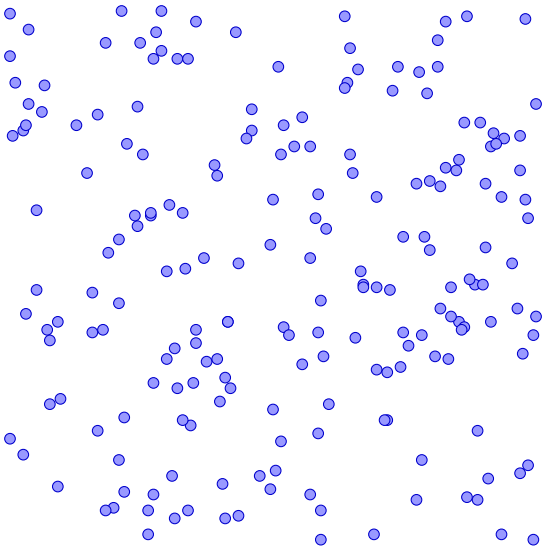
ESTRUTURAS DISCRETAS

Textos de apoio



Jorge Picado

Departamento de Matemática

Universidade de Coimbra

Segunda Edição

Agosto de 2014

Índice


Matemática Discreta
Fundamentos
Linguagens de primeira ordem
usando Tarski World
Lógica proposicional
usando Boole
Raciocínio matemático
Indução e recursão
Combinatória
Relações de recorrência
Técnicas de contagem avançadas
Contagem e probabilidade discreta
Algoritmos
Complexidade
Somatórios
Números inteiros
Criptografia
Aplicações
Aritmética modular
Teoria dos Grafos
Grafos: noções básicas
Árvores
Grafos eulerianos
Problemas famosos
Grafos hamiltonianos

Prefácio

Estes apontamentos incluem com algum pormenor os principais conceitos e resultados apresentados nas aulas, completados com exemplos, observações e exercícios. Neles vamos introduzir os conceitos básicos de matemática discreta, necessários para uma compreensão rigorosa da disciplina de informática e vamos motivar para o raciocínio matemático. Serão abordados temas que vão da lógica à álgebra, passando pela teoria das probabilidades e pela teoria dos grafos, através de uma articulação entre a teoria e a prática. Serão utilizados programas específicos para a parte da lógica: (`Tarski World` e `Boole`). Dada a extensão do programa será dada preferência a uma abordagem de ensino teórico "em largura", deixando para as aulas práticas, e trabalho em casa, o aprofundamento das diversas matérias.

Espera-se que estes apontamentos sejam um auxiliar valioso para o curso, que permita uma maior liberdade nas aulas, na explicação teórica dos assuntos, substituindo uma exposição com grande pormenor formal por uma que realce a motivação e os aspectos intuitivos desses mesmos conceitos e respectivas inter-relações, e que por outro lado sejam um estímulo à atenção e participação activa dos estudantes. Devem ser encarados como um mero guião das aulas, não sendo portanto um seu substituto. Na sua elaboração baseámo-nos fundamentalmente nos livros

- K. H. Rosen, *Discrete Mathematics and its Applications*, McGraw-Hill, 1995. (03A/ROS)[1]
- James Hein, *Discrete Structures, Logic and Computability*, Portland State University, 2002. (03D/HEI)
- Jon Barwise e John Etchemendy, *Language, Proof and Logic*, CSLI Publications, 1999. (03B/BAR.Lan)

Como material de estudo, além destes apontamentos recomendamos nalguns pontos do programa o livro

- C. André e F. Ferreira, *Matemática Finita*, Universidade Aberta, 2000. (05A/AND)

Podem ser encontradas mais informações sobre o curso (incluindo os apontamentos, restante material de apoio, sumários das aulas, etc.) em

`http://www.mat.uc.pt/~picado/ediscretas`

[1] Entre parênteses indica-se a cota do livro na Biblioteca do DMUC.

Que é a Matemática Discreta?

A matemática discreta (ou, como por vezes também é apelidada, matemática finita ou matemática combinatória) é a parte da Matemática devotada ao estudo de objectos e estruturas discretas ou finitas (*discreta* significa que é formada por elementos distintos desconexos entre si). O tipo de problemas que se resolvem usando matemática discreta incluem: De quantas maneira podemos escolher uma *password* válida para um computador? Qual é a probablidade de ganharmos o euromilhões? Qual é o caminho mais curto entre duas cidades para um determinado sistema de transporte? Como é que podemos ordenar uma lista de inteiros de modo a que os inteiros fiquem por ordem crescente? Em quantos passos podemos fazer essa ordenação? Como podemos desenhar um circuito para adicionar dois inteiros?

Genericamente, a matemática discreta é usada quando contamos objectos, quando estudamos relações entre conjuntos finitos e quando processos (algoritmos) envolvendo um número finito de passos são analisados. Nos últimos anos tornou-se uma disciplina importantíssima da Matemática porque nos computadores a informação é armazenada e manipulada numa forma discreta.

A matemática discreta aborda fundamentalmente três tipos de problemas que surgem no estudo de conjuntos e estruturas discretas:

I - Problemas de existência:
Existe algum arranjo de objectos de um dado conjunto satisfazendo determinada propriedade?

Exemplos:

(A1) Se num dado exame as notas foram dadas com aproximação até às décimas e a ele compareceram 202 alunos, existirão dois alunos com a mesma nota?

(A2) Escolham-se 101 inteiros entre os inteiros $1, 2, 3, \ldots, 200$. Entre os inteiros escolhidos, existirão dois tais que um é divisor do outro?

(A3) Se 101 (resp. n^2+1) pessoas se encontrarem alinhadas lado a lado numa linha recta, será possível mandar dar um passo em frente a 11 (resp. $n+1$) delas de tal modo que, olhando para este grupo da esquerda para a direita, as pessoas se encontrem por ordem crescente ou decrescente das suas alturas?

Ou seja, de uma sequência

$$a_1, a_2, \ldots, a_{n^2+1}$$

de números reais, será possível extrair uma subsequência crescente ou decrescente com $n+1$ elementos?

Por exemplo, a sequência $3, 2, 12, 8, 10, 1, 4, 11, 9, 7$ contém 10 termos. Note-se que $10 = 3^2+1$. Existem 2 subsequências crescentes de comprimento 4, nomeadamente $3, 8, 10, 11$ e $2, 8, 10, 11$. Existe também uma subsequência decrescente de comprimento 4 que é $12, 10, 9, 7$. Por outro lado, a sequência $3, 2, 12, 8, 10, 1, 4, 11, 7, 9$ já não contém nenhuma subsequência decrescente de comprimento 4. Em contrapartida, tem 5 subsequências crescentes de comprimento 4: $3, 8, 10, 11$; $3, 4, 7, 9$; $2, 8, 10, 11$; $2, 4, 7, 9$ e $1, 4, 7, 9$.

(A4) O Rio Pregel atravessa a cidade de Königsberg, na Prússia Oriental (actualmente Kaliningrado, na Rússia), dividindo-a em quatro regiões, como se pode ver na seguinte gravura[2] da cidade:

Conta-se que os habitantes de Königsberg se entretinham a tentar encontrar uma maneira de efectuar um passeio pela cidade, de modo a voltar ao ponto de partida, passando uma única vez por cada uma das 7 pontes. Como as suas tentativas saíram sempre goradas, muitos acreditavam ser impossível realizar tal trajecto. Contudo, só em 1736, com um artigo de L. Euler[3], o problema foi totalmente abordado de modo matemático, e tal impossibilidade foi provada. Vale a pena lermos os primeiros parágrafos desse artigo de Euler:

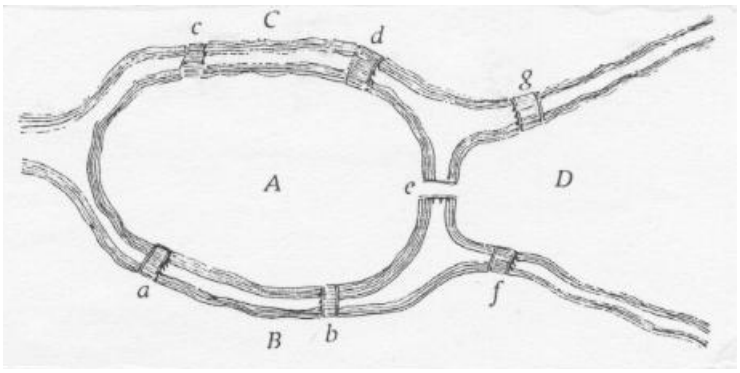
> *"1. Além do ramo da geometria que se preocupa com grandezas, e que sempre recebeu a maior atenção, existe outro ramo, quase desconhecido anteriormente, que Leibniz pela primeira vez mencionou, chamando-lhe 'geometria da posição'. Este ramo preocupa-se com a determinação de posições e suas propriedades; não envolve medidas, nem cálculos feitos com elas. Ainda não se determinou de modo satisfatório que tipo de problemas são relevantes para esta geometria*

[2] [M. Zeiller, *Topographia Prussiae et Pomerelliae*, Frankfurt, c. 1650], cópia em [1].

[3] No artigo [*Solutio Problematis ad Geometriam Situs Pertinentis*, Commentarii Academiae Scientiarum Imperialis Petropolitanae 8 (1736) 128-140], baseado numa comunicação apresentada à Academia em 26 de Agosto de 1735, e considerado por muitos o nascimento da Teoria dos Grafos. Euler foi um dos maiores génios da matemática; este ano comemoram-se os 300 anos do seu nascimento.

de posição, ou que métodos deverão ser utilizados para os resolver. Portanto, quando um problema foi recentemente mencionado, que parecia geométrico mas era tal que não requeria medir distâncias, nem realizar cálculos, não tive dúvida que tinha a ver com a geometria de posição — fundamentalmente porque a sua solução envolvia somente posição, e nenhuns cálculos eram úteis. Decidi então apresentar aqui o método que encontrei para resolver este tipo de problema, como um exemplo da geometria de posição.

2. O problema, que me foi dito ser muito popular, é o seguinte: em Königsberg na Prússia, existe uma ilha A, chamada 'Kneiphof'; o rio que a rodeia divide-se em dois braços, como pode ser visto na figura, e estes braços são atravessados por sete pontes a, b, c, d, e, f e g.



Pergunta-se se alguém consegue encontrar um trajecto de tal modo que atravesse cada ponte uma e uma só vez. Foi-me dito que algumas pessoas afirmaram tal ser impossível, enquanto outras tinham dúvidas; mas ninguém assegurou que tal trajecto existe. A partir disto, formulei o problema geral: qualquer que seja o arranjo e a divisão do rio em braços, e qualquer que seja o número de pontes, pode-se concluir se é possível ou não atravessar cada ponte exactamente uma vez?

3. Quanto ao problema das 7 pontes de Königsberg, este pode ser resolvido fazendo uma lista exaustiva de todos os trajectos possíveis, e verificando se cada trajecto satisfaz ou não as condições do problema. Por causa do número de possibilidades, este método de resolução seria muito complicado e laborioso, e noutros problemas com mais pontes totalmente impraticável. Além disso, se seguirmos este método até à sua conclusão, muitos trajectos irrelevantes serão encontrados, que é a razão da dificuldade deste método. Portanto rejeitei-o, e procurei outro, preocupado somente com o problema da existência do trajecto requerido; achei que um tal método seria mais simples."[4]

[4]O resto do artigo pode ser lido em [1]. O ponto de vista que Euler tomou, de não confinar a sua atenção ao caso particular do problema de Königsberg mas olhar para o problema geral, é típico de um matemático. Contudo Euler continuou com o caso particular em mente, voltando a ele mais do que uma vez, para interpretar e verificar as suas novas descobertas. Isto é muito interessante, ilustrando como a generalização e a especialização se complementam na investigação matemática. Outro aspecto muito interessante ocorre na Secção 4, quando

(A5) Imagine uma prisão com 64 celas, dispostas como os quadrados de um tabuleiro de xadrez (com 8 linhas e 8 colunas). Imagine ainda que entre cada duas celas vizinhas existe uma porta. É proposta, ao prisioneiro colocado na cela de um dos cantos, a sua liberdade caso consiga chegar à cela do canto diagonalmente oposto, depois de passar por todas as outras celas uma única vez. Conseguirá o prisioneiro obter a sua liberdade?

(A6) Consideremos um tabuleiro de xadrez e algumas peças (idênticas) de dominó tais que cada uma cobre precisamente 2 quadrados adjacentes do tabuleiro. Será possível dispor 32 dessas peças no tabuleiro de modo a cobri-lo, sem sobreposição de peças?[5]

E se o tabuleiro tiver mn quadrados em m linhas e n colunas?

II - Problemas de contagem (e enumeração):
Quantos arranjos (configurações) desse tipo existem?
Por vezes será importante ainda enumerá-los e/ou classificá-los.

Exemplos:

(B1) O problema (A6) de existência de uma cobertura perfeita de um tabuleiro de xadrez é muito simples; rapidamente se constroem diversas coberturas perfeitas. É no entanto muito mais difícil proceder à sua contagem. Tal foi feito pela primeira vez em 1961 por M. E. Fisher[6]: são

$$12\,988\,816 = 2^4 \times (901)^2.$$

Para outros valores de m e n já poderá não existir nenhuma cobertura perfeita. Por exemplo, não existe nenhuma no caso $m = n = 3$. Para que valores de m e n existem? Não é difícil concluir que um tabuleiro $m \times n$ possui uma cobertura perfeita se e só se pelo menos um dos números m ou n é par, ou equivalentemente, se e só se o número mn de quadrados do tabuleiro é par. Fischer determinou fórmulas gerais (envolvendo funções trigonométricas) para o cálculo do número exacto de coberturas perfeitas de um tabuleiro $m \times n$.

Este problema é equivalente a um problema famoso em Física Molecular, conhecido como o *Problema das moléculas diatómicas*[7].

(B2) Sejam $A = \{a_1, a_2, \ldots, a_t\} \subseteq \mathbb{N}$ e $n \in \mathbb{N}$. Quantos inteiros positivos, inferiores ou iguais a n, não são divisíveis por nenhum dos elementos de A? Quantos inteiros positivos inferiores a n são primos com n? Quantos números primos compreendidos entre 2 e $n \geq 2$ existem?

(B3) Um empregado de um restaurante, encarregue de guardar os n chapéus dos n clientes esqueceu-se de os identificar. Quando os clientes os pediram de volta, o empregado foi-os

Euler introduz a notação conveniente ("*o modo particularmente conveniente no qual o atravessamento de uma ponte pode ser representado*") e com ela obtém nas secções subsequentes um dispositivo muito útil para resolver o problema, que mostra como o cuidado na escolha da boa notação pode ser muitas vezes a chave do problema.

[5] Tal arranjo diz-se uma *cobertura perfeita* do tabuleiro por dominós.

[6] *Statistical Mechanics of Dimers on a Plane Lattice*, Physical Review 124 (1961) 1664-1672.

[7] Cf. [M. E. Fisher, *Statistical Mechanics of Dimers on a Plane Lattice*, Physical Review 124 (1961) 1664-1672].

devolvendo de forma aleatória! Qual é a probabilidade de nenhum cliente receber o seu chapéu de volta?

O caso $n = 52$ deste problema é equivalente ao célebre *Problème des rencontres* proposto por Montmort em 1708:

> *No chamado "jogo dos pares", as 52 cartas de um baralho são dispostas em linha, com o seu valor à vista. As cartas de um segundo baralho são dispostas também em linha por cima das outras. A pontuação é determinada contando o número de vezes em que a carta do segundo baralho coincide com a do primeiro sobre a qual foi colocada. Qual é a probabilidade de se obterem zero pontos ?*

(B4) O seguinte problema foi originalmente proposto por Leonardo de Pisa[8], mais conhecido por Fibonacci, no séc. XIII:

Suponhamos que, para estudar a reprodução profícua dos coelhos, colocámos um par de coelhos (sendo um de cada sexo) numa ilha. Passados dois meses, a fêmea deu à luz todos os meses um novo par de coelhos, de sexos opostos. Por sua vez, a partir dos dois meses de idade, cada novo par deu à luz um outro par, todos os meses. Quantos pares de coelhos existiam na ilha ao cabo de n meses, supondo que nenhum coelho morreu entretanto?

A população de coelhos pode ser descrita por uma *relação de recorrência*. No final do primeiro mês o número de pares de coelhos era 1. Como este par não reproduziu durante o segundo mês, no final deste o número de pares de coelhos continuou a ser 1. Durante o terceiro mês nasceu um novo par pelo que no final deste mês existiam 2 pares de coelhos. Durante o quarto mês só o par inicial deu origem a um novo par, logo no final do quarto mês existiam 3 pares de coelhos.

MÊS	Pares reprodutores	Pares jovens	Total de pares
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

Denotemos por f_n o número de pares de coelhos existentes no final do mês n. Este número é claramente igual à soma do número de pares de coelhos existentes no final do mês anterior, ou seja f_{n-1}, com o número de pares de coelhos entretanto nascidos durante o mês n, que é igual a f_{n-2}. Portanto a sequência $(f_n)_{n\in\mathbb{N}}$ satisfaz a relação

$$f_n = f_{n-1} + f_{n-2}$$

para $n \geq 3$, sendo $f_1 = f_2 = 1$.

[8] No seu livro *Liber Abacci* (literalmente, um livro sobre o ábaco), publicado em 1202.

Esta sucessão é a famosa *sucessão de Fibonacci*, e os seus termos são chamados *números de Fibonacci*[9].

Claro que para responder totalmente ao problema de Fibonacci teremos de encontrar um método para determinar uma fórmula explícita para o número f_n a partir daquela relação de recorrência.

III - Problemas de optimização:
De todas as possíveis configurações, qual é a melhor de acordo com determinado critério?

Exemplos:

(C1) A velocidade com que um gás flui através de uma tubagem depende do diâmetro do tubo, do seu comprimento, das pressões nos pontos terminais, da temperatura e de várias propriedades do gás. O desenho de uma rede de distribuição de gás envolve, entre outras decisões, a escolha dos diâmetros dos tubos, de modo a minimizar o custo total da construção e operação do sistema. A abordagem *standard* consiste em recorrer ao "bom senso" (método habitual da engenharia!) para a escolha de tamanhos razoáveis de tubagem e esperar que tudo corra pelo melhor. Qualquer esperança de fazer melhor parece, à primeira vista, não existir. Por exemplo, uma pequena rede com 40 ligações e 7 diâmetros possíveis de tubo, daria origem a 7^{40} redes diferentes. O nosso problema é o de escolher a rede mais barata de entre essas 7^{40} possibilidades (que é um número astronómico!). Trata-se assim de um problema de optimização, no qual procuramos o desenho (padrão ou arranjo) óptimo para um determinado desempenho.

Este problema, mesmo com o uso dos actuais computadores de grande velocidade, não parece tratável por exaustiva análise de todos os casos. Mesmo qualquer desenvolvimento esperado na velocidade daqueles não parece ter influência significativa nesta questão. Contudo, um procedimento simples implementado no Golfo do México[10], deu origem a um método que permite encontrar a rede óptima em $7 \times 40 = 280$ passos em vez dos tais 7^{40}, permitindo poupar alguns milhões de dólares. É um exemplo paradigmático das virtualidades da chamada Optimização Combinatória.

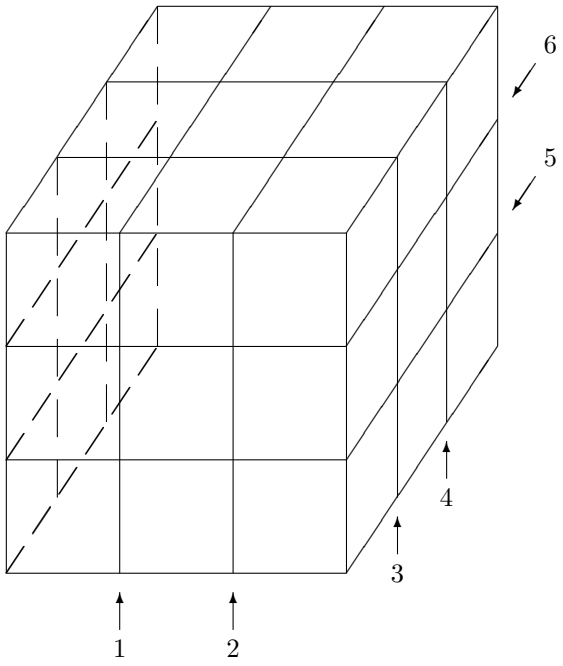
(C2) Suponha que se fazem n cortes numa *pizza*. Qual o número máximo de partes em que a *pizza* poderá ficar dividida?

(C3) Consideremos um cubo de madeira com 3 cm de lado. Se desejarmos cortar o cubo em 27 cubos de 1 cm de lado, qual é o número mínimo de cortes em que tal pode ser realizado?

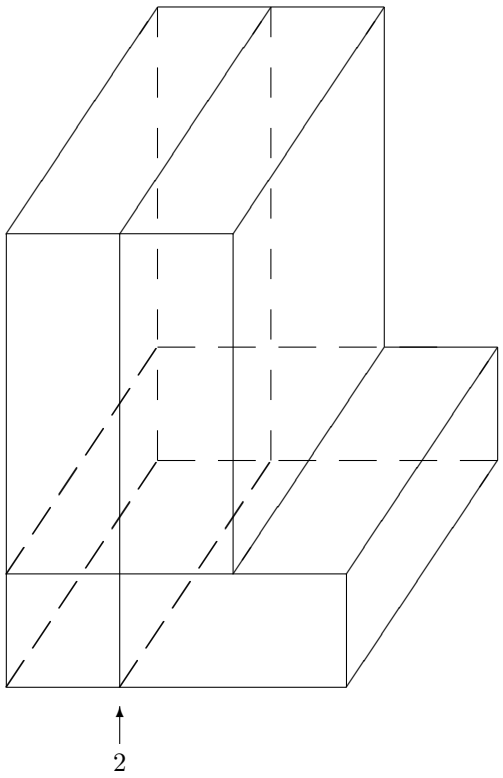
Uma maneira de cortar o cubo é fazendo 6 cortes, 2 em cada direcção (enquanto se mantém o cubo num só bloco):

[9] Estes números aparecem em variadíssimos problemas. Prova da sua importância é a existência da revista *Fibonacci Quartely*, revista da *Fibonacci Association*.

[10] Cf. [H. Frank e I. T. Frisch, *Network Analysis*, Sci. Amer. 223 (1970) 94-103], [D. J. Kleitman, *Comments on the First Two Days' Sessions and a Brief Description of a Gas Pipeline Network Construction Problem*, em F. S. Roberts (ed.), Energy: Mathematics and Models, SIAM, Filadélfia, 1976, p. 239-252], [Rothfarb *et al.*, *Optimal Design of Offshore Natural-Gas Pipeline Systems*, Oper. Res. 18 (1970) 992-1020] e [N. Zadeh, *Construction of Efficient Tree Networks: The Pipeline Problem*, Networks 3 (1973) 1-32].



Mas será possível realizar tal operação com menos cortes, se as peças puderem ser deslocadas entre cortes? Por exemplo, em



o segundo corte corta agora mais madeira do que cortaria se não tivéssemos rearranjado as peças depois do primeiro corte. Parece, pois, um problema difícil de analisar. Olhemos no entanto para ele de outro modo. As 6 faces do cubo do meio só se conseguem obter com cortes (independentes). Portanto, são sempre necessários 6 cortes e fazer rearranjos das peças entre os cortes não ajuda nada.

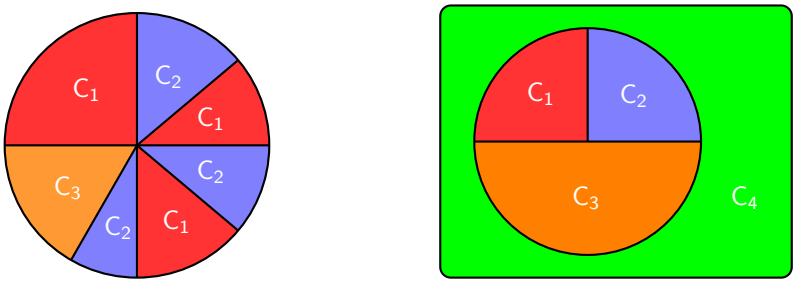
Agora outro problema (este de contagem) surge naturalmente: de quantas maneiras diferentes pode o cubo ser cortado, realizando somente 6 cortes?

(C4) Em 1852, Francis Guthrie reparou que no mapa de Inglaterra os condados poderiam ser coloridos, usando somente quatro cores, de modo a que condados vizinhos tivessem cores diferentes. Através do seu irmão perguntou a De Morgan se quatro cores chegariam para colorir, naquelas condições, qualquer mapa. Em 1878, num encontro da Sociedade Matemática de Londres, A. Cayley perguntou se alguém conseguia resolver o problema. Assim teve origem o famoso *Problema das 4 cores*. Somente em 1976, K. Appel e W. Hagen da Universidade do Illinois (E.U.A.), o conseguiriam resolver, com uma demonstração polémica[11], com a ajuda imprescindível do computador, que executou rotinas durante mais de 1000 horas consecutivas!

A demonstração deste resultado está muito longe de ser apresentável, pelo que nos limitamos a enunciar a solução[12]:

Em qualquer mapa sobre um plano ou uma esfera (representando um qualquer conjunto de regiões tais que, para quaisquer dois pontos numa mesma região, existe sempre um caminho, totalmente contido nessa região, ligando esses dois pontos), o menor número de cores necessárias para o colorir, de tal modo que duas regiões *adjacentes* (ou seja, com um número infinito de pontos fronteiros comuns) não tenham a mesma cor, é 4.

Por exemplo:



As origens da Matemática Combinatória datam do séc. XVII em estreita ligação com os jogos de azar e o cálculo das probabilidades; Pascal, Fermat, Jacob Bernoulli e Leibniz realizaram investigações de problemas combinatoriais relacionados com jogos de azar, constituindo estas as bases sobre as quais se desenvolveu o cálculo das probabilidades.

[11] Para uma história mais completa das origens e resolução deste problema consulte [R. Fritsch e G. Fritsch, *The Four-Color Theorem*, Springer, 1998].

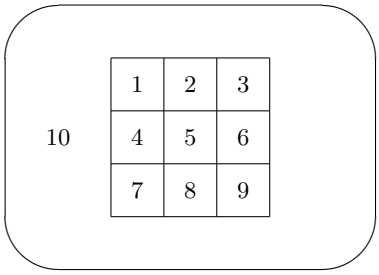
[12] K. Appel e W. Hagen, *Every planar map is four coulorable*, Bull. Amer. Math. Soc. 82 (1976) 711-712.

No séc. XVIII Euler fundou a Teoria dos Grafos com a resolução do famoso *problema das pontes de Königsberg*, como já referimos, e James Bernoulli publicou o primeiro livro[13] contendo métodos combinatoriais.

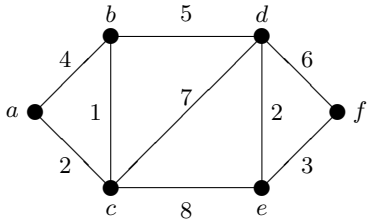
Com o desenvolvimento dos computadores, a Matemática Combinatória tornou-se uma disciplina autónoma dentro da matemática moderna, das que mais se tem desenvolvido, tendo inúmeras aplicações a diversas áreas da matemática, engenharias e outras ciências.

Exercícios

1. Mostre que um tabuleiro com $m \times n$ quadrados possui uma cobertura perfeita se e só se pelo menos um dos valores m ou n é par.
2. Para cada $n \in \mathbb{N}$, seja $f(n)$ o número de coberturas perfeitas de um tabuleiro $2 \times n$. Calcule $f(1), f(2), f(3), f(4)$ e $f(5)$. Tente encontrar uma relação que seja satisfeita pela função f e que lhe permita calcular $f(12)$.
3. Determine o número de coberturas perfeitas distintas de um tabuleiro 3×4.
4. Seja n um inteiro positivo. Dizemos que uma *n-coloração* de um mapa é uma coloração de todas as regiões do mapa, usando n cores, de tal modo que regiões adjacentes (isto é, regiões com um número infinito de pontos fronteiros comuns) têm cores diferentes. Prove que:
   (a) Um mapa formado no plano por um número finito de círculos possui uma 2-coloração.
   (b) Um mapa formado no plano por um número finito de linhas rectas também possui uma 2-coloração.
5. Mostre que o seguinte mapa de 10 países admite uma 3-coloração. Fixadas essas 3 cores, determine o número de colorações distintas possíveis.



6. Determine o caminho mais curto de a para f no mapa de estradas da figura



(Os valores junto de cada estrada representam os comprimentos destas, medidos numa determinada unidade.)

[13] *Ars Conjectandi.*

Referências

[1] N. L. Biggs, E. K. Lloyd e R. J. Wilson, *Graph Theory 1736-1936*, Clarendon Press, 1986. (05-01/BIG)

1. Fundamentos

1.1. Como raciocinamos? Lógica proposicional

A lógica é a base de todo o raciocínio. Portanto se quisermos estudar e fazer matemática precisamos de dominar os princípios básicos da lógica. Citando *Language, Proof and Logic* (de Jon Barwise e John Etchemendy):

> "(...) all rational inquiry depends on logic, on the ability of people to reason correctly most of the time."

> "(...) there is an overwhelming intuition that the laws of logic are somehow more irrefutable than the laws of the land, or even the laws of physics."

Porque deve um estudante de Informática estudar lógica? Porque precisa de dominar ferramentas lógicas que lhe permitam argumentar se um problema pode ou não ser resolvido num computador, traduzir proposições lógicas da linguagem comum em diversas linguagens computacionais, argumentar se um programa está correcto e se é eficiente. Os computadores baseiam-se em mecanismos lógicos e são programados de modo lógico. Os informáticos devem ser capazes de compreender e aplicar novas ideias e técnicas de programação, muitas das quais requerem conhecimento dos aspectos formais da lógica.

Todos nós raciocinamos enunciando factos e tirando conclusões baseadas nesses factos. O início de uma conclusão é habitualmente indicada por uma palavra como

Então, Logo, Portanto, Consequentemente, ...

Para chegarmos a uma conclusão aplicamos uma *regra de inferência* (ou *regra de dedução*). A mais comum é a chamada regra *modus ponens* (modo que afirma): sendo A e B afirmações, se A e "se A então B" são ambas verdadeiras, então podemos concluir que B é verdadeira.

`[Como aprendeu a regra modus ponens em criança?]`

Outra regra muito comum é a *modus tollens* (modo que nega): sendo A e B afirmações, se "se A então B" é verdadeira e B é falsa, então podemos concluir que A é falsa.

Por exemplo:

- Se ele foi a Coimbra então visitou a Universidade de Coimbra.
- Ele não visitou a Universidade de Coimbra.
- Logo não foi a Coimbra.

`[Como aprendeu a regra modus tollens em criança?]`

Quando tiramos uma conclusão que não decorre dos factos estabelecidos previamente, o raciocínio diz-se *non sequitur* (que não segue). Por exemplo:

– Sabe-se que ontem usei calças de ganga ou camisa branca.

– Além disso, ontem viram-me com calças de ganga.

– Logo não usei camisa branca.

`[Pense noutro exemplo de non sequitur que já tenha observado.]`

Neste primeiro capítulo começaremos por estudar um pouco de lógica. Algumas definições de *lógica* que podemos encontrar nos dicionários:

- Estudo dos princípios do raciocínio, particularmente da estrutura das afirmações e proposições e dos métodos de determinação da sua validade.
- Sistema de raciocínio.
- Raciocínio válido.

Um *cálculo* é uma linguagem de expressões, onde cada expressão tem um valor lógico e há regras para transformar uma expressão noutra com o mesmo valor. Aqui estudaremos um pouco do cálculo proposicional. O *cálculo proposicional* é a linguagem das *proposições*. Uma *proposição* é uma expressão da qual faz sentido dizer que é verdadeira ou que é falsa. Cada proposição tem um e um só valor lógico, entre dois possíveis: V (verdadeiro) ou F (falso).

Exemplo. "Coimbra é uma cidade portuguesa" é uma proposição com valor lógico verdadeiro. Mas atribuir um valor lógico à afirmação "Hoje está um belo dia!" já não faz sentido, pois trata-se duma expressão subjectiva que exprime um sentimento de alguém, não de uma afirmação objectiva.

Lógica e operações-bit: Os computadores representam informação por meio de bits. Um bit tem dois valores possíveis, 0 e 1. Um bit pode ser usado para representar os valores de verdade F e V, 0 representa F e 1 representa V. Há assim uma relação evidente entre a lógica e o sistema de funcionamento dos computadores.

O cálculo proposicional (tal como outros tipos de lógica) que vamos estudar pressupõe os seguintes princípios:

Princípio da não contradição: *Uma proposição não pode ser verdadeira e falsa ao mesmo tempo.*

Princípio do terceiro excluído: *Uma proposição é verdadeira ou falsa.*

A afirmação "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" pode decompor-se em duas afirmações: "Os alunos de Estruturas Discretas são de Engenharia Informática" e "Os alunos de Estruturas Discretas são de Comunicações e Multimédia". Estas duas últimas afirmações já não se podem decompor mais. Dizemos então que a proposição "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" é *composta* e as afirmações "Os alunos de Estruturas Discretas são de Engenharia Informática" e "Os alunos de Estruturas Discretas são de Comunicações e

Multimédia" são atómicas. As proposições compostas são construídas a partir de proposições atómicas ligando-as por *conectivos* (ou *operadores*). No exemplo anterior, esse conectivo é "ou". Se denotarmos por p a proposição "Os alunos de Estruturas Discretas são de Engenharia Informática" e por q a proposição "Os alunos de Estruturas Discretas são de Comunicações e Multimédia" e usarmos o símbolo $\vee$ para representar "ou", a afirmação "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" escreve-se simplesmente $p \vee q$. Temos assim a operação $\vee$ de *disjunção*.

A afirmação "Ele não visitou a Universidade de Coimbra" é a negação de "Ele visitou a Universidade de Coimbra". Se denotarmos por p esta última proposição e usarmos o símbolo $\neg$ para representar a operação de negação, a afirmação "Ele não visitou a Universidade de Coimbra" escreve-se $\neg p$.

A seguinte tabela contém uma lista destes e doutros conectivos lógicos[1]:

negação	$\neg p$ (não p)
conjunção	$p \wedge q$ (p e q)
disjunção	$p \vee q$ (p ou q)
implicação	$p \rightarrow q$ (se p então q; p só se q; p é condição suficiente para que q; q é condição necessária para que p)
equivalência (formal)	$p \leftrightarrow q$ (p é equivalente a q)
disjunção exclusiva	$p \dot{\vee} q$ (ou p ou q)

As proposições são representadas por fórmulas chamadas *fórmulas bem formadas* que são construídas a partir de um alfabeto constituído por:

- Símbolos de verdade: V e F.
- Variáveis proposicionais: letras do alfabeto $p, q, r, \ldots$
- Conectivos (operadores):

  $\neg$ ("não", negação)

  $\wedge$ ("e", conjunção)

  $\vee$ ("ou", disjunção)

  $\rightarrow$ ("implica", implicação).
- Símbolos de parênteses: (,).

Uma *fórmula bem formada* (abreviadamente, fbf) fica definida da seguinte forma:

- V e F são fbf's; toda a variável proposicional é uma fbf.
- Se A e B são fbf's, as seguintes são também fbf's: $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, (A).

[1] Em lógica é habitual designar estes conectivos por *conectivos booleanos*, em homenagem ao lógico britânico George Boole (1815-1864), que estudou as leis do pensamento usando métodos matemáticos em [*An investigation into the Laws of Thought*, 1854].

- Toda a fbf é construída por aplicação sucessiva das regras anteriores.

Exemplo. A expressão $p\neg q$ não é uma fbf. Mas cada uma das seguintes expressões é uma fbf: $p \wedge q \rightarrow r$, $(p \wedge q) \rightarrow r$, $p \wedge (q \rightarrow r)$.

Os parênteses funcionam como símbolos auxiliares que indicam como é formada a fbf. Para evitar um uso excessivo de parênteses e simplificar a escrita das expressões lógicas convenciona-se que as operações lógicas são consideradas pela seguinte ordem de prioridade: $\neg$, $\wedge$, $\vee$, $\rightarrow$. Convenciona-se ainda que na presença de uma só das três últimas operações, na ausência de parênteses as operações são realizadas da esquerda para a direita.

Exemplos.

$$\begin{array}{rcl} \neg p \wedge q & \text{significa} & (\neg p) \wedge q \\ p \vee q \wedge r & \text{significa} & p \vee (q \wedge r) \\ p \wedge q \rightarrow r & \text{significa} & (p \wedge q) \rightarrow r \\ p \rightarrow q \rightarrow r & \text{significa} & (p \rightarrow q) \rightarrow r. \end{array}$$

Relativamente a uma dada linguagem lógica podemos sempre estudar dois aspectos: a sintaxe e a semântica. A sintaxe diz respeito às regras de formação das expressões lógicas a utilizar, ou seja, as fórmulas bem formadas. Em cima, acabámos de descrever a sintaxe do cálculo proposicional.

A semântica estuda o significado das expressões.

$$\text{Linguagem (conjunto de símbolos)} \begin{cases} \text{sintaxe (fórmulas bem formadas)} \\ \\ \text{semântica (significado).} \end{cases}$$

Quanto à semântica, dada uma fbf, interpretando cada uma das suas variáveis proposicionais com os valores lógicos V ou F, é possível dar um significado à fórmula através da interpretação dos conectivos lógicos dada pelas respectivas *tabelas de verdade*. Cada conectivo tem uma tabela de verdade (que vai ao encontro da forma corrente do significado das operações "não", "e", "ou", etc.). A tabela de verdade faz corresponder aos possíveis valores lógicos das variáveis o correspondente valor lógico da operação[2]:

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$
V	V	F	V	V	V
V	F	F	F	V	F
F	V	V	F	V	V
F	F	V	F	F	V

Em conclusão:

[2]Nas aulas teórico-práticas usaremos o software `Boole` para nos ajudar a escrever tabelas de verdade. Consulte o apêndice Usando Boole.

- O significado de V é verdade e o de F é falso.
- O significado de qualquer outra fbf é dado pela sua tabela de verdade.

Exemplo. $\neg p \wedge q$ corresponde a $(\neg p) \wedge q$, cuja tabela de verdade é:

p	q	$\neg p$	$(\neg p) \wedge q$
V	V	F	F
V	F	F	F
F	V	V	V
F	F	V	F

É claro que a cada fbf corresponde uma e uma só tabela de verdade.

Uma fbf diz-se uma *tautologia* se for verdadeira para todos os possíveis valores lógicos das suas variáveis proposicionais. Uma fbf diz-se uma *contradição* se for falsa para todos os possíveis valores lógicos das suas variáveis proposicionais. Uma fbf diz-se uma *contingência* se não for uma tautologia nem uma contradição.

Exemplos. Suponhamos que queremos averiguar se $p \rightarrow p \vee q$ é ou não uma tautologia. Para isso basta construir a respectiva tabela de verdade:

p	q	$p \vee q$	$p \rightarrow p \vee q$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	V

Como para quaisquer valores de p e q toma sempre o valor de verdade, concluímos que é uma tautologia.

Mais exemplos: $p \vee \neg p$ é uma tautologia, $p \wedge \neg p$ é uma contradição e $p \rightarrow q$ é uma contingência.

Duas fbf's dizem-se (*logicamente*) *equivalentes* se tiverem o mesmo significado, isto é, a mesma tabela de verdade. Para indicar que duas fbf's A e B são equivalentes, escrevemos

$$A \equiv B.$$

Em vez do símbolo $\equiv$ também se costuma usar $\Leftrightarrow$. Note que dizer que A e B são logicamente equivalentes é o mesmo que dizer que as fórmulas $(A \rightarrow B)$ e $(B \rightarrow A)$ são tautologias (Prova: $A \equiv B$ sse A e B têm os mesmos valores de verdade sse $(A \rightarrow B)$ e $(B \rightarrow A)$ são tautologias).

Exemplos. As seguintes equivalências básicas são de fácil verificação e são fundamentais no cálculo proposicional:

$p \vee \neg p \equiv \mathsf{V}$	Lei do terceiro excluído
$p \wedge \neg p \equiv \mathsf{F}$	Lei da contradição
$p \wedge \mathsf{V} \equiv p$	Leis da identidade
$p \vee \mathsf{F} \equiv p$	
$p \vee \mathsf{V} \equiv \mathsf{V}$	Leis do elemento dominante
$p \wedge \mathsf{F} \equiv \mathsf{F}$	
$p \vee p \equiv p$	Leis da idempotência
$p \wedge p \equiv p$	
$\neg(\neg p) \equiv p$	Lei da dupla negação
$p \vee q \equiv q \vee p$	Leis da comutatividade
$p \wedge q \equiv q \wedge p$	

$p \vee (p \wedge q) \equiv p$	Leis da absorção
$p \wedge (p \vee q) \equiv p$	
$(p \vee q) \vee r \equiv p \vee (q \vee r)$	Leis da associatividade
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	Leis da distributividade
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	
$\neg(p \wedge q) \equiv \neg p \vee \neg q$	Leis de De Morgan
$\neg(p \vee q) \equiv \neg p \wedge \neg q$	
$p \rightarrow q \equiv \neg p \vee q$	

Por exemplo, construindo as tabelas de verdade de $\neg(p \wedge q)$ e $\neg p \vee \neg q$

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
V	V	V	F	F	F	F
V	F	F	V	F	V	V
F	V	F	V	V	F	V
F	F	F	V	V	V	V

concluímos que ambas as fbf's têm o mesmo valor lógico para os mesmos valores das variáveis proposicionais, pelo que são logicamente equivalentes.

É possível provar uma equivalência sem construir as tabelas de verdade por causa dos seguintes factos:

1. Se $A \equiv B$ e $B \equiv C$, então $A \equiv C$.
2. Se $A \equiv B$, então qualquer fbf C que contenha A é equivalente à fbf obtida de C substituindo uma ocorrência de A por B.

Exemplo. Use as equivalências básicas na tabela anterior para provar que $p \vee q \rightarrow p \equiv q \rightarrow p$.

Prova:

$$\begin{aligned} p \vee q \rightarrow p &\equiv \neg(p \vee q) \vee p \\ &\equiv (\neg p \wedge \neg q) \vee p \\ &\equiv (\neg p \vee p) \wedge (\neg q \vee p) \\ &\equiv \mathsf{V} \wedge (\neg q \vee p) \\ &\equiv \neg q \vee p \\ &\equiv q \rightarrow p. \qquad QED. \end{aligned}$$

Teste (1 minuto cada). Use equivalências conhecidas para provar:

1. $p \vee q \rightarrow r \equiv (p \rightarrow r) \wedge (q \rightarrow r)$.
2. $(p \rightarrow q) \vee (\neg p \rightarrow q) \equiv \mathsf{V}$ (isto é, $(p \rightarrow q) \vee (\neg p \rightarrow q)$ é uma tautologia).
3. $p \rightarrow q \equiv (p \wedge \neg q) \rightarrow \mathsf{F}$.

Teste (1 minuto cada). Use as leis da absorção para simplificar:

1. $(p \wedge q \wedge r) \vee (p \wedge r) \vee r$.
2. $(s \rightarrow t) \wedge (u \vee t \vee \neg s)$.

Se p é uma variável proposicional numa fbf A, denotemos por $A(p/\mathsf{V})$ a fbf que se obtém de A substituindo todas as ocorrências de p por V . De modo análogo podemos também definir a fórmula $A(p/\mathsf{F})$. As seguintes propriedades verificam-se:

- A é uma tautologia se e só se $A(p/\mathsf{V})$ e $A(p/\mathsf{F})$ são tautologias.
- A é uma contradição se e só se $A(p/\mathsf{V})$ e $A(p/\mathsf{F})$ são contradições.

O *Método de Quine* usa estas propriedades, conjuntamente com as equivalências básicas, para determinar se uma fbf é uma tautologia, uma contradição ou uma contingência. (Trata-se de um método alternativo à construção das tabelas de verdade.)

