alimenta infinitas vezes. Basta provarmos que P_e sorteia a garfo esquerdo infinitas vezes com probabilidade 1. De fato, se P_e sorteia o garfo esquerdo, ele o pega assim que estiver disponível e come, pois o da direita sempre estará disponível e uma espera indefinida pelo garfo esquerdo não ocorre pois nesse caso ele pegaria os garfos apenas um número finito de vezes. Sortear o garfo esquerdo um número finito de vezes depende de um número finito de coordenadas iniciais de ω, de modo que o evento E das computações $\textsc{comp}(S,\omega)$ em que P_e sorteia o garfo esquerdo um número finito de vezes é enumerável, portanto de probabilidade 0.

Então, em um computação em *deadlock* não pode ocorrer (na verdade, ocorre com probabilidade 0) o evento "um filósofo pega garfo finitas vezes e seu vizinho pega infinitas vezes" porque o vizinho certamente comerá.

Ademais, em uma computação em *deadlock* não é possível que todos os filósofos peguem algum garfo um número finito de vezes (por quê?).

A partir dos dois parágrafos acima concluímos o seguinte.

Proposição 1.90 *Numa computação em* deadlock *todos os filósofos pegam algum dos seus garfos um número infinito de vezes com probabilidade* 1. □

Lembremos que em cada instante da computação um bit aleatório pode ou não ser usado pelo processo da vez. Para um instante t fixo temos um sequência formada pelos bits aleatórios que foram de fato usados por algum dos processos (no sorteio de um garfo). Chamemos essa sequência de *configuração de sorteios aleatórios* e chamemos duas configurações A e uma posterior B de *disjuntas* caso entre A e B todos os filósofos utilizaram pelo menos um sorteio.

Lema 1.91 *Numa computação em* deadlock, *se para um dado instante* t *a configuração de sorteios aleatórios já efetuados é* A, *então com probabilidade* 1 *haverá num momento futuro uma configuração* B *disjunta de* A *em que o último sorteio de algum filósofo foi o garfo esquerdo e o último sorteio de seu vizinho a direita foi o garfo direito.*

Demonstração. Numa computação em *deadlock*, todos os filósofos pegam algum dos seus garfos um número infinito de vezes com probabilidade 1 pela Proposição 1.90.

Se A e B são duas configurações disjuntas e subsequentes então, no instante que ocorre B, a probabilidade com que o último sorteio de cada filósofo sejam iguais é $2(1/2)^n = 1/2^{n-1}$. Agora, se consideramos um intervalo de k configurações disjuntas subsequentes a partir de uma dada configuração, digamos $A_i, A_{i+1}, \ldots, A_{i+k}$, a probabilidade de que todos os filósofos tenham sorteado o mesmo valor em todos os respectivos últimos sorteios que antecedem imediatamente alguma configuração A_j, com $i < j \leqslant i + k$, é de $(1/2^{n-1})^k$. A probabilidade desse evento ao longo da computação é $\lim_{k\to\infty}(1/2^{n-1})^k = 0$, assim a partir de qualquer configuração A surgirá, com probabilidade 1, uma configuração disjunta B tal que, considerando o último sorteio de todos os filósofos, haverá algum filósofo que sorteou 0 (garfo da esquerda) e seu vizinho a direita sorteou 1 (garfo da direita). □

Lema 1.92 *Seja* F *um segmento inicial finito de uma computação composto por* t *instantes e tal que no instante* t *temos:* (*i*) *tanto* P_1 *quanto* P_2 *(seu vizinho a direita) estão tentando comer,* (*ii*) *o último sorteio de* P_1 *foi o garfo esquerdo e o último sorteio de* P_2 *foi o garfo direito. Seja* $C = \textsc{comp}(S, \omega)$ *uma continuação de* F. *Nessas condições, em* C *pelo menos um dentre* P_1 *e* P_2 *se alimenta antes da próxima configuração disjunta da atual com probabilidade* 1.

Demonstração. No instante t os filósofos P_1 e P_2 estão tentando comer, P_1 sorteou 0 e P_2 sorteou 1 (estão na linha 4 do Algoritmo 15), então antes do próximo sorteio cada um deles pode se encontrar em um dos seguintes estados:

1. o filósofo está esperando que o garfo sorteado seja disponibilizado, ou
2. o filósofo está em posse do garfo sorteado.