Exemplo. Seja A a fórmula $(p \wedge q \rightarrow r) \wedge (p \rightarrow q) \rightarrow (p \rightarrow r)$. Então:

$$\begin{aligned} A(p/\mathsf{F}) &= (\mathsf{F} \wedge q \rightarrow r) \wedge (\mathsf{F} \rightarrow q) \rightarrow (\mathsf{F} \rightarrow r) \\ &\equiv (\mathsf{F} \rightarrow r) \wedge \mathsf{V} \rightarrow \mathsf{V} \\ &\equiv \mathsf{V}. \end{aligned}$$

Portanto $A(p/\mathsf{F})$ é uma tautologia. A seguir olhemos para

$$\begin{aligned} A(p/\mathsf{V}) &= (\mathsf{V} \wedge q \rightarrow r) \wedge (\mathsf{V} \rightarrow q) \rightarrow (\mathsf{V} \rightarrow r) \\ &\equiv (q \rightarrow r) \wedge q \rightarrow r. \end{aligned}$$

Seja $B = (q \to r) \wedge q \to r$. Então

$$B(q/\mathsf{V}) = (\mathsf{V} \to r) \wedge \mathsf{V} \to r \equiv r \wedge \mathsf{V} \to r \equiv r \to r \equiv \mathsf{V}$$

e

$$B(q/\mathsf{F}) = (\mathsf{F} \to r) \wedge \mathsf{F} \to r \equiv \mathsf{F} \to r \equiv \mathsf{V},$$

o que mostra que B é uma tautologia. Portanto, A é uma tautologia.

Teste (2 minutos cada). Use o método de Quine em cada caso:

1. Mostre que $(p \vee q \to r) \vee p \to (r \to q)$ NÃO é uma tautologia.
2. Mostre que $(p \to q) \to r$ NÃO é equivalente a $p \to (q \to r)$.

No nosso dia a dia raciocinamos e tiramos conclusões usando determinadas regras. A lógica ajuda a compreender essas regras permitindo distinguir entre argumentos correctos e argumentos não correctos. Seguem-se alguns argumentos lógicos, cada um deles com um exemplo e a respectiva formalização.

(1)

1. Se o gato vê o peixe, então o gato apanha o peixe.
2. Se o gato apanha o peixe, então o gato come o peixe.

---

3. Se o gato vê o peixe, então o gato come o peixe.

1. $p \to q$
2. $q \to r$

---

3. $p \to r$

(2)

1. Se o João tem mais de 16 anos, então vai ao cinema.
2. O João tem mais de 16 anos.

---

3. O João vai ao cinema.

1. $p \to q$
2. p

---

3. q

(3)

1. A Maria vai aos testes ou faz o exame.
2. A Maria não faz o exame.

---

3. A Maria vai aos testes.

1. $p \vee q$
2. $\neg q$

---

3. p

Um argumento da forma "De $A_1, A_2, \ldots,$ e A_n deduz-se B", esquematicamente,

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \\ \hline B \end{array}$$

diz-se um *argumento correcto* se $A_1 \wedge A_2 \wedge \cdots \wedge A_n \rightarrow B$ for uma tautologia. Neste caso também se costuma escrever

$$A_1, A_2, \ldots, A_n \models B$$

(o símbolo $\models$ lê-se "de ... deduz-se ..."). Habitualmente nas aulas e na prática da matemática usa-se $A \Rightarrow B$ para indicar $A \models B$.

Um literal é uma variável proposicional ou a sua negação; por exemplo, p e $\neg p$ são literais (ditos *literais complementares*).

Uma fbf diz-se uma *forma normal disjuntiva* (FND) se for da forma $C_1 \vee C_2 \vee \cdots \vee C_n$, onde cada C_i é uma conjunção de literais (chamada *conjunção fundamental*).

Analogamente, uma fbf diz-se uma *forma normal conjuntiva* (FNC) se for da forma $D_1 \wedge D_2 \wedge \cdots \wedge D_n$, onde cada D_i é uma disjunção de literais (chamada *disjunção fundamental*).

Exemplos de formas normais disjuntivas:

$$p$$

$$\neg p$$

$$p \vee \neg q$$

$$p \wedge \neg q$$

$$(p \wedge q) \vee (p \wedge \neg q)$$

$$p \vee (p \wedge r)$$

Exemplos de formas normais conjuntivas:

$$p$$

$$\neg p$$

$$p \wedge \neg q$$

$$\neg p \vee q$$

$$p \wedge (q \vee r)$$

Como já observámos, para qualquer variável proposicional p temos

$$\mathsf{V} \equiv p \vee \neg p \quad \text{e} \quad \mathsf{F} \equiv p \wedge \neg p.$$

Ambas as formas são FND e FNC. Qualquer fbf tem uma FND e uma FNC. De facto, em qualquer fbf podemos usar equivalências básicas para obter uma forma normal conjuntiva:

1. "Removem-se" todas as $\rightarrow$.
2. Se a expressão contém negações de conjunções ou negações de disjunções, fazem-se desaparecer usando as leis de De Morgan, e simplifica-se onde necessário.
3. Agora basta usar as duas propriedades distributivas

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

$$(p \wedge q) \vee r \equiv (p \vee r) \wedge (q \vee r)$$

e simplificar onde necessário.

Para obter uma forma normal disjuntiva procede-se de forma análoga, usando agora em 3 as propriedades

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

$$(p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r).$$

Teste (2 minutos). Transforme $(p \wedge q) \vee \neg(r \rightarrow s)$ numa FND e numa FNC.

Exemplo. $(p \rightarrow q \vee r) \rightarrow (p \wedge s)$

$$\begin{aligned}
&\equiv \neg(p \rightarrow q \vee r) \vee (p \wedge s) && (x \rightarrow y \equiv \neg x \vee y)\\
&\equiv (p \wedge \neg(q \vee r)) \vee (p \wedge s) && (\neg(x \rightarrow y) \equiv x \wedge \neg y)\\
&\equiv (p \wedge \neg q \wedge \neg r) \vee (p \wedge s) && (\neg(x \vee y) \equiv \neg x \wedge \neg y)) \text{ (FND)}\\
&\equiv ((p \wedge \neg q \wedge \neg r) \vee p) \wedge ((p \wedge \neg q \wedge \neg r) \vee s) && (\vee \text{ é distributiva relativamente a } \wedge)\\
&\equiv p \wedge ((p \wedge \neg q \wedge \neg r) \vee s) && \text{(absorção)}\\
&\equiv p \wedge (p \vee s) \wedge (\neg q \vee s) \wedge (\neg r \vee s) && (\vee \text{ é distributiva relativamente a } \wedge) \text{ (FNC)}\\
&\equiv p \wedge (\neg q \vee s) \wedge (\neg r \vee s) && \text{(absorção) (FNC).}
\end{aligned}$$

Uma *função de verdade* (ou *função lógica*) é uma função que só pode tomar os valores lógicos V ou F e cujos argumentos também só podem tomar esses valores. Por exemplo,

$$f(p,q) = \begin{cases} \mathsf{V} \text{ se } p \text{ é } \mathsf{V} \\ \mathsf{V} \text{ se } p \text{ e } q \text{ são ambas } \mathsf{F} \\ \mathsf{F} \text{ se } p \text{ é } \mathsf{F} \text{ e } q \text{ é } \mathsf{V} \end{cases}$$

é uma função de verdade.

É claro que toda a função de verdade pode ser representada por uma tabela de verdade. No exemplo anterior:

p	q	$f(p,q)$
V	V	V
V	F	V
F	V	F
F	F	V

De seguida vamos ver que

Toda a função de verdade é equivalente a uma fbf.

A metodologia a seguir será encontrar uma fbf com a mesma tabela de verdade (podemos construir quer uma FND quer uma FNC).

Técnica. Para construir uma FND, tendo em conta que a disjunção de um número finito de fbf's é V se e só se uma delas o for, basta tomar cada linha da tabela que tenha valor V e construir uma conjunção fundamental que só seja verdadeira nessa linha. De modo análogo, para construir uma FNC, basta considerar cada linha que tenha valor F e construir uma disjunção fundamental que só seja falsa nessa linha.

Exemplo. Na função f acima,

p	q	$f(p,q)$	Partes FND	Partes FNC
V	V	V	$p \wedge q$	
V	F	V	$p \wedge \neg q$	
F	V	F		$p \vee \neg q$
F	F	V	$\neg p \wedge \neg q$	

Assim, $f(p,q)$ pode ser escrita nas formas:

$$f(p,q) \equiv (p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge \neg q) \ \text{(FND)}$$

$$f(p,q) \equiv p \vee \neg q \ \text{(FNC)}.$$

Uma FND para uma fbf A é uma *FND plena* (*forma normal disjuntiva plena*) se cada conjunção fundamental contém o mesmo número de literais, um por cada variável proposicional de A. Uma FNC para uma fbf A é uma *FNC plena* (*forma normal conjuntiva plena*) se cada disjunção fundamental contém o mesmo número de literais, um por cada variável proposicional de A.

Exemplo. No exemplo anterior obtivemos uma FND plena e uma FNC plena.

Podemos usar a técnica das funções de verdade para determinar uma FND plena ou uma FNC plena de qualquer fbf com a excepção das tautologias (não têm uma FNC plena) e das contradições (não têm uma FND). Por exemplo:

$\mathsf{V} \equiv p \vee \neg p$, que é uma FND plena e uma FNC, mas não é uma FNC plena,

$\mathsf{F} \equiv p \wedge \neg p$, que é uma FNC plena e uma FND, mas não é uma FND plena.

Os conectivos lógicos que usámos para definir as fbf's do cálculo proposicional são $\neg$, $\wedge$, $\vee$ e $\rightarrow$. É evidente que o símbolo $\rightarrow$ não é absolutamente necessário (pela última lei da tabela das equivalências básicas): qualquer fbf pode ser substituída por outra logicamente equivalente e onde não figura o símbolo $\rightarrow$.

Um conjunto de conectivos lógicos diz-se *completo* se toda a fbf do cálculo proposicional é equivalente a uma fbf onde figuram apenas conectivos desse conjunto. É claro que

$$\{\neg, \wedge, \vee, \rightarrow\}$$

é completo, por definição.

Exemplos. Cada um dos seguintes conjuntos é um conjunto completo de conectivos:

$$\{\neg, \wedge, \vee\}, \{\neg, \wedge\}, \{\neg, \vee\}, \{\neg, \rightarrow\}, \{\mathsf{F}, \rightarrow\}.$$

Teste (2 minutos). Mostre que $\{\neg, \rightarrow\}$ é completo.

Apêndice: sistemas formais. Como vimos, as tabelas de verdade são suficientes para determinar quando uma fbf é uma tautologia. Contudo, quando uma proposição tem mais do que duas variáveis e contém vários conectivos, a tabela de verdade pode começar a ficar muito complicada. Nesses casos, o método alternativo que vimos de encontrar uma prova de equivalência usando as leis de equivalência básicas ou ainda uma combinação dos dois (por exemplo, o método de Quine) pode ser mais prático.

Quando usamos uma prova de equivalência, em vez de uma tabela de verdade, para verificar se duas fbf's são equivalentes, isso parece de certo modo mais parecido com o modo como comunicamos habitualmente. Embora não seja necessário raciocinar formalmente desse modo no cálculo proposicional, há outros tipos de sistemas lógicos onde isso já é necessário para averiguar da validade das fbf's pois aí as tabelas de verdade não funcionam. Para esses casos existe uma ferramenta: os sistemas de raciocínio formal. Quais são as ideais básicas destes sistemas?

Um *sistema formal* consiste em:

(1) Um conjunto (numerável) de símbolos.

(2) Um conjunto de sequências finitas destes símbolos que constituem as chamadas *fórmulas bem formadas*.

(3) Um determinado conjunto de fbf's, chamadas *axiomas*, que se assumem ser verdadeiras.

(4) Um conjunto finito de "regras de dedução" chamadas *regras de inferência* que permitem deduzir uma fbf como consequência directa de um conjunto finito de fbf's.

Um sistema formal requere algumas regras que ajudem à obtenção de novas fórmulas, as chamadas *regras de inferência*. Uma regra de inferência (que corresponde sempre a uma tautologia do cálculo proposicional) aplica uma ou mais fbf's, chamadas *premissas*, *hipóteses* ou *antecedentes*, numa só fórmula, chamada *conclusão* ou *consequente*. Algumas regras de inferência úteis:

MP (modus ponens)	MT (modus tollens)	Conj (conjunção)
$\dfrac{p\to q,\ p}{\therefore q}$	$\dfrac{p\to q,\ \neg q}{\therefore \neg p}$	$\dfrac{p,\ q}{\therefore p\wedge q}$
Ad (adição)	**SD (silogismo disjuntivo)**	**SH (silogismo hipotético)**
$\dfrac{p}{\therefore p\vee q}$	$\dfrac{p\vee q,\ \neg p}{\therefore q}$	$\dfrac{p\to q,\ q\to r}{\therefore p\to q}$

Aqui o conceito crucial é o de dedução. Uma dedução de uma certa conclusão — digamos S — a partir de premissas $P_1, P_2, \ldots, P_n$ é feita passo a passo. Numa dedução, estabelecem-se conclusões intermédias, cada uma delas conclusão imediata das premissas e conclusões intermédias anteriores. Podemos dizer que uma dedução consiste numa sucessão de afirmações, que são premissas ou conclusões intermédias, e que termina, ao fim de um número finito de passos, quando se obtém a conclusão S.

Cada passo de dedução é correcto, i.e., não oferece dúvidas quanto à validade de cada conclusão intermédia, em consequência da validade das premissas e das conclusões intermédias anteriores.

Uma dedução de uma afirmação S a partir de premissas $P_1, P_2, \ldots, P_n$ é uma demonstração passo a passo que permite verificar que S tem que ser verdadeira em todas as circunstâncias em que as premissas sejam verdadeiras. Uma dedução formal assenta num conjunto fixo de regras de dedução e tem uma apresentação rígida — um pouco à semelhança dos programas escritos numa dada linguagem de programação.

Seja $\mathcal{C}$ um conjunto de fbf's e seja P uma fbf em S. Diz-se que P *é dedutível a partir de* C em S, e escreve-se

$$C \models_S P$$

(ou apenas $C \models P$ se não houver dúvidas sobre o sistema S a que nos referimos) se existir uma sequência finita de fbf's, $P_1, P_2, \ldots, P_n$ tal que:

- $P_n = P$.
- Para cada $i \in \{1, \ldots, n\}$, P_i é um axioma de S ou uma fbf em C ou uma consequência dos P_i's anteriores através de aplicação das regras de inferência.

A sequência dos P_i's diz-se uma *prova formal* de P a partir de C. Se P é dedutível de um conjunto vazio escreve-se $\models_S P$. Neste caso, P diz-se um teorema de S.

Exemplo. $A\to(B\to C), A\to B, A \models C$.

Prova:	1. $A\to(B\to C)$	premissa	
	2. $A\to B$	premissa	
	3. A	premissa	
	4. B	MP(2,3)	
	5. $B\to C$	MP(1,3)	
	6. C	MP(4,5)	*QED.*

Leituras suplementares:

- James Hein, *Discrete Structures, Logic and Computability*, Secção 6.3.
- Jon Barwise e John Etchemendy, *Language, Proof and Logic*, Capítulo 6.

1.2. Linguagens de primeira ordem: Lógica dos predicados

O cálculo proposicional providencia ferramentas adequadas para raciocinarmos sobre fbf's que são combinações de proposições atómicas. Mas uma proposição atómica é uma sentença tomada como um todo o que faz com que o cálculo proposicional não sirva para todo o tipo de raciocínio que precisamos de fazer no dia a dia. Por exemplo, no argumento seguinte é impossível encontrar no cálculo proposicional um método formal para testar a correcção da dedução sem uma análise mais profunda de cada sentença:

(1) Todos os alunos de Engenharia Informática têm um computador portátil.

(2) Boole não tem um computador portátil.

(3) Então Boole não é um aluno de Engenharia Informática.

Para discutir e analisar este argumento precisamos de partir as sentenças em partes. As palavras "Todos", "têm" e "não" são relevantes para a compreensão do argumento.

Por outro lado, a afirmação "x tem um computador portátil" não é uma proposição porque o seu valor de verdade não é absoluto, depende de x. De um ponto de vista gramatical, a propriedade "ter um computador portátil" é um *predicado* (ou seja, a parte da frase que enuncia uma propriedade do sujeito). Do ponto de vista matemático, um predicado é uma relação (unária, binária, ternária, etc.). Por exemplo, "ter um computador portátil" é um predicado unário que podemos designar por CP; então $CP(x)$ significa que x tem um computador portátil (por exemplo, acima afirma-se que $\neg CP(Boole)$).

Portanto, para analisarmos argumentos deste tipo (necessários nas aulas de Análise Matemática e de Álgebra Linear) precisamos de um cálculo de predicados, que inclua dois *quantificadores*:

existe pelo menos um , para todos

A maioria das proposições matemáticas são de uma das duas formas

- Existe um objecto x satisfazendo a propriedade P
- Para todos os objectos x, a propriedade P verifica-se.

Os matemáticos usam o símbolo $\exists$ (*quantificador existencial*) para a primeira e o símbolo $\forall$ (*quantificador universal*) para a segunda:

- $\exists x\ P(x)$.
- $\forall x\ P(x)$.

Por exemplo, a afirmação

Existe um número real x tal que $x^2 + 2x + 1 = 0$

pode ser escrita simbolicamente como

$$\exists x\,(x^2 + 2x + 1 = 0)$$

(desde que especifiquemos à partida que a variável x se refere a números reais, senão teremos que escrever $(\exists x \in \mathbb{R})\ldots$). Para dizermos que

O quadrado de qualquer número real é maior do que ou igual a 0

podemos escrever

$$(\forall x \in \mathbb{R})\,(x^2 \geq 0) \quad \text{ou simplesmente} \quad \forall x\,(x^2 \geq 0).$$

Se quisermos representar formalmente a afirmação

Existe um conjunto de números naturais que não contém o 4 $(*)$

podemos começar por escrever

$\exists S(S$ é um subconjunto de $\mathbb{N}$ e $\neg S(4))$.

Podemos continuar com a formalização pois uma afirmação do tipo "A é um subconjunto de B" pode ser formalizada como $\forall x(A(x) \rightarrow B(x))$. Então podemos formalizar a afirmação $(*)$ na forma

$$\exists S(\forall x(S(x) \rightarrow \mathbb{N}(x)) \wedge \neg S(4)).$$

Note que a ordem dos quantificadores pode ser decisiva. Por exemplo, trocando a ordem em

$$(\forall m \in \mathbb{N})(\exists n \in \mathbb{N})(n > m)$$

(a afirmação de que não existe nenhum número natural máximo – uma afirmação verdadeira) obtemos

$$(\exists n \in \mathbb{N})(\forall m \in \mathbb{N})(n > m),$$

uma afirmação muito diferente: existe um número natural que é maior que todos os naturais – uma afirmação claramente falsa!

As seguintes equivalências (óbvias!) são muito úteis quando queremos negar um quantificador e escrever a afirmação na positiva:

$$\neg[\exists x\, P(x)] \equiv \forall x\,[\neg P(x)], \quad \neg[\forall x\, P(x)] \equiv \exists x\,[\neg P(x)]. \qquad (**)$$

Para aprendermos um pouco melhor a raciocinar com quantificadores e predicados utilizaremos o `Tarski World`[3], com a qual podemos construir mundos tridimensionais habitados por blocos geométricos de diversos tipos e tamanhos, e usar uma linguagem de primeira ordem muito simples para descrever esses mundos e testar o valor lógico (verdadeiro ou falso) de sentenças de primeira ordem elaboradas sobre esses mundos.

Um mundo de Tarski (tridimensional) consiste num tabuleiro de xadrez (8×8) juntamente com figuras geométricas diversas (tetraedros, cubos e octaedros, de três tamanhos diferentes) dispostas nas casas do tabuleiro. Relativamente a estas figuras consideram-se os seguintes predicados unários, binários e ternários, que descrevem o tamanho, o tamanho relativo e a posição relativa das figuras no mundo:

[3]Consulte o apêndice Usando Tarksi.

Predicados unários

Sentença atómica	Interpretação
$Tet(a)$	a é um tetraedro
$Cube(a)$	a é um cubo
$Dodec(a)$	a é um dodecaedro
$Small(a)$	a é pequeno
$Medium(a)$	a é médio
$Large(a)$	a é grande

Predicados binários

Sentença atómica	Interpretação
$SameSize(a, b)$	a tem o mesmo tamanho que b
$SameShape(a, b)$	a tem a mesma forma que b
$Larger(a, b)$	a é maior que b
$Smaller(a, b)$	a é menor que b
$SameCol(a, b)$	a está na mesma coluna que b
$SameRow(a, b)$	a está na mesma linha que b
$Adjoins(a, b)$	a e b estão localizados em casas adjacentes (mas não na diagonal)
$LeftOf(a, b)$	a está numa coluna à esquerda de b
$RightOf(a, b)$	a está numa coluna à direita de b
$FrontOf(a, b)$	a está numa linha à frente de b
$BackOf(a, b)$	a está numa linha atrás de b

Predicados ternários

Sentença atómica	Interpretação
$Between(a, b, c)$	a, b e c estão na mesma coluna, linha ou diagonal, e a está entre b e c.

Exemplo. Se quisermos negar a afirmação

Todos os cubos são grandes,

ou seja,

$$\forall x\,(Cube(x) \to Large(x)),$$

usando $(**)$ obtemos

$$\exists x\,\neg(Cube(x) \to Large(x)),$$

que é ainda equivalente a

$$\exists x\,(Cube(x) \wedge \neg Large(x)).$$

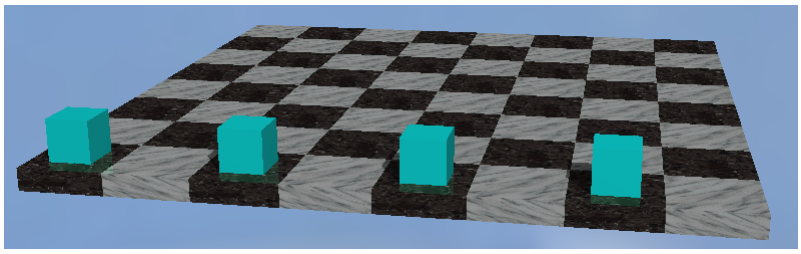
Por palavras, *existe um cubo que não é grande.*

Teste. Mostre que a negação da fórmula $\forall x\ \forall y\ (\neg SameShape(y, x) \vee Tet(y) \vee Cube(x))$ é equivalente a $\exists x\,\exists y\,(Dodec(x) \wedge Dodec(y))$.

Exemplo. Qual é o valor lógico das fórmulas

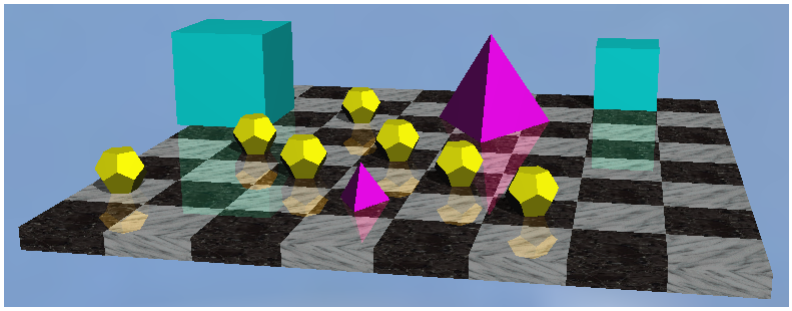
$$\forall x\,\forall y\,[(Cube(x)\wedge Cube(y)) \rightarrow (LeftOf(x,y)\vee RightOf(x,y))] \quad \text{e} \quad \exists x\,\exists y\,(Cube(x)\wedge Cube(y))$$

no seguinte mundo?:



Note que uma afirmação do tipo $\forall x\,\forall y...$ será verdadeira quando for verdadeira para todos os pares de objectos x e y no mundo (logo, em particular, quando $y = x$). Portanto, a primeira fórmula é falsa porque falha no caso $y = x$ (é impossível um objecto estar à esquerda ou à direita de si próprio). Quanto à segunda é claramente verdadeira (há diversos pares de objectos (x, y) no mundo que a satisfazem) mas observe que continua a ser verdadeira mesmo quando o mundo só contém um cubo (nesse caso, tome para x e y esse cubo).

Porque é que no mundo



as fórmulas

$$\forall x\,\forall y\,((Cube(x)\wedge Cube(y)) \rightarrow \neg SameRow(x,y))$$

e

$$\forall x\,\forall y\,[(Tet(x)\wedge Tet(y)) \rightarrow \neg SameSize(x,y)]$$

são falsas?

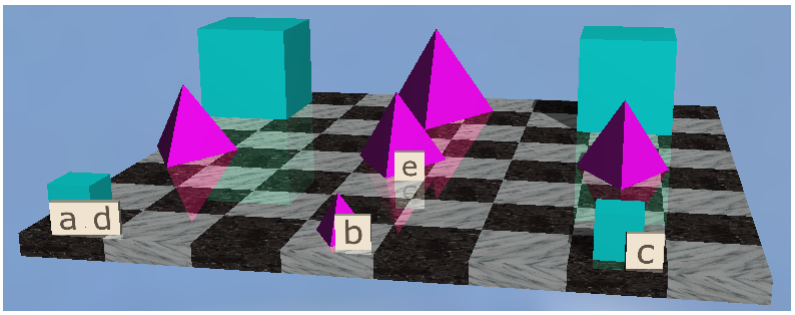
Teste.

1. Pode a fórmula $\exists x\,\forall y\,Smaller(x,y)$ ser verdadeira nalgum mundo?
2. Tente exprimir a fórmula $\exists y\,\forall x\,(Dodec(x) \rightarrow Smaller(x,y))$ em Português, de modo claro e sem ambiguidades. Não é fácil, pois não?

3. Qual é a diferença entre as fórmulas $\forall x\,((Cube(x) \wedge Medium(x)) \rightarrow \neg\exists y\, BackOf(y,x))$ e $\forall x\,((Cube(x) \wedge Medium(x)) \rightarrow \exists y\, \neg BackOf(y,x))$? (Não afirmam a mesma coisa.)

4. As fórmulas $\forall x\,(Tet(x) \rightarrow \exists y\,\exists z\, Between(x,y,z))$ e $\exists y\,\forall x\,(Tet(x) \rightarrow \exists z\, Between(x,y,z))$ significam coisas muito diferentes. O quê, precisamente?

Exemplo. No mundo



a fórmula

$$\forall x\,\forall y\,((Tet(x) \wedge Small(x) \wedge Tet(y) \wedge Small(y)) \rightarrow x = y)$$

é verdadeira (porquê? percebe o que a fórmula afirma?) e

$$\forall x\,\forall y\,((Dodec(x) \wedge Small(x) \wedge Dodec(y) \wedge Small(y)) \rightarrow x = y)$$

também é verdadeira (porquê?). Por outro lado, $\forall x\,(Dodec(x) \rightarrow x = b)$ parece afirmar uma patetice mas é verdadeira, enquanto $\forall x\,(Dodec(x) \leftrightarrow x = b)$ é falsa (sob que condições seria verdadeira?). Claro que $\forall x\,((Tet(x) \wedge Small(x)) \leftrightarrow x = b)$ já é verdadeira.

Exemplo. As fórmulas

$$\exists x\,\exists y\,(Tet(x) \wedge Larger(x,y))$$

e

$$\exists x\,(Tet(x) \wedge \exists y\, Larger(x,y))$$

afirmam a mesma coisa de modos diferentes. As fórmulas

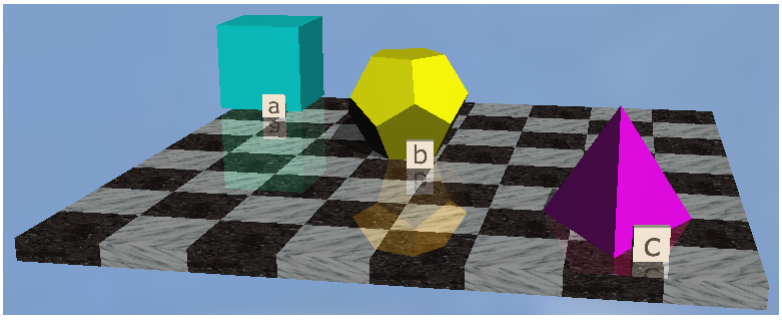
$$\exists x\,\exists y\,(Cube(x) \wedge Tet(y) \wedge Larger(x,y))$$

e

$$\exists x\,(Cube(x) \wedge \exists y\,(Tet(y) \wedge Larger(x,y)))$$

também afirmam a mesma coisa de modos diferentes.

Teste. Quantos objectos teremos que acrescentar ao mundo



para que a fórmula $\exists x \, \exists y \, (Tet(x) \wedge Tet(y) \wedge Larger(x, y) \wedge BackOf(x, y))$ seja verdadeira?

Apêndice: linguagens formais de primeira ordem e de ordem superior. Costuma-se classificar o cálculo proposicional como uma *lógica de ordem zero*. Porquê? Porque não permite que os conjuntos sejam quantificados e que sejam elementos de outros conjuntos. O cálculo de predicados já é uma lógica de ordem superior: permite que os conjuntos sejam quantificados e que sejam elementos de outros conjuntos. De que ordem?

Diz-se que um predicado tem *ordem* 1 se todos os seus argumentos são termos (isto é, constantes, variáveis individuais ou valores de funções). Caso contrário, diz-se que tem *ordem* $n+1$, onde n é a maior ordem entre os seus argumentos que não são termos. Por exemplo, em $S(x) \wedge T(S)$, o predicado S tem ordem 1 e o predicado T tem ordem 2. Em $p(f(x)) \wedge q(f)$, p tem ordem 1, f tem ordem 1 e q tem ordem 2.

Uma *lógica de ordem n* é uma lógica em cujas fbf's todos os seus predicados têm ordem $\leq n$.

Tal como na linguagem Tarski ou na lógica de predicados, uma linguagem de primeira ordem (LPO) serve para descrever mundos da seguinte maneira:

- Cada nome deve designar um objecto.
- Nenhum nome pode designar mais do que um objecto.
- Um objecto pode ter vários nomes, mas também pode não ter nome.

As constantes são símbolos referentes a objectos previamente fixados.

Língua Portuguesa	LPO
nome	constante (designação, termo)

Os *símbolos de predicado* ou *relacionais* são símbolos que designam propriedades dos objectos ou relações entre objectos.

Por exemplo, podemos usar o símbolo EEI, um símbolo de predicado unário (ou seja, de aridade um), para designar, no universo dos alunos da FCTUC, `ser Estudante de Engenharia Informática`. Outro exemplo: o símbolo $<$, um símbolo de predicado binário (ou seja, de aridade dois), representa, no universo dos números reais, `ser menor do que`.

Língua Portuguesa	LPO
O Pedro é estudante de Eng. Inf.	$EEI(Pedro)$
2 é menor que 1	$2 < 1$

Portanto, numa linguagem de primeira ordem, a cada símbolo de predicado está associado exactamente um número natural — o número de argumentos que ocorre no predicado, que se designa por *aridade*. Além disso, cada símbolo de predicado ou relacional é interpretado por uma propriedade bem determinada ou uma relação com a mesma aridade que o símbolo.

Uma *sentença atómica* é uma sequência finita de símbolos, escolhidos entre as constantes, os símbolos de predicados, os parênteses "(" e ")" e a vírgula, da forma

$$P(c_1), \quad T(c_1, c_2), \quad R(c_1, c_2, c_3)$$

onde c_1, c_2, c_3 são constantes e P, T, R são símbolos de predicados num vocabulário fixado.

Exemplos.

Língua Portuguesa	LPO
A Rita é estudante de Matemática	$M(Rita)$
O Nuno é mais velho do que o Pedro	$MaisVelho(Nuno, Pedro)$

A notação usual é a *prefixa*: o símbolo de predicado escreve-se à esquerda. Excepções: com o símbolo de igualdade $=$ utiliza-se a notação habitual $a = b$; com os símbolos $<, >$ também se utiliza a notação habitual: $1 < 2$, $2 > 1$.

Portanto, com algumas excepções (predicados $=, <, >$), numa linguagem de primeira ordem as sentenças atómicas são expressões que se obtêm escrevendo um símbolo de predicado de aridade n, seguido de n constantes, delimitadas por parênteses e separadas por vírgulas: $P(c_1, c_2, \ldots, c_n)$. As excepções $(=, <, >)$ podem ser estendidas a outros símbolos. Note que a ordem em que as constantes ocorrem é fundamental.

Especifica-se uma linguagem de primeira ordem fixando as constantes, os símbolos de predicado e os símbolos funcionais. Cada símbolo de predicado e cada símbolo funcional tem uma aridade bem determinada. Uma linguagem de primeira ordem pode não incluir símbolos funcionais, mas necessita sempre de símbolos relacionais. No entanto, em vários exemplos, o único símbolo relacional considerado é o da igualdade $=$.

As linguagens de primeira ordem podem assim distinguir-se entre si através das respectivas constantes, símbolos de predicado e símbolos funcionais. Partilham os conectivos $\neg, \wedge, \vee, \rightarrow$ e $\leftrightarrow$ e os quantificadores $\forall, \exists$.

Quando se traduz uma frase escrita em Português para uma sentença numa linguagem de primeira ordem, tem-se em geral uma linguagem previamente definida, em que se conhecem à partida as constantes, os símbolos relacionais e (caso existam) os símbolos funcionais. No entanto, há situações em que há que decidir quais as constantes, os símbolos relacionais e (caso existam) os símbolos funcionais adequados para expressar o que se pretende.

Exemplo. Consideremos a frase O Nuno explicou o Tarski World ao Pedro.

(1) Tomando o símbolo de predicado binário $ExplicouTarskiWorld$ podemos escrever

$$ExplicouTarskiWorld(Nuno, Pedro).$$

(2) Tomando o símbolo de predicado ternário $Explicou$ podemos escrever

$$Explicou(Nuno, TarskiWorld, Pedro).$$

O poder expressivo da linguagem (2) é maior do que o da linguagem (1). De facto, considerando a frase A Rita explicou o Boole ao Miguel, esta pode ser traduzida usando o símbolo de predicado ternário $Explicou$ — teríamos $Explicou(Rita, Boole, Miguel)$ — mas não pode ser traduzida usando o símbolo de predicado $ExplicouTarskiWorld$. O símbolo de predicado $Explicou$ é mais versátil do que os símbolos de predicado $ExplicouTarskiWorld$ ou $ExplicouBoole$.

Para considerar as frases A Rita explicou o Boole ao Miguel no sábado e No domingo, o Miguel explicou o Boole ao João podemos considerar um predicado quaternário $Explicou(x, y, z, w)$ — que se lê "x explicou y a z no w" — e traduzir as duas frases consideradas para LPO:

$$Explicou(Rita, Boole, Miguel, sábado)$$

$$Explicou(Miguel, Boole, João, domingo).$$

Os *símbolos funcionais* são símbolos que permitem obter outras designações para objectos.

Exemplos. (1) Jorge é pai do Nuno. Supondo que a afirmação é verdadeira, $Jorge$ e $pai(Nuno)$ são duas designações diferentes para o mesmo indivíduo; *pai* é um *símbolo funcional unário.*

(2) As expressões 3 e $((1 + 1) + 1)$ são duas designações diferentes do mesmo número natural; $+$ é um *símbolo funcional binário.*

As expressões 1, $(1 + 1)$ e $((1 + 1) + 1)$ são termos. A definição de *termo* é a seguinte:

- Constantes são termos.
- Se F é um símbolo funcional de aridade n e $t_1, t_2, \ldots, t_n$ são n termos, então a expressão $F(t_1, t_2, \ldots, t_n)$ é um termo.

Numa linguagem de primeira ordem com símbolos funcionais

- Termos complexos obtêm-se colocando um símbolo funcional n-ário antes de um n-tuplo de n termos. Excepção: certos símbolos funcionais binários escrevem-se entre termos, como por exemplo $+$ (por exemplo, $(1 + 1)$).
- Termos usam-se como nomes ou designações na formação de sentenças atómicas.

Exemplos de linguagens de primeira ordem usadas em Matemática.
(1) `A Linguagem de Primeira Ordem da Teoria de Conjuntos`. Na linguagem de primeira ordem da Teoria de Conjuntos tem-se apenas dois símbolos de predicados, ambos binários, $=$ e $\in$. As sentenças atómicas nesta linguagem são da forma $a = b$ (lê-se "a é igual a b") e $a \in b$ (lê-se "(o elemento) a pertence ao (conjunto) b"), sendo a e b constantes individuais.

Por exemplo, supondo que a designa 2, b designa o conjunto $\mathbb{N}$ dos números naturais e c designa o conjunto dos números ímpares, tem-se:

$a \in a$	sentença falsa
$a \in b$	sentença verdadeira
$a \in c$	sentença falsa
$b = c$	sentença falsa.

(2) `A Linguagem de Primeira Ordem da Aritmética`. A linguagem de primeira ordem da Aritmética contém duas constantes 0 e 1, dois símbolos relacionais binários $=$ e $<$ e dois símbolos funcionais binários $+$ e $\times$. São termos desta linguagem $0, 1, (1+1), ((1+1)+1), (0\times(1+1)), \ldots$

Os termos na aritmética de primeira ordem formam-se segundo as regras:

- As constantes 0, 1 são termos.
- Se t_1 e t_2 são termos, também são termos as expressões $(t_1 + t_2)$ e $(t_1 \times t_2)$.
- São termos apenas as expressões que possam ser obtidas por aplicação sucessiva dos passos anteriores um número finito de vezes.

As sentenças atómicas na aritmética de primeira ordem são as expressões que se podem escrever usando os termos (no lugar das constantes) e os símbolos relacionais $=, <$:

- Se t_1 e t_2 são termos, são sentenças atómicas as expressões $t_1 = t_2$ e $t_1 < t_2$.

Leituras suplementares:

- James Hein, *Discrete Structures, Logic and Computability*, Capítulos 6, 7, 8.

1.3. Raciocínio matemático, indução e recursão

Para entendermos um texto matemático temos que compreender o que faz com que um argumento esteja matematicamente correcto, isto é, seja uma prova. Uma *prova* é uma demonstração de que alguma afirmação é verdadeira. Normalmente apresentamos provas escrevendo frases em Português misturadas com equações e símbolos matemáticos.

Quando é que um argumento matemático está correcto?

Um *teorema* é uma afirmação que se pode demonstrar ser verdadeira. Demonstra-se que um teorema é verdadeiro com uma sequência de afirmações que formam um argumento, chamada *prova*. Para construir provas, precisamos de métodos que nos permitam deduzir novas afirmações a partir de afirmações já comprovadas. As afirmações usadas numa prova incluem os *axiomas* ou *postulados* da teoria, as hipóteses do teorema a provar e teoremas previamente provados. As *regras de inferência* são as ferramentas para deduzir novas conclusões e ligam os diversos passos da prova. Recorde de 1.1 que um argumento do tipo

$$\begin{array}{l} A_1 \\ A_2 \\ \vdots \\ \underline{A_n} \\ \therefore B \end{array}$$

é uma regra de inferência se $A_1 \wedge A_2 \wedge \cdots \wedge A_n \rightarrow B$ for uma tautologia. Por exemplo, é a tautologia $(p \wedge (p \rightarrow q)) \rightarrow q$ que está por trás da regra *modus ponens*, como vimos na primeira secção. Vimos na altura, também, uma lista das regras de inferência mais usadas no raciocínio matemático:

Regras de inferência	**Tautologia**	**Nome**
$\begin{array}{l} p \\ \underline{p \rightarrow q} \\ \therefore q \end{array}$	$(p \wedge (p \rightarrow q)) \rightarrow q$	Modus ponens (MP)
$\begin{array}{l} \neg q \\ \underline{p \rightarrow q} \\ \therefore \neg p \end{array}$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$	Modus tollens (MT)
$\begin{array}{l} \underline{p} \\ \therefore p \vee q \end{array}$	$p \rightarrow (p \vee q)$	Adição (Ad)
$\begin{array}{l} \underline{p \wedge q} \\ \therefore p \end{array}$	$(p \wedge q) \rightarrow p$	Simplificação (S)
$\begin{array}{l} p \vee q \\ \underline{\neg p} \\ \therefore q \end{array}$	$((p \vee q) \wedge \neg p) \rightarrow q$	Silogismo disjuntivo (SD)
$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \\ \therefore p \rightarrow r \end{array}$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	Silogismo hipotético (SH)

Qualquer argumento elaborado com regras de inferência diz-se *válido*. Quando todas as afirmações usadas num argumento válido são verdadeiras, podemos ter a certeza de chegar a uma conclusão correcta. No entanto, um argumento válido pode conduzir a conclusões incorrectas se uma ou mais proposições falsas são usadas no argumento. Por exemplo,

> "*Se* 101 *é divisível por* 3 *então* 101^2 *é divisível por* 9. 101 *é divisível por* 3. *Logo,* 101^2 *é divisível por* 9."

é um argumento válido (baseado na regra MP) mas a conclusão é falsa: 9 não divide $101^2 = 10\,201$.

Noutro tipo de falácias muito comum as conclusões estão incorrectas porque os argumentos não são válidos: apesar de aparentarem ser regras de inferência, na realidade não o são (baseiam-se em contingências e não em tautologias).

Exemplo 1. A proposição $((p \to q) \wedge q) \to p$ não é uma tautologia (é falsa quando p é falsa e q é verdadeira). No entanto, por vezes é usada como se fosse uma tautologia (este tipo de argumento incorrecto chama-se *falácia de afirmar a conclusão*):

> *Se resolver todos os exercícios destes apontamentos, então aprenderá matemática discreta. Aprendeu matemática discreta. Logo, resolveu todos os exercícios.*

(É claro que pode aprender matemática discreta sem precisar de resolver todos os exercícios destes apontamentos!)

Exemplo 2. A proposição $((p \to q \wedge \neg p) \to \neg q$ não é uma tautologia, pois é falsa quando p é falsa e q é verdadeira. É outro exemplo de proposição que por vezes é usada como regra de inferência em argumentos incorrectos (a chamada *falácia de negar a hipótese*):

> *Se resolver todos os exercícios destes apontamentos, então aprenderá matemática discreta. Não resolveu todos os exercícios. Logo, não aprendeu matemática discreta.*

(É claro que pode aprender matemática discreta sem ter resolvido todos os exercícios destes apontamentos!)

Que métodos podemos usar para elaborar argumentos matemáticos correctos?

Os matemáticos usam diversos métodos para provar a validade de uma proposição ou argumento. Façamos uma breve digressão, com exemplos, pelos mais comuns.

(1) Verificação exaustiva. Algumas proposições podem ser provadas por verificação exaustiva de um número finito de casos.

Exemplo 1. *Existe um número primo entre 890 e 910.*

Prova. Verificando exaustivamente descobrirá que o 907 é primo. □

Exemplo 2. *Cada um dos números 288, 198 e 387 é divisível por 9.*

Prova. Verifique que 9 divide cada um desses números. □

É claro que uma proposição enunciada para um número infinito de casos não poderá ser provada directamente por verificação exaustiva (por mais casos que consigamos verificar nunca conseguiremos verificar todos...). Por exemplo, se tentarmos comprovar, com a ajuda do computador, a famosa **conjectura de Goldbach**, que afirma que

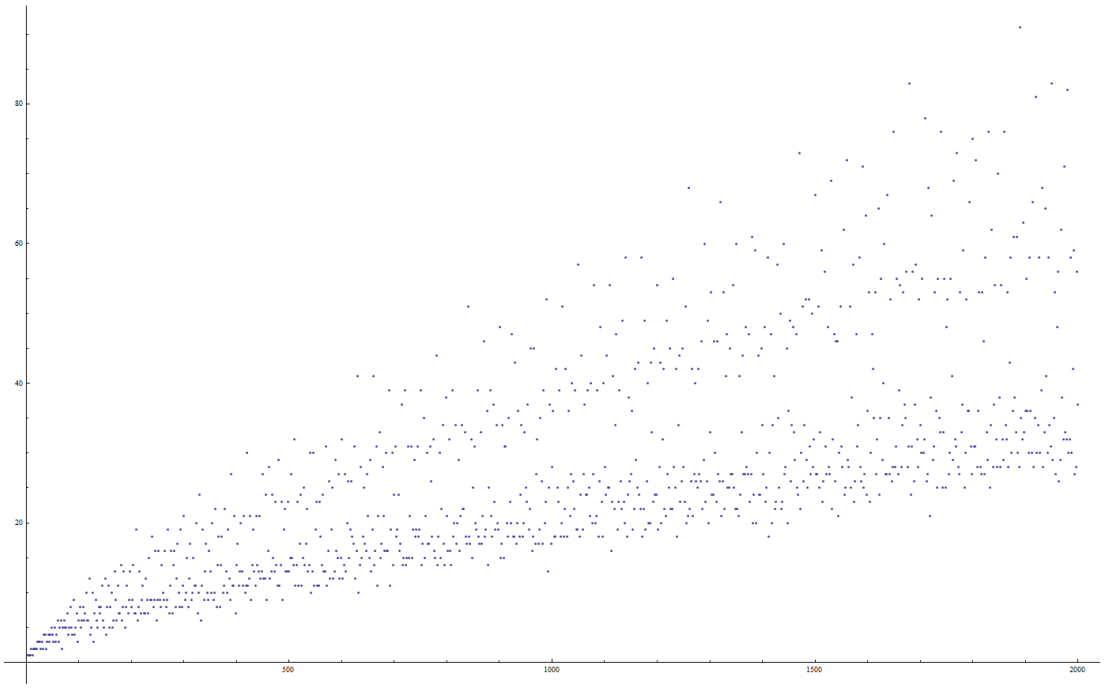
qualquer inteiro par maior do que 2 pode escrever-se como soma de dois primos

não encontraremos nenhum *contra-exemplo* (isto é, um exemplo que refute a conjectura). Pelo contrário, à medida que o inteiro par cresce, mais soluções vamos encontrando para a partição do inteiro em dois primos.