Se algum dos filósofos, dentre P_1 e P_2, está no estado 2 então um deles irá comer antes do próximo sorteio. De fato, no caso em que tanto P_1 quanto P_2 se encontram no estado 2 o próximo filósofo a ser ativado irá se alimentar antes do seu próximo sorteio pois encontrará o garfo compartilhado pelos dois disponível. No caso em que P_1 se encontra no estado 2 e P_2 no estado 1, se P_1 for o próximo dentre os dois a ser ativado, ele encontrará o garfo compartilhado disponível e comerá antes de ter feito algum sorteio; se P_2 for ativado, ele pode tanto permanecer no estado 1 e voltamos para a condição inicial, quanto progredir para o estado 2 e recaímos no caso anterior. Finalmente, o caso em que P_1 se encontra no estado 1 e P_2 no estado 2 é análogo ao anterior.

Por outro lado, no caso em que ambos os filósofos se encontram no estado 1, antes de algum sorteio de algum deles, um deverá avançar para o estado 2 e recaímos nos casos acima. □

Com as propriedades dadas nos lemas acima provaremos o resultado final desse capítulo. Seja S um escalonamento justo. Denotemos por D o evento "a computação $\text{comp}(S,\omega)$ está em *deadlock*" e suponha que $\mathbb{P}(D) > 0$. Podemos então nos referir às probabilidades dos eventos condicionados ao *deadlock*. Pelo Lema 1.91, com probabilidade 1 ocorre uma sequência infinita, digamos $A_1, A_2, \ldots, A_n, \ldots$, de configurações disjuntas de sorteios aleatórios satisfazendo as hipóteses do Lema 1.92, donde tiramos que algum filósofo come entre A_n e A_{n+1}, para todo n, com probabilidade 1. Chegamos então à conclusão de que, condicionado ao evento "computação em *deadlock*", computações livres de *deadlock* têm probabilidade 1. Desta maneira, a ocorrência de deadlock deve ter probabilidade zero.

Teorema 1.93 *Para todo escalonamento* S *justo,* $\text{comp}(S,\omega)$ *está em* deadlock *com probabilidade* 0. □

1.9 EXERCÍCIOS

Exercício 1.94. Um algoritmo para testar se n é da forma a^b é com segue: seja k tal que $2^{k-1} \leqslant n < 2^k$, então uma b-ésima raiz de n pertence ao intervalo $2^{\lfloor (k-1)/b \rfloor} \leqslant n^{1/b} < 2^{\lceil k/b \rceil}$, faça uma busca binária nesse intervalo. Descreva o algoritmo, verifique que usando a estratégia do Algoritmo 7 a potência x^y pode ser calculada em tempo $O((y \log(x))^2)$ e conclua que testar se n é da forma a^b com a estratégia acima tem tempo de execução $O((\log n)^3)$.

Exercício 1.95. Dados inteiros $m_1, m_2, \ldots, m_k > 1$, sejam $m = m_1 m_2 \cdots m_k$ e $m_i' = m/m_i$. Para $a \in \mathbb{Z}_n$, mostre como computar $a^{m_1'}, a^{m_2'}, \ldots a^{m_k'}$ com $O(\log(k)\log(m))$ multiplicações.

Exercício 1.96. Dez dados equilibrados são lançados. Supondo que os resultados são independentes, use o princípio da decisão adiada para determinar a probabilidade da soma dos resultados ser divisível por seis.

Exercício 1.97. Um baralho comum de 52 cartas é embaralhado de modo que a disposição final é qualquer uma dentre as 52! possibilidades com igual probabilidade. Denote por E o evento "a carta do topo é de espadas", que ocorre com probabilidade 1/4. Use o princípio da decisão adiada para provar que $\mathbb{P}(F) = \mathbb{P}(E)$ para F o evento "a quarta carta a partir do topo é espada".

Exercício 1.98. Um baralho comum de 52 cartas é embaralhado de modo que a disposição final é qualquer uma as 52! possibilidades com igual probabilidade e em seguida é dividido em 13 montes de 4 cartas cada. Todo monte tem um único rótulo tomado em $\{A, 2, 3, \ldots, 9, 10, J, Q, K\}$ arbitrariamente. No primeiro movimento abrimos uma carta do monte K e o resultado indica o próximo monte donde abriremos uma carta e assim por diante seguimos. O jogo acaba quando uma jogada indica abrir a carta de um monte vazio. Use o princípio da decisão adiada para provar que a probabilidade de abrirmos todas as cartas do baralho é 1/13.

Exercício 1.99. Prove que se o laço da linha 1 no algoritmo para corte mínimo, Algoritmo 9 na página 76, for executado $n^2\lceil \log(n) \rceil$ vezes, então a probabilidade do algoritmo não encontrar um corte de tamanho $\leqslant k$ é menor que $1/n$.