Verificação da conjectura para os inteiros pares entre 4 e 50:

```
4 pode ser expresso como 2+2
6 pode ser expresso como 3+3
8 pode ser expresso como 3+5
10 pode ser expresso como 3+7 ou 5+5
12 pode ser expresso como 5+7
14 pode ser expresso como 3+11 ou 7+7
16 pode ser expresso como 3+13 ou 5+11
18 pode ser expresso como 5+13 ou 7+11
20 pode ser expresso como 3+17 ou 7+13
22 pode ser expresso como 3+19 ou 5+17 ou 11+11
24 pode ser expresso como 5+19 ou 7+17 ou 11+13
26 pode ser expresso como 3+23 ou 7+19 ou 13+13
28 pode ser expresso como 5+23 ou 11+17
30 pode ser expresso como 7+23 ou 11+19 ou 13+17
32 pode ser expresso como 3+29 ou 13+19
34 pode ser expresso como 3+31 ou 5+29 ou 11+23 ou 17+17
36 pode ser expresso como 5+31 ou 7+29 ou 13+23 ou 17+19
38 pode ser expresso como 7+31 ou 19+19
40 pode ser expresso como 3+37 ou 11+29 ou 17+23
42 pode ser expresso como 5+37 ou 11+31 ou 13+29 ou 19+23
44 pode ser expresso como 3+41 ou 7+37 ou 13+31
46 pode ser expresso como 3+43 ou 5+41 ou 17+29 ou 23+23
48 pode ser expresso como 5+43 ou 7+41 ou 11+37 ou 17+31 ou 19+29
50 pode ser expresso como 3+47 ou 7+43 ou 13+37 ou 19+31
```

A seguinte figura, que mostra o *cometa de Goldbach* até ao número 2000 (isto é, o número de maneiras possíveis de escrever o inteiro par n como soma de 2 primos, para n de 4 até 2000) reforça isso mesmo:



Mesmo assim, não poderemos garantir que a conjectura é verdadeira para qualquer inteiro par maior do que 2 (tal como ninguém o conseguiu fazer até hoje!). De facto, verificar que uma dada proposição sobre os inteiros n é válida para muitos valores de n, não significa automaticamente que a proposição seja verdadeira para todo o n.

A *conjectura de Polya* é um exemplo relevante disso mesmo:

Um número natural n diz-se de *tipo par* caso a sua factorização em primos tenha um número par de factores, caso contrário diz-se de *tipo ímpar*. Por exemplo, os números 4 e 24 são de tipo par (pois $4 = 2 \times 2$ e $24 = 2 \times 2 \times 2 \times 3$) enquanto 30 e 18 são de tipo ímpar (pois $30 = 2 \times 3 \times 5$ e $18 = 2 \times 3 \times 3$). Seja $P(n)$ o número de naturais $\leq n$ de tipo par e seja $I(n)$ o número de naturais $\leq n$ de tipo ímpar. Se verificarmos para alguns valores de $n \geq 2$ constataremos sempre que

$$I(n) \geq P(n).$$

Em 1919, o famoso matemático Polya conjecturou a possibilidade desta proposição ser verdadeira. Depois de verificada para todo o n inferior a 1 milhão, mais matemáticos se convenceram dessa possibilidade. No entanto, em 1962, R. Sherman Lehman encontrou um contra-exemplo: para $n = 906\,180\,359$, tem-se $I(n) = P(n) - 1$. O contra-exemplo mais pequeno foi entretanto encontrado por Minoru Tanaka em 1980: é o número $n = 906\,150\,257$. Portanto, para qualquer inteiro n no intervalo $[2, 906\,150\,256]$ tem-se de facto $I(n) \geq P(n)$.

Mais adiante, estudaremos o método de indução matemática que nos permite garantir a validade de muitas afirmações para uma lista infinita de inteiros sem grande dificuldade.

(2) Prova de implicações (condicionais). A maioria dos teoremas que se provam em matemática são implicações (ou equivalências, que são conjunções de duas implicações). A im-

plicação ("se p então q" ou "p implica q") é uma afirmação *condicional* com *hipótese* p e *conclusão* q. A sua *contraposta* é a afirmação "se não q então não p" e a sua *recíproca* é "se q então p".

(2a) Prova directa. Como a maioria dos teoremas utilizados na prática da matemática são implicações, as técnicas para provar implicações são muito importantes. Para provar que $p \Rightarrow q$, ou seja, que $p \to q$ é uma tautologia, mostra-se que se p é verdadeira também q o é: começamos por assumir que a hipótese p é verdadeira; depois tentamos encontrar uma proposição que resulte da hipótese e/ou factos conhecidos; continuamos deste modo até chegarmos à conclusão q.

Exemplo 3. *Se m é ímpar e n é par então $m - n$ é ímpar.*

Prova. Suponhamos que m é ímpar e n é par. Então $m = 2k + 1$ e $n = 2l$ para alguns inteiros k e l. Portanto, $m - n = 2k + 1 - 2l = 2(k - l) + 1$, que é um inteiro ímpar. □

Exemplo 4. *Se n é ímpar então n^2 é ímpar.*

Prova. Suponhamos que n é ímpar. Então $n = 2k + 1$ para algum inteiro k. Portanto, $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$, que é um inteiro ímpar pois $2k^2 + 2k$ é um inteiro. □

(2b) Prova indirecta. A seguinte tabela de verdade mostra que uma condicional e a sua contraposta são equivalentes:

p	q	$\neg q$	$\neg p$	$p \to q$	$\neg q \to \neg p$
V	V	F	F	V	V
V	F	V	F	F	F
F	V	F	V	V	V
F	F	V	V	V	V

Esta equivalência proporciona um método alternativo para provar uma implicação (o chamado *método indirecto*).

Exemplo 5. *Se n^2 é par então n é par.*

Prova. A contraposta desta afirmação é "se n é ímpar então n^2 é ímpar", que é verdadeira pelo Exemplo anterior. □

Teste (1 minuto). Prove as recíprocas dos Exemplo 4 e 5.

(2c) Prova por contradição (redução ao absurdo). Da tabela de verdade da implicação decorre imediatamente que "se p então q" é equivalente a "p e não q implica falso":

p	q	$p \to q$	$p \wedge \neg q$	$p \wedge \neg q \to \mathsf{F}$
V	V	V	F	V
V	F	F	V	F
F	V	V	F	V
F	F	V	F	V

Portanto temos aqui mais um método alternativo de demonstrar uma implicação: para provar "se p então q" é suficiente provar "p e não q implica falso", ou seja, assumir p e $\neg q$ e depois argumentar de modo a chegar a uma contradição (proposição sempre falsa, como vimos em 1.1). Chama-se a esta técnica de demonstração *prova por contradição* (ou *por redução ao absurdo*).

Exemplo 6. *Se n^2 é ímpar então n é ímpar.*

Prova. Suponhamos, *por absurdo*, que n^2 é ímpar e n é par. Então $n = 2k$ para algum inteiro k pelo que $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$ é também um inteiro par. Chegamos assim à conclusão que n^2 é simultaneamente um inteiro par e ímpar, o que é uma contradição. □

Exemplo 7. *Se $2|5n$ então n é par.*

Prova. Suponhamos, *por absurdo*, que $2|5n$ e n é ímpar. Então $5n = 2d$ para algum inteiro d e $n = 2k + 1$ para algum inteiro k. Juntando tudo obtemos $2d = 5n = 5(2k + 1) = 10k + 5$. Logo $5 = 2d - 10k = 2(d - 5k)$, o que é uma contradição pois afirma que 5 é um número par! □

Exemplo 8. *$\sqrt{2}$ é um número irracional.*

Prova. Suponhamos, *por absurdo*, que $\sqrt{2}$ é racional. Então $\sqrt{2} = \frac{p}{q}$ para algum par de inteiros p e q (podemos assumir que a fracção p/q está já escrita na sua forma reduzida, isto é, $\text{mdc}(p, q) = 1$). Elevando ao quadrado ambos os membros da igualdade anterior obtemos $2q^2 = p^2$, pelo que p^2 é par. Então, pelo Exemplo 5, p é par. Sendo p par, é claro que p^2 é um múltiplo de 4, ou seja, $p^2 = 4k$ para algum inteiro k. Consequentemente, $2q^2 = 4k$, isto é, $q^2 = 2k$ é par, pelo que q também é par. Chegámos aqui a uma contradição: p e q são pares mas $\text{mdc}(p, q) = 1$. □

(3) Prova de equivalências ("se e só se"; abreviadamente "sse"). Uma proposição da forma $p \Leftrightarrow q$ ("p se e só se q") significa a conjunção de "p implica q" e "q implica p". Portanto, é preciso apresentar duas provas. Por vezes, estas provas podem ser escritas como uma só prova na forma "p sse r sse s sse ... sse q", onde cada "... sse ..." é conclusão evidente da informação anterior.

Exemplo 9. *n é par se e só se $n^2 - 2n + 1$ é ímpar.*
Prova.

n é par	sse	$n = 2k$ para algum inteiro k	(definição)
	sse	$n - 1 = 2k - 1$ para algum inteiro k	(álgebra)
	sse	$n - 1 = 2(k - 1) + 1$ para algum inteiro $k - 1$	(álgebra)
	sse	$n - 1$ é ímpar	(definição)
	sse	$(n - 1)^2$ é ímpar	(Exemplo 4 e Teste)
	sse	$n^2 - 2n + 1$ é ímpar	(álgebra) □

Muitas vezes um teorema afirma que determinadas proposições $p_1, p_2, \ldots, p_n$ são equivalentes, isto é, que $p_1 \leftrightarrow p_2 \leftrightarrow \cdots \leftrightarrow p_n$ é uma tautologia (o que assegura que as n proposições têm a mesma tabela de verdade). Uma maneira de provar o teorema é usar a tautologia

$$(p_1 \leftrightarrow p_2 \leftrightarrow \cdots \leftrightarrow p_n) \leftrightarrow ((p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \cdots \wedge (p_n \rightarrow p_1))$$

que assegura a equivalência

$$(p_1 \leftrightarrow p_2 \leftrightarrow \cdots \leftrightarrow p_n) \equiv ((p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \cdots \wedge (p_n \rightarrow p_1)).$$

Exemplo 10. *Para cada inteiro n as proposições seguintes são equivalentes:*

(i) *n é ímpar.*

(ii) *n^2 é ímpar.*

(iii) *$n^2 - 2n + 1$ é par.*

Prova. Para provar a equivalência das três asserções, basta provar que as implicações (i)→(ii), (ii)→(iii) e (iii)→(i) são verdadeiras:
(i)→(ii): Provada no Exemplo 4.
(ii)→(iii): Se n^2 é ímpar então $n^2 + 1$ é par. Como $2n$ é sempre par, então $n^2 - 2n + 1$ é par.
(iii)→(i): No Exemplo 6 provámos que se n é par então $n^2 - 2n + 1$ é ímpar. A implicação (iii)→(i) é a sua contraposta, pelo que também é verdadeira. □

(4) Prova de proposições com quantificadores. A maneira mais óbvia de provar uma afirmação de existência $\exists x P(x)$ é determinar um objecto particular a para o qual $P(a)$ é V. Por exemplo, para provar que existe um número irracional basta mostrar que $\sqrt{2}$ é irracional (como fizemos no Exemplo 8). Mas às vezes não é fácil ou possível determinarmos explicitamente esse objecto e temos então que adoptar uma estratégia menos directa, como o exemplo seguinte ilustra:

Exemplo 11. *Existem irracionais r, s tais que r^s é racional.*

Prova. Consideremos dois casos.
Caso 1. Se $\sqrt{2}^{\sqrt{2}}$ é racional, podemos tomar $r = s = \sqrt{2}$ e o resultado está provado.
Caso 2. Se $\sqrt{2}^{\sqrt{2}}$ é irracional, podemos considerar $r = \sqrt{2}^{\sqrt{2}}$ e $s = \sqrt{2}$, pois nesse caso

$$r^s = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^{\sqrt{2}\cdot\sqrt{2}} = (\sqrt{2})^2 = 2,$$

e o resultado está provado. □

Observe que nesta prova não sabemos qual das duas possibilidades se verifica pelo que não exibimos um par específico de irracionais r, s tais que r^s é racional. Limitámo-nos a mostrar que tal par existe. Esta prova é também um exemplo de uma *prova por casos*, outra técnica de demonstração muito útil.

Como poderemos provar uma afirmação universal $\forall x P(x)$? Uma possibilidade é tomarmos um x arbitrário e mostrar que satisfaz a propriedade P. Por exemplo:

Exemplo 12. $(\forall n \in \mathbb{N})(\exists m \in \mathbb{N})(m > n^2)$.

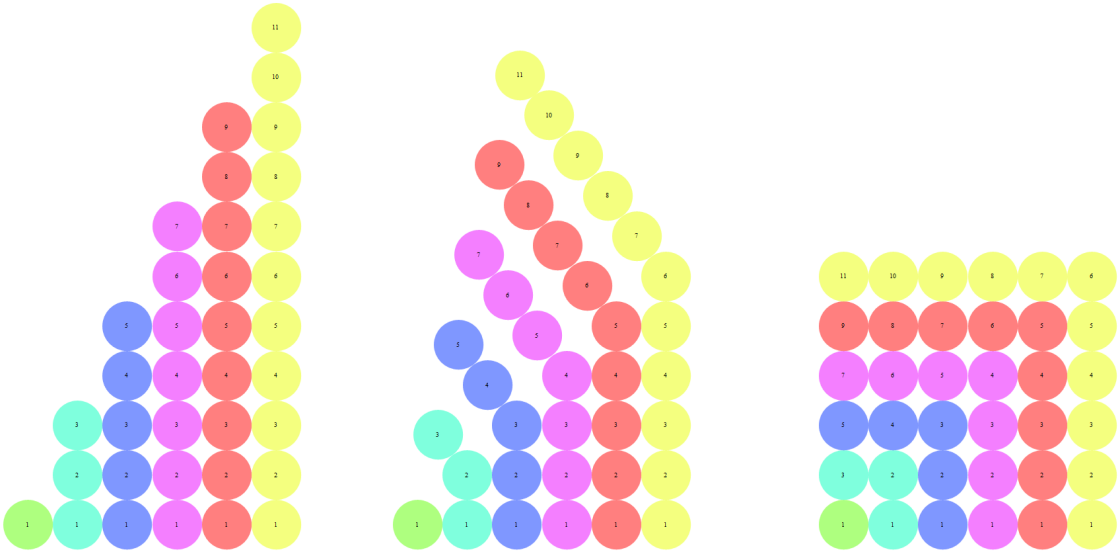
Prova. Seja n um número natural arbitrário. Então n^2 é um número natural e $m = n^2 + 1$ também. Como $m > n^2$, isto mostra que $(\exists m \in \mathbb{N})(m > n^2)$ é uma proposição verdadeira. □

Note que ao iniciarmos a prova dizendo "Seja n um número natural arbitrário", então usamos sempre o símbolo n ao longo da prova para representar esse número e assumimos que o seu valor permanece constante (não impondo no entanto nenhuma restrição a esse valor).

Proposições da forma $\forall x P(x)$ são às vezes provadas pelo método da contradição: assumindo $\neg \forall x P(x)$, obtemos um x tal que $\neg P(x)$ (porque $\neg \forall x P(x) \equiv \exists x \neg P(x)$). Temos assim uma maneira de começar a prova. A dificuldade pode estar em terminá-la, isto é, em obter uma contradição...

(5) Prova por indução matemática. A que é igual a soma dos n primeiros inteiros positivos ímpares? Para $n = 1, 2, 3, 4, 5, 6$ tem-se

$$\begin{aligned} &1 = 1, \\ &1 + 3 = 4, \\ &1 + 3 + 5 = 9, \\ &1 + 3 + 5 + 7 = 16, \\ &1 + 3 + 5 + 7 + 9 = 25. \end{aligned}$$

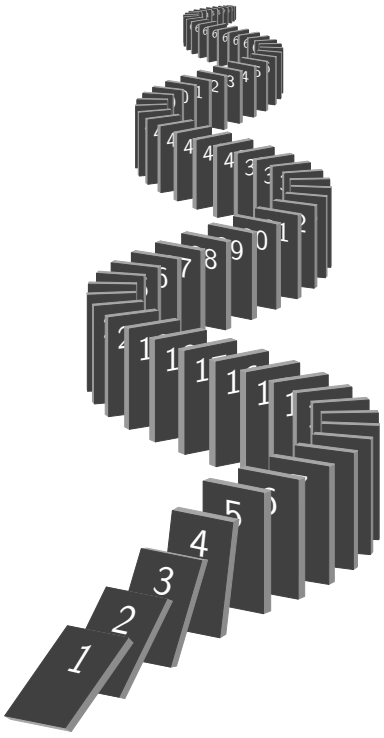


Destes valores particulares é razoável conjecturar[4] que a soma, para qualquer n, deverá ser igual a n^2. O *método de indução matemática* permite-nos provar facilmente que esta conjectura está correcta[5]. Trata-se de um método muito potente para provar asserções deste tipo, enunciadas sobre o conjunto $\mathbb{N}$ dos naturais. Baseia-se na observação óbvia que todo o subconjunto não vazio de $\mathbb{N}$ tem um elemento mínimo e no seguinte:

[4] Veja `www.mat.uc.pt/~picado/ediscretas/somatorios/Matematica_sem_palavras_files/soma_impares.html`.
[5] Assim como todas as outras em `www.mat.uc.pt/~picado/ediscretas/somatorios`.

Base do Princípio de Indução Matemática. *Seja $S \subseteq \mathbb{N}$, $1 \in S$ e suponhamos que $k \in S$ implica $k+1 \in S$. Então $S = \mathbb{N}$.*

Prova. Suponhamos, por absurdo, que $S \neq \mathbb{N}$. Então $\mathbb{N} \setminus S \neq \varnothing$ logo tem um elemento mínimo m. Como $1 \in S$ e $m \notin S$, então $m > 1$. Portanto, $m-1$ é um natural e pertence a S (pois m é o mínimo de $\mathbb{N} \setminus S$). Logo, por hipótese, $(m-1)+1 \in S$, isto é, $m \in S$. Chegamos assim a uma contradição: $m \notin S$ e $m \in S$. Em conclusão, $S = \mathbb{N}$. □



Princípio de Indução Matemática (PIM). *Seja $P(n)$, $n \in \mathbb{N}$, uma proposição. Para provar que $P(n)$ é verdadeira para qualquer $n \in \mathbb{N}$ basta:*

(1) (Passo inicial) *Mostrar que $P(1)$ é verdadeira.*

(2) (Passo indutivo) *Mostrar que a implicação $P(k) \rightarrow P(k+1)$ é verdadeira para qualquer $k \in \mathbb{N}$.*

Prova. Suponhamos que os dois passos foram provados. Seja $S = \{n \mid P(n) \text{ é verdadeira}\}$. O passo inicial garante que $1 \in S$; por outro lado, o passo indutivo garante que $k \in S$ implica $k+1 \in S$. Então, pela Base do PIM, podemos concluir que $S = \mathbb{N}$. □

No passo indutivo, $P(n)$ chama-se *hipótese de indução.*

Exemplo 13. *Para qualquer natural n, a soma dos n primeiros inteiros positivos ímpares é igual a n^2.*

Prova. Seja $P(n)$ a proposição

$$\underbrace{1+3+5+\cdots+(2n-1)}_{n \text{ parcelas}} = n^2.$$

$P(1)$ é claramente verdadeira: $1 = 1^2$. Assumindo que $P(k)$ é verdadeira, provemos que $P(k+1)$ é verdadeira (para qualquer $k \geq 1$). O membro esquerdo de $P(k+1)$ é:

$$\begin{aligned} 1+3+5+\cdots+(2k-1)+(2k+1) &= (1+3+5+\cdots+(2nk-1))+(2k+1) \\ &= k^2+(2k+1) && \text{(hip. indução)} \\ &= (k+1)^2 && \text{(álgebra)} \end{aligned}$$

que é o membro direito de $P(k+1)$. Portanto, $P(k+1)$ é verdadeira e, pelo PIM, segue que $P(n)$ é verdadeira para qualquer n. □

Exemplo 14. *Para qualquer natural n, a soma dos n primeiros inteiros positivos pares é igual a n^2+n.*

Prova. Seja $P(n)$ a identidade

$$\underbrace{2+4+6+\cdots+2n}_{n \text{ parcelas}} = n^2+n.$$

Queremos mostrar que $P(n)$ é V para qualquer $n \in \mathbb{N}$. Pelo método de indução matemática teremos que mostrar duas coisas:

(1) $P(1)$ é V: É óbvio, pois a identidade $P(1)$ resume-se a $2 = 1^2+1$.

(2) A implicação $P(k) \to P(k+1)$ é V para qualquer $k \geq 1$: Suponhamos que $P(k)$ é V, isto é,

$$2+4+6+\cdots+2k = k^2+k.$$

Então

$$2+4+6+\cdots+2k+(2k+2) = k^2+k+2k+2 = (k+1)^2+k+1,$$

o que mostra precisamente que $P(k+1)$ também é V. □

Exemplo 15. *Seja $f : \mathbb{N} \to \mathbb{N}$ definida por*

$$f(n) = \begin{cases} 1 & \text{se } n = 1 \\ f(n-1)+n^2 & \text{se } n \neq 1. \end{cases}$$

Então $f(n) = \dfrac{n(n+1)(2n+1)}{6}$ para qualquer $n \in \mathbb{N}$.

Prova. Seja $P(n)$ a proposição $f(n) = n(n+1)(2n+1)/6$. Como $f(1) = 1$ e $1(1+1)(2+1)/6 = 1$, então $P(1)$ é verdadeira. Assumindo que $P(k)$ é verdadeira, provemos que $P(k+1)$ é verdadeira.

O membro esquerdo de $P(k+1)$ é:

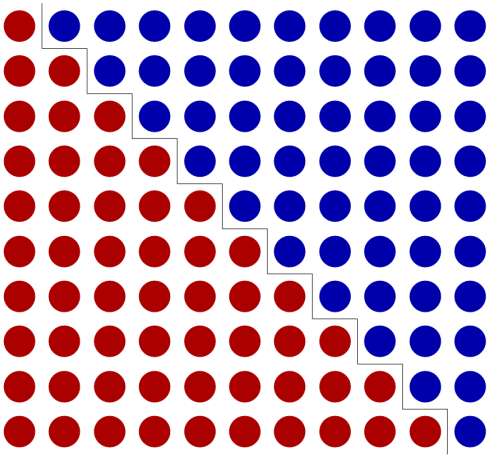
$$\begin{aligned}
f(k+1) &= f(k+1-1) + (k+1)^2 && \text{(definição de } f\text{)} \\
&= f(k) + (k+1)^2 && \text{(álgebra)} \\
&= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 && \text{(hip. indução)} \\
&= \frac{(k+1)(2k^2+7k+6)}{6} && \text{(álgebra)} \\
&= \frac{(k+1)(k+2)(2k+3)}{6} && \text{(álgebra)} \\
&= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6} && \text{(álgebra)}
\end{aligned}$$

que é o membro direito de $P(k+1)$. Portanto, $P(k+1)$ é verdadeira e, pelo PIM, segue que $P(n)$ é verdadeira para qualquer n. □

Teste. Usando indução matemática, prove as fórmulas:

(a) $1+2+3+\cdots+n = \frac{1}{2}n(n+1)$.

(b) $1+2+3+\cdots+(n-1)+n+(n-1)+\cdots+3+2+1 = n^2$.



$$1+2+3+4+5+6+7+8+9+10 = \frac{1}{2}10(10+1)$$

Exemplo 16. *Para qualquer natural* n, $\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^n} = 1 - \frac{1}{2^n}$.

Prova. Seja $P(n)$ a identidade

$$\underbrace{\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^n}}_{n \text{ parcelas}} = 1 - \frac{1}{2^n}.$$

Queremos mostrar que $P(n)$ é V para qualquer $n \in \mathbb{N}$. Pelo método de indução matemática teremos que mostrar duas coisas:

(1) $P(1)$ é V: É óbvio, pois a identidade $P(1)$ resume-se a $\frac{1}{2} = 1 - \frac{1}{2}$.
(2) A implicação $P(k) \to P(k+1)$ é V para qualquer $k \geq 1$: Suponhamos que $P(k)$ é V, isto é,

$$\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^k} = 1 - \frac{1}{2^k}.$$

Então

$$\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^k} + \frac{1}{2^{k+1}} = 1 - \frac{1}{2^k} + \frac{1}{2^{k+1}} = 1 - \left(\frac{1}{2^k} - \frac{1}{2^{k+1}}\right) = 1 - \frac{2-1}{2^{k+1}} = 1 - \frac{1}{2^{k+1}},$$

o que mostra precisamente que $P(k+1)$ também é V. □

Exemplo 17. *Para qualquer natural n,*

$$1 + 3 + 5 + 7 + \cdots + (2n-3) + (2n-1) + (2n-3) + \cdots + 7 + 5 + 3 + 1 = n^2 + (n-1)^2.$$

Prova. Seja $P(n)$ a identidade

$$1 + 3 + 5 + 7 + \cdots + (2n-3) + (2n-1) + (2n-3) + \cdots + 7 + 5 + 3 + 1 = n^2 + (n-1)^2.$$

Queremos mostrar que $P(n)$ é V para qualquer $n \in \mathbb{N}$. Pelo método de indução matemática teremos que mostrar duas coisas:
(1) $P(1)$ é V: É óbvio, pois a identidade $P(1)$ resume-se a $1 = 1^2 + 0^2$.
(2) A implicação $P(k) \to P(k+1)$ é V para qualquer $k \geq 1$: Suponhamos que $P(k)$ é V, isto é,

$$1 + 3 + 5 + 7 + \cdots + (2k-3) + (2k-1) + (2k-3) + \cdots + 7 + 5 + 3 + 1 = k^2 + (k-1)^2.$$

Então

$$\begin{aligned} 1 + 3 + 5 + \cdots + (2k-3) + (2k-1) + (2k+1) + (2k-1) + (2k-3) \cdots + 5 + 3 + 1 = \\ = k^2 + (k-1)^2 + (2k+1) + (2k-1) = k^2 + (k-1)^2 + 4k = k^2 + k^2 - 2k + 1 + 4k = \\ = k^2 + k^2 + 2k + 1 = (k+1)^2 + k^2 \end{aligned}$$

o que mostra precisamente que $P(k+1)$ também é V. □

É fácil adaptar a demonstração da Base do PIM para que este funcione também para proposições $P(n)$ onde $n \in \{a, a+1, \ldots\}$:

Princípio de Indução Matemática (PIM). *Seja $P(n)$, $n \in \{a, a+1, \ldots\}$, uma proposição. Para provar que $P(n)$ é verdadeira para qualquer $n \geq a$ basta:*

(1) (Passo inicial) *Mostrar que $P(a)$ é verdadeira.*

(2) (Passo indutivo) *Mostrar que a implicação $P(k) \to P(k+1)$ é verdadeira para qualquer $k \geq a$.*

Em algumas situações pode ser difícil definir um objecto de modo explícito e ser mais fácil defini-lo em função dele próprio. A este processo chama-se *recursão* e pode ser usado para definir sequências, sucessões e conjuntos. Por exemplo, a sequência das potências de 2 pode ser definida explicitamente por $a_n = 2^n$ para $n = 0, 1, 2, \ldots$, mas também pode ser definida recursivamente pelo primeiro termo $a_0 = 1$ e pela regra que permite definir um termo à custa dos anteriores: $a_{n+1} = 2a_n$ para $n = 0, 1, 2, \ldots$.

Portanto, podemos definir uma função sobre os inteiros não negativos

- especificando o valor da função em 0 e
- dando uma regra que permita calcular o seu valor num inteiro a partir dos valores em inteiros menores.

Tal definição chama-se uma *definição recursiva* ou *indutiva*.

Teste (1 minuto). Encontre uma definição recursiva da função factorial

$$f(n) = n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1.$$

As definições recursivas são também utilizadas frequentemente para definir conjuntos. Nesse caso, especifica-se um colecção inicial de elementos como pertencendo ao conjunto que se pretende definir, e depois especificam-se as regras de construção dos elementos do conjunto a partir de elementos que já se sabe estarem no conjunto. Foi o que fizemos, quando logo na Secção 1.1 definimos deste modo o conjunto das fórmulas bem formadas do cálculo proposicional. Os conjuntos definidos deste modo ficam bem definidos e os teoremas sobre eles podem ser provados usando a definição recursiva.

Exemplo. Seja S o conjunto definido recursivamente por

- $3 \in S$,
- Se $x \in S$ e $y \in S$ então $x + y \in S$.

Mostre que S é o conjunto dos inteiros positivos divisíveis por 3. (Assume-se implicitamente neste tipo de definições que um elemento só pertence a S se puder ser gerado usando as duas regras na definição de S.)

Prova. Seja C o conjunto de todos os inteiros positivos divisíveis por 3. Para provar a igualdade $C = S$ temos que verificar as inclusões $C \subseteq S$ e $S \subseteq C$.

$C \subseteq S$: Provemos por indução matemática que todo o inteiro positivo divisível por 3 pertence a S. Para isso seja $P(n)$ a proposição "$3n \in S$". O passo inicial $P(1)$ é verdadeiro pela primeira regra da definição recursiva. Para estabelecer o passo indutivo, assumimos que $P(n)$ é verdadeira, ou seja, que $3n \in S$. Mas então, como 3 também pertence a S, pela segunda regra da definição recursiva, $3n + 3 = 3(n + 1)$ também está em S.

$S \subseteq C$: Basta mostrar que as regras de definição de S só geram elementos que estão contidos em C. A primeira é evidente: $3 \in C$. Quanto à segunda, se $x, y \in S$ são divisíveis por 3 então $x + y$ também é divisível por 3, o que completa a prova. □

2. Algoritmos

2.1. Algoritmos e sua complexidade

Na matemática discreta abordamos muitos tipos de problemas. Em muitos deles, para chegarmos à solução, temos que seguir um procedimento que, num número finito de passos, conduz à tão desejada solução. A uma tal sequência chama-se algoritmo[6]. Um *algoritmo* é um procedimento para resolver um problema num número finito de passos.

Exemplo. O problema da determinação do maior elemento numa sequência finita de inteiros pode ser facilmente descrito por um algoritmo, formulado em português da seguinte maneira:

- Tome o *máximo temporário* igual ao primeiro inteiro da sequência. (O máximo temporário será o maior inteiro encontrado até ao momento, em cada passo do procedimento.)
- Compare o inteiro seguinte na sequência com o máximo temporário, e se for maior, tome o máximo temporário igual a esse inteiro.
- Repita o passo anterior se existirem mais inteiros na sequência.
- Pare quando chegar ao fim da sequência. O máximo temporário será então o maior inteiro da sequência.

Abreviadamente:

```
procedure max(a_1, a_2, ..., a_n : inteiros)
max := a_1
for i := 2 to n
if max < a_i then max := a_i
{max é o maior elemento}
```

É claro que um algoritmo pode ser formulado explicitamente numa qualquer linguagem de computação, mas nesse caso só poderemos utilizar expressões válidas dessa linguagem. Exemplifiquemos isso convertendo cada linha do algoritmo acima em código `Maple`[7]. Podemos considerar uma lista de números como um vector (matriz). Para usar esta funcionalidade no `Maple` temos primeiro que abrir a package de Álgebra Linear `linalg`:

[6] O termo *algoritmo* deriva do nome *al-Khowarizmi* de um matemático persa do século IX, cujo livro sobre numerais hindus esteve na base da notação decimal moderna que hoje utilizamos.

[7] Se estiver interessado no programa de cálculo simbólico `Maple` recomendamos a leitura do manual *Maple Experiments in Discrete Mathematics* de James Hein (`www.cs.pdx.edu/~jhein/books/MapleLabBook09.pdf`). Alternativamente, recomendamos, na biblioteca do DMUC, o livro *Exploring Discrete Mathematics with Maple* de Kenneth Rosen (o mesmo autor do manual indicado na Bibliografia do curso).

```
> with(linalg):
Warning, the protected names norm and trace have been redefined and unprotected
```

Continuando:

```
> with(linalg):
Warning, the protected names norm and trace have been redefined and unprotected

> Max := proc(t::array)
>   local max, max_temp;
>   max := t[1];
>   for max_temp from 1 to vectdim(t) do
>      if t[max_temp] > max then
>         max := t[max_temp]
>      fi;
>      od;
>      RETURN([max]);
> end:
```

Fica assim traduzido o algoritmo em `Maple`. Dando como input uma qualquer sequência t

```
> t := array(1..10, [1,20,45,3,2,10,99,98,45,32]);
```

$$t := [1, 20, 45, 3, 2, 10, 99, 98, 45, 32]$$

basta mandar calcular $Max(t)$:

```
> Max(t);
```

$$[99]$$

Em geral, os algoritmos têm características comuns:

- `Entrada (Input)`: conjunto de valores de entrada, definidos num determinado conjunto.
- `Saída (Output)`: a partir de cada conjunto de valores de entrada, um algoritmo produz valores de saída num determinado conjunto. Estes valores de saída contêm a solução do problema.
- `Precisão`: os passos do algoritmo têm que estar definidos com precisão.
- `Finitude`: o algoritmo deve produzir os valores de saída ao cabo de um número finito de passos.
- `Realizável`: deve ser possível realizar cada passo do algoritmo em tempo útil.
- `Generalidade:` o procedimento deve ser aplicável a todos os problemas da forma desejada, e não somente a um conjunto particular de valores de entrada.

Algoritmos recursivos. Encontramos muitas vezes algoritmos recursivos quando pretendemos resolver problemas discretos. Um algoritmo chama-se *recursivo* se resolve um problema reduzindo-o a uma instância do mesmo problema com input mais pequeno.

Uma definição recursiva exprime o valor de uma função num inteiro positivo em termos dos valores da função em inteiros mais pequenos. Isto significa que podemos ter sempre um algoritmo recursivo para calcular o valor de uma função definida por recursão. Por exemplo:

Teste. Especifique um algoritmo recursivo para calcular a potência a^n de um número real a para qualquer inteiro não negativo n.

Solução. A definição recursiva de a^n diz-nos que $a^{n+1} = a \times a^n$ e $a^0 = 1$ (condição inicial). Então podemos fazer:

```
procedure potencia(a : número real ≠ 0, n : inteiro não negativo)
if n = 0 then potencia(a, n) := 1
else potencia(a, n) := a * potencia(a, n − 1)
```

Também não é difícil especificar um algoritmo recursivo para o cálculo do máximo divisor comum de dois inteiros:

```
procedure mdc(a, b : inteiros não negativos com a < b)
if a = 0 then mdc(0, b) := b
else mdc(a, b) := mdc(b mod a, a)
```

Há outro modo de calcular a função potência $f(n) = a^n$ a partir da sua definição recursiva: em vez de reduzir sucessivamente o cálculo a inteiros mais pequenos, podemos começar com o valor da função em 1 e aplicar sucessivamente a definição recursiva para encontrar os valores da função em números sucessivamente maiores. Tal procedimento diz-se *iterativo*. Por outras palavras, para calcular a^n usando um processo iterativo, começamos em 1 e multiplicamos sucessivamente por cada inteiro positivo $\leq n$:

```
procedure potencia iterativa(a : número real ≠ 0, n : inteiro positivo)
x := 1
for i := 1 to n
   x := a * x
{x é a^n}
```

Apresentemos mais um exemplo. A famosa **conjectura de Collatz** (ou conjectura "$3x+1$"), que até hoje ninguém conseguiu provar (!), afirma que se pegarmos num inteiro x arbitrário, e se iterarmos sucessivamente a função

$$f(x) = \begin{cases} \dfrac{x}{2} & \text{se } x \text{ é par} \\ \\ 3x+1 & \text{se } x \text{ é ímpar} \end{cases}$$

chegaremos inevitavelmente ao inteiro 1 ao cabo de um número finito de passos.

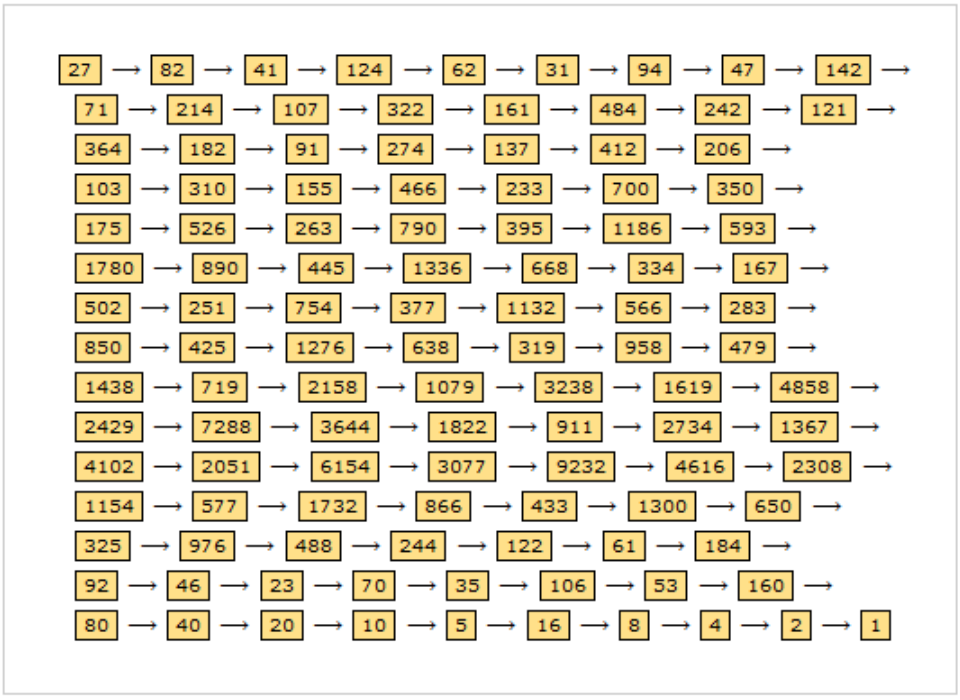
```
procedure Collatz(n : inteiro positivo)
x := n
while x ≠ 1 do
  if x mod 2 = 0 then x := x/2
  else x := 3x + 1
```

Se simularmos este algoritmo para os primeiros 200 inteiros e contarmos o número de iterações necessárias para levar a função até 1 obtemos a seguinte tabela:

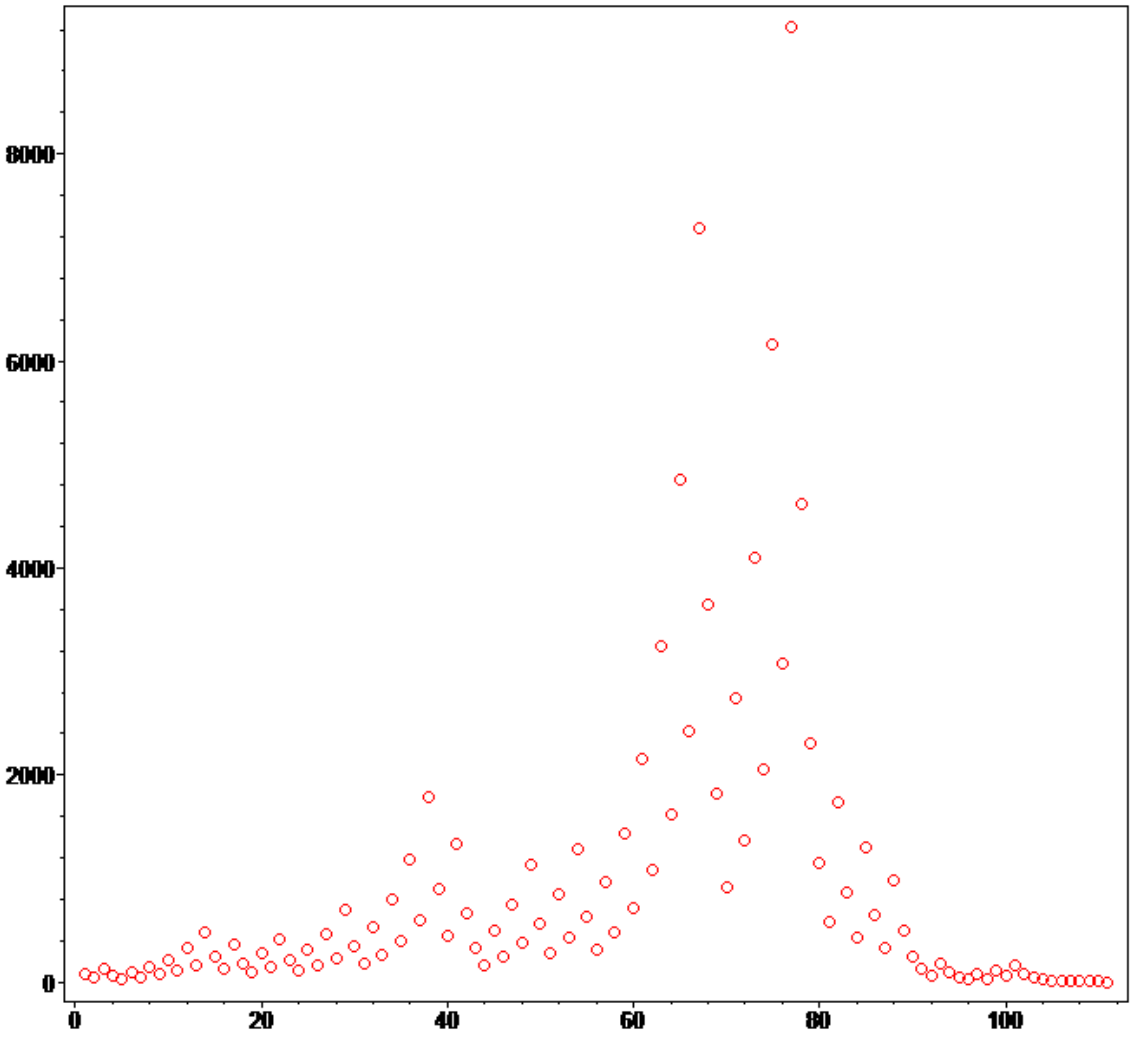
Verificação da conjectura para os primeiros 200 inteiros:

```
0,   1,   7,   2,   5,   8,   16,  3,   19,  6,   14,  9,   9,   17,  17,  4,   12
20,  20,  7,   7,   15,  15,  10,  23,  10,  111, 18,  18,  18,  106, 5,   26,  13,
13,  21,  21,  21,  34,  8,   109, 8,   29,  16,  16,  16,  104, 11,  24,  24,  24,
11,  11,  112, 112, 19,  32,  19,  32,  19,  19,  107, 107, 6,   27,  27,  27,  14,
14,  14,  102, 22,  115, 22,  14,  22,  22,  35,  35,  9,   22,  110, 110, 9,   9,
30,  30,  17,  30,  17,  92,  17,  17,  105, 105, 12,  118, 25,  25,  25,  25,  25,
87,  12,  38,  12,  100, 113, 113, 113, 69,  20,  12,  33,  33,  20,  20,  33,  33,
20,  95,  20,  46,  108, 108, 108, 46,  7,   121, 28,  28,  28,  28,  28,  41,  15,
90,  15,  41,  15,  15,  103, 103, 23,  116, 116, 116, 23,  23,  15,  15,  23,  36,
23,  85,  36,  36,  36,  54,  10,  98,  23,  23,  111, 111, 111, 67,  10,  49,  10,
124, 31,  31,  31,  80,  18,  31,  31,  31,  18,  18,  93,  93,  18,  44,  18,  44,
106, 106, 106, 44,  13,  119, 119, 119, 26,  26,  26,  119, 26
```

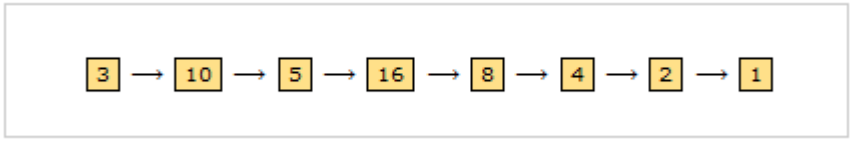
Por exemplo, para $n = 27$ são necessárias 111 iterações:



Desde o valor inicial 27 até 1 a função atinge um pico na 77^a iteração com o valor 9232. Graficamente:



Para $n = 3$ a sequência é muito mais curta, bastam 7 iterações:



Eficiência de algoritmos. Além de produzir uma solução satisfatória e precisa para o problema que pretende resolver, um algoritmo tem que ser eficiente (em termos de velocidade de execução). Um dos objectivos da *algoritmia* consiste em medir a eficiência de algoritmos. Muitas vezes dispomos de diferentes algoritmos que resolvem correctamente o problema, mas algum poderá

ser mais eficiente que os outros. Uma medida de eficiência será, claro, o tempo dispendido por um computador para resolver o problema executando o algoritmo.

Seja P um problema e A um algoritmo para resolver P. O *tempo de execução* de A pode ser analisado contando o número de determinadas operações que são efectuadas durante a sua execução. Esta contagem pode depender do tamanho do input.

Exemplos. (1) No problema da determinação do elemento máximo de uma sequência com n elementos, será natural tomar como medida de eficiência o número de comparações entre elementos, que dependerá evidentemente de n. Calculemos esse número. Para encontrar o elemento máximo, o máximo temporário começa por ser igual ao termo inicial da sequência. Em seguida, depois de uma comparação ter sido feita para verificar que o final da sequência ainda não foi atingido, o máximo temporário é comparado com o segundo termo da sequência, actualizando o máximo temporário para este valor, caso seja maior. O procedimento continua, fazendo mais duas comparações no passo seguinte. Como são feitas duas comparações em cada um dos passos (desde o segundo termo da sequência até ao último) e mais uma comparação é feita para sair do ciclo (quando $i = n + 1$), são feitas ao todo

$$2(n-1) + 1 = 2n - 1$$

comparações.

(2) Se P consiste em verificar se um determinado objecto pertence a uma dada lista, será também natural contar o número de comparações efectuadas por A, que dependerá do tamanho da lista.

(3) Para resolver o problema da *ordenação de listas* existem diversos algoritmos de ordenação, entre os quais o chamado *algoritmo da inserção*. A ideia por detrás deste algoritmo consiste em dividir a lista L que se pretende ordenar em duas sublistas. A sublista L_1 inclui os elementos de L já ordenados e a sublista L_2, que é um sufixo da lista inicial, inclui os elementos de L ainda não analisados. Cada passo do algoritmo consiste na inserção do primeiro elemento de L_2 ordenadamente na lista L_1 e, claro, na sua remoção da lista L_2. O algoritmo inicia-se com

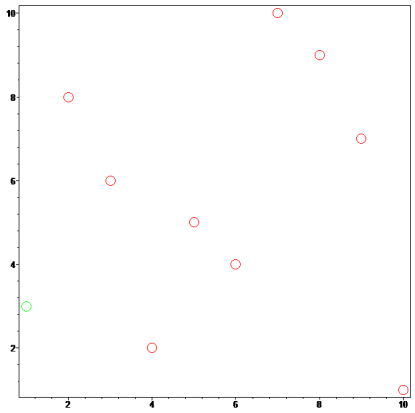
$$L_1 = \{Primeiro[L]\} \text{ e } L_2 = Resto[L]$$

e termina quando $L_2 = \varnothing$. No caso dos algoritmos de ordenação é típico tomar-se como medida de eficiência o número de comparações entre elementos. Claro que o número de comparações depende da lista dada inicialmente.

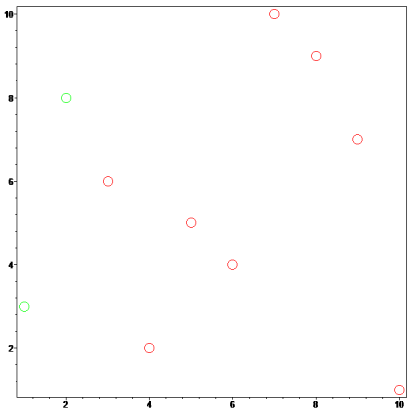
Simulemos o algoritmo para a lista inicial

```
[3,8,6,2,5,4,10,9,7,1]
```

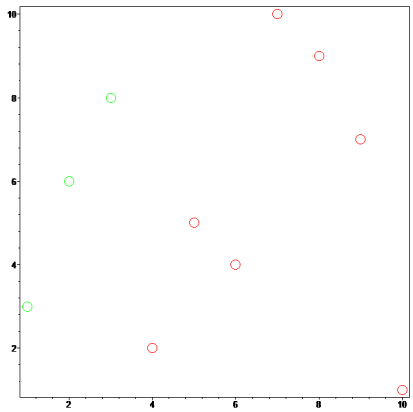
As seguintes figuras dão uma visão dinâmica da ordenação sucessiva efectuada pelo algoritmo. As bolas verdes correspondem a elementos já ordenados (ou seja, elementos da sublista L_1), enquanto que as vermelhas correspondem aos elementos ainda não ordenados da sublista L_2.



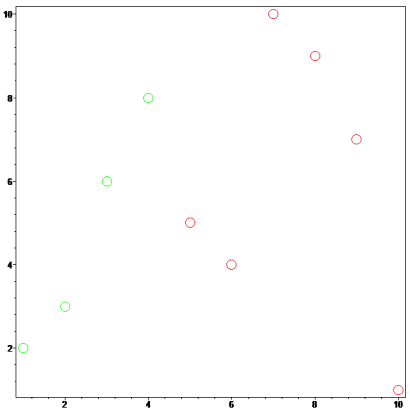
3, 8, 6, 2, 5, 4, 10, 9, 7, 1



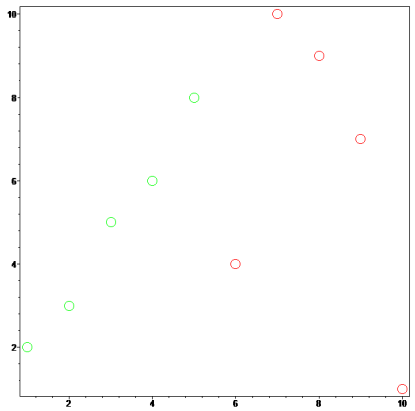
3, 8, 6, 2, 5, 4, 10, 9, 7, 1



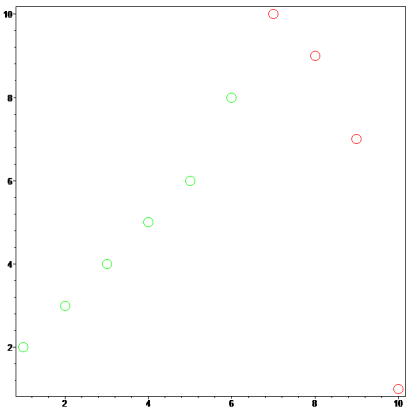
3, 6, 8, 2, 5, 4, 10, 9, 7, 1



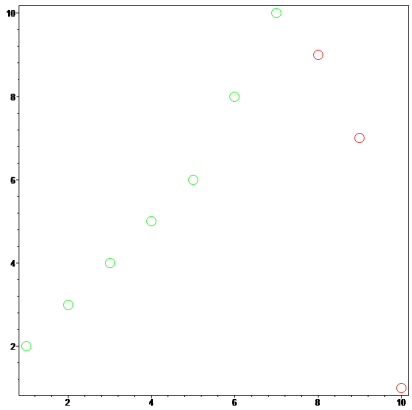
2, 3, 6, 8, 5, 4, 10, 9, 7, 1



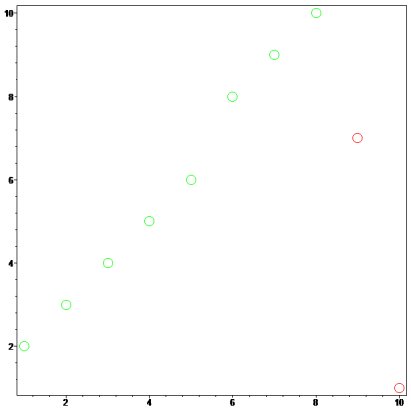
2, 3, 5, 6, 8, 4, 10, 9, 7, 1



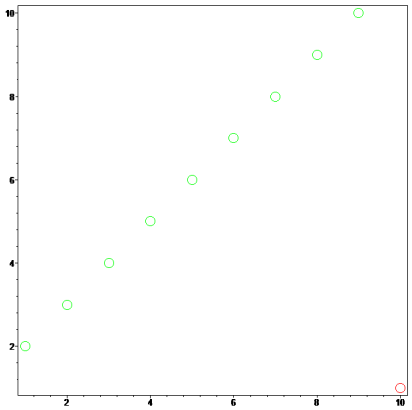
2, 3, 4, 5, 6, 8, 10, 9, 7, 1



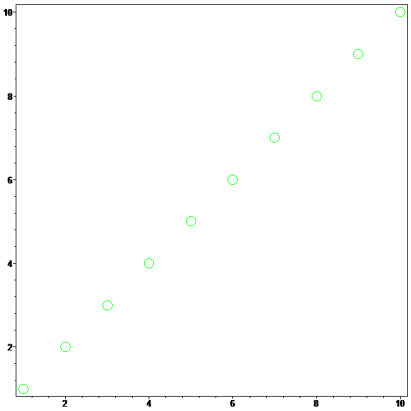
2, 3, 4, 5, 6, 8, 10, 9, 7, 1



2, 3, 4, 5, 6, 8, 9, 10, 7, 1



2, 3, 4, 5, 6, 7, 8, 9, 10, 1



1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Em todos estes exemplos, a análise da eficiência baseia-se na contagem do número de comparações a realizar, o que obviamente depende do tamanho da lista. Para determinar essa eficiência é costume considerar dois casos:

- pior situação (ou seja, situação mais desfavorável dos dados)
- situação média.

Um input no caso da pior situação é um input que leva A a executar o maior número de operações. No caso da determinação do elemento máximo de uma sequência (exemplo (1)), fixado o comprimento n da sequência, a pior situação acontece para qualquer lista de números (pois o número de comparações é constante, igual a $2n - 1$). No exemplo (2) a pior situação será uma lista que não contém o objecto procurado.

No caso do algoritmo de inserção (exemplo (3)), a pior situação é quando a lista dada está ordenada por ordem inversa.

A análise na situação média obriga a considerações probabilísticas pois é necessário atribuir a cada situação uma probabilidade. Muitas vezes considera-se que as situações têm todas a mesma probabilidade (a distribuição das mesmas é então uniforme).

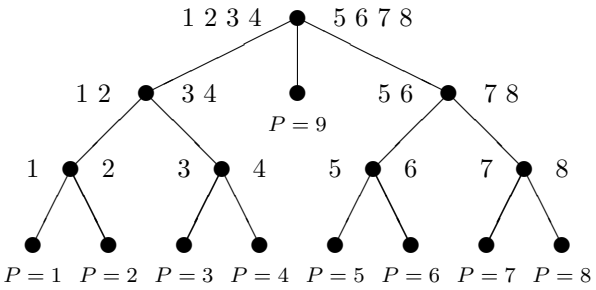
Estudemos um pouco o caso da pior situação. Seja $T_A(n)$ o tempo de execução máximo de A para inputs de tamanho n. A função T_A chama-se a *função da pior situação* para A. Um algoritmo A para resolver um problema $\mathcal{P}$ diz-se *optimal na pior situação* se qualquer algoritmo B que resolve $\mathcal{P}$ satisfaz

$$T_A(n) \leq T_B(n) \quad \text{para qualquer } n \in \mathbb{N}.$$

Uma *árvore de decisão* para um algoritmo é uma árvore cujos nós representam pontos de decisão no algoritmo e cujas folhas representam os resultados.

Teste. Dado um conjunto de nove moedas, uma das quais é mais pesada que as outras, use uma balança de dois pratos (sem pesos) para determinar a moeda mais pesada.

Solução. Denotemos as moedas por $1, 2, \ldots, 9$ (e por P a mais pesada). Podemos fazer a seguinte sequência de pesagens (cada nó interno representa uma pesagem; os números de cada lado de cada nó interno representam os conjuntos de moedas colocados em cada prato da balança, na respectiva pesagem; $P = 1, P = 2, \ldots, P = 9$ são os 9 resultados possíveis):



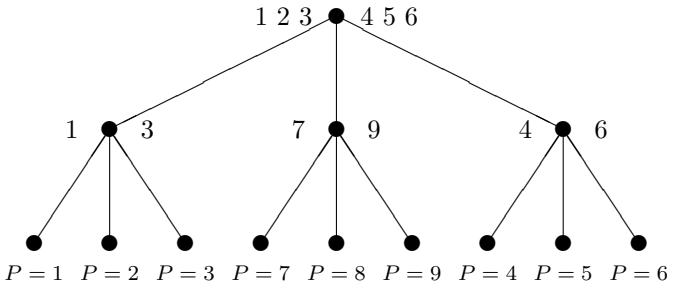
Esta árvore tem *profundidade* três[8], o que significa que, neste algoritmo, o caso da pior situação são 3 pesagens.

Será o algoritmo optimal na pior situação? Em cada pesagem pode acontecer uma de três coisas: o prato da esquerda está mais pesado, o prato da direita está mais pesado ou os pratos estão equilibrados. Portanto, neste tipo de problemas com balanças, as árvores de decisão são ternárias (em cada nó haverá no máximo três ramos). Uma árvore ternária de profundidade d tem, no máximo, 3^d folhas. Como há 9 possíveis resultados, então

$$3^d \geq 9, \text{ isto é, } d \geq \log_3 9 = 2.$$

Portanto, poderá haver, eventualmente, algum algoritmo cuja árvore tenha profundidade 2. E, de facto, há:

[8] A *profundidade* de uma árvore é o comprimento do maior caminho desde a raiz da árvore até às folhas. Na árvore em questão, desde a raiz até às folhas há caminhos com um ramo (no caso da folha $P = 9$) ou três ramos (nas restantes folhas). Mais tarde, quando estudarmos os grafos, estudaremos as árvores com mais cuidado.



Pela discussão acima podemos agora concluir que este é um algoritmo optimal na pior situação (2 pesagens).

Teste. Dado um conjunto de nove moedas, uma das quais é defeituosa (mais pesada ou mais leve que as outras), determine um algoritmo optimal na pior situação para balanças de dois pratos (sem pesos) que determine a moeda defeituosa e dê como output se a moeda é mais pesada ou mais leve.

Solução. Comecemos por determinar um minorante para a profundidade da árvore de decisão. Denotemos as moedas por $1, 2, \ldots, 9$ e usemos a letra P quando a moeda defeituosa for mais pesada e a letra L caso contrário. Existem assim 18 resultados possíveis:

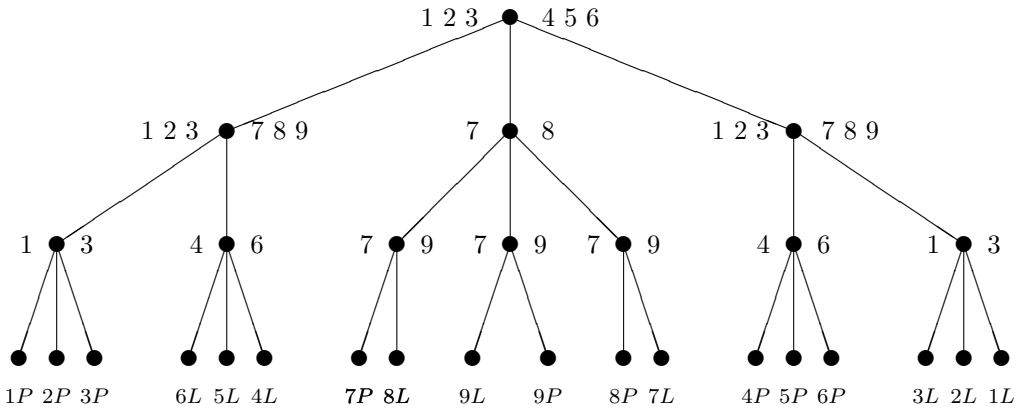
$$1P,\ 1L, \ldots, 9P, 9L.$$

Uma árvore de decisão ternária com profundidade d terá no máximo 3^d folhas, donde $3^d \geq 18$, isto é,

$$d \geq \log_3 18 \geq \lceil \log_3 18 \rceil = 3.$$

(A função $\lceil - \rceil : \mathbb{R} \to \mathbb{N}$ nos números reais é a chamada função *tecto* (*ceiling*) e aplica cada número real x no menor inteiro $\lceil x \rceil$ tal que $\lceil x \rceil \geq x$. De modo análogo, podemos definir a função *chão* (*floor*) $\lfloor - \rfloor : \mathbb{R} \to \mathbb{N}$ que a cada real x faz corresponder o maior inteiro $\lfloor x \rfloor \leq x$. Por exemplo, $\lfloor \frac{1}{2} \rfloor = 0$, $\lceil \frac{1}{2} \rceil = 1$, $\lfloor -\frac{1}{2} \rfloor = -1$, $\lceil -\frac{1}{2} \rceil = 0$.)

Portanto, qualquer algoritmo que resolva o problema terá sempre que efectuar pelo menos 3 pesagens. No algoritmo seguinte esse valor é igual a 3:



Portanto, este é um algoritmo optimal para a pior solução.

O `Maple` tem algumas ferramentas que permitem medir a *performance* de um algoritmo. Nomeadamente, com a função `time(-)` que indica a hora actual, é fácil medir o tempo de CPU (Unidade de Processamento Central) que uma função leva a calcular um resultado:

```
> ha := time():
> Funcaoqualquer(x):
> time() - ha;
```

Vamos agora usar estas funções para comparar dois algoritmos (ver Exercícios 11 e 12 da ficha prática) que calculam o valor de um polinómio $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ num ponto específico c, ou seja, o número $p(c) = a_0 + a_1c + a_2c^2 + \cdots + a_nc^n$. Os valores de entrada são o número c e a lista de coeficientes $a_0, a_1, a_2, \ldots, a_n$ do polinómio.

```
> Polinomio := proc(c::float, coef::list)
>   local potencia, i, y;
>   potencia := 1;
>   y := coef[1];
>   for i from 2 to nops(coef) do
>     potencia := potencia*c;
>     y := y + coef[i] * potencia;
>   od;
>   RETURN(y);
> end:

> Horner := proc(c::float, coef::list)
>   local i, y;
>   y := coef[nops(coef)];
>   for i from nops(coef)-1 by -1 to 1 do
>     y := y * c + coef[i];
>   od;
>   RETURN(y);
> end:
```

Por exemplo, para o polinómio $p(x) = 4 + 3x + 2x^2 + x^3$, o valor de $p(5)$ é igual a 194:

```
> input_lista := [4,3,2,1];


> Polinomio(5.0, input_lista);
                                194.000


> Horner(5.0, input_lista);
                                194.000
```

De modo a testarmos um algoritmo contra o outro precisamos de gerar listas de coeficientes. O comando seguinte gera aleatoriamente um polinómio de grau 2000:

```
> p2000 := randpoly(x,degree=2000,dense):
```

e se fizermos

```
> q2000 := subs(x=1,convert(p2000,list)):
```

obtemos a lista dos coeficientes correspondentes, que podemos usar como input nos algoritmos `Polinomio` e `Horner`. Agora, usando as ferramentas para medir o tempo de execução, obtemos:

```
> ha := time():
> Horner(104567890000000.0, q2000);
```

$$0.3913255222\ 10^{27971}$$

```
> time() - ha;
```

$$0$$

```
> ha := time():
> Polinomio(104567890000000.0, q2000);
```

$$0.3913255222\ 10^{27971}$$

```
> time() - ha;
```

$$0$$

Experimentando polinómios de grau superior obtemos os tempos de execução seguintes:

	4000	6000	8000
`Polinomio`	0.031	0.047	0.063
`Horner`	0	0.015	0.031

Podemos assim concluir que o método de Horner de cálculo polinomial é marginalmente mais rápido que o método tradicional da substituição da indeterminada x pelo valor onde queremos calcular a função polinomial.

Voltemos ao primeiro algoritmo `Max` desta secção (determinação do elemento máximo de uma sequência). Na altura verificámos que para uma sequência de comprimento n, o algoritmo efectua $2n - 1$ comparações. Usando a chamada *notação assimptótica* da seguinte definição, diz-se que o número de comparações no pior caso para este algoritmo é $O(n)$.

Definição. Diz-se que uma função $f : \mathbb{N} \to \mathbb{R}$ é *da ordem de* $g : \mathbb{N} \to \mathbb{R}$, o que se denota por $f(n) = O(g(n))$ se existe uma constante real c e um $k \in \mathbb{N}$ tais que

$$|f(n)| \leq c\,|g(n)| \quad \text{para } n \geq k.$$

Teste. Mostre que qualquer função polinomial p com coeficientes reais, dada por $p(n) = a_t n^t + a_{t-1} n^{t-1} + \cdots + a_1 n + a_0$, é da ordem de q onde $q(n) = n^t$.

No algoritmo `Max`, tomando $f(n) = 2n - 1$ então $f(n) = O(n)$: basta tomar $c = 2$ e $k = 1$, pois $2n - 1 \leq 2n$ para qualquer $n \in \mathbb{N}$. Portanto, o algoritmo `Max` tem complexidade $O(n)$, ou seja, *linear*, na pior situação, atendendo à seguinte classificação:

Complexidade	**Terminologia**
$O(1)$	Complexidade constante
$O(\log n)$	Complexidade logarítmica
$O(n)$	Complexidade linear
$O(n \log n)$	Complexidade $n \log n$
$O(n^b)$	Complexidade polinomial
$O(b^n)$, onde $b > 1$	Complexidade exponencial
$O(n!)$	Complexidade factorial

Qual é a complexidade do algoritmo de inserção (de ordenação de listas)? No pior caso, onde a lista está ordenada por ordem inversa, para cada i é necessário fazer $i - 1$ comparações. Como i varia de 2 até ao comprimento n da lista, temos que o número de comparações no pior caso é igual a

$$\sum_{i=2}^{n} (i-1) = 1 + 2 + 3 + \cdots + (n-1),$$

ou seja, é dado pela soma dos $n - 1$ primeiros números naturais (termos de uma progressão aritmética de razão 1):

$$\sum_{i=2}^{n} (i-1) = 1 + 2 + 3 + \cdots + (n-1) = \frac{n^2 - n}{2}.$$

Poderíamos ter calculado este somatório com o auxílio do `WolframAlpha`[9]:

```
> sum((i-1), i=2..n)
```

$$\frac{1}{2}(n-1)n.$$

Demonstrámos, assim, o seguinte resultado:

Proposição. *O número de comparações na pior situação para o algoritmo da inserção é $O(n^2)$. Tem assim complexidade polinomial.*

Façamos agora um exemplo de análise do caso médio no caso do algoritmo da inserção.

[9] Calculadora simbólica de acesso público em `http://www.wolframalpha.com`, baseada no `Mathematica`.

Suponhamos que queremos inserir um elemento na parte da lista já ordenada (a sublista L_1 com $i-1$ elementos) de modo que a lista resultante com i elementos esteja ordenada. Potencialmente, existem i posições onde esse elemento pode ser colocado (para além das $i-1$ ocupadas pela lista dada, ainda existe a possibilidade do novo elemento ser maior que todos os outros). Vamos impor uma hipótese probabilística que consiste em assumir que todas as posições são equiprováveis para colocar o novo elemento. Isto é, cada posição tem probabilidade $1/i$.

O número de comparações necessárias se o novo elemento tiver que ser colocado na posição $k \geq 2$ é $i-k+1$ e na posição $k=1$ é $i-1$. Logo *o número médio de comparações para introduzir o novo elemento numa das i posições é dado pelo somatório*

$$\left(\sum_{k=2}^{i} \frac{1}{i}(i-k+1)\right) + \frac{1}{i}(i-1).$$

Acedendo ao `WolframAlpha`, podemos calcular este somatório:

```
>  sum((i-k+1)/i, k=2...i) + (i-1)/i
```

que é igual a

$$\frac{i-1}{i} + \frac{i-2}{2}$$

$$\frac{i^2+i-2}{2i}.$$

Logo, o número médio de comparações para que a lista fique ordenada é dado pelo somatório

$$\sum_{i=2}^{n} \left(\frac{1}{2} - \frac{1}{i} + \frac{i}{2}\right).$$

2.2. Somatórios

Como estamos a ver, ao determinar as medidas de eficiência de algoritmos surgem somatórios, às vezes não triviais, para calcular. Portanto, se quisermos fazer análise de algoritmos temos que saber calcular somatórios. Estudaremos, em seguida, algumas técnicas de cálculo de somatórios.

Proposição. *Seja I um conjunto finito. Os somatórios gozam das seguintes propriedades*:

(1) *Distributividade:* $\sum_{i\in I} c\, a_i = c \sum_{i\in I} a_i$.

(2) *Associatividade:* $\sum_{i\in I} (a_i + b_i) = \sum_{i\in I} a_i + \sum_{i\in I} b_i$.

(3) *Comutatividade:* $\sum_{i\in I} a_i = \sum_{i\in I} a_{p(i)}$ *para qualquer bijecção (permutação)* $p : I \to I$.

(4) *Progressão constante:* $\sum_{i\in I} c = c\,|I|$.

(5) *Aditividade dos índices:* $\sum_{i\in I} a_i + \sum_{i\in J} a_i = \sum_{i\in(I\cup J)} a_i + \sum_{i\in(I\cap J)} a_i$ *(sendo J um conjunto finito também).*

(6) *Mudança de variável:* $\sum_{i\in I} a_{f(i)} = \sum_{j\in J} a_j$*, para qualquer função bijectiva $f : I \to J$; mais geralmente, para qualquer função $f : I \to J$,* $\sum_{i\in I} a_{f(i)} = \sum_{j\in J} \big(a_j \cdot \#(f^{-1}(\{j\}))\big)$.

Teste. Mostre que $\sum_{0\le i<n}(a_{i+1} - a_i)b_i = a_n b_n - a_0 b_0 - \sum_{0\le i<n} a_{i+1}(b_{i+1} - b_i)$.

Vejamos alguns exemplos de aplicação destas propriedades.

Exemplo: progressão aritmética de razão r. Provemos que

$$\boxed{\sum_{i=0}^{n}(a+ri) = \left(a + \frac{1}{2}rn\right)(n+1) = (2a+rn)\frac{n+1}{2} = [a + (a+rn)]\frac{n+1}{2}}$$

$$\sum_{i=0}^{n}(a+ri) = \sum_{i=0}^{n}(a+r(n-i)) \qquad \text{(por comutatividade, com } p(i) = n-i)$$

$$\Leftrightarrow \quad \sum_{i=0}^{n}(a+ri) = \sum_{i=0}^{n}(a+rn-ri)$$

$$\Leftrightarrow \quad 2\sum_{i=0}^{n}(a+ri) = \sum_{i=0}^{n}(a+rn-ri) + \sum_{i=0}^{n}(a+ri)$$

$$\Leftrightarrow \quad 2\sum_{i=0}^{n}(a+ri) = \sum_{i=0}^{n}((a+rn-ri)+(a+ri)) \qquad \text{(por associatividade)}$$

$$\Leftrightarrow \quad 2\sum_{i=0}^{n}(a+ri) = \sum_{i=0}^{n}(2a+rn)$$

$$\Leftrightarrow \quad 2\sum_{i=0}^{n}(a+ri) = (2a+rn)\sum_{i=0}^{n}1 \qquad \text{(por distributividade)}$$

$$\Leftrightarrow \quad 2\sum_{i=0}^{n}(a+ri) = (2a+rn)(n+1) \qquad \text{(por progressão constante)}$$

$$\Leftrightarrow \quad \sum_{i=0}^{n}(a+ri) = (a+\frac{1}{2}rn)(n+1).$$

Alternativamente, se tivermos já na mão a prova da fórmula

$$\sum_{i=1}^{n-1} i = \frac{n^2 - n}{2}$$

(é o caso $a = 0, r = 1$) que usámos anteriormente, podemos simplificar muito a demonstração do caso geral:

$$\begin{aligned}
\sum_{i=0}^{n}(a+ri) &= \sum_{i=0}^{n} a + \sum_{i=0}^{n} ri && \text{(por associatividade)}\\
&= a\sum_{i=0}^{n} 1 + r\sum_{i=0}^{n} i && \text{(por distributividade)}\\
&= a\sum_{i=0}^{n} 1 + r\left(\sum_{i=1}^{n-1} i + n\right) && \text{(por progressão constante)}\\
&= a(n+1) + r\left(\frac{n^2-n}{2} + n\right)\\
&= a(n+1) + r\left(\frac{n^2+n}{2}\right).
\end{aligned}$$

Exemplo: progressão geométrica de razão r. Provemos que

$$\boxed{\sum_{i=0}^{n} ar^i = \frac{ar^{n+1} - a}{r-1}}$$

$$\begin{aligned}
& \sum_{i=0}^{n} ar^i + ar^{n+1} = \sum_{i=0}^{n+1} ar^i && \text{(por aditividade com } I = \{0,\ldots,n\} \text{ e } J = \{n+1\})\\
\Leftrightarrow\ & \sum_{i=0}^{n} ar^i + ar^{n+1} = ar^0 + \sum_{i=1}^{n+1} ar^i && \text{(por aditividade com } I = \{0\} \text{ e } J = \{1,\ldots,n+1\})\\
\Leftrightarrow\ & \sum_{i=0}^{n} ar^i + ar^{n+1} = ar^0 + \sum_{i=0}^{n} ar^{i+1} && \text{(por mudança de variável com } I = \{0,\ldots,n\})\\
\Leftrightarrow\ & \sum_{i=0}^{n} ar^i + ar^{n+1} = a + r\sum_{i=0}^{n} ar^i && \text{(por distributividade)}\\
\Leftrightarrow\ & \sum_{i=0}^{n} ar^i = \frac{ar^{n+1} - a}{r-1}.
\end{aligned}$$

Leituras suplementares. Como vimos, muitos problemas algorítmicos relativamente simples requerem por vezes o cálculo de somatórios um pouco complicados. Não se conhecem métodos gerais que permitam resolver qualquer somatório, a maior parte das vezes os problemas têm que

ser atacados de forma *ad hoc* (esta é uma das características de muitas áreas da matemática discreta: a não existência de métodos gerais de resolução que obrigam uma abordagem *ad hoc* ao problema; aqui o conhecimento e a destreza na manipulação dos diversos métodos particulares de resolução, que poderão só funcionar em alguns casos, é crucial).

Nestas observações finais indicaremos muito resumidamente algumas das técnicas mais elegantes e importantes para resolver somatórios.

Método 1: Método *ad hoc*. Calculando as primeiras somas parciais do somatório (a_1+a_2, a_1+a_2+a_3, etc.), é por vezes possível adivinhar a correspondente fórmula geral. A ilustração deste método em muitos exemplos interessantes pode ser vista e experimentada (interactivamente) no módulo *Somatórios*[10] na página da disciplina.

A fórmula deverá depois ser confirmada com uma prova formal, para termos a certeza da sua validade. Essa prova pode ser feita de modo análogo como fizemos nalguns exemplos acima, usando as propriedades dos somatórios que enunciámos, ou, mais facilmente, pelo método de indução matemática estudado na secção anterior.

Método 2: Método da perturbação. A ideia por detrás deste método é a seguinte: tentar obter duas expressões diferentes que tenham o mesmo valor, "perturbando" levemente a soma a calcular. Um exemplo ilustra o funcionamento deste método:

Suponhamos que pretendemos calcular o valor de $q(n) = \sum_{i=1}^{n} i^2$. Vamos perturbar levemente a sua definição e tentar escrever $q(n+1)$ de duas formas diferentes. Por um lado, $q(n+1) = q(n) + (n+1)^2$ e, por outro lado, por uma mudança de variável,

$$\begin{aligned} q(n+1) &= \sum_{i=0}^{n} (i+1)^2 \\ &= \sum_{i=0}^{n} (i^2 + 2i + 1) \\ &= \sum_{i=0}^{n} i^2 + 2\sum_{i=0}^{n} i + \sum_{i=0}^{n} 1 \\ &= q(n) + 2\sum_{i=0}^{n} i + (n+1). \end{aligned}$$

Comparando ambos os resultados obtemos

$$q(n) + (n+1)^2 = q(n) + 2\sum_{i=0}^{n} i + (n+1),$$

ou seja,

$$(n+1)^2 = 2\sum_{i=1}^{n} i + (n+1) \Leftrightarrow \sum_{i=1}^{n} i = \frac{(n+1)^2 - (n+1)}{2}.$$

Parece que não fizemos muitos progressos! Limitámo-nos a obter a soma $\sum_{i=1}^{n} i$ (note também que temos aqui uma prova formal, rigorosa, da fórmula para $\sum_{i=1}^{n-1} i$ que utilizámos anteriormente, na página 40). Mas isto sugere imediatamente o seguinte: se perturbando um pouco a

[10] `www.mat.uc.pt/~picado/ediscretas/somatorios`.

soma $q(n)$ dos quadrados conseguimos obter uma fórmula para $\sum_{i=1}^{n} i$, será que perturbando a soma $c(n) = \sum_{i=1}^{n} i^3$ dos cubos conseguimos uma fórmula para $q(n)$?

A fórmula de recorrência de $c(n)$ é $c(n+1) = c(n) + (n+1)^3$ e, por outro lado, por uma mudança de variável,

$$\begin{aligned} c(n+1) &= \sum_{i=0}^{n} (i+1)^3 \\ &= \sum_{i=0}^{n} (i^3 + 3i^2 + 3i + 1) \\ &= \sum_{i=0}^{n} i^3 + 3\sum_{i=0}^{n} i^2 + 3\sum_{i=0}^{n} i + \sum_{i=0}^{n} 1 \\ &= c(n) + 3q(n) + \frac{3}{2}n(n+1) + (n+1). \end{aligned}$$

Igualando ambas as expressões, obtemos

$$c(n) + (n+1)^3 = c(n) + 3q(n) + \frac{3}{2}n(n+1) + (n+1),$$

ou seja,

$$\begin{aligned} (n+1)^3 = 3q(n) + \frac{3}{2}n(n+1) + (n+1) &\Leftrightarrow 3q(n) = (n+1)^3 - \frac{3}{2}n(n+1) - (n+1) \\ &\Leftrightarrow q(n) = \frac{2(n+1)^3 - 3n(n+1) - 2(n+1)}{6} \\ &\Leftrightarrow q(n) = \frac{2n^3 + 6n^2 + 6n + 2 - 3n^2 - 3n - 2n - 2}{6} \\ &\Leftrightarrow q(n) = \frac{2n^3 + 3n^2 + n}{6} \\ &\Leftrightarrow q(n) = \frac{1}{6}n(n+1)(2n+1). \end{aligned}$$

Em `www.mat.uc.pt/~picado/ediscretas/somatorios/Matematica_sem_palavras_files/soma_quadrados.html` pode ver uma "prova" geométrica, sem palavras, desta fórmula.

Método 3: Método do integral. Este método consiste em aproximar o somatório por um integral. Por exemplo, para calcular $q(n) = \sum_{i=0}^{n} i^2$, aproximamos $q(n)$ por $\int_0^n x^2\,dx$ que podemos calcular facilmente no `WolframAlpha`:

```
> int(x^2, x=0...n)
```

$$\frac{n^3}{3}$$

Analisemos agora o erro $e(n) = q(n) - \frac{n^3}{3}$ desta aproximação:

$$\begin{aligned} e(n) &= q(n-1) + n^2 - \frac{n^3}{3} \\ &= q(n-1) - \frac{(n-1)^3}{3} + n^2 - \frac{n^3}{3} + \frac{(n-1)^3}{3} \\ &= e(n-1) + n - \frac{1}{3}. \end{aligned}$$

De modo análogo, podemos concluir que $e(n-1) = e(n-2) + (n-1) - \frac{1}{3}$. Portanto,

$$e(n) = 0 + \sum_{i=1}^{n}(i - \frac{1}{3}) = \frac{(n+1)n}{2} - \frac{n}{3}.$$

Consequentemente,

$$q(n) = \frac{n^3}{3} + \frac{(n+1)n}{2} - \frac{n}{3}.$$

Coincide com o resultado calculado anteriormente pelo método da perturbação? Basta reduzir ao mesmo denominador e simplificar:

$$\frac{n(n+1)(2n+1)}{6}.$$

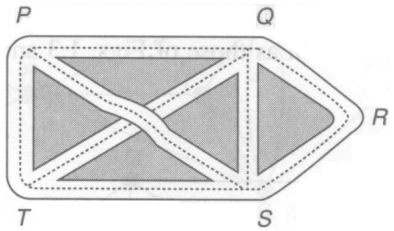
Sim, coincide, claro!

3. Teoria dos Grafos

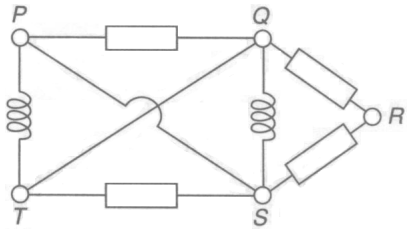
3.1. Noções básicas

A Teoria dos Grafos é actualmente uma das áreas mais importantes da matemática discreta. Tendo as suas raízes em jogos e recreações matemáticas, atribui-se a sua criação a Euler, ao resolver o problema das pontes de Königsberg em 1736, mas foram os problemas acerca de fórmulas de estrutura de compostos químicos, que A. Cayley resolveu na segunda metade do século XIX, que a começaram a desenvolver. Hoje, a Teoria dos Grafos tem sido aplicada a muitas áreas (Informática, Investigação Operacional, Economia, Sociologia, Genética, etc.), pois um grafo constitui o modelo matemático ideal para o estudo das relações entre objectos discretos de qualquer tipo.

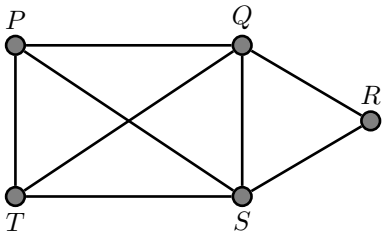
Por exemplo, a seguinte secção de um mapa de estradas



ou a seguinte secção de uma rede eléctrica



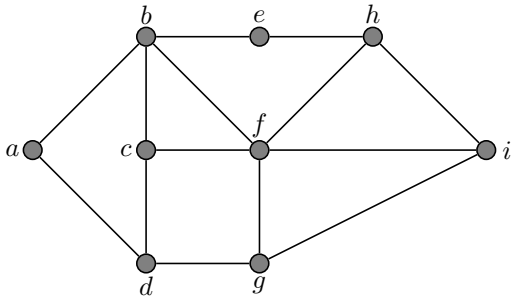
podem ser ambas representadas por meio de pontos e segmentos de recta do seguinte modo:



Um *grafo simples* G consiste num conjunto finito e não vazio $V(G)$ de elementos chamados *vértices* e num conjunto finito $A(G)$ de pares não ordenados de elementos distintos de $V(G)$, chamados *arestas*.

Dois vértices a e b de G dizem-se *adjacentes* se o par $\{u, v\}$ pertence a $A(G)$. Habitualmente representa-se um grafo simples $G = (V(G), A(G))$ por um diagrama no qual os vértices são representados por pontos e as arestas por linhas unindo vértices adjacentes.

Por exemplo, o diagrama



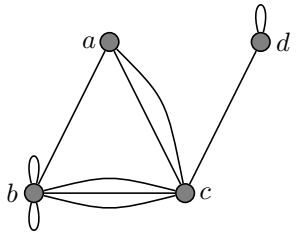
representa o grafo G definido por

$$V(G) = \{a, b, c, d, e, f, g, h, i\} \quad \text{e}$$

$$A(G) = \Big\{\{a,b\},\{a,d\},\{b,c\}\{b,e\},\{b,f\},\{c,d\},\{c,f\},\{d,g\},\{e,h\},\{f,g\},\{f,h\},\{f,i\},\{g,i\},\{h,i\}\Big\}.$$

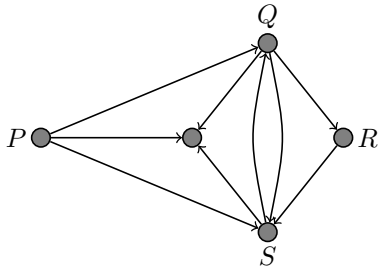
Muitas vezes chamaremos grafo simples ao diagrama que o representa.

Em qualquer grafo simples, existe no máximo uma aresta unindo cada par de vértices. No entanto, muitos resultados envolvendo grafos simples podem ser estendidos a grafos mais gerais nos quais dois vértices podem ter várias arestas (*arestas múltiplas*) unindo-os. Podemos ainda remover a restrição que impõe que as arestas unam vértices distintos, admitindo *lacetes*, ou seja, arestas unindo um vértice a ele próprio. O grafo daí resultante, no qual lacetes e arestas múltiplas são admitidas, diz-se um *pseudografo*. Por exemplo,

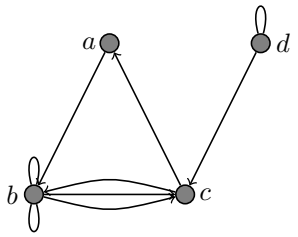


é um pseudografo mas não é um grafo simples.

Embora por vezes tenhamos necessidade de nos restringirmos a grafos simples, provaremos os resultados para pseudografos, sempre que tal seja possível. Muitas vezes, na modelação de certos problemas convirá considerar um sentido para as arestas. Por exemplo, na modelação de mapas de estradas com sentido único:



Um *grafo dirigido* (ou, abreviadamente, *digrafo*) D consiste num conjunto finito não vazio $V(D)$ de elementos chamados *vértices*, e num conjunto finito $A(D)$ de arestas orientadas (eventualmente múltiplas), chamadas *aros*. Por exemplo:



Um digrafo diz-se *simples* se não contiver lacetes e os seus arcos forem todos distintos. Muitas das definições que iremos estudar para pseudografos podem ser imitadas nos digrafos.

A tabela seguinte resume as definições dos vários tipos de grafos:

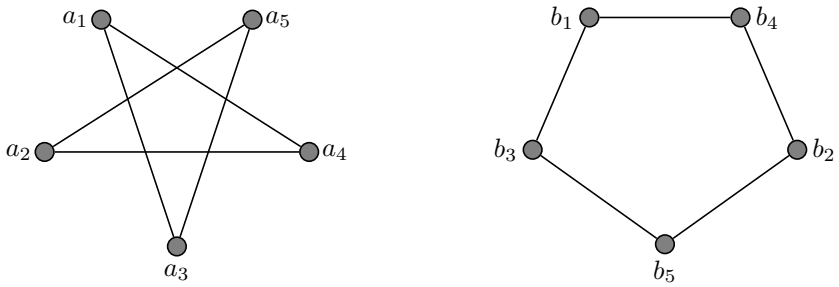
Tipo	**Arestas**	**Arestas Múltiplas?**	**Lacetes?**
Grafo simples	sem direcção	não	não
Multigrafo	sem direcção	sim	não
Pseudografo	sem direcção	sim	sim
Grafo dirigido	dirigidas	sim	sim
Grafo dirigido simples	dirigidas	não	não

Dois grafos G_1 e G_2 dizem-se *isomorfos* se existir uma bijecção

$$f\colon V(G_1) \to V(G_2)$$

preservando a adjacência de vértices, isto é, tal que o número de vezes em que $\{u, v\}$ ocorre em $A(G_1)$ é igual ao número de vezes que $\{f(u), f(v)\}$ ocorre em $A(G_2)$. Neste caso f diz-se um *isomorfismo de grafos*. Escreveremos $G_1 \cong G_2$ para indicar que G_1 e G_2 são isomorfos.