Exercício 1.100. Na solução probabilística para o PIT, Algoritmo 13, qual é a probabilidade de uma resposta errada em k repetições (independentes) do algoritmo se as escolhas aleatórias são garantidas ser sem repetição.

Exercício 1.101 (emparelhamento perfeito em grafos bipartidos). Seja $G = (A \cup B, E)$ um grafo com $|A| = |B| = n$ e todas as arestas em E tem um vértice em A e o outro em B, isto é G é um **grafo bipartido**. Defina a *matriz de adjacências* $A = (a_{i,j})$ pondo

$$a_{i,j} := \begin{cases} x_{i,j} & \text{se } \{a_i, b_j\} \in E \\ 0 & \text{caso contrário.} \end{cases}$$

Um **emparelhamento** M em G é um subconjunto de E formado por arestas não adjacentes, isto é, $e \cap d = \varnothing$ para quaisquer arestas $e, d \in M$ distintas. O emparelhamento M é dito **perfeito** se $|M| = n$.

Prove que G tem um emparelhamento perfeito se, e somente se, $\det(A) \neq 0$ (como polinômios). Escreva um algoritmo baseado no teorema de Schwartz–Zippel que determina um emparelhamento perfeito caso exista. Analise a probabilidade de erro e o tempo de execução do algoritmo.

Exercício 1.102. Seja G um grafo bipartido e $\mathcal{F} := \{M_1, M_2, \ldots, M_k\}$ o conjunto dos emparelhamentos perfeitos de G. Tome $p\colon E(G) \to \{1, \ldots, 2|E|\}$ uma atribuição de pesos escolhidos uniformemente e independentemente para as arestas de G e defina o peso de um emparelhamento como a soma dos pesos de suas arestas. Na matriz A do Exercício 1.101 tome $x_{i,j} = 2^{p(a_i, b_j)}$. Suponha, sem perda de generalidade, que M_1 é o único emparelhamento de peso mínimo. Prove que a maior potência de 2 que divide $\det(A)$ é o peso de M_1. Prove que para cada aresta $\{a_i, b_j\}$, o determinante da matriz resultante da eliminação da linha i e da coluna j de A vezes $2^{p(a_i, b_j) - p(M_1)}$

é ímpar se, e somente se, $\{a_i, b_j\} \in M_1$. Baseado no Lema do Isolamento (Exercício 1.58), escreva um algoritmo probabilístico que ou devolve um emparelhamento perfeito ou falha. Analise seu algoritmo.

Exercício 1.103. Escreva um algoritmo aleatorizado que recebe um inteiro $M \geqslant 2$ e devolve uma escolha aleatória uniforme $N \in \{1, \ldots, M-1\}$ tal que $\mathrm{mdc}(M, N) = 1$. Analise o algoritmo.

Exercício 1.104 (Arvind e Mukhopadhyay (2008) e Klivans e Spielman (2001)). Prove a seguinte versão do Lema do Isolamento (Exercício 1.58, página 50). Sejam C, ε constantes positivas e $\{\sum_i c_i x_i\}$ uma família de formas lineares distintas em que $0 \leqslant c_i \leqslant C$ são inteiros para todo i. Se cada x_i é escolhido aleatoriamente em $\{0, 1, \ldots, Cn/\varepsilon\}$, então existe uma única forma linear de valor mínimo com probabilidade $1 - \varepsilon$. Use esse resultado para dar um algoritmo probabilístico para o problema da identidade de polinômios.

Exercício 1.105. Prove que a definição de probabilidade dada na equação (1.28), página 88, é consistente, isto é, se existem $B \subseteq \{0,1\}^k$ e $B' \subseteq \{0,1\}^{k'}$, com $k \neq k'$, tais que $A := B \times \{0,1\}^{\mathbb{N}} = B' \times \{0,1\}^{\mathbb{N}}$, então $P(A)$ definido na equação (1.28) coincide nas duas representações de A.

Exercício 1.106. Em geral, a parte difícil da aplicação do teorema de Carathéodory (descrito informalmente na página 88) é provar que a função que se quer estender é enumeravelmente aditiva. O que se faz, normalmente, é provar que a função é finitamente aditiva e contínua no sentido da seção 1.1.3. Prove que se $\mathbb{P}$ é não-negativa, finitamente aditiva e $\mathbb{P}(\Omega) = 1$ então são equivalentes

1. $\mathbb{P}$ é uma medida de probabilidade.
2. Para toda sequência decrescente $(A_n : n \geqslant 1)$ de elementos de $\mathcal{A}$ tal que $\bigcap_{n \geqslant 1} A_n = \varnothing$ vale que $\lim_{n \to \infty} \mathbb{P}(A_n) = 0$.