Exemplo. Os grafos

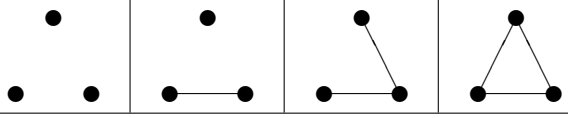


(G_1) (G_2)

são isomorfos. O isomorfismo é dado por

$$\begin{array}{rccccc} f: & V(G_1) & \to & V(G_2) & & \\ & a_i & \mapsto & b_i & & (i=1,2,3,4,5). \end{array}$$

Observações. (1) Dois grafos isomorfos têm o mesmo número de vértices e o mesmo número de arestas.
(2) No caso em que G_1 e G_2 são grafos simples, uma bijecção $f: V(G_1) \to V(G_2)$ é um isomorfismo se e só se $\{u,v\} \in A(G_1)$ exactamente quando $\{f(u), f(v)\} \in A(G_2)$.
(3) Se não fizermos distinção entre grafos isomorfos, os grafos simples com menos de 4 vértices são determinados pelo seu número de vértices e de arestas. Sendo p o número de vértices e q o número de arestas, o quadro

	$q=0$	$q=1$	$q=2$	$q=3$
$p=1$				
$p=2$				
$p=3$				

dá-nos todos os grafos simples com menos de 4 vértices.
(4) Não podemos afirmar o mesmo no caso do número de vértices ser superior ou igual a 4; neste caso o número de vértices e de arestas não é suficiente para determinar esses grafos. Por exemplo,

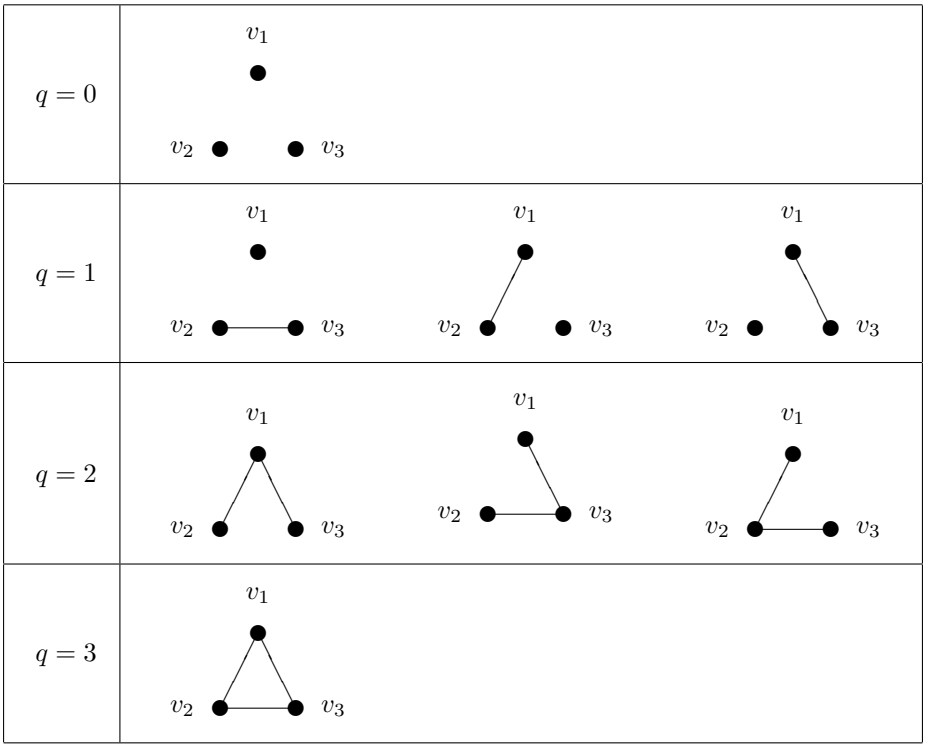
não são isomorfos apesar de terem ambos 4 vértices e 2 arestas.
(5) O número de grafos simples com p vértices $v_1, v_2, \ldots, v_p$ é igual a

$$2^{C(p,2)}$$

pois é o número de subconjuntos do conjunto de todos os pares não ordenados de $\{v_1, v_2, \ldots, v_p\}$. O número desses grafos que contêm q arestas $(0 \leq q \leq C(p,2))$ é igual a

$$C(C(p,2), q).$$

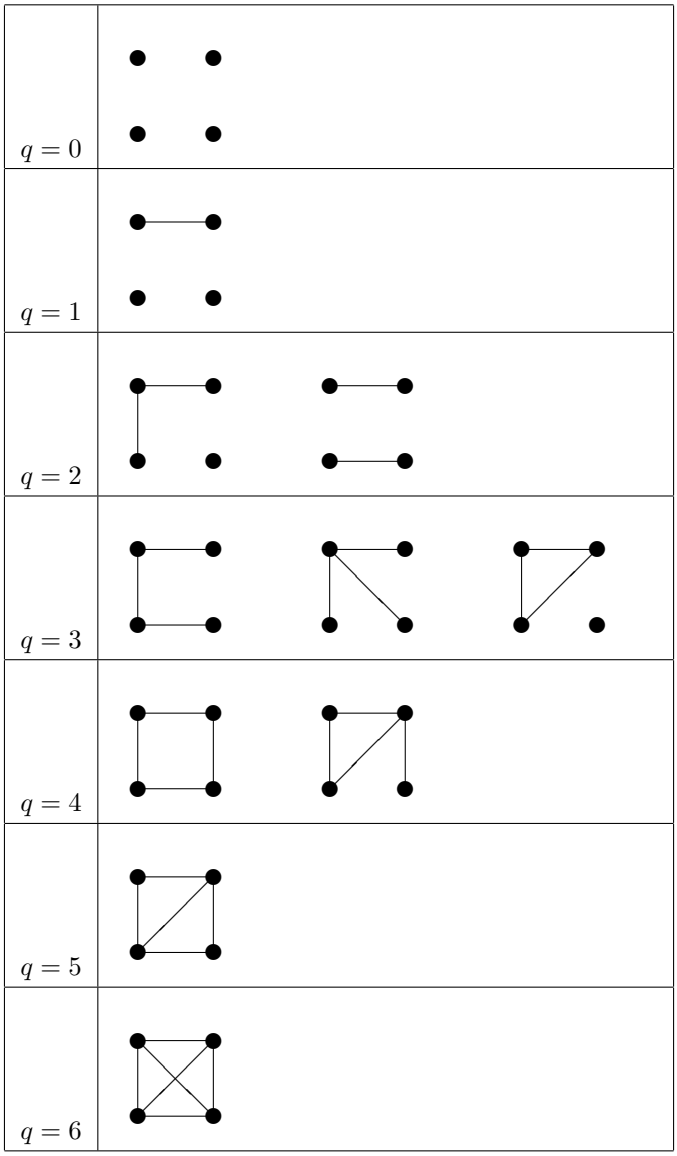
Por exemplo, para $p = 3$:



É claro que os grafos da figura anterior com uma aresta são isomorfos entre si, o mesmo acontecendo com os de duas arestas. A relação de isomorfismo particiona assim o conjunto dos 2^3 grafos simples com vértices v_1, v_2, v_3, em 4 classes de equivalência, cada uma constituída respectivamente pelos grafos simples com 0 arestas, 1 aresta, 2 arestas e 3 arestas. Representa-se cada uma dessas classes por um (qualquer) dos grafos simples nela contidos, sem designação dos vértices:

Todo o grafo simples com 3 vértices pertence a uma destas 4 classes.

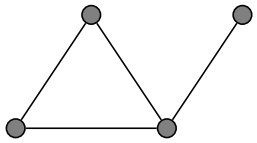
(6) Enumeremos as classes de equivalência dos grafos simples com 4 vértices:



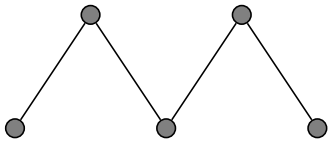
O número de grafos simples em cada classe, com vértices v_1, v_2, v_3, v_4, é dado pela seguinte tabela:

q	**n.º de grafos**	**n.º de grafos**	**n.º de grafos**	**Total**
0	1			1
1	6			6
2	$4 \times C(3,2) = 12$	$C(4,2)/2 = 3$		15
3	12	4	4	20
4	3	12		15
5	6			6
6	1			1

Um grafo G_1 é um *subgrafo* de G se $V(G_1) \subseteq V(G)$ e $A(G_1) \subseteq A(G)$. Se, além disso, $V(G_1) = V(G)$, G_1 diz-se um *subgrafo gerador* de G. Por exemplo, os grafos

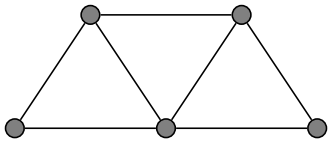


(G_1)



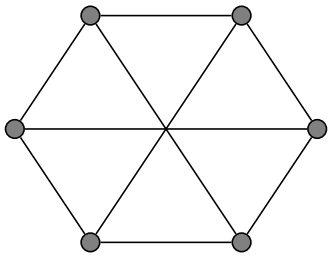
(G_2)

são ambos subgrafos de



sendo G_2 gerador e G_1 não.

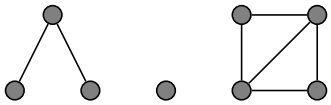
No entanto G_1 não é subgrafo de



embora G_2 o seja.

Podemos combinar dois grafos de modo a obter um grafo maior. Se G_1 e G_2 são dois grafos tais que $V(G_1) \cap V(G_2) = \emptyset$, podemos definir a sua *união* $G_1 \cup G_2$ como sendo o grafo G tal que $V(G) = V(G_1) \cup V(G_2)$ e $A(G) = A(G_1) \cup A(G_2)$.

Um grafo é *conexo* se não puder ser expresso como união de dois grafos, e *desconexo* caso contrário. Evidentemente qualquer grafo desconexo G pode ser expresso como união de grafos conexos, cada um destes dizendo-se uma *componente* de G. Por exemplo,

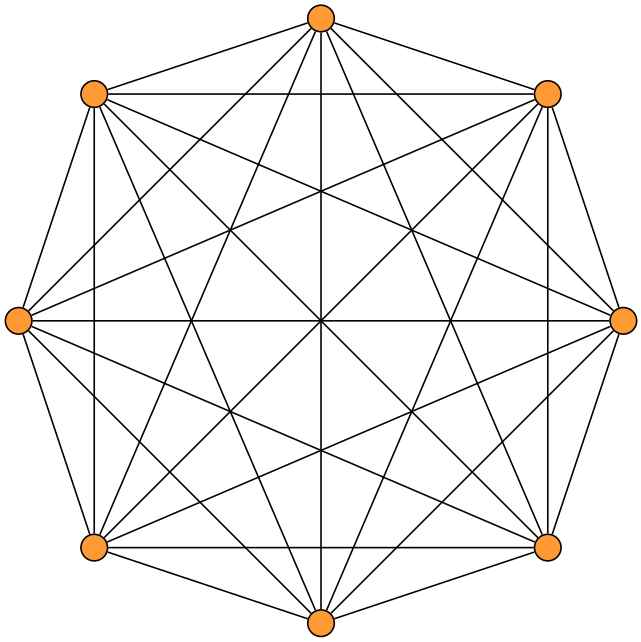


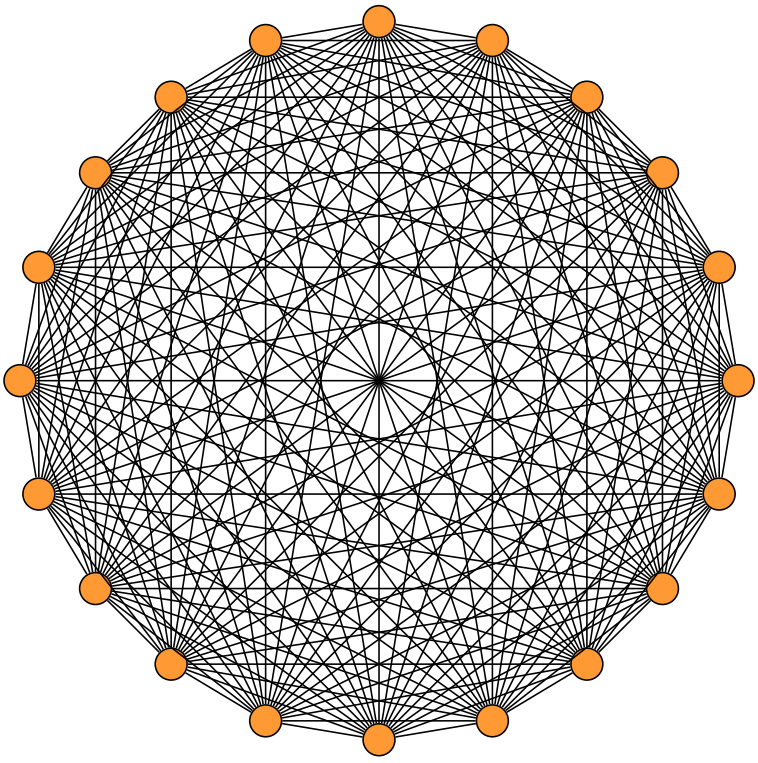
é um grafo desconexo com três componentes.

Se $a = \{v_1, v_2\}$ for uma aresta de um grafo, dizemos que a é *incidente* em v_1 e em v_2. Designamos por *grau* de um vértice v o número de arestas incidentes em v. Denotaremos esse número por $g(v)$.

Um grafo simples com p vértices no qual todos tenham o mesmo grau $p-1$ diz-se um *grafo completo* e denota-se por K_p.

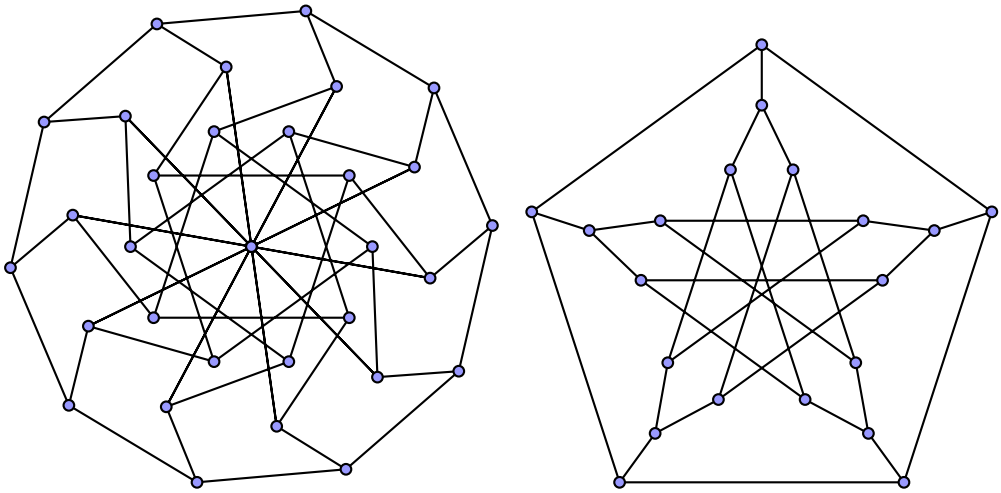
K_8





K_{20}

Um grafo diz-se *regular* se todos os seus vértices tiverem o mesmo grau. Se esse grau for r diz-se que o grafo é regular de grau r. Na figura seguinte, o grafo da direita é regular (de grau 3), o da esquerda não.



Um vértice de grau 0 diz-se *isolado* e um de grau 1 chama-se *terminal*.

Proposição 1. **[Euler (1736)]** *Em qualquer grafo a soma dos graus dos vértices é o dobro do número de arestas, sendo portanto um número par.*

Prova. É óbvio, pois cada aresta é incidente em dois vértices. □

Este resultado é habitualmente apelidado de *Lema dos apertos de mão*, pelo facto de implicar que se um grupo de pessoas apertar as mãos entre si, o número total de mãos apertadas será par — precisamente porque exactamente duas mãos estão envolvidas em cada aperto de mão.

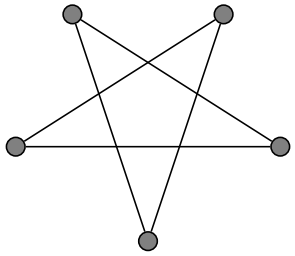
Esta proposição implica imediatamente o seguinte:

Corolário 1. *Em qualquer grafo, o número de vértices com grau ímpar é par.* □

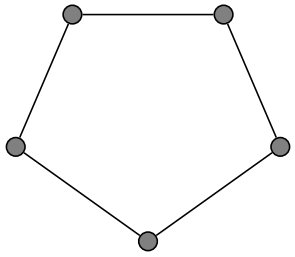
Corolário 2. *Seja G um grafo regular tal que todo o vértice tem grau 3. Então $|V(G)|$ é par.*

Prova. Designemos os vértices de G por $v_1, v_2, \ldots, v_p$. Como $g(v_i) = 3$ para cada $i \in \{1, 2, \ldots, p\}$, $\sum_{i=1}^{p} g(v_i) = 3p$. Mas, pela Proposição 1, $\sum_{i=1}^{p} g(v_i) = 2q$, sendo q o número de arestas de G. Logo $3p = 2q$ e, consequentemente, p é par. □

Seja G um grafo simples. O *grafo complementar* de G, que denotaremos por $\overline{G}$, é definido por $V(\overline{G}) = V(G)$ e $\{u, v\} \in A(\overline{G})$ se e só se $\{u, v\} \notin A(G)$. Por exemplo,



é o complementar de

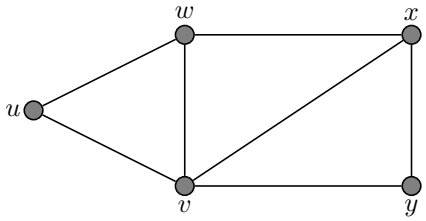


Teorema 2. *Para qualquer grafo simples G com 6 vértices, G ou $\overline{G}$ admitem K_3 como subgrafo (ou seja, G ou $\overline{G}$ contêm um triângulo).*

Prova. Seja v um vértice de G. A soma dos graus de v nos grafos G e $\overline{G}$ é 5. Portanto, num dos grafos G ou $\overline{G}$, v está unido com, pelo menos, outros 3 vértices. Suponhamos, sem perda de generalidade, que isto se passa em G, isto é, que há 3 vértices em G unidos a v por uma aresta. Se dois destes vértices forem adjacentes em G então eles formam com v um triângulo. Se, pelo contrário, não houver arestas em G entre quaisquer dois destes 3 vértices então eles são adjacentes em $\overline{G}$ e formam, pois, um triângulo em $\overline{G}$. □

Utilizando este teorema podemos provar que se 6 pessoas participam numa festa, então 3 delas conhecem-se mutuamente ou desconhecem-se mutuamente (basta traduzir esta situação por um grafo com 6 vértices, representando as 6 pessoas, fazendo dois vértices adjacentes se as pessoas que representam se conhecem).

Embora seja muito conveniente representar um grafo por um diagrama de pontos ligados por arestas, tal representação pode ser inconveniente se a pretendermos armazenar em computador. Um modo alternativo de representar um grafo simples é por listagem dos vértices adjacentes a cada vértice do grafo. Por exemplo, o grafo



pode ser representado por

$$\begin{aligned} u: &\quad v, w \\ v: &\quad u, w, y \\ w: &\quad v, x, u \\ x: &\quad w, y, v \\ y: &\quad v, x \end{aligned}$$

Contudo as representações mais úteis são as que usam matrizes. Seja G um grafo com vértices $v_1, v_2, \ldots, v_n$. A *matriz de adjacência* de G é a matriz $A = [\alpha_{ij}]$, de ordem $n \times n$, onde α_{ij} é o número de arestas que ligam o vértice v_i ao vértice v_j. Se G tiver arestas $a_1, a_2, \ldots, a_m$, a *matriz de incidência* de G é a matriz $B = [\beta_{ij}]$, de ordem $n \times m$, onde $\beta_{ij} = 1$ caso a_j seja incidente em v_i e $\beta_{ij} = 0$ caso contrário.

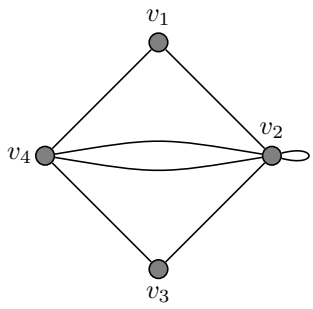
Por exemplo,

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$

e

$$B = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

são as matrizes de adjacência e de incidência do grafo



Observações. (1) Toda a matriz de adjacência é simétrica. Se o grafo não possuir lacetes então os elementos da diagonal principal são nulos. No caso do grafo não possuir arestas múltiplas as entradas da matriz só podem tomar os valores 0 e 1. As matrizes de adjacência representam de forma completa os grafos, na medida em que é possível recuperar toda a informação sobre um grafo a partir da sua matriz de adjacência. Toda a matriz de números inteiros positivos, simétrica, determina um grafo.
(2) Se na matriz de adjacência de um grafo G fizermos uma troca de colunas acompanhada da respectiva troca de linhas, isso equivale, no grafo G, a renumerar os seus vértices.

(3) Uma matriz de incidência de um grafo sem lacetes tem em cada coluna exactamente dois elementos não nulos. No caso de haver lacetes, a respectiva coluna possui só um elemento não nulo.
(4) Toda a matriz de elementos no conjunto $\{0,1\}$, tal que em cada coluna há entre um e dois elementos não nulos, determina um grafo.

Num grafo G, chama-se *caminho* a uma sequência

$$v_0, a_1, v_1, a_2, v_2, \ldots, v_{m-1}, a_m, v_m$$

com $v_i \in V(G)$ $(i \in \{0,1,\ldots,m\})$ e $a_j = \{v_{j-1}, v_j\} \in A(G)$ $(j \in \{1,2,\ldots,m\})$. O caminho diz-se *fechado* caso $v_0 = v_m$. Se $v_0 \neq v_m$ diz-se *aberto*. Se todas as arestas são distintas, diz-se um *caminho sem repetição de arestas*. Se todos os vértices forem distintos (com excepção do primeiro e do último no caso de caminhos fechados), diz-se um *caminho sem repetição de vértices*.

Note-se que um grafo é conexo se e só se todo o par de vértices estiver ligado por um caminho sem repetição de vértices. Um subgrafo S de G é uma componente de G se for conexo e se não existir um subgrafo S_1 de G que seja conexo e tal que $S_1 \neq S$ e S é subgrafo de S_1 (isto é, S é um subgrafo conexo *maximal* de G).

Por vezes denotaremos abreviadamente o caminho

$$v_0, a_1, v_1, a_2, v_2, \ldots, v_{m-1}, a_m, v_m$$

por $v_0\, v_1\, v_2\, \ldots\, v_m$. O seu *comprimento* é o número m de arestas que possui. Um caminho com um só vértice tem comprimento zero.

Um caminho fechado sem repetição de vértices, de comprimento $m \geq 1$, é chamado *ciclo*. Qualquer lacete ou par de arestas múltiplas é um ciclo.

Proposição. *Seja G um grafo com vértices $v_1, v_2, \ldots, v_n$ e matriz de adjacência A. O elemento na linha i e coluna j de A^m $(m \in \mathbb{N})$ é o número de caminhos de comprimento m unindo v_i e v_j.*

Prova. Basta reparar que esse elemento é igual a

$$\sum_{k_{m-1}=1}^{n} \sum_{k_{m-2}=1}^{n} \cdots \sum_{k_2=1}^{n} \sum_{k_1=1}^{n} a_{ik_1} a_{k_1k_2} a_{k_2k_3} \cdots a_{k_{m-2}k_{m-1}} a_{k_{m-1}j}.$$

□

Em particular, para $i \neq j$, o elemento (i,j) de A^2 é igual ao número de caminhos, sem repetição de arestas, de comprimento 2, unindo v_i e v_j.

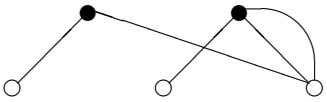
Se G não contiver arestas múltiplas, o elemento (i,i) de A^2 é igual ao grau de v_i. De facto, neste caso, este elemento é dado por

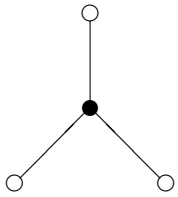
$$\sum_{j=1}^{n} a_{ij} a_{ji} = \sum_{j=1}^{n} a_{ij}^2 = \sum_{j=1}^{n} a_{ij}$$

(pois, não havendo arestas múltiplas, $a_{ij} \in \{0,1\}$) e $\sum_{j=1}^{n} a_{ij} = g(v_i)$.

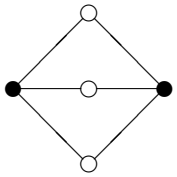
Um *grafo bipartido* G é um grafo cujo conjunto de vértices admite uma partição em dois subconjuntos não vazios, V_1 e V_2, de tal modo que toda a aresta de G é incidente num elemento de V_1 e noutro de V_2. Se todo o vértice de V_1 estiver ligado por uma (e uma só) aresta a cada vértice de V_2, G diz-se um *grafo bipartido completo*. Neste caso, se $|V_1| = m$ e $|V_2| = n$, G denota-se por $K_{m,n}$. Se $|V_1| = 1$, G diz-se uma *estrela*.

Apresentemos alguns exemplos de grafos bipartidos:

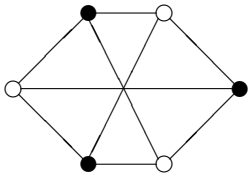




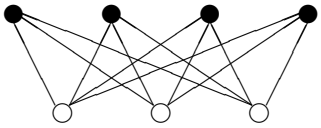
$K_{1,3}$



$K_{2,3}$



$K_{3,3}$



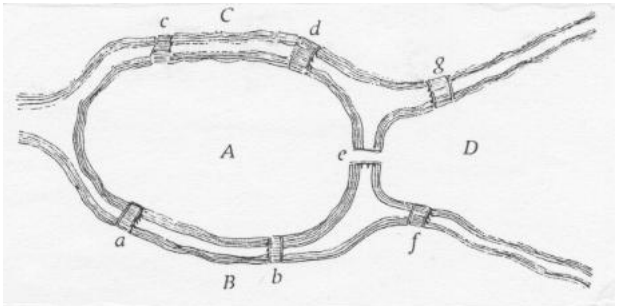
$K_{4,3}$

Proposição. *Se G é um grafo bipartido, cada ciclo de G tem comprimento par.*

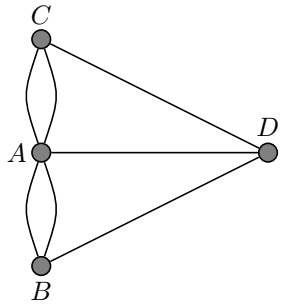
Prova. Seja $V(G) = V_1 \cup V_2$ uma partição de $V(G)$ em dois subconjuntos não vazios, tais que toda a aresta de G une um elemento de V_1 com um de V_2. Consideremos um ciclo $v_1 v_2 \dots v_m v_1$ e suponhamos (sem perda de generalidade) que $v_1 \in V_1$. Então todos os vértices de índice ímpar do ciclo estão em V_1 e os de índice par pertencem a V_2. Como v_m tem que estar em V_2, m é par. □

3.2. Grafos eulerianos

Recordemos o problema das pontes de Königsberg onde se pergunta se será possível atravessar cada uma das 7 pontes na figura



exactamente uma vez e voltar ao ponto de partida. Isto é equivalente a perguntar se no grafo



existe um caminho fechado, sem repetição de arestas, contendo todas as arestas.

Um grafo diz-se *euleriano* se admite um caminho fechado sem repetição de arestas, contendo todas as arestas. Designa-se esse caminho por *caminho euleriano*.

Portanto a questão que se põe é a de saber se o grafo acima é euleriano. Problemas sobre grafos eulerianos aparecem frequentemente em passatempos recreativos (um problema típico é o de saber se determinada figura geométrica pode ser desenhada sem levantar a ponta do lápis do papel e sem passar por nenhuma linha mais do que uma vez).

Lema. *Se G é um grafo no qual o grau de qualquer vértice é pelo menos 2, G contém um ciclo.*

Prova. Se G possui lacetes ou arestas múltiplas, o resultado é óbvio. Podemos pois assumir que G é simples. Seja v um vértice de G e construamos um caminho $v\, v_1\, v_2\, \ldots$, escolhendo v_1 entre os vértices adjacentes a v e, para cada $i > 1$, escolhendo v_{i+1} entre os vértices adjacentes a v_i diferentes de v_{i-1} (a existência de tal vértice é garantida pela hipótese). Como G contém somente um número finito de vértices, teremos que a dada altura ter como única hipótese a escolha de um vértice que já o tinha sido anteriormente. Se v_k for o primeiro destes vértices então a parte do caminho entre as duas ocorrências de v_k é o ciclo requerido. □

Todo o grafo euleriano é obviamente conexo. O seguinte teorema, demonstrado por Euler em 1736, permitindo a resolução imediata do problema das pontes de Königsberg, caracteriza os grafos conexos que são eulerianos.

Teorema. [Euler (1736)] *Um grafo conexo G é euleriano se e só se o grau de qualquer vértice de G for par.*

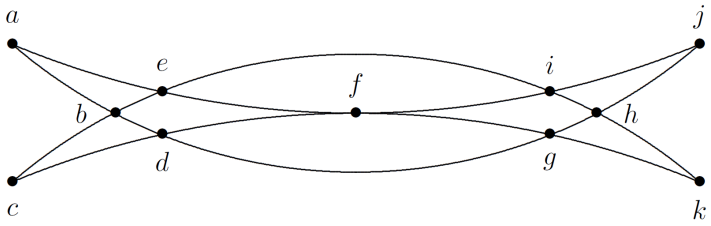
Prova. Seja E um caminho euleriano em G. Cada vez que um vértice aparece em E tem duas arestas incidentes. Como cada aresta ocorre precisamente uma vez em E, o grau de cada vértice é par.

Provaremos a recíproca por indução sobre o número de arestas de G. Suponhamos então que o grau de cada vértice de G é par. O caso em que G não possui arestas é trivial. Portanto, como hipótese de indução, admitiremos que o resultado é válido se G possuir menos de n arestas e nessas condições provaremos que o resultado é válido no caso de G possuir n arestas ($n \geq 1$).

Como G é conexo, cada vértice terá pelo menos grau 2 e, portanto, pelo Lema, G contém um ciclo C. Se C contiver todas as arestas de G, a prova está terminada. Senão, removamos de G todas as arestas de C, formando um novo grafo H, eventualmente desconexo, com menos arestas que G e no qual todo o vértice continua a ter grau par. Pela hipótese de indução, cada componente de H possui um caminho euleriano. Como cada componente de H possui pelo menos um vértice em comum com C (pela conexidade de G) obtemos o caminho euleriano de G seguindo as arestas de C até um vértice não isolado de H ser alcançado, traçando o caminho euleriano da componente de H que contêm tal vértice e, de seguida, continuando pelas arestas de C até encontrar um vértice não isolado pertencendo a outra componente de H, traçando o caminho euleriano desta, e assim sucessivamente. O processo terminará quando voltarmos ao vértice inicial. □

É importante notar que a demonstração do Teorema de Euler nos dá um algoritmo para construirmos um caminho euleriano num grafo euleriano. O seguinte exemplo mostra-nos como pode ser utilizada para resolver os tais passatempos com lápis e papel referidos anteriormente.

Exemplo. *Será que se consegue desenhar a cimitarra de Mohammed sem levantar a ponta do lápis do papel e sem passar por nenhum traço mais do que uma vez?*

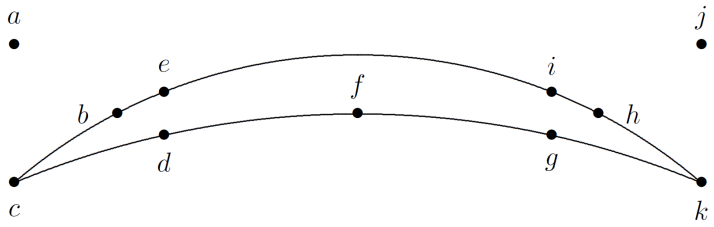


É claro que o Teorema nos diz imediatamente que tal é possível, pois o grau de cada vértice de G é par. Usando a respectiva demonstração podemos obter um caminho euleriano em G, que nos diz como realizar tal desenho:

Primeiro consideramos o ciclo

$$a\,b\,d\,g\,h\,j\,i\,f\,e\,a.$$

O subgrafo H obtido por remoção das arestas contidas neste ciclo é o grafo



que é evidentemente euleriano. Por exemplo,

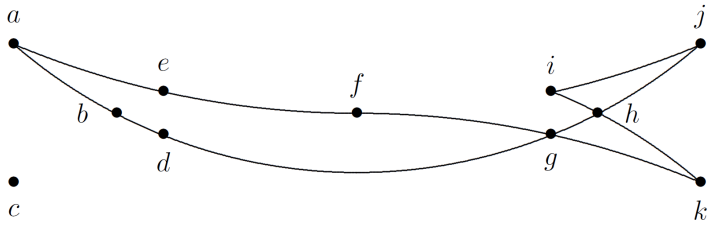
$$c\,d\,f\,g\,k\,h\,i\,e\,b\,c$$

é um caminho euleriano em H. Pondo este circuito no ciclo inicial, no lugar apropriado, produz o caminho euleriano

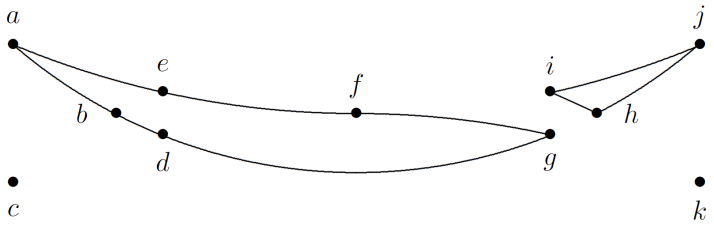
$$a\,b\,c\,d\,f\,g\,k\,h\,i\,e\,b\,d\,g\,h\,j\,i\,f\,e\,a:$$

Ciclo C	a	b							d	g	h	j	i	f	e	a
Caminho		c	d	f	g	k	h	i	e	b						
euler. em H																

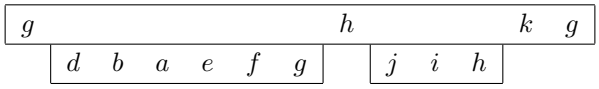
Claro que, como as escolhas dos ciclos em cada passo não são únicas, muitas outras soluções se podem obter. Por exemplo, se tomarmos inicialmente o ciclo $i\,f\,d\,c\,b\,e\,i$, o subgrafo H é, neste caso,



Para determinar um caminho euleriano de H consideremos o ciclo $g\,h\,k\,g$. Neste caso obtemos o grafo



cujas duas componentes têm caminhos eulerianos $a\,e\,f\,g\,d\,b\,a$ e $j\,i\,h$. Portanto



é um caminho euleriano de H e, consequentemente,

$$i\,h\,k\,g\,d\,b\,a\,e\,f\,g\,h\,j\,i\,f\,d\,c\,b\,e\,i$$

é um caminho euleriano de G.

A prova do Teorema de Euler pode ser ligeiramente modificada de modo a obtermos o seguinte resultado:

Corolário. *Um grafo conexo é euleriano se e só se o conjunto das suas arestas pode ser particionado em ciclos.*

Prova. Seja G um grafo euleriano. O caso em que G não possui arestas é trivial. Sendo G conexo e tendo pelo menos uma aresta, todo o seu vértice tem, pelo menos, grau 2. Portanto, pelo Teorema de Euler, possui um ciclo C_1. Retirando a G as arestas de C_1 obtemos um subgrafo gerador G_1 cujos vértices têm ainda todos grau par. Se G_1 não tem arestas, está terminada a demonstração desta implicação. Caso contrário, G_1 tem um ciclo C_2 e a repetição do argumento anterior conduz-nos a um grafo G_2, subgrafo gerador de G_1, cujos vértices têm grau par. Se G_2 não tem arestas terminamos, caso contrário repete-se o argumento. E continuamos com este raciocínio sucessivamente até obtermos um grafo G_n totalmente desconexo (isto é, sem arestas). Aí teremos uma partição das arestas de G em n ciclos.

Reciprocamente, suponhamos que o conjunto das arestas de G admite uma partição em ciclos. Seja C_1 um desses ciclos. Se G se reduz a este ciclo então, evidentemente, G é euleriano. Senão existe outro ciclo C_2 da partição, com um vértice comum a C_1. Sejam

$$C_1 \equiv v_0\, v_1\, \ldots\, v_n,$$

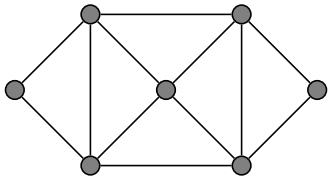
com $v_0 = v_n = v$, e

$$C_2 \equiv w_0\, w_1\, \ldots\, w_m$$

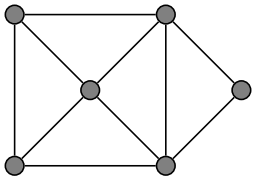
com $w_0 = w_m = v$. Então $v_0\, v_1\, \ldots\, v_n\, w_1\, \ldots\, w_m$ é um caminho sem repetição de arestas, fechado, contendo todos os vértices e arestas de C_1 e C_2. Se G se reduzir aos ciclos C_1 e C_2 a demonstração está terminada. Não sendo esse o caso, bastará continuarmos com um raciocínio análogo, agora para três ciclos C_1, C_2 e C_3.

Continuando este processo podemos construir um caminho fechado sem repetição de arestas, contendo todas as arestas e vértices de G. □

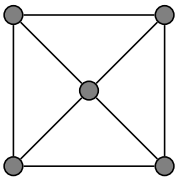
Por vezes, a questão que se põe é a da averiguação da existência de um *caminho semi-euleriano*, isto é, um caminho aberto sem repetição de arestas, contendo todas as arestas. Os grafos onde tal caminho exista chamam-se *grafos semi-eulerianos*. Por exemplo,



é euleriano, mas não é semi-euleriano, enquanto que



é semi-euleriano, mas não é euleriano, e



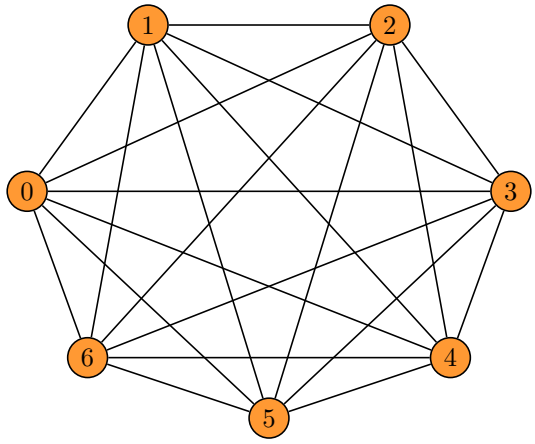
não é uma coisa nem outra.

Corolário. *Um grafo conexo é semi-euleriano se e só se possuir exactamente dois vértices de grau ímpar. Neste caso o caminho semi-euleriano inicia-se num desses vértices e termina no outro.*

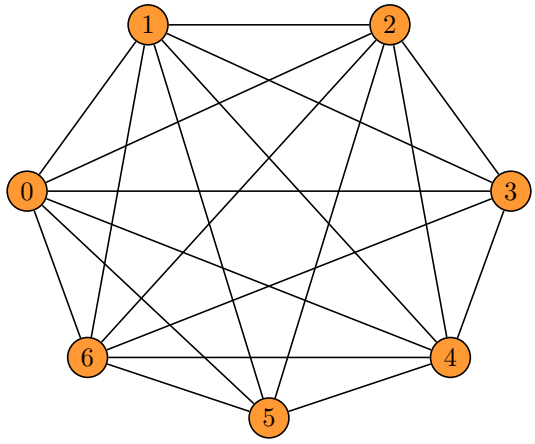
Prova. Suponhamos que G possui um caminho semi-euleriano E começando num vértice v e terminando num vértice w. Como $v \neq w$ é claro que v e w têm ambos grau ímpar. Cada vez que um dos outros vértices aparece em E tem 2 arestas incidentes. Como cada aresta ocorre precisamente uma vez em E, o grau desses vértices é par.

Reciprocamente, suponhamos que G é conexo e possui exactamente 2 vértices, v e w, de grau ímpar. Consideremos o grafo G^* que se obtém de G por junção de uma nova aresta ligando v a w. A este novo grafo podemos aplicar o Teorema de Euler e concluir que admite um caminho euleriano. Apagando deste caminho a aresta previamente adicionada a G obtemos um caminho semi-euleriano ligando v e w, como desejávamos. □

Exemplo. O Dominó tem 28 peças. Seguindo a sua regra básica, é possível dispor as 28 peças na mesa, formando um circuito fechado. Isso pode ser comprovado rapidamente com a ajuda de grafos. Representando cada peça por uma aresta (podemos ignorar os *dobles*, peças com igual número de pintas nas duas metades, pois isso é irrelevante para o problema) o grafo correspondente ao Dominó é o K_7:



(Se quisermos considerar os dobles basta acrescentar um lacete a cada vértice.) Como todos os vértices têm grau par, existe um circuito euleriano. Este facto está na base de um truque de magia muito conhecido: o Mágico esconde uma peça e entrega as 27 peças restantes a um voluntário e pede a este que as coloque numa sequência, respeitando a regra básica do Dominó, sem a mostrar ao Mágico. Este consegue adivinhar as pintas das extremidades de tal sequência! Porquê? Por exemplo, se o Mágico guardou a peça (3,5), o grafo correspondente às 27 peças restantes é semi-euleriano, somente com dois vértices de grau ímpar: o 3 e o 5. Claro que são estes os extremos da sequência (caminho semi-euleriano).

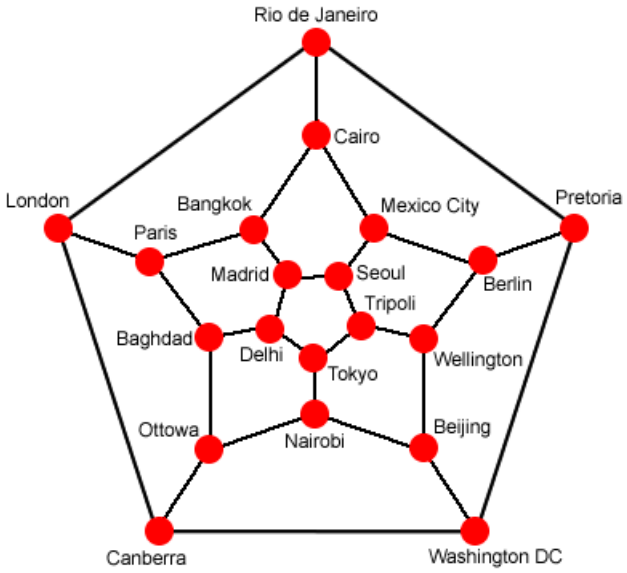


3.3. Grafos hamiltonianos

Em 1857, o matemático irlandês W. R. Hamilton inventou um *puzzle*[11] cujo objectivo é o de determinar um certo caminho através das arestas de um dodecaedro.

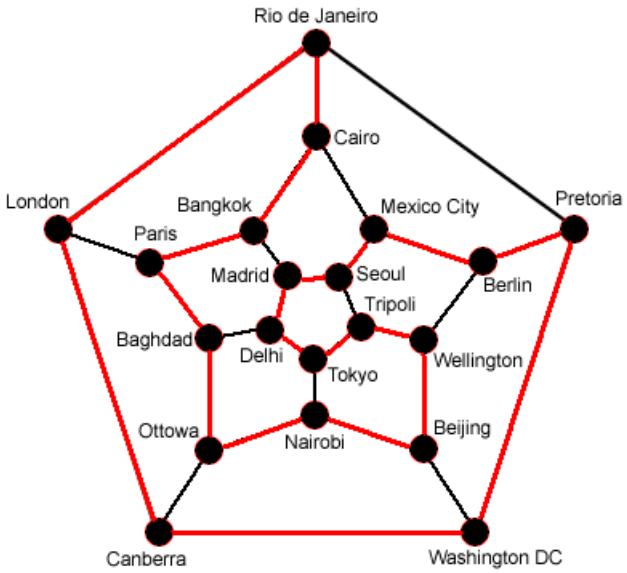
Os vértices do dodecaedro representam 20 cidades importantes: Bruxelas, Cantão, Deli, etc., acabando em Zanzibar. Cada vértice é marcado por um grampo, e um pedaço de fio é usado para ligar os grampos uns aos outros, para indicar um caminho. Um circuito completo, passando por cada cidade uma única vez era chamado "uma viagem à volta do mundo".

Os vértices e as arestas do dodecaedro podem ser representados no plano pelo seguinte grafo:



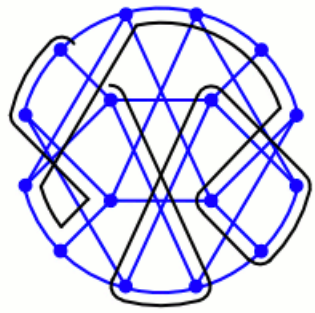
No grafo seguinte, as arestas a vermelho determinam um caminho fechado que, começando num vértice arbitrário, permite-nos voltar a ele depois de visitarmos cada um dos outros vértices uma vez. Este caminho mostra como o *puzzle* de Hamilton tem resposta afirmativa.

[11] Conhecido por "Viagem à volta do mundo" ou "Dodecaedro do viajante".

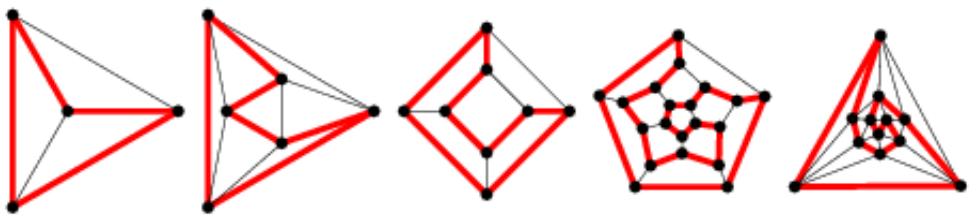


Na terminologia moderna dos grafos, um circuito destes é chamado um *caminho hamiltoniano*. Esta noção de caminho hamiltoniano é relevante para um problema de grande importância prática, chamado "problema do caixeiro-viajante", que abordaremos mais adiante.

O *puzzle* de Hamilton pode evidentemente ser formulado para qualquer outro grafo: um grafo diz-se *hamiltoniano* se admite um caminho hamiltoniano. Por exemplo, no seguinte grafo (a azul) está traçado um caminho hamiltoniano que mostra que o grafo é hamiltoniano.



Mais exemplos:



Se o número de vértices de um grafo hamiltoniano G for n, então qualquer caminho hamiltoniano em G tem comprimento n.

Apesar da semelhança entre caminhos hamiltonianos e caminhos eulerianos não se conhecem condições necessárias e suficientes para que um grafo seja hamiltoniano. A procura de tais caracterizações é um dos mais importantes problemas da Teoria dos Grafos, ainda por resolver. Com efeito, muito pouco é conhecido sobre grafos hamiltonianos. Os únicos resultados conhecidos são do tipo "se G possui arestas suficientes então G é hamiltoniano". Provavelmente o mais importante destes resultados é devido a G. A. Dirac e conhecido como Teorema de Dirac. Vamos deduzi-lo a partir do seguinte resultado de O. Ore:

Teorema de Ore (1960). *Seja $n \geq 3$. Se G é um grafo simples com n vértices e $g(v)+g(w) \geq n$ para cada par de vértices não adjacentes v e w, então G é hamiltoniano.*

Prova. Observemos antes de mais que G é conexo. Se não fosse, teria pelo menos duas componentes. Suponhamos que uma tinha n_1 vértices e a outra n_2, onde $n_1 + n_2 \geq n$. Sendo v um vértice da primeira componente e w um vértice da outra, v e w não estariam ligados por nenhuma aresta. Além disso, $g(v) \geq n_1 - 1$ e $g(w) \geq n_2 - 1$, pelo que $g(v) + g(w) \geq n_1 + n_2 - 2 \geq n - 2$, o que contradiz a hipótese. Portanto G é conexo.

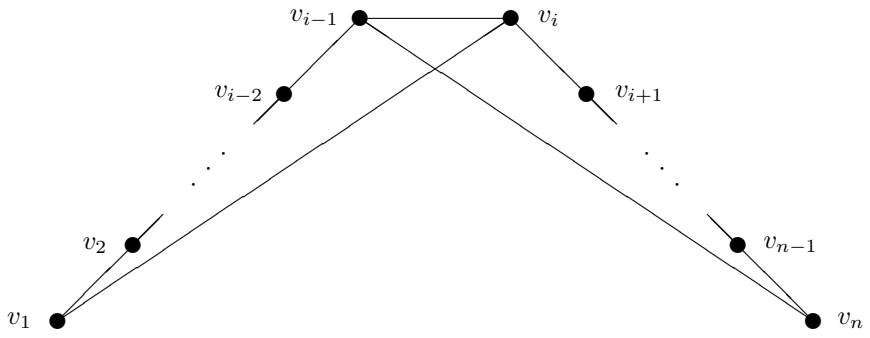
Seja agora $C = v_1\, v_2\, \ldots\, v_r$ um caminho sem repetição de vértices com o maior comprimento possível.

CASO 1: $r = n$ Se v_1 e v_n estiverem ligados por uma aresta então $v_1\, v_2\, \ldots\, v_n\, v_1$ é um caminho hamiltoniano.

Se v_1 e v_n não estiverem ligados por uma aresta, seja $p \geq 1$ o número de vértices à qual v_1 está ligado e $q \geq 1$ o número de vértices à qual v_2 está ligado. Por hipótese $p+q \geq n$. Se existir um vértice v_i, entre os vértices $v_2, \ldots, v_n$, à qual v_1 esteja ligado e tal que v_n está ligado a v_{i-1} então

$$v_1\, v_i\, v_{i+1}\, \ldots\, v_n\, v_{i-1}\, v_{i-2}\, \ldots\, v_2\, v_1$$

é um caminho hamiltoniano:



Mostremos que tal vértice terá que existir, o que completará a prova deste caso. Se v_n não estivesse unido a nenhum dos vértices de C imediatamente precedentes a um dos p vértices à qual v_1 está ligado, então os q vértices à qual v_n está ligado fariam parte de um conjunto de $(n-1)-p$ vértices. Consequentemente $(n-1)-p \geq q$, ou seja, $n-1 \geq p+q$, o que contradiz o facto $p+q \geq n$.

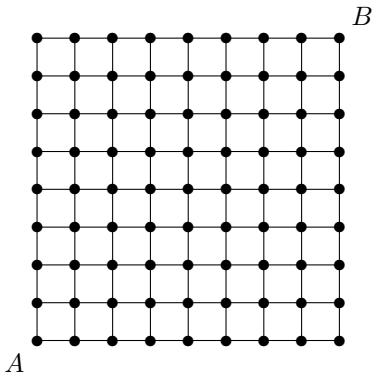
CASO 2: $r < n$ Suponhamos que v_1 está ligado a um vértice v que não é um vértice de C. Então $v\, v_1\, v_2\, \ldots\, v_r$ seria um caminho sem repetição de arestas, de comprimento maior do que C. Portanto v_1 está ligado somente a vértices de C. Analogamente, poderemos concluir que v_r está também ligado somente a vértices de C. Podemos então repetir um raciocínio análogo ao realizado no caso 1 e concluir que existe um ciclo C' de comprimento r, $v_1\, v_2\, \ldots\, v_r\, v_1$ ou

$$v_1\, v_i\, v_{i+1}\, \ldots\, v_r\, v_{i-1}\, v_{i-2}\, \ldots\, v_1,$$

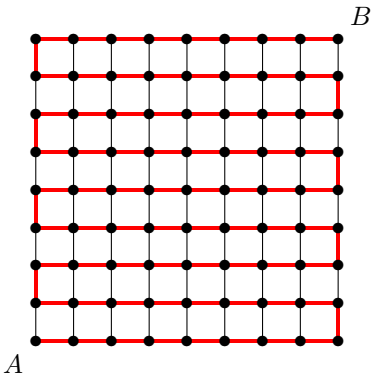
conforme o caso. Representemo-lo por $w_1\, w_2\, \ldots\, w_r\, w_1$. Como G é conexo, existe um vértice v que não está em C' mas que está unido a algum vértice w_j. Então $v\, w_j\, \ldots\, w_r\, w_1\, \ldots\, w_{j-1}$ é um caminho sem repetição de vértices de comprimento $r+1$, o que não pode existir, da maneira como tomámos C. Em conclusão, o caso 2 nunca pode acontecer. □

Corolário. [Dirac (1952)] *Seja $n \geq 3$. Se G é um grafo simples com n vértices e $g(v) > \frac{n}{2}$ para qualquer vértice v, então G é hamiltoniano.* □

Para terminar recordemos o Problema (A5) da Introdução. Representando cada cela por um vértice e cada porta por uma aresta obtemos o grafo



O problema resume-se à existência ou não de um caminho unindo A e B, passando exactamente uma vez por cada vértice, ou seja, um caminho *semi-hamiltoniano* entre A e B. Não é difícil encontrar uma solução:



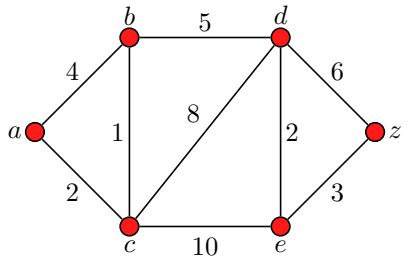
3.4. Problemas famosos

Os avanços mais importantes da Teoria dos Grafos têm sido motivados, quase sempre, pela tentativa de resolução de problemas práticos muito específicos — Euler e o problema das pontes de Königsberg, Cayley e a enumeração de compostos químicos, Kirchoff e problemas de redes eléctricas, etc.

Abordemos sucintamente alguns desses problemas com importância na vida real.

O problema do caminho mais curto. Consideremos o seguinte problema:

Qual é o caminho mais curto de a para z no grafo seguinte ?



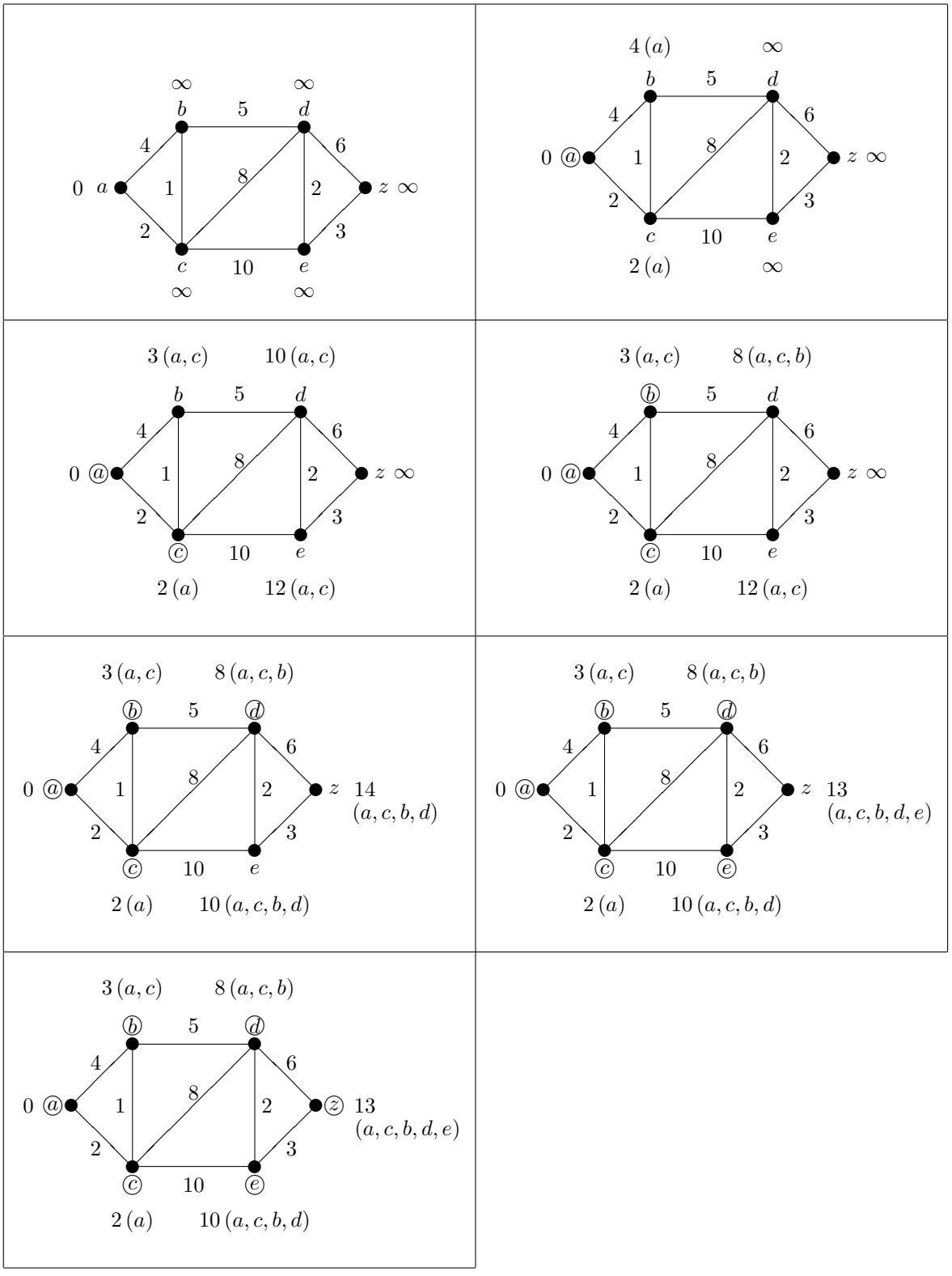
Notemos que noutros problemas, os números no grafo poderão representar, não os comprimentos das estradas, mas sim os tempos gastos a percorrê-las ou os custos de as percorrer. Portanto, possuindo um algoritmo para resolver o problema do caminho mais curto, este algoritmo pode também ser utilizado para determinar o caminho mais rápido, o mais económico, etc.

Nestes problemas o nosso mapa pode ser visto como um grafo conexo no qual um número não negativo é atribuído a cada aresta. Tais grafos chamam-se *grafos com pesos* e o número atribuído a cada aresta a chama-se o *peso* de a.

Existe um algoritmo eficiente, isto é, um procedimento com um número finito de passos que rapidamente nos conduz à solução. A ideia deste algoritmo consiste em movermo-nos ao longo do grafo, da esquerda para a direita, associando a cada vértice v um número $L(v)$ indicando a distância mínima entre a e v. Isto significa que, quando chegarmos por exemplo ao vértice d, $L(d)$ é o menor dos números $L(b)+5$ ou $L(c)+8$.

Para aplicar o algoritmo começamos por definir $L(a)=0$ e damos a b, c, d, e as etiquetas temporárias $L(b)=L(c)=L(d)=L(e)=\infty$. Em seguida consideramos os vértices adjacentes a a. O vértice b fica com a etiqueta temporária $L(a)+4=4$ e o vértice c com $L(a)+2=2$. Consideramos a menor destas, que será a etiqueta definitiva de c: $L(c)=2$. Em seguida consideramos os vértices adjacentes a c ainda não etiquetados definitivamente. O vértice e fica etiquetado com $L(c)+10=12$, o vértice d com $L(c)+8=10$ e podemos descer a etiqueta de b para $L(c)+1=3$. A menor destas etiquetas é agora 3 (em b). Será esta a etiqueta permanente de b. Agora consideramos os vértices adjacentes a b. O vértice d desce a sua etiqueta temporária para $L(b)+5=8$. A menor das etiquetas temporárias é agora 8 (em d). Escreveremos então $L(d)=8$. Continuando deste modo, obtemos sucessivamente as etiquetas permanentes $L(e)=10$ e $L(z)=13$. Portanto o caminho mais curto entre a e z mede 13, que é o caminho $a\,c\,b\,d\,e$.

Resumindo:



Este algoritmo deve-se a Dijkstra (1959) e a sua formulação geral diz o seguinte:

Seja $V(G) = \{v_1, \ldots, v_n\}$. Denotemos por $c(v_i, v_j)$ o comprimento da aresta entre v_i e v_j. O algoritmo começa por etiquetar v_1 com um zero e os outros vértices com ∞. Usamos a notação $L_0(v_1) = 0$ e $L_0(v) = \infty$ para estas etiquetas. Estes são os comprimentos dos caminhos mais curtos de v_1 aos diferentes vértices que contêm somente o vértice v_1 (∞ indica simplesmente que não existe nenhum caminho nessas condições).

O algoritmo prossegue formando uma família de vértices específica. Seja S_k tal família após k iterações. Começamos com $S_0 = \emptyset$. O conjunto S_k forma-se a partir de S_{k-1} acrescentando-lhe o vértice w com menor etiqueta entre os que não estão em S_{k-1}. Uma vez acrescentado este vértice a S_{k-1}, actualizam-se as etiquetas dos vértices que não pertencem a S_k de modo a que essa etiqueta $L_k(v)$ (a etiqueta do vértice v no k-ésimo passo) seja o comprimento do caminho mais curto de v_1 a v que contem somente vértices de S_k. Notemos que

$$L_k(v) = \min\{L_{k-1}(v), L_{k-1}(w) + c(w, v)\}.$$

Algoritmo de Dijkstra (1959). Determinação do caminho mais curto do vértice v_1 aos outros vértices do grafo, onde os comprimentos das arestas são positivos:

1. Faça
$$c(v_i, v_j) = \begin{cases} \text{Comprimento da aresta } v_iv_j & \text{caso ela exista} \\ \infty & \text{caso contrário} \end{cases}$$
2. Faça $L(v_1) := 0$, $L(v_2) := \infty, \ldots, L(v_n) := \infty$, $S := \emptyset$.
3. Enquanto $v_n \notin S$ faça
   (a) $w :=$ vértice em $V(G) \setminus S$ com etiqueta $L(w)$ mínima;
   (b) $S := S \cup \{w\}$;
   (c) Para todos os vértices em $V(G) \setminus S$, se $L(w) + c(w, v) < L(v)$ faça $L(v) := L(w) + c(w, v)$.
4. O comprimento do caminho mais curto de v_1 a v_n é então $L(v_n)$.

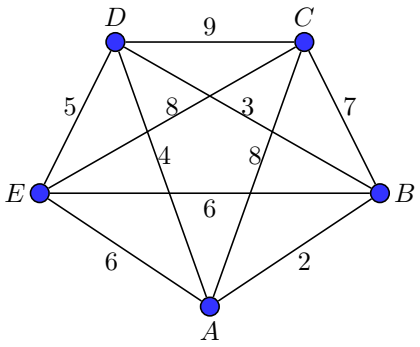
Existem muitas variantes deste algoritmo para aplicação, por exemplo, aos digrafos, à determinação do caminho mais longo, etc.

O problema do carteiro chinês. Neste problema, originalmente estudado pelo matemático chinês Mei-Ku Kwan, em 1962, um carteiro tem que distribuir cartas pelas casas de um bairro, voltando depois ao ponto de partida na estação dos correios. Qual é a menor distância que terá o carteiro de percorrer? Evidentemente terá de percorrer cada rua pelo menos uma vez, mas deverá evitar percorrê-las mais do que uma vez.

Este problema pode ser formulado em termos de grafos com pesos, onde o grafo corresponde à rede de ruas e o peso de cada aresta é o comprimento da respectiva rua. Se o grafo tiver um caminho euleriano, o carteiro pode iniciar esse caminho na estação dos correios, percorrê-lo e voltar aos correios. Nenhum outro trajecto tem comprimento menor. Tal caminho euleriano

pode ser obtido, como vimos, pelo algoritmo do Teorema de Euler. Se o grafo não for euleriano, o problema é muito mais complicado, embora se conheça um algoritmo eficiente para o resolver, que não apresentaremos aqui. A ideia é acrescentar a G cópias de algumas das suas arestas de modo a obter um multi-grafo que tenha um caminho euleriano. Assim, o problema de determinar o trajecto óptimo para o carteiro é equivalente à determinação do menor número de cópias de arestas de G a juntar a G de maneira a obter um multigrafo com um caminho euleriano.

O problema do caixeiro-viajante. Neste problema, como já referimos, um caixeiro-viajante pretende visitar várias cidades e voltar ao ponto de partida, percorrendo a menor distância possível. Por exemplo, se existirem 5 cidades A, B, C, D e E e as distâncias forem como em



o trajecto mais curto é $A \to B \to D \to E \to C \to A$ e mede 26.

Este problema pode, como estamos a ver, ser formulado em termos de grafos com pesos. Neste caso o que se pretende é encontrar um caminho hamiltoniano de menor peso possível.

Um método possível consiste em calcular a distância total de todos os caminhos hamiltonianos, mas isto torna-se muito pouco prático. Com efeito, qualquer trajecto começando e terminando na cidade C_1 corresponde a uma permutação das $n-1$ cidades restantes $C_2, \ldots, c_n$. Existem assim $(n-1)!$ trajectos diferentes. Como, para qualquer permutação $i_2 i_3 \ldots i_n$ dos números $2, 3, \ldots, n$, o trajecto $C_1 C_{i_2} c_{i_3} \ldots C_{i_n} C_1$ tem o mesmo comprimento que $C_1 C_{i_n} c_{i_{n-1}} \ldots C_{i_2} C_1$, será suficiente considerar

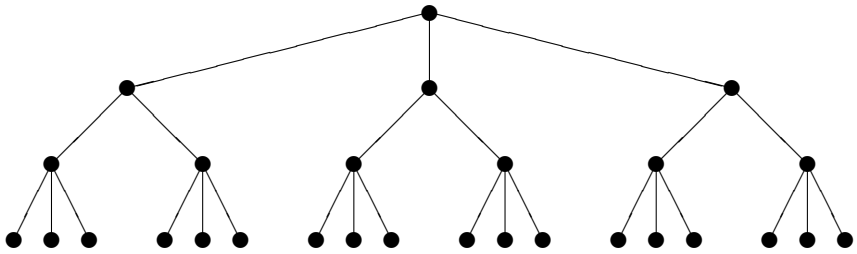
$$\frac{(n-1)!}{2}$$

trajectos. Mas, mesmo assim, este número pode ser muito elevado o que torna este método impraticável para mais do que 5 cidades. Por exemplo, no caso de 20 cidades, o número de caminhos hamiltonianos a avaliar é $19!/2 \approx 6 \times 10^{16}$. Outros algoritmos têm sido propostos mas levam muito tempo a executar. Na verdade, não se conhece nenhum algoritmo suficientemente geral e eficiente para determinar o trajecto mais económico[12].

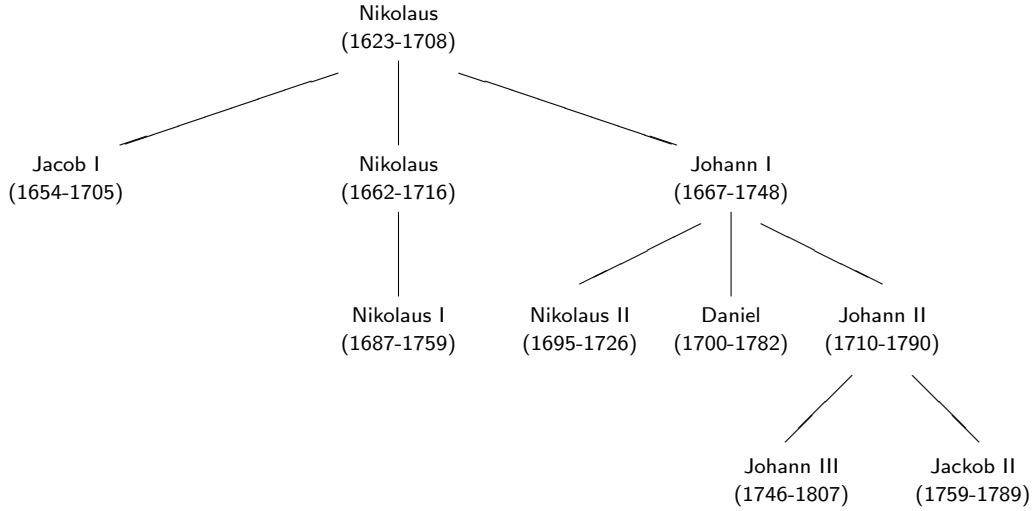
[12] Este problema do caixeiro-viajante é um exemplo de problema que tem desafiado os investigadores na procura de um bom algoritmo. Pertence a uma classe de problemas conhecidos como *NP-completos* ou *NP-difíceis*, para os quais não se acredita ser possível encontrar um algoritmo de complexidade polinomial. Uma das actividades mais importantes na Matemática Discreta é a procura de algoritmos eficientes que forneçam uma boa aproximação da solução óptima destes problemas. É o que acontece com o problema do caixeiro-viajante para o qual já existem alguns algoritmos heurísticos que fornecem rapidamente uma solução aproximada.

3.5. Árvores

Todos nós estamos familiarizados com a ideia de árvore genealógica:

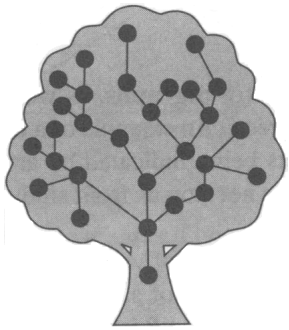


Tal diagrama é um grafo no qual os vértices representam membros da família e as arestas representam relações de parentesco (descendência). Por exemplo,

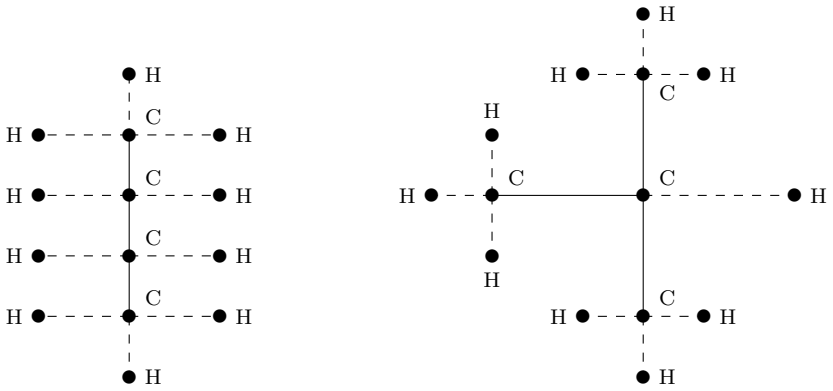


representa a famosa família de matemáticos suíços Bernoulli.

Os grafos que representam árvores genealógicas são exemplos de um tipo especial de grafo, que abordaremos nesta secção, chamado *árvore*.



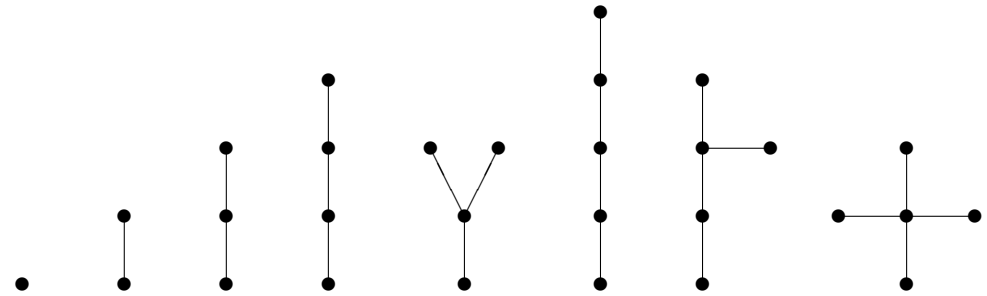
Outros exemplos de árvores são dados por algumas moléculas orgânicas — os vértices representando os átomos e as arestas as ligações entre eles:



A. Cayley foi o primeiro a estudar árvores de modo sistemático[13]. Mais tarde[14], aplicou esse estudo à química orgânica, mostrando a sua utilidade na enumeração de compostos químicos. Esta enumeração conduziu-o à descoberta de compostos desconhecidos.

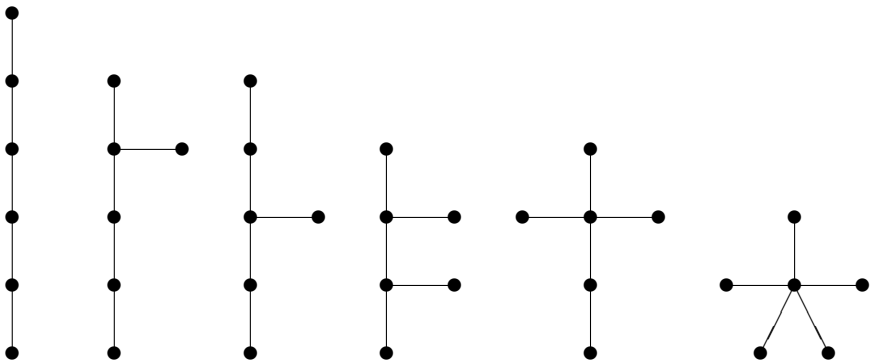
Por *árvore* entende-se um grafo conexo sem ciclos.

A figura seguinte contém todas as árvores estruturalmente diferentes (ou seja, não isomorfas) com 1, 2, 3, 4, 5 e 6 vértices.



[13]Nos artigos [A. Cayley, *On the theory of the analytical forms called trees*, Philosophical Magazine 13 (1857) 172-176] e [A. Cayley, *On the theory of the analytical forms called trees, part II*, Philosophical Magazine 18 (1859) 374-378].

[14]Nos artigos [A. Cayley, *On the mathematical theory of isomers*, Philosophical Magazine 47 (1874) 444-446] e [A. Cayley, *On the analytical forms called trees, with applications to the theory of chemical combinations*, Rep. Brit. Advance Sci. 45 (1875) 257-305].



(6)

Como veremos, as árvores têm "boas" propriedades. Muitas vezes, na tentativa de provar um resultado geral para grafos, começa-se por tentar prová-lo para árvores. De facto, existem muitas conjecturas que ainda não foram provadas para grafos arbitrários mas que já se sabe serem verdadeiras para as árvores.

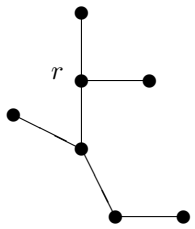
Em qualquer grafo conexo, dados dois vértices arbitrários distintos, existe sempre um caminho sem repetição de vértices ligando-os. O resultado seguinte diz-nos que as árvores são precisamente os grafos conexos nos quais cada par de vértices distintos está ligado por exactamente um caminho sem repetição de vértices:

Teorema 1. *Um grafo simples G é uma árvore se e só se quaisquer dois vértices de G estão ligados por um único caminho sem repetição de vértices.*

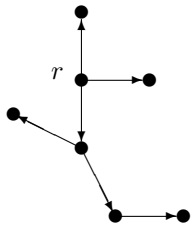
Prova. Seja G uma árvore e sejam x e y dois vértices de G. Como G é conexo, existe um caminho sem repetição de vértices que os liga. Resta provar que este caminho é único. Se existisse outro caminho, o caminho formado pela combinação do primeiro, de x para y, com o caminho de y para x obtido seguindo o segundo caminho na direcção de y para x, formaria um ciclo, o que seria uma contradição.

Reciprocamente, suponhamos que existe um único caminho sem repetição de vértices unindo quaisquer dois vértices de G. Então G é claramente conexo. Além disso, não poderá ter ciclos: se contivesse um ciclo, contendo os vértices x e y, existiriam evidentemente dois caminhos sem repetição de vértices unindo x a y (pois qualquer ciclo que passe por x e y é constituído por dois caminhos sem repetição de vértices, um unindo x a y, o outro unindo y a x). □

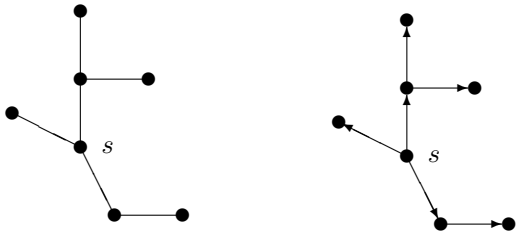
Fixando um vértice qualquer r de uma árvore é possível, usando o teorema anterior, dar uma direcção a todas as arestas do seguinte modo: como existe um único caminho de r para cada um dos restantes vértices do grafo, direccionamos cada aresta usando esses caminhos. Por exemplo, na árvore



fixando o vértice r indicado, obtemos



Este grafo dirigido diz-se uma *árvore com raiz* r. Outra escolha de raiz produzirá uma outra árvore com raiz:



Teorema 2. *Uma árvore com n vértices possui $n-1$ arestas.*

Prova. Fixando um vértice qualquer r, e construindo a respectiva árvore com raiz r, é evidente que existe uma correspondência bijectiva entre as arestas da árvore e os vértices diferentes de r (a cada seta corresponde o respectivo vértice terminal). Como há $n-1$ vértices diferentes de r, a árvore tem $n-1$ arestas. □

A terminologia para as árvores inspira-se na botânica e na genealogia. Seja G uma árvore com raiz r. Se v é um vértice diferente de r, o *pai* (ou *ascendente* de v é o único vértice u para o qual existe uma aresta dirigida de u para v. Nesse caso, v diz-se um *descendente* ou *filho* de u. Um vértice de G é uma *folha* se não tiver descendentes. Os vértices que têm descendentes dizem-se *vértices internos*.

G diz-se uma *árvore m-ária* se todo o vértice interno não tiver mais de m descendentes. No caso $m = 2$, a árvore chama-se uma *árvore binária*. A árvore diz-se uma *árvore m-ária plena* se todo o vértice interno tiver exactamente m descendentes.

Sabemos pelo Lema dos apertos de mão (Proposição 1 da Secção 2.1) que a soma dos graus dos vértices de um grafo é o dobro do número das arestas. Se G for uma árvore com vértices $v_1, v_2, \ldots, v_n$ e m arestas, então $m = n - 1$ pelo Teorema 2. Logo

$$\sum_{i=1}^{n} g(v_i) = 2m = 2(n-1).$$

Consequentemente, no caso de G não ser K_1, como não existem vértices isolados, existem pelo menos dois vértices de grau 1. Podemos assim afirmar que toda a árvore diferente de K_1 possui pelo menos dois vértices de grau 1.

Teorema 3. *Uma árvore m-ária plena com i vértices internos possui $n = m\,i + 1$ vértices.*

Prova. Todo o vértice, com excepção da raiz, é descendente de um vértice interno. Como cada um dos i vértices internos tem m descendentes, existem $m\,i$ vértices na árvore além da raiz. Assim, no total existem $m\,i + 1$ vértices. □

Sejam n o número de vértices, i o número de vértices internos e l o número de folhas de uma árvore com raiz. Se a árvore for m-ária plena é possível, a partir de qualquer um dos números n, i ou l determinar os outros dois:

Teorema 4. *Uma árvore m-ária plena com*

(a) *n vértices tem $i = \dfrac{n-1}{m}$ vértices internos e $l = \dfrac{(m-1)n+1}{m}$ folhas,*

(b) *i vértices internos tem $n = m\,i + 1$ vértices e $l = (m-1)i + 1$ folhas,*

(c) *l folhas tem $n = \dfrac{ml-1}{m-1}$ vértices e $i = \dfrac{l-1}{m-1}$ vértices internos.*

Prova. Evidentemente $n = i + l$. Esta igualdade, em conjunto com a do Teorema 3, permite provar facilmente as três afirmações. Provaremos somente a primeira, uma vez que as outras duas se podem provar de modo análogo:

Pelo Teorema 3, $n = m\,i + 1$, ou seja $i = \dfrac{n-1}{m}$. Então

$$l = n - i = n - \frac{n-1}{m} = \frac{(m-1)n+1}{m}.$$

□

4. Números inteiros

4.1. Aritmética modular

Se eu escrever

$$11 + 22 = 9$$

dirão *ele não sabe somar!*

Por outro lado, se eu disser *são 11 horas, daqui a 22 horas serão 33 horas* dirão *e além disso, não sabe que um dia tem só 24 horas!* São 11 horas, dentro de 22 horas serão $33 - 24 = 9$ horas.

Temos então que decidir,

$$11 + 22 = 33 \quad \text{ou} \quad 11 + 22 = 9\,?$$

Bem, na aritmética usual

$$11 + 22 = 33$$

mas quando calculamos as horas, $11 + 22 = 9$. Portanto, a aritmética que usamos para calcular as horas é uma aritmética um pouco diferente da habitual, na qual 24 conta como zero, isto é, $24 = 0$. Esta aritmética chama-se *aritmética módulo 24*. Para a distinguir da aritmética habitual escrevemos

$$11 +_{24} 22 =_{24} 9.$$

Note que isto é verdade porque $11 + 22 = 33 = 24 + 9$.

Quanto à multiplicação, $11 \times_{24} 22 =_{24} 2$ pois $11 \times 22 = 242 = (10 \times 24) + 2$. Para obter $242 = (10 \times 24) + 2$ basta fazer a divisão de 242 por 24:

$$\begin{array}{rl} 242 & \big|\,24 \\ 002 & \\ 02 & \end{array} \qquad\qquad \begin{array}{r|l} 242 & 24 \\ \cline{2-2} 240 & 10 \\ \cline{1-1} 2 & \end{array}$$

O quociente é 10 e o resto é 2.

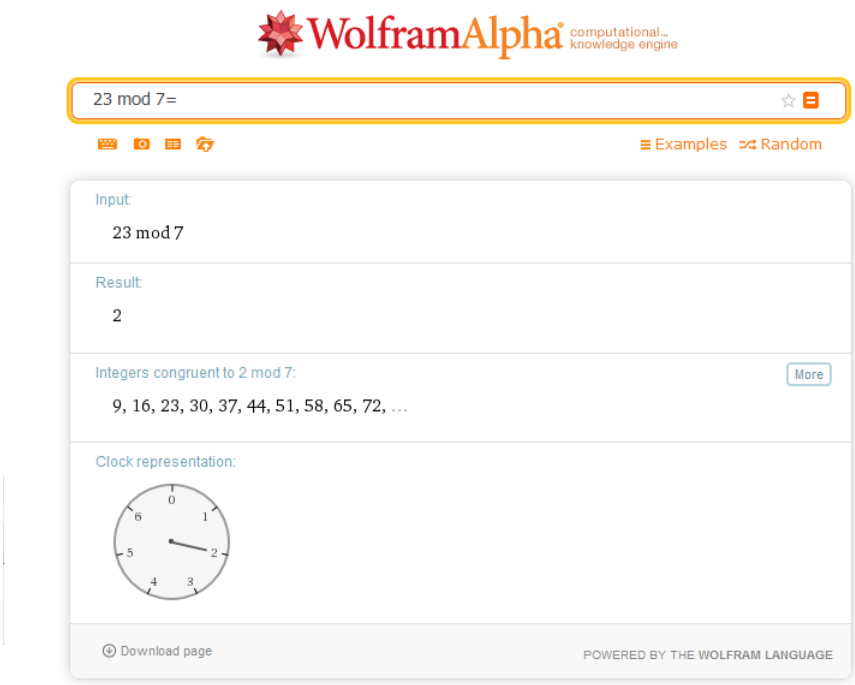
Definição. Dados dois inteiros m e n, diz-se que r é o *resto* da divisão inteira de n por m, e denota-se por $r = n \bmod m$, se $0 \leq r < |m|$ e $n = q \times m + r$ para algum inteiro q. No caso particular em que $r = 0$, diz-se que m *divide* n e escreve-se $m \mid n$.

No `WolframAlpha`[15] basta escrevermos por exemplo

```
> 23 mod 7
```

para obtermos

[15] `http://www.wolframalpha.com`




```
> 23 mod (-7)
                2
> -23 mod 7
                5
```

Portanto

- $a +_{24} b = (a + b) \mod 24$.
- $a \times_{24} b = (a \times b) \mod 24$.

Naturalmente, o número 24 não tem nada de particular. Podemos considerar a aritmética módulo 43, onde $43 = 0$, ou a aritmética módulo 10, onde $10 = 0$; ou naturalmente a aritmética módulo um número n muito grande,por exemplo

$$n = 346901634552179002110238294056748995 3,$$

a aritmética onde este número n é igual a zero.

Assim, podem-se definir operações $+_n$ e $\times_n$ sobre o conjunto $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$:

$$a +_n b = (a + b) \bmod n, \qquad a \times_n b = (a \times b) \bmod n.$$

É esta a chamada *aritmética modular*[16] nos inteiros.

[16]Para mais informação e manipulação destas operações no caso $1 \leq n \leq 10$ vá a `www.atractor.pt/mat/alg_controlo/arit_modular/mod_texto.htm`.

Na aritmética modular todas as propriedades usuais da soma e da multiplicação dos números inteiros continuam válidas. Por exemplo

$$\begin{aligned}(a +_n b) +_n c &= a +_n (b +_n c)\\ (a \times_n b) \times_n c &= a \times_n (b \times_n c)\\ a \times_n (b +_n c) &= (a \times_n b) +_n (a \times_n c).\end{aligned}$$

Mas a particularidade da aritmética módulo n é que algumas vezes um inteiro a pode ter um inverso $\frac{1}{a}$ que é ainda um inteiro! Isto é, para um determinado número a módulo n, pode existir um número b módulo n tal que

$$a \times_n b = 1.$$

Por exemplo, para $n = 10$, como $7 \times 3 = 21 = (2 \times 10) + 1$, então $7 \times_{10} 3 = 1$.

Dados inteiros a e m, a notação

$$a \bmod m$$

representa o resto da divisão inteira de a por m. Por exemplo,

$$10 \bmod 5 = 0, \quad 7 \bmod 5 = 2 \quad \text{e} \quad 2 \bmod 5 = 2.$$

Isto permite definir a chamada *relação de congruência módulo n* entre inteiros:

Definição. Dados inteiros a e b e um natural n, diz-se que a é *congruente com b módulo n*, e escreve-se $a \equiv_n b$, se $a \bmod n = b \bmod n$, isto é, $(a - b) \bmod n = 0$ (ou seja, quando $a - b$ é múltiplo de n).

Verifique que esta relação satisfaz as seguintes propriedades:

(C1) Trata-se de uma relação de equivalência, isto é, para quaisquer a, b, c, d em $\mathbb{Z}$:

- $a \equiv_n a$ (reflexiva).
- Se $a \equiv_n b$ então $b \equiv_n a$ (simétrica).
- Se $a \equiv_n b$ e $b \equiv_n c$ então $a \equiv_n c$ (transitiva).

(C2) Se $a \equiv_n b$ e $c \equiv_n d$ então $a + c \equiv_n b + d$.

(C3) Se $a \equiv_n b$ e $c \equiv_n d$ então $a \times c \equiv_n b \times d$.

Aplicações.

(1) Códigos. Com esta aritmética temos já uma maneira simples de codificar e descodificar uma informação! Efectivamente, podemos multiplicar por 7 módulo 10 para codificar a informação e depois, multiplicar por 3 módulo 10 para descodificar. Fazer estas duas operações consecutivamente corresponde exactamente a multiplicar por 1 módulo 10, isto é, não fazer nada! Recuperamos assim a informação inicial!

Consideremos um exemplo. Um cartão de multibanco tem um código secreto, em geral um número de quatro algarismos. Claro que não é prudente escrevermos este código sobre o cartão. Mas porque não? Por exemplo, para o código 7938, posso multiplicar cada algarismo por 7 módulo 10

$$7 \times_{10} 7 = 9, \quad 9 \times_{10} 7 = 3, \quad 3 \times_{10} 7 = 1, \quad 8 \times_{10} 7 = 6$$

e escrever os resultados sobre o cartão: 9316. Para recuperar o código correcto, basta multiplicar por 3 módulo 10:

$$9 \times_{10} 3 = 7, \quad 3 \times_{10} 3 = 9, \quad 1 \times_{10} 3 = 3, \quad 6 \times_{10} 3 = 8.$$

(2) Números aleatórios. Em qualquer software de matemática é possível gerar números aleatórios. No `WolframAlpha` basta escrever `random integer(n)` para gerar aleatoriamente um inteiro entre 0 e n. A geração destes números é muito útil em simulações computacionais. Diversos métodos têm sido criados para gerar números destes. Em rigor, nenhum destes métodos gera números perfeitamente aleatórios, por isso é habitual chamar-lhes *números pseudo-aleatórios*.

O método mais comum é o chamado *método das congruências lineares*. Escolhemos quatro inteiros: o módulo m, o multiplicador a, o incremento c e a raiz x_0, com $2 \leq a < m$, $0 \leq c < m$ e $0 \leq x_0 < m$. Gera-se uma sequência de números pseudo-aleatórios $\{x_n\}$, com $0 \leq x_n < m$ para qualquer n, usando sucessivamente a fórmula

$$x_{n+1} = (ax_n + c) \bmod m.$$

Por exemplo, a sequência de números pseudo-aleatórios gerada escolhendo $m = 9$, $a = 7$, $c = 4$ e $x_0 = 3$ é a seguinte:

$$\begin{aligned}
x_1 &= (7x_0 + 4) \bmod 9 = 25 \bmod 9 = 7 \\
x_2 &= (7x_1 + 4) \bmod 9 = 53 \bmod 9 = 8 \\
x_3 &= (7x_2 + 4) \bmod 9 = 60 \bmod 9 = 6 \\
x_4 &= (7x_3 + 4) \bmod 9 = 46 \bmod 9 = 1 \\
x_5 &= (7x_4 + 4) \bmod 9 = 11 \bmod 9 = 2 \\
x_6 &= (7x_5 + 4) \bmod 9 = 18 \bmod 9 = 0 \\
x_7 &= (7x_6 + 4) \bmod 9 = \ \ 4 \bmod 9 = 4 \\
x_8 &= (7x_7 + 4) \bmod 9 = 32 \bmod 9 = 5 \\
x_9 &= (7x_8 + 4) \bmod 9 = 39 \bmod 9 = 3
\end{aligned}$$

Como $x_9 = x_0$ e cada termo na sequência só depende do anterior, a sequência terá nove números diferentes antes de se começar a repetir:

$$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$$

A maioria dos computadores usa este método para gerar números pseudo-aleatórios. Por exemplo, é muito utilizado o sistema módulo $m = 2^{31} - 1$ com incremento $c = 0$ e multiplicador $a = 7^5 = 16\,807$, que permite gerar $2^{31} - 2$ números antes que a repetição comece.

(3) Cálculo do máximo divisor comum. O algoritmo mais antigo que se conhece, e que aparece no livro VII dos *Elementos* de Euclides (c. 325 a.C. - 265 a.C.), calcula o *máximo divisor comum* $\mathrm{mdc}(a, b)$ de dois inteiros a e b.

Exemplo. Consideremos os inteiros $a = 252$ e $b = 54$. Dividindo a por b obtemos $a = 252 = 54 \times 4 + 36$. Tornando a dividir, agora b pelo resto 36, $54 = 36 \times 1 + 18$. Continuando o processo, chegamos a uma divisão exacta (porquê?), e o processo pára: $36 = 18 \times 2 + 0$. É fácil ver que 18, o último resto não nulo, é o máximo divisor comum de a e b.

$$\begin{aligned} 252 &= 54 \times 4 + 36 \\ 54 &= 36 \times 1 + \boxed{18} \\ 36 &= 18 \times 2 + \boxed{0} \end{aligned}$$

$$\therefore \quad \text{mdc}(252, 54) = 18$$

Não se trata de uma coincidência: não é difícil provar que, seguindo este procedimento para quaisquer outro par de inteiros positivos, o último resto não nulo é sempre igual a $\text{mdc}(a, b)$. Este é o algoritmo de Euclides:

```
procedure mdc(a, b : inteiros positivos)
x := a
y := b
while y ≠ 0
begin
   r := x mod y
   x := y
   y := r
end {x é o mdc(a, b)}
```

O algoritmo de Euclides é um dos resultados básicos dos números inteiros.

(4) Criptografia[17]: cifra de César. As congruências utilizam-se muito na criptografia. O exemplo mais simples (e muito antigo, remonta a Júlio César) é a chamada *cifra de César*. Ele usava um método de escrita de mensagens secretas transladando cada letra do alfabeto para três casas mais à frente:

A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	T	U	V	X	Z	A	B	C

Este sistema de encriptação pode ser descrito matematicamente de forma muito abreviada: substituímos cada letra por um inteiro de 0 até 22, baseado na sua posição no alfabeto:

[17] A *criptografia* é a parte da *criptologia* (ciência dos códigos) que se dedica ao estudo de mensagens secretas e dos processos de *encriptação*, ou seja, de escrita de mensagens secretas.

A	B	C	D	E	F	G	H	I	J	L	M
0	1	2	3	4	5	6	7	8	9	10	11

N	O	P	Q	R	S	T	U	V	X	Z	
12	13	14	15	16	17	18	19	20	21	22	

Portanto o método de César é definido pela função f que aplica cada inteiro n, $0 \leq n \leq 22$, no inteiro

$$f(n) = (n + 3) \bmod 23.$$

Por exemplo,

$$\begin{array}{c} \mathrm{X} \longleftrightarrow 21 \\ \downarrow \\ f(21) = 24 \bmod 23 = 1 \longleftrightarrow \mathrm{B}. \end{array}$$

Teste. Como fica a mensagem "DESCOBRI A SOLUCAO" depois de encriptada pela cifra de César ?

Para recuperar a mensagem original a partir da mensagem encriptada basta considerar a função inversa f^{-1} que transforma um inteiro n, $0 \leq n \leq 22$, em $f^{-1}(n) = (n - 3) \bmod 23$.

Podemos generalizar a cifra de César transladando b casas em vez de três:

$$f(n) = (n + b) \bmod 23.$$

É claro que a cifra de César é um método de encriptação muito pouco seguro. Podemos melhorá-lo um pouco definindo, mais geralmente, $f(n) = (an + b) \bmod 23$, com a e b inteiros escolhidos de modo a garantir que f é uma bijecção.

Teste. Que letra substitui J com a função encriptadora $f(n) = (7n + 3) \bmod 23$?

Teste. Descodifique a mensagem

$$*\mathrm{PIBE}* \smile \mathrm{D} \smile \mathrm{W@P}$$

(onde o símbolo $\smile$ indica um espaço em branco) que foi encriptada utilizando o alfabeto da figura seguinte e a função $f(p) = (22p + 25) \bmod 29$.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z	W	$*$	@	$\smile$
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Números primos.

Quais são os elementos de $\mathbb{Z}_n$, chamados *invertíveis*, que têm inverso relativamente à operação $\times_n$? Estes elementos estão intimamente ligados aos números primos.

Definição. Um natural $p \geq 2$ diz-se *primo* quando, para qualquer natural n, se n divide p então $n = 1$ ou $n = p$.

No `WolframAlpha` se perguntarmos `is n a prime number?` permite testar se o número n é primo[18] e a função `prime(n)` lista o n-ésimo número primo:

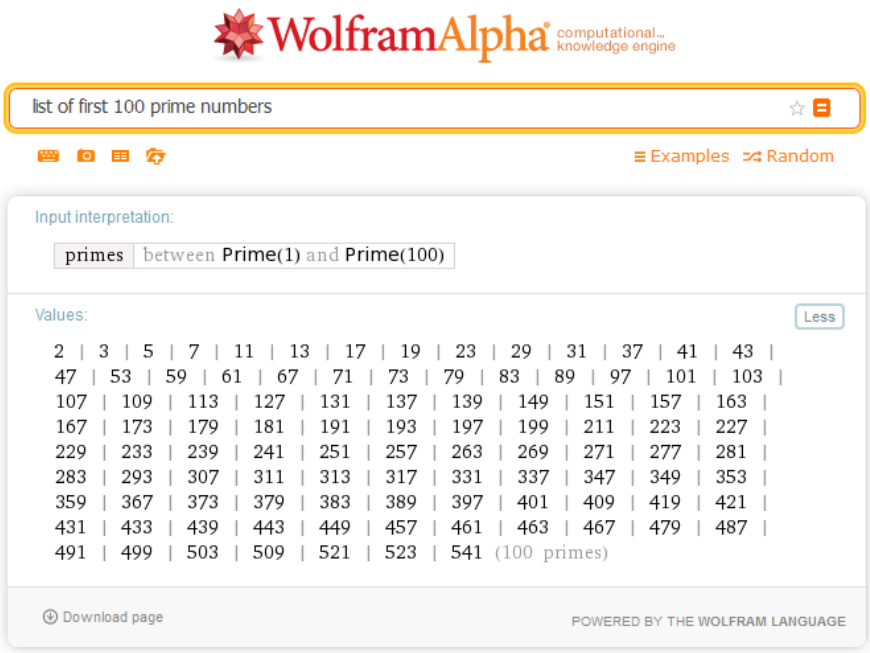
```
> is 15 485 863 a prime number?
```

Result: 15 485 863 is a prime number

```
> prime(23)
```

$p_{23} = 83$ (p_n is the n^{th} prime number)

```
> list of first 100 prime numbers
```



Os números primos têm um papel fundamental relativamente aos outros números naturais por causa do seguinte resultado:

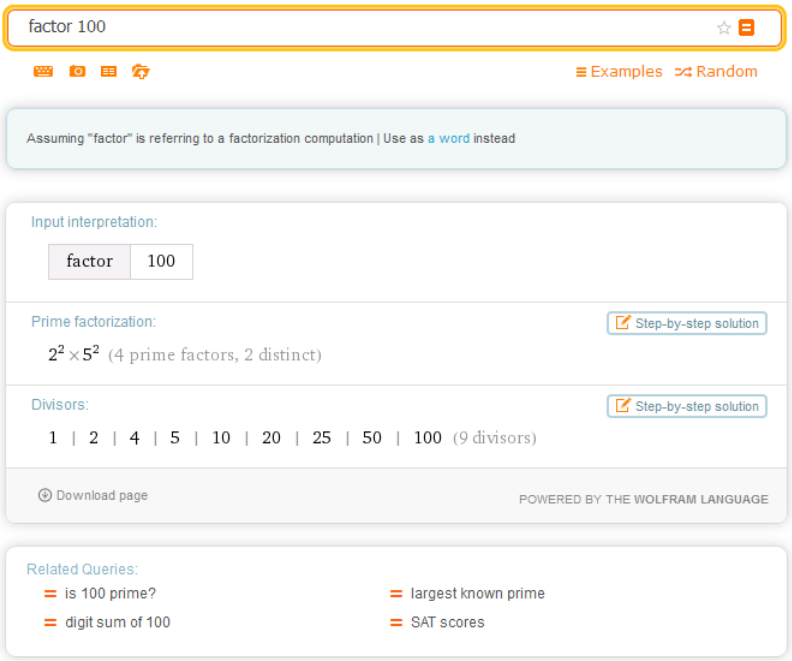
Teorema Fundamental da Aritmética. *Todo o natural $n \geq 2$ pode escrever-se de maneira única, a menos da ordem dos factores, da seguinte forma:*

$$n = p_1^{e_1} \times p_2^{e_2} \times \cdots \times p_k^{e_k}$$

onde cada p_i é primo e $e_i > 0$.

No `WolframAlpha` podemos obter rapidamente estas fcatorizações:

[18] Testar se um número é primo é um problema de *complexidade polinomial*. Este resultado só foi provado muito recentemente (em 2002) por uma equipa de investigadores do *Indian Institute of Technology*, depois de muitas tentativas goradas durante o século passado.



Ao contrário do que acontece com o problema da verificação da primalidade de um número, não se conhece nenhum algoritmo que, em tempo polinomial, consiga realizar esta decomposição em primos. Mais, conjectura-se que um tal algoritmo não existe.

Definição. Dois naturais m e n dizem-se *primos entre si* se $\text{mdc}(m,n) = 1$ (diz-se também que m é *coprimo* de n).

Podemos agora caracterizar facilmente os elementos de $\mathbb{Z}_n$ que são invertíveis para a multiplicação $\times_n$:

Proposição. *Um natural $a \in \mathbb{Z}_n$ é invertível se e só se a e n são primos entre si.*

Prova. "⇒": $ab \equiv_n 1$ significa que $ab = nq+1$, isto é, $ab-nq = 1$, para algum inteiro q. Então, se d é um divisor comum de a e n, será um divisor de $ab - nq = 1$, pelo que necessariamente $d = 1$ ou $d = -1$. Isto mostra que $\text{mdc}(a,n) = 1$.
"⇐": Se $\text{mdc}(a,n) = 1$ então, pelo algoritmo de Euclides (usado em ordem inversa — ver exemplo na página seguinte), existem inteiros s e t tais que $1 = sa+tn$, ou seja, $sa = n\times(-t)+1$, o que mostra que $sa \equiv_n 1$. Consequentemente, todos os números da forma $s + kn$ ($k \in \mathbb{Z}$) são solução da equação $xa \equiv_n 1$ e um deles pertence necessariamente a $\mathbb{Z}_n$. Esse será o inverso de a em $\mathbb{Z}_n$. □

Portanto, em $\mathbb{Z}_5 = \{0,1,2,3,4\}$ todos os elementos não nulos são invertíveis:

$$1 \times_5 1 = 1, \quad 2 \times_5 3 = 1, \quad 3 \times_5 2 = 1, \quad 4 \times_5 4 = 1.$$

Mais geralmente, se n é primo todos os elementos não nulos de $\mathbb{Z}_n$ são invertíveis. Em $\mathbb{Z}_4 = \{0,1,2,3\}$ só o 1 e o 3 são invertíveis; em $\mathbb{Z}_6 = \{0,1,2,3,4,5\}$ só o 1 e o 5 são invertíveis. Em $\mathbb{Z}_{10}$, como vimos no início da secção, o 7 é invertível.

Para cada n, o número de coprimos de n menores ou iguais a n é dado pela *função de Euler* (ou função *tociente*)

$$\begin{array}{rcl} \phi : \mathbb{N} & \to & \mathbb{N} \\ n & \mapsto & \#\{m : m \leq n \text{ e } m \text{ é coprimo de } n\} \end{array}$$

cujos valores podemos calcular com o `WolframAlpha`, recorrendo à função `totient` ou `phi`:

```
> phi(107)
```

106

```
> phi(106)
```

52

Portanto, em $\mathbb{Z}_{107}$ há 106 elementos invertíveis e em $\mathbb{Z}_{106}$ só há 52.

Acabámos de obter uma solução para as congruências do tipo $ax \equiv_n 1$. Mais geralmente:

Resolvendo uma congruência. Sejam a, b, n inteiros. *Se* $mdc(a,n) = 1$ *então a equação* $ax \equiv_n b$ *tem uma solução*:

> (1) `Determine inteiros` s `e` t `tais que` $1 = as + nt$
> (usando o algoritmo de Euclides por ordem inversa)
>
> (2) `Então` $x = bs$ `é uma solução da congruência`
> (o conjunto completo de soluções é $\{bs + nk \mid k \in \mathbb{Z}\}$)

Teste. Resolva a equação $10x \equiv_{27} 5$.

Solução. Usemos o algoritmo de Euclides para determinar mdc$(10, 27)$:

$$\begin{array}{rcl} 27 & = & 10 \times 2 + 7 \\ 10 & = & 7 \times 1 + 3 \\ 7 & = & 3 \times 2 + 1 \end{array}$$

Portanto mdc$(10, 27) = 1$. Agora invertamos o processo para encontrar inteiros s e t tais que $1 = 10s + 27t$:

$$\begin{array}{rcl} 1 & = & 7 - 3 \times 2 \\ & = & 7 - (10 - 7 \times 1) \times 2 \\ & = & 7 \times 3 - 10 \times 2 \\ & = & (27 - 10 \times 2) \times 3 - 10 \times 2 \\ & = & 10 \times (-8) + 27 \times 3. \end{array}$$

Portanto $s = -8$ e $t = 3$. Logo $x = bs = -40$ é uma solução, e o conjunto completo de soluções é $\{-40 + 27k \mid k \in \mathbb{Z}\}$.

Teorema de Fermat-Euler.[19]

(a) *Se p é um primo que não divide x então $x^{p-1} \equiv_p 1$.*

(b) *Se p e q são primos que não dividem x então $x^{(p-1)(q-1)} \equiv_{pq} 1$.*

Prova. Provamos somente (a). Para isso consideremos os inteiros

$$x \bmod p, \quad 2x \bmod p, \quad \ldots \quad , \quad (p-1)x \bmod p. \quad (*)$$

Estes números são todos distintos, dois a dois:

Se $nx \bmod p = mx \bmod p$, com n, m entre 1 e $p-1$, então $nx \equiv_p mx$, isto é, $nx - mx \equiv_p 0$, ou seja, p divide $(n-m)x$. Mas p é primo e não divide x logo terá que dividir o outro factor $n-m$. Como $n-m$ é um número entre 0 e $p-1$, o único número destes que é múltiplo de p é o zero. Portanto, $n - m = 0$, ou seja, $n = m$.

Assim, como todos estes $p-1$ números são distintos e pertencem ao conjunto $\{1, 2, \ldots, p-1\}$, a lista $(*)$ constitui um rearranjo (permutação) dos números $1, 2, \ldots, p-1$. Portanto, o produto dos números da lista é igual ao produto dos números $1, 2, \ldots, p-1$:

$$x^{p-1}(p-1)! \bmod p = (p-1)! \bmod p.$$

Daqui decorre que $x^{p-1}(p-1)! \equiv_p (p-1)! \Leftrightarrow (p-1)!(x^{p-1} - 1) \equiv_p 0$. Portanto p divide $(p-1)!(x^{p-1} - 1)$. Como p não pode dividir $(p-1)!$, terá então que dividir $x^{p-1} - 1$, o que significa que $x^{p-1} \equiv_p 1$. □

4.2. Criptografia: o sistema RSA de chave pública

Os resultados sobre os inteiros que acabámos de estudar estão na base de toda a criptografia actual, que permite a troca de mensagens confidenciais por intermédio de um canal público, supondo que os agentes comunicantes, digamos a Alice e o Bruno, não partilham segredo nenhum.

Se a Alice quiser enviar uma mensagem x ao Bruno, pede-lhe para ele gerar um par de chaves, uma chave pública u (conhecida por toda a gente) e uma chave privada v (conhecida apenas pelo Bruno). As chaves u e v são aplicações do espaço das mensagens para o espaço das mensagens e, para que o sistema funcione bem e permita manter o secretismo na comunicação, devem ter as seguintes propriedades:

(P1) $v(u(x)) = x$ para qualquer mensagem x.

(P2) deve ser difícil obter x conhecendo $u(x)$ e não conhecendo v.

O protocolo funciona do seguinte modo:

[19] Este teorema (proposição (a)) foi obtido pelo matemático francês Fermat por volta de 1630, e generalizado pelo matemático suíço Euler um século mais tarde (proposição (b)).

(1) A Alice envia a mensagem $u(x)$ ao Bruno pelo canal público.

(2) O Bruno recupera a mensagem original x aplicando v a $u(x)$.

Ao definirmos um *sistema criptográfico* deveremos explicitar o espaço das mensagens bem como as aplicações u e v. Um dos sistemas mais utilizados hoje em dia é o *sistema RSA* cujo nome deriva dos seus criadores (Rivest, Shamir e Adleman, em 1976).

A família de sistemas criptográficos RSA é definida do seguinte modo:

- espaço de mensagens: $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$, onde $n = p \times q$ para algum par de primos p, q,
- $u(x) = x^a \bmod n$, para qualquer $x \in \mathbb{Z}_n$,
- $v(y) = y^b \bmod n$, para qualquer $y \in \mathbb{Z}_n$,

onde a e b são tais que $a \times b \bmod (p-1) \times (q-1) = 1$.

Será de facto um bom sistema, isto é, satisfaz as propriedades (P1) e (P2) ?

Os conceitos e resultados relativos aos números inteiros apresentados nesta secção permitem-nos verificar a propriedade (P1), como veremos mais adiante. Quanto à propriedade (P2), a sua confirmação é ainda hoje um problema em aberto[20].

Este sistema permite enviar mensagens encriptadas por uma chave pública a, mas para desencriptar a mensagem o receptor precisa de ter uma chave privada b (só do seu conhecimento).

Sejam

$$p \text{ e } q \text{ primos}, \quad n = pq, \quad m = (p-1)(q-1).$$

Consideremos ainda a tal que $\text{mdc}(a, m) = 1$ e seja b a solução da congruência $ab \equiv_m 1$.

No sistema RSA, podemos começar por traduzir as mensagens (sequências de letras) em sequências de inteiros (como na cifra de César):

A	B	C	D	E	F	G	H	I	J	L	M
00	01	02	03	04	05	06	07	08	09	10	11
N	O	P	Q	R	S	T	U	V	X	Z	
12	13	14	15	16	17	18	19	20	21	22	

O inteiro x daí resultante é depois transformado, com a ajuda da chave pública a, num inteiro

$$u(x) = x^a \bmod n.$$

Teste. Codifique a mensagem HELP usando o sistema RSA com $p = 43$, $q = 59$ e $a = 13$.

[20] Trata-se de um dos problemas em aberto mais importantes da matemática, com grandes implicações práticas: até hoje, ninguém conseguiu demonstrar se o sistema RSA verifica a propriedade (P2), apesar de todos os especialistas conjecturarem que isso seja verdadeiro. Portanto, toda a criptografia actual assenta, não numa certeza absoluta, mas numa conjectura.

Solução. Neste sistema $n = 43 \times 59 = 2537$. Note que, como 13 é primo, $\text{mdc}(13, 42 \times 58) = 1$. Traduzindo as letras no seu valor numérico, H→07, E→04, L→10 e P→14. A mensagem corresponde então ao número $x = 07041014$. Se aplicarmos já a chave pública u a x obtemos uma mensagem só com quatro algarismos: $u(x) = 07041014^{13} \bmod 2537 = 1507$. Para manter o número de algarismos na mensagem encriptada, como $n = 2537$ tem quatro algarismos, agrupam-se os algarismos de x em blocos de quatro e só depois se aplica u a cada um desses blocos[21]:

$$0704 \xrightarrow{u} 704^{13} \bmod 2537 = 0981,$$

$$1014 \xrightarrow{u} 1014^{13} \bmod 2537 = 1175.$$

A mensagem encriptada é então 0981 1175.

O receptor quando recebe a mensagem desencripta-a com a ajuda da chave privada b que só ele conhece:

$$v(u(x)) = u(x)^b \bmod n.$$

Teste. Descodifique a mensagem seguinte, recebida usando o sistema RSA do exemplo anterior: 2128 2431.

Solução. Temos que resolver a congruência $13b \equiv_{42\times58} 1$ para determinar a chave privada b:

$$\begin{aligned} 42 \times 58 = 2436 &= 13 \times 187 + 5 \\ 13 &= 5 \times 2 + 3 \\ 5 &= 3 \times 1 + 2 \\ 3 &= 2 \times 1 + 1 \end{aligned}$$

donde

$$\begin{aligned} 1 &= 3 - 2 \times 1 \\ &= 3 - (5 - 3 \times 1) \\ &= 3 \times 2 - 5 \\ &= (13 - 5 \times 2) \times 2 - 5 \\ &= 13 \times 2 - 5 \times 5 \\ &= 13 \times 2 - (2436 - 13 \times 187) \times 5 \\ &= 13 \times (2 + 187 \times 5) - 2436 \times 5 \\ &= 13 \times 937 - 2436 \times 5. \end{aligned}$$

Portanto $b = 937$. Então $2128^b \bmod 2537 = 2128^{937} \bmod 2537 = 1718$ e $2431^b \bmod 2537 = 2431^{937} \bmod 2537 = 1314$, pelo que a mensagem original é, na versão numérica, 1718 1314, ou seja STOP.

[21] Portanto, deveremos ter o cuidado de ter sempre $x < n$.

Já sabemos como encriptar e desencriptar mensagens no sistema RSA. Falta assegurar, como tínhamos anunciado, que o RSA satisfaz a propriedade (P1), isto é, que a desencriptação v é de facto inversa da encriptação u:

Proposição. *Sejam p e q primos distintos, $n = pq$ e $m = (p-1)(q-1)$. Se a e b são inteiros tais que $ab \equiv_m 1$, então $v(u(x)) = x$ para qualquer inteiro $x < p, q$.*

Prova. Uma vez que $v(u(x)) = v(x^a \bmod n) = (x^a \bmod n)^b \bmod n = x^{ab} \bmod n$, e $ab = k(p-1)(q-1)+1$ para algum inteiro k, temos

$$v(u(x)) = x^{k(p-1)(q-1)+1} \bmod n = ((x^{(p-1)(q-1)})^k \times x) \bmod n.$$

Mas como x é menor do que p e q, não é divisível por p e q logo, pelo Teorema de Fermat-Euler (b), $(x^{(p-1)(q-1)})^k \bmod n = 1$. Portanto $v(u(x)) = x \bmod n = x$. □

Basta assim usarmos mensagens com número x inferior a p, q para termos a certeza que a função de desencriptação v recupera a mensagem original.

Como é que o processo de troca de mensagens secretas entre a Alice e o Bruno se desenrola na realidade? O Bruno, o receptor, faz o seguinte:

(1) escolhe dois números primos p e q,
(*pode ser difícil quando se procuram números p, q muito grandes*)
(2) calcula os produtos $n = pq$ e $m = (p-1)(q-1)$,
(*muito fácil*)
(3) escolhe $a \in \mathbb{Z}_m$ (a chave pública) tal que $\text{mdc}(a, m) = 1$,
(*fácil: conhecemos um algoritmo eficaz para calcular o mdc de 2 números*)
(4) usando o algoritmo de Euclides, determina $b \in \mathbb{Z}_m$ (a chave privada) tal que $ab \equiv_m 1$.
(*fácil: conhecemos um algoritmo eficaz para calcular o inverso de um elemento em $\mathbb{Z}_m$*)
(5) Envia os valores de n e a para a Alice, mantendo a chave privada b só do seu conhecimento.
(a partir daqui, não havendo garantias de segurança no canal de comunicação, os valores de n e a passam a ser eventualmente públicos.)

A Alice tem agora os elementos para encriptar as suas mensagens com a função u e enviá-las ao Bruno. Como só este conhece o valor de b, só ele poderá decifrar a mensagem aplicando a função v.

E que trabalho tem que fazer uma terceira pessoa mal intencionada, que conhece só a função de encriptação, para desencriptar uma mensagem?

(1) factorizar o número n para recuperar os primos p e q,
(*pode ser muito difícil quando n é muito grande*)
(2) usando o algoritmo de Euclides, determinar $b \in \mathbb{Z}_m$ (a chave privada) tal que $ab \equiv_m 1$,
(*fácil: conhecemos um algoritmo eficaz para calcular o inverso de um elemento em $\mathbb{Z}_m$*).

Vemos assim que a diferença principal é:

- para criar o código é preciso encontrar dois números primos p, q;
- para decifrar o código é preciso factorizar o produto $n = pq$.

Comparemos a dificuldade de ambas as operações. Actualmente, os grandes computadores utilizam números primos com mais de 80 algarismos. Claramente, se um número p de 80 algarismos não é primo, podemos escrever $p = a \times b$ e é impossível que ambos os factores tenham mais de 40 algarismos. Portanto, um dos factores tem menos de 40 algarismos. Assim, para saber se p é primo, basta dividir p por todos os números de 40 algarismos ou menos. E se o número não é primo, recomeçar o trabalho com um outro número, e assim sucessivamente, até achar finalmente um número primo.

Existe uma fórmula famosa que dá uma aproximação do número de números primos inferiores a um certo número n:

Existem cerca de $\frac{n}{\ln n}$ *números primos inferiores a* n.

Isto implica que, entre os números de 80 algarismos, aproximadamente um em cada 185 números é primo. Então, para encontrar dois números primos de 80 algarismos, o Bruno tem que fazer cerca de

$$2 \times 185 \times 10^{40} = 370 \times 10^{40}$$

operações.

Quando uma terceira pessoa mal intencionada conhece $n = pq$, sabendo que p e q são números primos de 80 algarismos, tem que dividir n por todos os números de 80 algarismos para descobrir os factores p, q. Há 10^{80} números de 80 algarismos ou menos, e 10^{79} números de 79 algarismos ou menos. Então há

$$10^{80} - 10^{79} = 10^{79}(10 - 1) = 9 \times 10^{79}$$

números de exactamente 80 algarismos. Em conclusão, o trabalho da pessoa mal intencionada é

$$\frac{9 \times 10^{79}}{370 \times 10^{40}} = \frac{9}{37} \times 10^{38} \approx 2,4 \times 10^{37}$$

vezes mais difícil do trabalho do Bruno. Isto significa que se o computador do Bruno gastar um segundo a encontrar os primos p, q, um computador do mesmo tipo tem que trabalhar durante aproximadamente 10^{37} segundos para quebrar o código. É muito tempo?

A Terra tem cerca de 3.500.000.000 anos, ou seja,

$$3.500.000.000 \times 365 \times 24 \times 60 \times 60 = 110.376.000.000.000.000 \approx 1,1 \times 10^{17}$$

segundos. Então 10^{37} segundos são 10^{20} vezes a idade da Terra!!!

É claro que o exemplo de RSA que apresentamos na página 95 é (matematicamente) inseguro, um intruso facilmente quebraria o sistema: conhecendo $n = 2537$, facilmente obteria a sua factorização prima $pq = 43 \times 49$, calcularia $(p - 1)(q - 1) = 2436$, e usaria o algoritmo de Euclides para descobrir a chave privada $b = 937$.

É aqui que reside a grande segurança do sistema RSA: a aparente dificuldade em resolver este problema, desde que p e q sejam números com muitos algarismos. Neste caso, mesmo sendo

pública a informação de n e a, um intruso não deverá ser capaz de descobrir o expoente b de desencriptação[22].

Exemplo. O Bruno começa por procurar dois primos grandes p e q, por exemplo com 80 algarismos. No `WolframAlpha` com qualquer expressão próxima de `random integer (n)` obtemos um inteiro aleatório r com 80 algarismos e depois `next prime (r)` calcula o menor primo maior do que o inteiro r:

```
>  random integer (10^80)
```

r: 19669081321110693270343633073697474256143563558458718976746753830538032062222085

```
>  next prime (r)
```

p: 196690813211106932703436330736974742561435635584587189767467538305380320622222257

```
>  random integer (10^80)
```

s: 74121768604305613921745580037409259811952655310075487163797179490457039169594160

```
>  next prime (s)
```

q: 741217686043056139217455800374092598119526553100754871637971794904570391695942 13

A selecção primeiro de dois inteiros aleatórios r e s tem a intenção de tornar mais difícil de adivinhar os primos p e q. Em seguida, o Bruno calcula $n = pq$ e $m = (p-1)(q-1)$.

```
>  pxq
```

n: 145790709434263657193410815968586298032651591491182486164339752298049755073623061549604680218687683561183675344052519958769801995483916593242784227837370699874 1

Este valor de n com 160 algarismos permite encriptar mensagens até 80 letras num só bloco.

```
> (p-1)x(q-1);
```

m: 145790709434263657193410815968586298032651591491182486164339752298049755073623052170519687677056964352262364233379113149147915142063302538849452128330247518272 72

Em seguida, decide a escolha de a. Como será um valor público, não se preocupa em gerar números aleatoriamente. A única preocupação é que satisfaça $\text{mdc}(a, m) = 1$. Por exemplo, pode fazer $a = 2^{16} + 1 = 65537$. Depois calcula b tal que $ab \equiv_m 1$:

```
>  a^-1 mod(m)
```

[22] A não ser que descubra um algoritmo de factorização que torne o problema realizável em tempo útil, o que os matemáticos acreditam (conjecturam) não existir. Mas, até hoje, ninguém foi capaz de provar isso. Por esta razão a factorização, para a qual se julga não existir algoritmo polinomial, é o calcanhar de Aquiles do RSA.

b: 34180298922096847472065507840720943425419102236324807359431775852717312155060777
829318324017852209549910908745378489609482547509922679456023648197991886310291 3

Está pronto para definir as função de encriptação u e desencriptação v:

$$u\colon (x,a,n) \mapsto x^a \mathrm{mod}(n), \quad v\colon (x,b,n) \mapsto x^b \mathrm{mod}(n).$$

Verifiquemos num exemplo (mensagem $x =$ "5") que os procedimentos u e v são inversos um do outro:

```
>  5^a mod(n)
```

y: 44669982652857045772784970284788245601106395884543657540636484577393488318578590
496921573836272232443097862919568742270009616466410734910391539516416953887426 1

```
>  y^b mod(n)
```

5

Teste. Descodifique a mensagem 144527134207733385081058793072124611963727630008654292399 9910113230948208915317126596566680325137342547829329336431874588144606598803386096533964035759 67077856790 recebida pelo Bruno.

Solução.

```
>  144527134207733385081058793072124611963727630008654292399101132309482089153171265965666803251373425478293293364318745881446065988033860965339640357596707785679 0^b mod(n)
```

417181903041104171816191819160017030817021604180017

Como o primeiro par de algarismos, 41, não corresponde a nenhuma letra (ver tabela da página 94), o par original deverá ser 04 (o `WolframAlpha` não escreveu o 0), ou seja, é a letra E. Continuando, 17→S, 18→T, 19→U, etc. A mensagem original é

"ESTUDEMESTRUTURASDISCRETAS".

Observe que, em geral, no sistema RSA assume-se que quase tudo é do conhecimento público, incluindo a forma da função encriptadora. Isto significa que um intruso que intersecta uma mensagem RSA sabe que esta foi formada com a função $u(x) = x^a \mod n$, e conhece os valores a e n. A vantagem desta informação ser pública reside no facto da Alice e do Bruno para comunicarem entre si numa linha de comunicação insegura não precisarem de pensar numa maneira de trocarem entre si secretamente o expoente de encriptação a e o módulo n. Só a chave privada b nunca pode circular entre ambos pelo canal de comunicação, para que não possa ser interceptada.

Leituras suplementares. **(1)** Até agora utilizámos um método muito simples de conversão de letras em números, que representa A pelo número 0, B pelo número 1, etc., até Z. Este

método tem uma desvantagem óbvia: não funciona conjuntamente com maiúsculas e minúsculas, com espaços, acentos e outros caracteres. Existe um método muito utilizado de conversão de caracteres no código ASCII, usado pela maioria dos computadores, permitindo a conversão de cadeias de caracteres alfanuméricos em inteiros e vice-versa. No `WolframAlpha` e `Mathematica` a instrução é

`ToCharacterCode`[“xxx”]

mas bastará por exemplo escrever qualquer coisa parecida com[23]

`convert` “xxx” `to ASCII`

```
> convert "Bom dia, a vossa missão para hoje é codificar esta mensagem."  to ASCII
```

[66, 111, 109, 32, 100, 105, 97, 44, 32, 97, 32, 118, 111, 115, 115, 97, 32, 109, 105, 115, 115, 227, 111, 32, 112, 97, 114, 97, 32, 104, 111, 106, 101, 32, 233, 32, 99, 111, 100, 105, 102, 105, 99, 97, 114, 32, 101, 115, 116, 97, 32, 109, 101, 110, 115, 97, 103, 101, 109, 46]

Em sentido inverso:

```
> convert {66, 111, 109, 32, 100, 105, 97, 44, 32, 97, 32, 118, 111, 115, 115,
97, 32, 109, 105, 115, 115, 227, 111, 32, 112, 97, 114, 97, 32, 104, 111, 106,
101, 32, 233, 32, 99, 111, 100, 105, 102, 105, 99, 97, 114, 32, 101, 115, 116,
97, 32, 109, 101, 110, 115, 97, 103, 101, 109, 46} to characters
```

“Bom dia, a vossa missão para hoje é codificar esta mensagem.”

Para mais informação sobre conversão de mensagens consulte o texto[24] `conversaoRSA2.mws` de Mike May (2002).

(2) O sistema RSA tem resistido a ataques de criptoanalistas, à custa do aumento da dimensão das chaves, mas é necessário ir acompanhando os desenvolvimentos mais recentes. Apesar da matemática subjacente ser há muito conhecida, a cifra RSA surgiu apenas nos anos 70 porque é aplicável apenas com números primos de grande dimensão e só nos anos 70 apareceram computadores potentes de custo aceitável. Ataques mais conhecidos em 2006:

- Números até 100 bits consegue-se quebrar, com PCs.
- Em computação paralela são conhecidos ataques até 640 bits (actualmente recomenda-se o uso de números RSA-1024, ou RSA-2048)[25].

Qual é a complexidade do algoritmo de factorização do número n em primos pq?
Factorização à força bruta: testar todos os primos até $n/2$ (é de complexidade $O(\sqrt{n})$).

[23] É a grande vantagem do `WolframAlpha`: não precisamos de conhecer as instruções exactas, ele reconhece as expressões em inglês aproximadas.
[24] Na página da disciplina.
[25] A notação RSA-xxxx refere-se a um número RSA com tamanho xxxx em bits.

Exemplo. $n := 408508091$.

1.	`Divisível por 3?` Não!
2.	`Divisível por 5?` Não!
3.	`etc.`
⋮	
2099.	`Divisível por 18 313 (é o` 2099^o `primo)?` Sim, está identificado o primo p.
2100.	$q = n/18313 = 22307$.

Demorou 2099 passos, e n só tem 9 algarismos! Imagine RSA-640 com 193 algarismos decimais...

A empresa norte-americana *RSA Security*[26] submete a concurso a factorização de números RSA com prémios até 200 mil dólares (para o caso RSA-2048):

- Primeiro prémio (100 dólares) atribuído em Abril 1994, pela factorização de números RSA-129.
- O prémio mais elevado (20 000 dólares) foi ganho em Novembro 2005 na factorização do RSA-640

  310741824049004372135075003588856793003734602284272754572016194882320644051808150455634682967172328678243791627283803341547107310850191954852900733772482278352574238645401469173660247765234660 9

  por F. Bahr, M. Boehm, J. Franke, T. Kleinjung (Univ. Bona, Alemanha). Os factores são

  163473364580925384844313388386509085984178367003309231218111085238933310010450815121211816751157 9

  e

  190087128166482211312685157393541397547189678996851549366663853908802710380210449895719126146557 1

  Os cálculos foram efectuados durante 540 dias por um conjunto de 80 AMD64 Opteron (CPU que equipa cerca de 10% dos supercomputadores mais rápidos do mundo).

Desafio. Se conseguir descobrir a factorização do RSA-704

74037563479561712828046796097429573142593188889231289084936232638972765034028266276891996419625117843995894330502127585370118968098286733173273108930900552505116877063299072396380786710086096962537934650563796359

[26] `www.rsa.com/rsalabs/node.asp?id=2093`.

terá 20 valores à disciplina e pode candidatar-se ao prémio de 30 000 dólares da *RSA Security*.

Exemplos de chaves públicas usadas em algumas páginas web.

Chave pública Departamento de Matemática (tamanho: 140 Bytes / 1120 Bits)

```
> DMUC:    "30 81 89 02 81 81 00 db 35 3c 03 49 fc e0 48 e4 b6 7c 55 66 f3 52 08
39 b9 d8 bf eb 9c c7 e8 7a 32 54 fc 88 66 19 de a0 08 b1 19 ad a0 34 75 0c 2b 0d
f5 6d 3b 9d f4 78 2e 2e fe 45 d3 7e b5 ff 7c f6 9a 3b d7 13 46 a9 e1 ab 0b 01 d8
d8 3c 65 d7 ce a3 e7 32 c3 59 54 97 54 b7 a4 6e be 07 61 25 0d 32 04 3a 99 15 19
23 9d 97 61 e1 66 5d b7 ef 81 e9 1e cd bc 25 fd 39 9b 74 6a 86 09 07 9a 98 bd 45
f2 ba 84 a1 02 03 01 00 01"
```

Conversão desta representação hexadecimal para a sua representação decimal:

```
DMUC:    26986751662544233897390996989562786998209640195002153770034468141230169
50163931124306976585375250715400911398515483295152423812242464904341510349824376
57607908609525327637173783415854826421482431563077009840890804022964491227968726
58542504958657945923684483016763566353046590571882008173675016517836292426828436
78677672254248775317520385
```

Chave pública VISA eCommerce (tamanho: 270 Bytes / 2160 Bits)

```
> nVISA := "30 82 01 0a 02 82 01 01 00 af 57 de 56 1e 6e a1 da 60 b1 94 27 cb 17
db 07 3f 80 85 4f c8 9c b6 d0 f4 6f 4f cf 99 d8 e1 db c2 48 5c 3a ac 39 33 c7 1f
6a 8b 26 3d 2b 35 f5 48 b1 91 c1 02 4e 04 96 91 7b b0 33 f0 b1 14 4e 11 6f b5 40
af 1b 45 a5 4a ef 7e b6 ac f2 a0 1f 58 3f 12 46 60 3c 8d a1 e0 7d cf 57 3e 33 1e
fb 47 f1 aa 15 97 07 55 66 a5 b5 2d 2e d8 80 59 b2 a7 0d b7 46 ec 21 63 ff 35 ab
a5 02 cf 2a f4 4c fe 7b f5 94 5d 84 4d a8 f2 60 8f db 0e 25 3c 9f 73 71 cf 94 df
4a ea db df 72 38 8c f3 96 bd f1 17 bc d2 ba 3b 45 5a c6 a7 f6 c6 17 8b 01 9d fc
19 a8 2a 83 16 b8 3a 48 fe 4e 3e a0 ab 06 19 e9 53 f3 80 13 07 ed 2d bf 3f 0a 3c
55 20 39 2c 2c 00 69 74 95 4a bc 20 b2 a9 79 e5 18 89 91 a8 dc 1c 4d ef bb 7e 37
0b 5d fe 39 a5 88 52 8c 00 6c ec 18 7c 41 bd f6 8b 75 77 ba 60 9d 84 e7 fe 2d 02
03 01 00 01":
```

5. Contagem

5.1. Técnicas básicas

Neste capítulo começaremos por abordar os dois princípios gerais, intuitivamente claros, que fundamentam os raciocínios básicos que se fazem na resolução de problemas elementares de contagem.

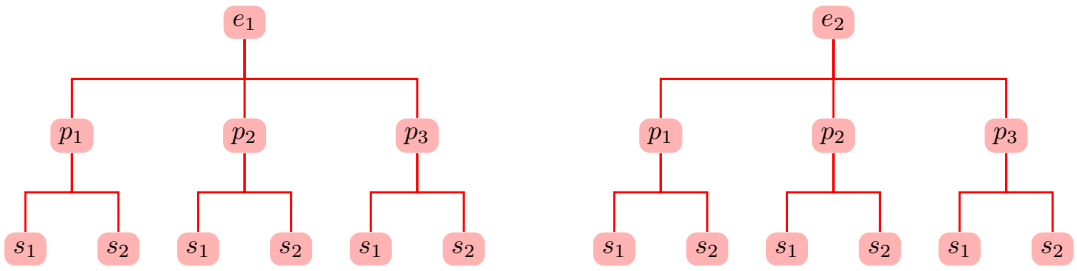
O princípio fundamental da contagem (chamado *princípio da multiplicação*) diz que se há p maneiras de fazer uma escolha E_1 e, feita a escolha E_1, há q maneiras de fazer a escolha E_2, então o número de maneiras de fazer sucessivamente as escolhas E_1 e E_2 é $p \times q$.

Mais geralmente:

> *Quando pretendemos realizar m escolhas múltiplas e existem p_1 possibilidades para a primeira escolha, p_2 possibilidades para a segunda escolha, etc., p_m possibilidades para a m-ésima escolha então se as escolhas forem combinadas livremente, o número total de possibilidades para o conjunto total das escolhas é igual a $p_1 \times p_2 \times \cdots \times p_m$.*

Exemplo. *O menu de um restaurante apresenta duas entradas, três pratos principais e duas sobremesas. Quantas ementas diferentes (com uma entrada, um prato principal e uma sobremesa) podemos escolher?*

Num problema tão simples podemos esquematizar as várias possibilidades e contá-las; se designarmos por $E = \{e_1, e_2\}$ o conjunto das entradas, por $P = \{p_1, p_2, p_3\}$ o conjunto dos pratos principais e por $S = \{s_1, s_2\}$ o conjunto das sobremesas, o seguinte quadro mostra os resultados possíveis:



Portanto, $2 \times 3 \times 2 = 12$ é a solução do problema. O quadro dá-nos também imediatamente a enumeração de todos os casos possíveis: $\{e_1, p_1, s_1\}, \{e_1, p_1, s_2\}, \ldots, \{e_2, p_3, s_2\}$.

A justificação para o Princípio da Multiplicação é a seguinte:

Fazer a escolha E_1 significa escolher um elemento de um conjunto S_1 de cardinal p_1, fazer a escolha E_2 significa escolher um elemento de um conjunto S_2 de cardinal p_2, e assim sucessivamente, pelo que fazer a escolha sucessiva $E_1, E_2, \cdots, E_m$ significa tomar um elemento do produto cartesiano $S_1 \times S_2 \times \cdots \times S_m$. Logo o número de maneiras de fazer tal escolha é igual ao cardinal $|S_1 \times S_2 \times \cdots \times S_m|$. Portanto o Princípio da Multiplicação assenta no seguinte facto, facilmente demonstrável por indução:

Princípio da Multiplicação. *Sejam $S_1, S_2, \ldots, S_m$ conjuntos finitos e $S = S_1 \times S_2 \times \cdots \times S_m$ o seu produto cartesiano. Então*

$$|S| = |S_1| \times |S_2| \times \cdots \times |S_m|.$$

Por outro lado, é evidente que o número de maneiras diferentes de escolher uma entrada ou um prato principal ou uma sobremesa é igual a $2 + 3 + 2 = 7$. Este raciocínio é um caso particular do chamado Princípio da Adição:

Princípio da Adição. *Se $S_1, S_2, \ldots, S_m$ formarem uma partição de um conjunto finito S, ou seja, se $S = \bigcup_{i=1}^{m} S_i$ e $S_i \cap S_j = \emptyset$ para quaisquer $i, j \in \{1, 2, \ldots, m\}$, $i \neq j$, então*

$$|S| = \sum_{i=1}^{m} |S_i|.$$

Caso alguns dos subconjuntos $S_1, S_2, \ldots, S_m$ tenham intersecção não vazia, um princípio mais geral (o chamado Princípio da Inclusão-Exclusão) será necessário para contar os elementos de S. Estudaremos esse princípio mais adiante. Os princípios da multiplicação e da adição podem ser facilmente demonstrados pelo Princípio de Indução Matemática.

Teste 1. *Uma bandeira é formada por 7 listras que devem ser coloridas usando apenas as cores verde, amarela e vermelha. Se cada listra deve ter apenas uma cor e não se pode usar cores iguais em listras adjacentes, de quantas maneiras se pode colorir a bandeira?*

Solução. Colorir a bandeira equivale a escolher a cor de cada listra. Há 3 maneiras de escolher a cor da primeira listra e, a partir daí, 2 maneiras de escolher a cor de cada uma das outras 6 listras. Portanto a resposta é $3 \times 2^6 = 192$.

Teste 2. *Quantos são os números de três algarismos distintos?*

Solução. O primeiro algarismo pode ser escolhido de 9 maneiras, pois não pode ser igual a 0. O segundo algarismo pode ser escolhido de 9 maneiras, pois não pode ser igual ao primeiro algarismo. O terceiro algarismo pode ser escolhido de 8 maneiras, pois não pode ser igual ao primeiro e segundo algarismos. A resposta é $9 \times 9 \times 8 = 648$.

Estes exemplos mostram-nos qual deve ser a estratégia para resolver problemas de contagem. Citando Elon Lages Lima[27]:

(1) *Postura.* Devemos sempre colocar-nos no papel da pessoa que deve fazer a acção solicitada pelo problema e ver que decisões devemos tomar. No Teste 2, colocámo-nos no papel da pessoa que deveria escrever o número de três algarismos; no Teste 1, colocámo-nos no papel da pessoa que deveria colorir a bandeira.

[27] *A matemática do ensino médio*, Sociedade Brasileira de Matemática, 2000.

(2) *Divisão.* Devemos, sempre que possível, dividir as decisões a serem tomadas em decisões mais simples. Colorir a bandeira foi dividido em colorir cada listra; formar um número de três algarismos foi dividido em escolher cada um dos três algarismos.

(3) *Não adiar dificuldades.* Pequenas dificuldades adiadas costumam transformar-se em grandes dificuldades. Se uma das decisões a serem tomadas for mais restrita que as demais, essa é a decisão que deve ser tomada em primeiro lugar. No Teste 2, a escolha do primeiro algarismo é uma decisão mais restrita do que as outras, pois o primeiro algarismo não pode ser igual a 0. Essa é portanto a decisão que deve ser tomada em primeiro lugar; adiá-la só serve para causar problemas. Com efeito, começando a escolha dos algarismos pelo último, há 10 maneiras de escolher o último algarismo. Em seguida, há 9 maneiras de escolher o algarismo central, pois não podemos repetir o algarismo já usado. Agora temos um impasse: de quantas maneiras podemos escolher o primeiro algarismo? A resposta é "depende". Se antes não tivermos usado o zero, haverá 7 maneiras de escolher o primeiro algarismo, pois não poderemos usar nem o zero nem os dois algarismos já usados; se já tivermos usado o zero, haverá 8 maneiras de escolher o primeiro algarismo. Isto mostra como algumas pessoas conseguem, por erros de estratégia, tornar complicadas as coisas mais simples.

Teste. *Quantos são os números pares de três algarismos distintos?*

Solução. Há 5 maneiras de escolher o último algarismo. Note que começamos pelo último algarismo, que é o mais restrito; o último algarismo só pode ser 0,2,4,6 ou 8. Em seguida, vamos ao primeiro algarismo. De quantas maneiras se pode escolher este algarismo? A resposta é "depende": se não tivermos usado o 0, haverá 8 maneiras de escolher o primeiro algarismo, pois não poderemos usar nem o 0 nem o algarismo já usado na última posição; se já tivermos usado o 0, haverá 9 maneiras de escolher o primeiro algarismo, pois apenas o 0 não poderá ser usado na primeira posição.

Este tipo de impasse é comum na resolução de problemas e há dois métodos para ultrapassá-lo. O primeiro método consiste em voltar atrás e contar separadamente os números que terminam em 0 e os que não terminam em 0. Comecemos pelos que terminam em 0. Há uma maneira de escolher o último algarismo, 9 maneiras de escolher o primeiro e 8 maneiras de escolher o algarismo central. Há assim $1 \times 9 \times 8 = 72$ números terminados em 0. Para os que não terminam em 0, há 4 maneiras de escolher o último algarismo, 8 maneiras de escolher o primeiro e 8 maneiras de escolher o algarismo central. Há pois $4 \times 8 \times 8 = 256$ números que não terminam em 0. A resposta final é $72 + 256 = 328$.

O segundo método consiste em ignorar uma das restrições do problema, o que nos fará contar em demasia. Depois descontaremos o que tiver sido contado indevidamente. Em primeiro lugar fazemos de conta que o 0 pode ser usado na primeira posição do número. Procedendo assim, há 5 maneiras de escolher o último algarismo (só pode ser 0,2,4,6, ou 8), 9 maneiras de escolher o primeiro algarismo (não podemos repetir o algarismo usado na última casa) e 8 maneiras de escolher o algarismo central. Há $5 \times 9 \times 8 = 360$ números, aí incluídos os que começam por 0. Por fim vamos determinar quantos desses números começam por 0; são esses os números que foram contados indevidamente. Há só uma maneira de escolher o primeiro algarismo (tem que ser 0), 4 maneiras de escolher o último (só pode ser 2,4,6, ou 8 — lembre-se que os algarismos

são distintos) e 8 maneiras de escolher o algarismo central (não podemos repetir os algarismos já usados). Há assim $1 \times 4 \times 8 = 32$ números começados por 0. A resposta final é $360 - 32 = 328$.

É claro que este problema poderia ter sido resolvido com um truque. Para determinar quantos são os números pares de três algarismos distintos, poderíamos calcular os números de três algarismos distintos menos os números ímpares de três algarismos distintos.

Para os números de três algarismos distintos, há 9 maneiras de escolher o primeiro algarismo, 9 maneiras de escolher o segundo e 8 maneiras de escolher o último. Portanto há $9 \times 9 \times 8 = 648$ números de três algarismos distintos.

Para os números ímpares de três algarismos distintos, há 5 maneiras de escolher o último algarismo, 8 maneiras de escolher o primeiro e 8 maneiras de escolher o algarismo central. Há pois $5 \times 8 \times 8 = 320$ números ímpares de três algarismos distintos.

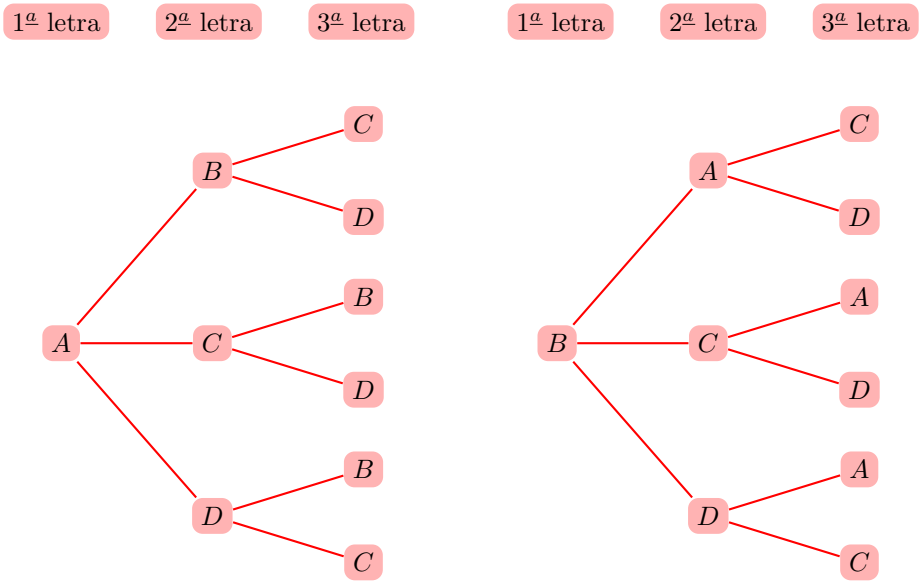
A resposta final é $648 - 320 = 328$.

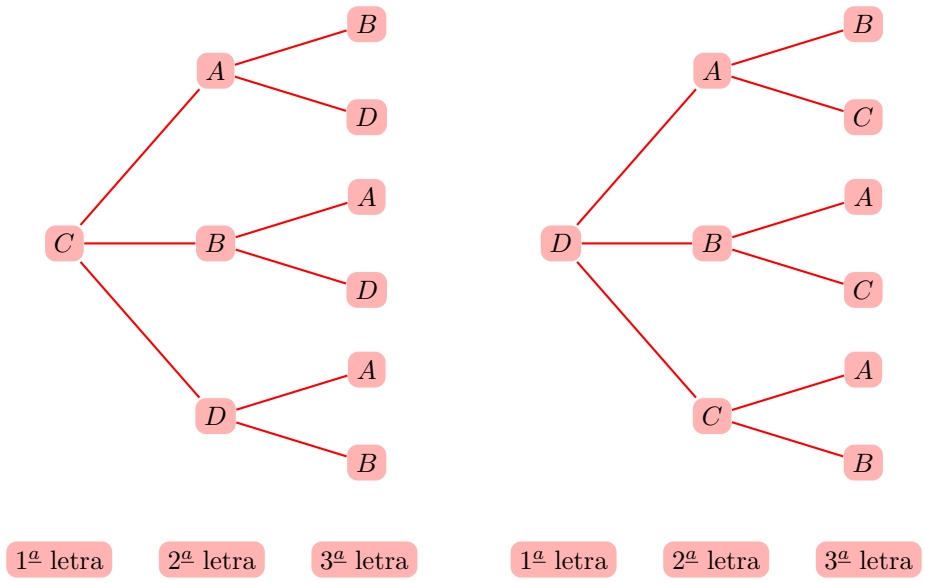
Alguns tipos de problemas de contagem, embora sejam aplicações do Princípio da Multiplicação, aparecem recorrentemente com muita frequência. Para esses problemas, vale a pena conhecer de cor fórmulas que forneçam imediatamente a resposta.

(1) *De um conjunto de 4 letras $\{A, B, C, D\}$, quantas sequências de 3 letras se podem formar se repetições de uma mesma letra não forem permitidas?*

(2) *Considere 4 pontos A, B, C, D num plano, tais que nenhum grupo de 3 esteja situado sobre uma mesma recta. Quantos triângulos diferentes podem ser construídos usando esses pontos como vértices?*

No primeiro problema temos $4 \times 3 \times 2$ hipóteses diferentes:





Note-se que neste caso a ordem pela qual se escrevem as letras na sequência é fundamental:

$$ABC \neq BAC \neq CAB.$$

No segundo problema já a ordem não interessará pois, por exemplo, as sequências de vértices

$$ABC,\ BAC,\ CAB$$

definem o mesmo triângulo (o que define um triângulo é o conjunto dos seus três vértices, e não a ordem pela qual os poderemos escrever).

Estes dois problemas revelam-nos duas estruturas diferentes, que ocorrem frequentemente, e que abordaremos de seguida, de uma maneira mais formal e sistemática.

Seja $S = \{a_1, a_2, \ldots, a_n\}$ um conjunto com n elementos. Uma *permutação dos n elementos de S, r a r* $(0 < r \leq n)$ é uma sequência **ordenada** $(a_1, a_2, \ldots, a_r)$ de elementos de S. Assumimos que não há repetição de elementos nas sequências ordenadas. Denotaremos o número de permutações dos n elementos de S, r a r, por $P(n, r)$. Se $r = n$ diremos simplesmente que se trata de *permutações de n elementos*.

No exemplo (1) acima pedia-se o cálculo de $P(4, 3)$, que vimos ser igual a 24. Seja $S = \{a, b, c\}$. As permutações dos 3 elementos de S, 2 a 2, são

$$(a, b), (a, c), (b, a), (b, c), (c, a), (c, b).$$

Logo $P(3, 2) = 6$. As permutações de 3 elementos são $(a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b)$ e (c, b, a) pelo que $P(3, 3) = 6$.

É possível fazer estes cálculos no `WolframAlpha`:

```
> permutation (3,2)

    6
```

```
> permutation (3,3)
```

6

Proposição 1. *Para quaisquer inteiros positivos n e r tais que $r \leq n$,*

$$P(n,r) = n \times (n-1) \times \cdots \times (n-r+1).$$

Prova. O primeiro elemento da sequência ordenada pode ser escolhido de entre n elementos diferentes. O segundo de entre $n-1$, e assim sucessivamente, até ao elemento na r-ésima posição que poderá ser escolhido de entre $n-(r-1) = n-r+1$ elementos diferentes. Logo, pelo Princípio da Multiplicação, a construção da sequência pode ser realizada de

$$n \times (n-1) \times \cdots \times (n-r+1)$$

maneiras diferentes, ou seja, $P(n,r) = n \times (n-1) \times \cdots \times (n-r+1)$. □

Convencionando que $0! = 1$, podemos reescrever a Proposição 1 do seguinte modo:

$$P(n,r) = \frac{n!}{(n-r)!} \quad (n \geq r > 0).$$

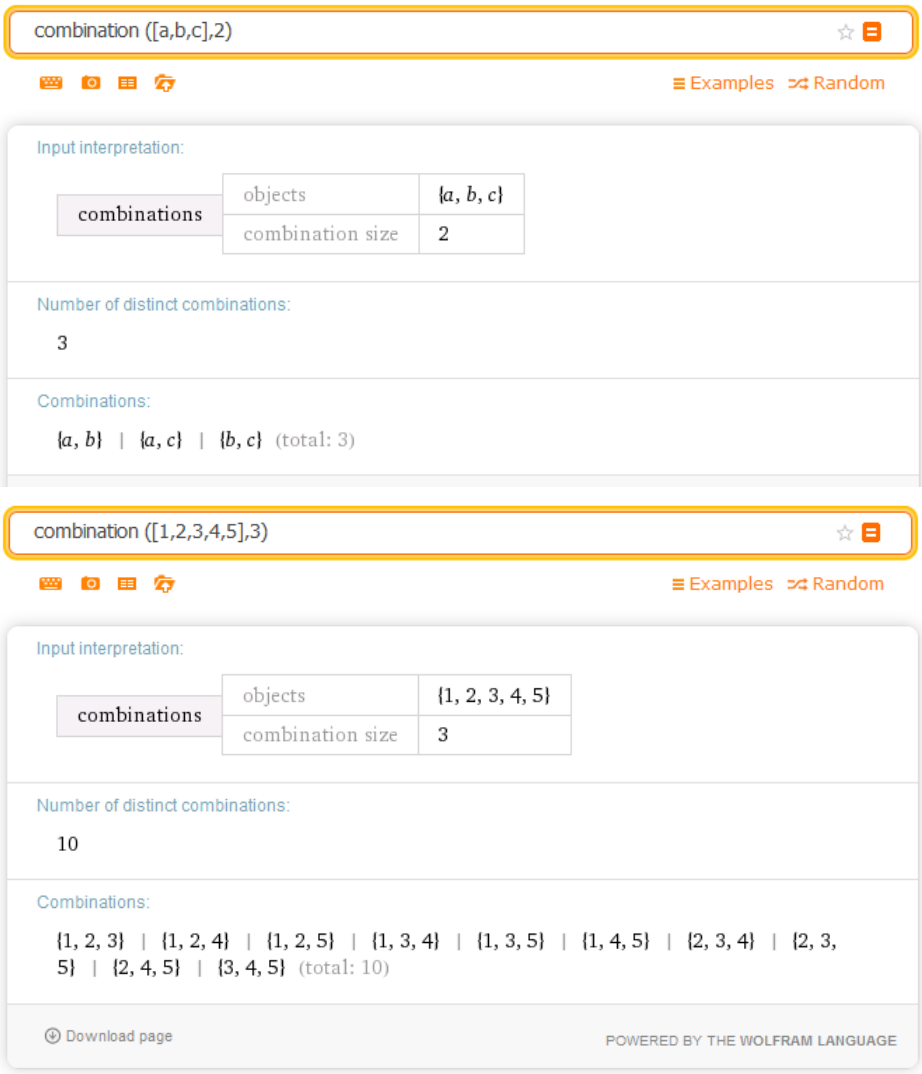
Esta fórmula continua válida para $r = 0$ se definirmos $P(n,0)$ $(n \geq 0)$ como sendo igual a 1 (correspondendo à permutação vazia). O caso particular $r = n$ diz-nos que o número $P(n,n)$ de permutações de n elementos é igual a $n!$.

Seja S um conjunto com n elementos. Uma *combinação dos n elementos de S, r a r*, com $0 < r \leq n$, é um subconjunto de S com r elementos (distintos, evidentemente). Denotaremos o número de combinações de n elementos, r a r, por $C(n,r)$ ou $\binom{n}{r}$.

Exemplo. As combinações dos elementos de $S = \{a,b,c\}$, dois a dois, são $\{a,b\}, \{a,c\}$ e $\{b,c\}$. Portanto $C(3,2) = 3$. As combinações dos elementos de S três a três reduzem-se a $\{a,b,c\}$. Logo $C(3,3) = 1$.

```
> combination (3,2)
```

3



Proposição 2. *Para quaisquer inteiros positivos n e r tais que $r \leq n$ temos*

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}.$$

Prova. Seja S um conjunto com n elementos. Cada permutação dos elementos de S, r a r, pode ser obtido em 2 passos:

1. Seleccionando um subconjunto de S com r elementos;
2. Reordenando esses r elementos de modo a formar a permutação desejada.

Como $C(n,r)$ representa o número de subconjuntos de S com r elementos, podemos efectuar o passo 1 de $C(n,r)$ maneiras diferentes. Uma vez seleccionado um determinado subconjunto de r elementos, estes podem ser reordenados de $P(r,r) = r!$ maneiras diferentes. Atendendo ao Princípio da Multiplicação, concluímos que $P(n,r) = C(n,r) \times r!$, isto é,

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}. \qquad \square$$

Note que $C(n,n) = 1$ e $C(n,1) = n$. Convencionando que $C(n,0) = 1$ para $n \geq 0$ a fórmula da Proposição 2 continua válida para $n \geq r = 0$.

A combinatória[28] e a teoria das probabilidades partilham raízes comuns e estão muito ligadas. De facto, o cálculo de uma *probabilidade discreta* (probabilidade de um acontecimento num espaço de resultados finito[29]) é um mero problema de contagem: numa experiência aleatória com um espaço $\mathcal{S}$ de resultados equiprováveis e finito, a *probabilidade* $p(A)$ de um acontecimento A é igual a $|A|/|\mathcal{S}|$, ou seja,

$$\boxed{p(A) = \frac{\text{número dos resultados } \textbf{favoráveis} \text{ a } A}{\text{número total de resultados } \textbf{possíveis}}}$$

Basta então contar o número total de resultados possíveis e, de entre esses, quais são favoráveis à realização do acontecimento.

Exemplo 1. *Existem várias lotarias, como o totoloto, que dão prémios avultados a pessoas que acertam correctamente em 6 números escolhidos entre os primeiros n inteiros positivos (habitualmente, n está entre 30 e 50). Qual é a probabilidade de uma pessoa ganhar o prémio no caso $n = 40$?*

Solução. Só existe uma combinação vencedora. O número total de resultados possíveis é igual ao número de maneiras diferentes de escolher um subconjunto de 6 números entre os primeiros 40 inteiros positivos, ou seja, é igual a

$$C(40,6) = \frac{40!}{34!\,6!} = 3\,838\,380.$$

Consequentemente, a probabilidade de acertar na combinação vencedora é igual a

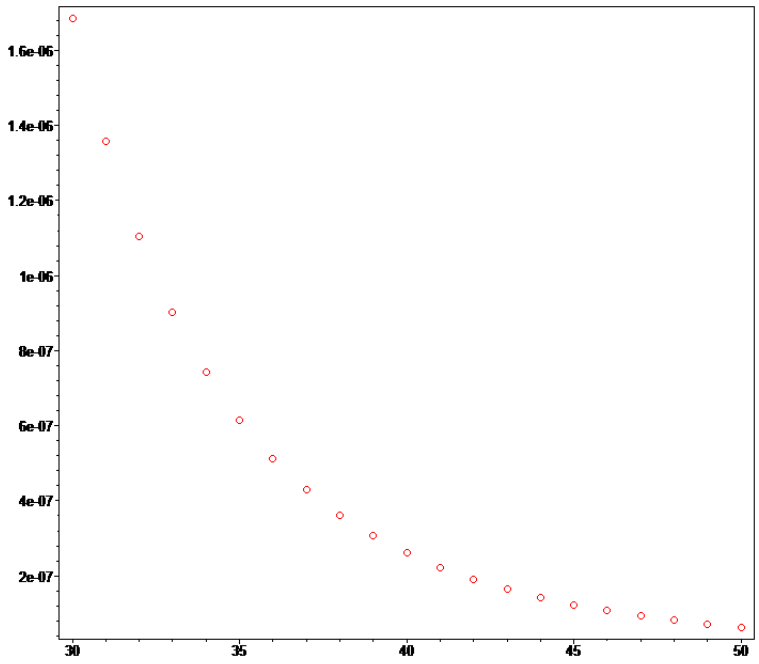
$$1/3\,838\,380 \sim 0.00000026.$$

Cálculo dessa probabilidade para todos os valores de n entre 30 e 50:

[28] Área da matemática que trata dos problemas de contagem.

[29] Uma *experiência aleatória* é um procedimento aleatório donde resulta um de entre vários resultados possíveis. O *espaço dos resultados* é o conjunto dos resultados *possíveis* da experiência. Um *acontecimento aleatório* é um subconjunto do espaço dos resultados, formado pelos elementos que são *favoráveis* à realização desse acontecimento.

n	casos possíveis	probabilidade	n	casos possíveis	probabilidade
30	593775	0.168413961510^{-5}			
31	736281	0.135817710910^{-5}	41	4496388	0.222400735910^{-6}
32	906192	0.110351890110^{-5}	42	5245786	0.190629202210^{-6}
33	1107568	0.902879100910^{-6}	43	6096454	0.164029778610^{-6}
34	1344904	0.743547494810^{-6}	44	7059052	0.141662081510^{-6}
35	1623160	0.616082210010^{-6}	45	8145060	0.122773804010^{-6}
36	1947792	0.513401841710^{-6}	46	9366819	0.106759829610^{-6}
37	2324784	0.430147489010^{-6}	47	10737573	0.931309151510^{-7}
38	2760681	0.362229464410^{-6}	48	12271512	0.814895507610^{-7}
39	3262623	0.306501854510^{-6}	49	13983816	0.715112384210^{-7}
40	3838380	0.260526576310^{-6}	50	15890700	0.629298898110^{-7}



Exemplo 2. *Qual é a probabilidade que uma mão de cinco cartas no póquer contenha quatro cartas do mesmo tipo?*

Solução. Pela regra da multiplicação, o número de mãos de cinco cartas com quatro cartas do mesmo tipo é o produto do número de maneiras de escolher um tipo (de entre os 13 tipos de carta diferentes) pelo número de maneiras de escolher quatro cartas desse tipo de entre todas as

cartas do baralho desse tipo (4 também) e pelo número de maneiras de escolher a quinta carta: $C(13,1) \times C(4,4) \times C(48,1)$. Como existem, no total, $C(52,5)$ mãos diferentes de cinco cartas, a probabilidade pedida é igual a

$$\frac{C(13,1) \times C(4,4) \times C(48,1)}{C(52,5)} = \frac{13 \times 1 \times 48}{2\,598\,960} \sim 0.00024.$$

Exemplo 3. *Através de um informador, a polícia sabe o local de encontro de um grupo de malfeitores. A identidade dos diferentes elementos do grupo é, no entanto, desconhecida. A tarefa do inspector Costa é prender o chefe do grupo. O inspector sabe que o chefe do grupo é o mais baixo dos cinco elementos do grupo, todos eles de diferentes alturas, que estarão presentes na reunião. Terminada a reunião, os bandidos, como medida de precaução, deixam o edifício separadamente, com um intervalo de 15 minutos. Como o inspector não sabe qual deles é o mais baixo, decide deixar sair os dois primeiros bandidos, e prender o primeiro dos seguintes que seja mais baixo do que os que até esse momento saíram. Qual é a probabilidade do inspector Costa prender a pessoa certa?*

Solução. Designemos pelas letras a, b, c, d, c os cinco bandidos de modo que as respectivas alturas satisfaçam $alt(a) < alt(b) < alt(c < alt(d) < alt(e)$. O objectivo do inspector Loureiro é, portanto, prender o bandido a. A probabilidade de ele realizar tal evento é igual ao quociente do número de permutações *favoráveis* do conjunto $\{a, b, c, d, e\}$ (isto é, as permutações $x_1x_2x_3x_4x_5$ tais que o elemento de $\{x_3, x_4, x_5\}$ com menor índice que seja mais baixo do que x_1 e x_2 seja exactamente o bandido a) pelo número de permutações total (que é igual a $5! = 120$). Determinemos então o número de permutações favoráveis:

Claro que nenhuma permutação na qual a aparece na 1^a ou 2^a posições é favorável. Aquelas em que a aparece na 3^a posição são todas favoráveis e são em número de $4! = 24$. Contemos agora as permutações favoráveis nas quais a aparece na 4^a posição: as 6 nas quais b está na 1^a posição são favoráveis; analogamente as 6 nas quais b está na 2^a posição são também favoráveis; nenhuma das que b aparece na 3^a posição é favorável; das que b aparece na 5^a posição somente 4 são favoráveis (*cdcab*, *dceab*, *ccdab*, *ecdab*). Portanto ao todo temos 16 permutações favoráveis nas quais a esta na 4^a posição.
Finalmente contemos as permutações favoráveis nas quais a aparece na 5^a posição: obviamente são aquelas em que b aparece na 1^a ou 2^a posições; portanto, são $3 \times 2 + 3 \times 2 = 12$ permutações. Em conclusão, o número de permutações favoráveis é igual a $24 + 16 + 12 = 52$ e, consequentemente, a probabilidade do inspector Loureiro apanhar o chefe do bando é igual a $\frac{52}{120} \sim 0,433333$.

Os números $\binom{n}{r} = C(n,r)$ chamam-se números (ou coeficientes) binomiais (por razões que serão evidentes mais adiante) e têm muitas propriedades importantes (e fascinantes!). Em fórmulas que aparecem na análise de algoritmos, em problemas de probabilidades, etc., estes números ocorrem variadas vezes, revelando-se uma necessidade saber manipulá-los.

Da Proposição 2 conclui-se imediatamente:

Corolário 3. *Para quaisquer inteiros n e r tais que $0 \leq r \leq n$ tem-se $\binom{n}{r} = \binom{n}{n-r}$.* □

Fórmula de Pascal. *Para quaisquer inteiros n e r tais que $0 \leq r \leq n-1$ tem-se*

$$\binom{n}{r} + \binom{n}{r+1} = \binom{n+1}{r+1}.$$

□

Utilizando a Fórmula de Pascal e observando que $\binom{n}{0} = \binom{n}{n} = 1$, podemos imediatamente calcular os números $\binom{n}{r}$ para $0 \leq r \leq n$, sem necessitar de utilizar a Proposição 2. Dispondo esses números do seguinte modo

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$	$\cdots$
0	1									
1	1	1								
2	1	2	1							
3	1	3	3	1						
4	1	4	6	4	1					
5	1	5	10	10	5	1				
6	1	6	15	20	15	6	1			
7	1	7	21	35	35	21	7	1		
8	1	8	28	56	70	56	28	8	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

obtemos o chamado *Triângulo de Pascal*.

Muitas das relações envolvendo coeficientes binomiais podem ser descobertas através da simples observação do Triângulo de Pascal. Por exemplo:

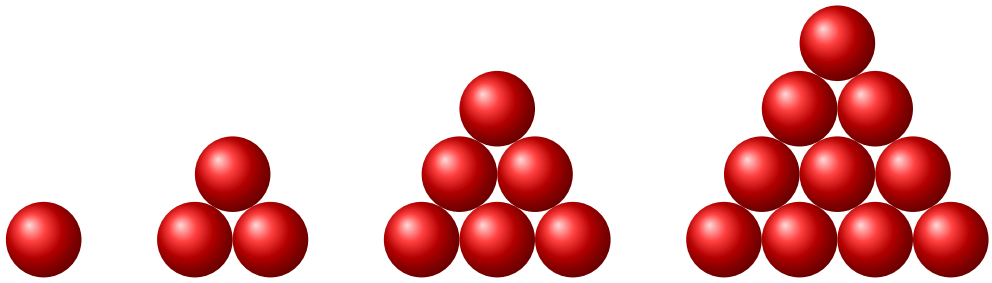
1. Se adicionarmos os elementos em cada linha n obtemos o valor 2^n, ou seja,

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n.$$

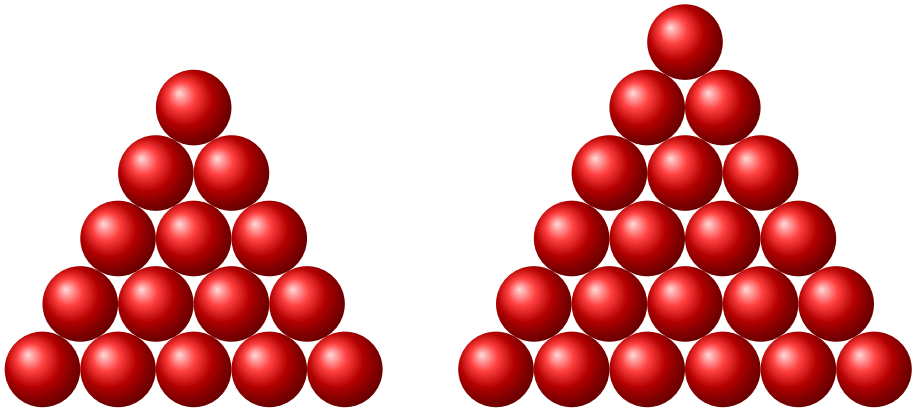
   Sendo S um conjunto com n elementos, como $\binom{n}{r}$ é o número de subconjuntos de S com r elementos, então podemos concluir que o número de subconjuntos de S é igual a 2^n.

2. Facilmente se observa, pela simetria em cada linha, que $\binom{n}{r} = \binom{n}{n-r}$ (Corolário 3).

3. Na terceira coluna aparecem os chamados *números triangulares*, correspondentes ao número de bolas nas seguintes figuras triangulares:



1 3 6 10



15 21

A validade destas identidades pode depois ser facilmente verificada utilizando o Princípio de Indução Matemática.

Teorema Binomial (ou Fórmula do Binómio de Newton[30]). *Para quaisquer* $x, y \in \mathbb{R}$ *e* $n \in \mathbb{N}$*,*

$$(x+y)^n = \sum_{r=0}^{n} \binom{n}{r} x^r y^{n-r}.$$

Prova. Não é difícil provar o Teorema Binomial por indução matemática. No entanto, a seguinte prova, puramente combinatorial, é mais curta e elegante:

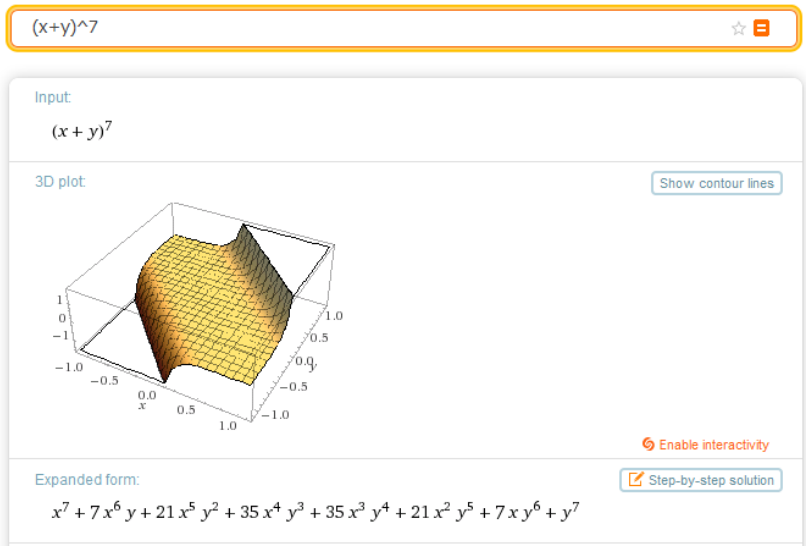
Quando efectuamos a multiplicação $(x+y)(x+y)\cdots(x+y)$ até não restarem mais parênteses, cada um dos factores $(x+y)$ contribui com um x ou um y para cada parcela. Resultam portanto 2^n parcelas e cada uma delas pode ser escrita na forma $x^r y^{n-r}$ para algum $r \in \{0, 1, \ldots, n\}$. Obtemos a parcela $x^r y^{n-r}$ precisamente quando escolhemos x em r dos factores e y nos restantes $n-r$. Então o número de vezes que a parcela $x^r y^{n-r}$ ocorre na expansão é igual ao número de maneiras diferentes de seleccionar r dos n factores $(x+y)$, ou seja, ao número $\binom{n}{r}$ de combinações de n elementos r a r. □

Expansão do Binómio de Newton para $n = 1, 2, \ldots, 7$**:**

$x + y$
$x^2 + 2xy + y$
$x^3 + 3xy + 3xy + y^3$
$x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$
$x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5$

[30] O Teorema Binomial dá-nos uma fórmula para o desenvolvimento de $(x+y)^n$ com $x, y \in \mathbb{R}$, $n \in \mathbb{N}$. Em 1676, Newton generalizou-o, obtendo um desenvolvimento para $(x+y)^\alpha$ com $\alpha \in \mathbb{R}$. Para se obter esta forma é necessário estender o domínio de definição dos números binomiais $\binom{n}{r}$, permitindo que $n \in \mathbb{N}$ e $r \in \mathbb{Z}$. Neste caso geral o desenvolvimento torna-se uma série infinita e consequentemente algumas questões de convergência se levantam, por isso não vamos sequer enunciar esse resultado.

$x^6 + 6x^5y + 15x^4y^2 + 20x^3y^3 + 15x^2y^4 + 6xy^5 + y^6$
$x^7 + 7x^6y + 21x^5y^2 + 35x^4y^3 + 35x^3y^4 + 21x^2y^5 + 7xy^6 + y^7$



O Teorema Binomial justifica a designação de coeficientes binomiais para os números $\binom{n}{r}$: estes números são precisamente os coeficientes da expansão do Binómio de Newton. Note que a fórmula do binómio ainda é válida para $n = 0$.

Do Binómio de Newton podemos obter, como casos particulares, algumas identidades úteis. Por exemplo:

Para $x = 1$ e $y = 1$,

$$2^n = \sum_{r=0}^{n} \binom{n}{r} \text{ , para } n \geq 0.$$

Para $y = 1$,

$$(x+1)^n = \sum_{r=0}^{n} \binom{n}{r} x^r = \sum_{r=0}^{n} \binom{n}{n-r} x^r \text{ , para } n \geq 0.$$

Para $x = -1$ e $y = 1$,

$$0 = \sum_{r=0}^{n} (-1)^r \binom{n}{r} \text{ , para } n \geq 1.$$

Em resumo, temos à disposição vários métodos que podemos usar para obter identidades envolvendo os números binomiais:

(1) Definição;

(2) Indução matemática;

(3) Triângulo de Pascal;

(4) Fórmula de Pascal;

(5) Argumentos combinatoriais;

(6) Teorema Binomial.

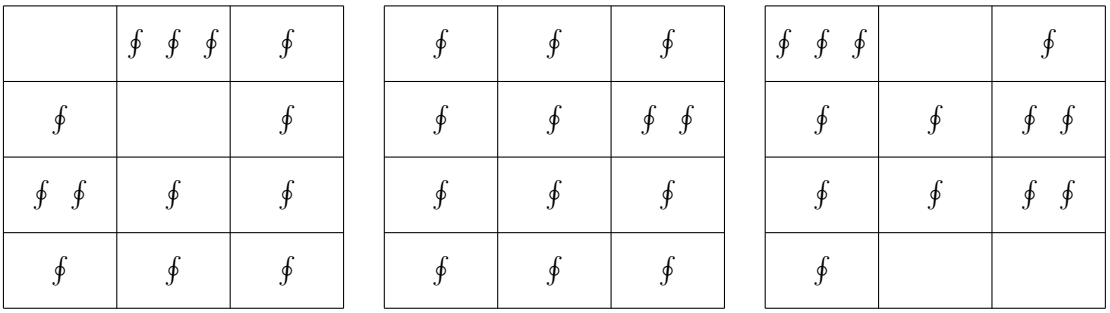
Apêndice: O Princípio dos Pombais

Há um outro princípio combinatorial básico muito intuitivo que, apesar de elementar, permite a resolução de muitos problemas (de existência de determinadas configurações), alguns surpreendentes e difíceis.

Princípio dos Pombais (ou de Dirichlet). *Se $n+1$ objectos forem colocados em n caixas, pelo menos uma das caixas ficará com dois ou mais objectos.*

Prova. Faremos a demonstração por redução ao absurdo. Suponhamos que em cada caixa ficava, no máximo, um objecto. Então o número de objectos seria no máximo n, o que contradiz a hipótese. Portanto alguma caixa conterá, pelo menos, dois objectos. □

Formulado em termos de pombos este princípio diz que se n pombos voarem para $n-1$ pombais, necessariamente um pombal será ocupado por dois ou mais pombos. Por exemplo, no caso de 13 pombos e 12 pombais,



são algumas configurações possíveis.

Solução do Problema (A2).[31] Escolhendo 101 inteiros entre os inteiros $1, 2, \ldots, 200$, vamos aplicar o Princípio dos Pombais para mostrar que entre os inteiros escolhidos existem dois tais que um é divisor do outro.

Qualquer inteiro pode ser escrito na forma $2^k a$, com $k \in \mathbb{N}_0$ e a ímpar. Para qualquer inteiro entre 1 e 200, a é um dos números $1, 3, 5, \ldots, 199$. Logo, entre os 101 escolhidos, dois são da forma $2^{k_1}a_1$ e $2^{k_2}a_2$ com $a_1 = a_2$. Se $k_1 \leq k_2$ então $2^{k_1}a_1$ é divisor de $2^{k_2}a_2$. Caso $k_1 > k_2$, $2^{k_2}a_2$ é divisor de $2^{k_1}a_1$. □

Vamos agora apresentar uma forma mais geral do Princípio dos Pombais.

Proposição 4. *Sejam $p_1, p_2, \ldots, p_n$ inteiros positivos. Se $p_1 + p_2 + \cdots + p_n - n + 1$ objectos forem colocados em n caixas, pelo menos uma das caixas ficará com p_i ou mais objectos, para algum $i \in \{1, 2, \ldots, n\}$.*

[31] No capítulo *Que é a Matemática Discreta?*

Prova. Suponhamos por absurdo que, para cada $i \in \{1, 2, \ldots, n\}$, a i-ésima caixa ficava com, no máximo, $p_i - 1$ elementos. Então o número total de objectos não excederia

$$(p_1 - 1) + (p_2 - 1) + \cdots + (p_n - 1) = p_1 + p_2 + \cdots + p_n - n,$$

o que é absurdo. Logo existe $i \in \{1, 2, \ldots, n\}$ tal que a i-ésima caixa conterá pelo menos p_i objectos. □

Observações. (1) Se $p_1 = p_2 = \cdots = p_n = 2$ obtemos o Princípio dos Pombais.
(2) Fazendo $p_1 = p_2 = \cdots = p_n = r \in \mathbb{N}$, podemos afirmar que

> "*se* $n(r-1)+1$ *objectos forem colocados em* n *caixas, pelo menos uma das caixas ficará com* r *ou mais objectos*".

Por exemplo, no problema (A1), como o número de caixas é igual ao número de notas possíveis, ou seja, 201, podemos assegurar que se comparecerem $201(r-1)+1 = 201r - 200$ alunos ao exame, r de entre eles terão a mesma nota.

Solução do Problema (A3).[32] Provemos, utilizando a Observação (2), que de uma sequência $a_1, a_2, \ldots, a_{n^2+1}$ de números reais é possível extrair uma subsequência crescente ou decrescente com $n+1$ elementos.

Suponhamos que não existe nenhuma subsequência crescente com $n+1$ elementos. Para $k \in \{1, 2, \ldots, n^2+1\}$ seja m_k o número de elementos da maior subsequência crescente que começa em a_k. É evidente que, para cada $k \in \{1, 2, \ldots, n^2+1\}$, $m_k \geq 1$ e $m_k \leq n$. Temos então n^2+1 inteiros, $m_1, m_2, \ldots, m_{n^2+1}$, entre 1 e n. Como $n(r-1)+1 = n^2+1$ para $r = n+1$, podemos concluir que $n+1$ desses inteiros são iguais entre si. Sejam eles

$$m_{k_1}, m_{k_2}, \ldots, m_{k_{n+1}},$$

onde

$$1 \leq k_1 < k_2 < \cdots < k_{n+1} \leq n^2 + 1.$$

Se existisse algum $i \in \{1, 2, \ldots, n\}$ tal que $a_{k_i} < a_{k_{i+1}}$, seria possível construir uma subsequência crescente começando em a_{k_i} com $m_{k_{i+1}}+1$ elementos, o que é absurdo uma vez que $m_{k_i} = m_{k_{i+1}}$. Consequentemente,

$$a_{k_1} \geq a_{k_2} \geq \cdots \geq a_{k_{n+1}},$$

isto é,

$$a_{k_1}, a_{k_2}, \cdots, a_{k_{n+1}}$$

é uma subsequência decrescente com $n+1$ elementos.

Mostrámos assim que existe uma subsequência crescente ou uma subsequência decrescente com $n+1$ elementos. □

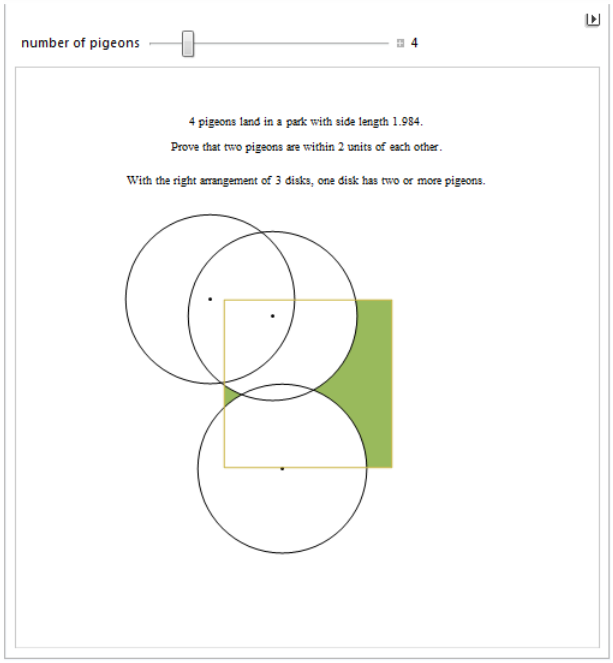
Em particular, nos primeiros 101 números naturais, dispostos por qualquer ordem, será sempre possível encontrar 11 números que formam ou uma sequência crescente ou uma sequência

[32] Difícil!

decrescente. Isto já não acontece se tomarmos apenas os primeiros 100 números naturais. Como se poderá ordenar esses números de forma a não ser possível encontrar a desejada sequência de 11 elementos? Bastará começar com 91, 92, 93 até 100, depois 81, 82, 83 até 90 e assim sucessivamente:

91	92	93	94	95	96	97	98	99	100
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

The Pigeonhole Principle - Disk Coverings



In 1834, Johann Dirichlet noted that if there are five objects in four drawers then there is a drawer with two or more objects. The Schubfachprinzip, or drawer principle, got renamed as the pigeonhole principle, and became a powerful tool in mathematical proofs.

In this Demonstration, n pigeons land in a park. If $n-1$ unit disks completely cover the park, then there must be a disk with two or more pigeons. These two pigeons must be within 2 units of each other. Move the disks to complete the proof.

Contributed by: Ed Pegg Jr

Wolfram Demonstrations Project™

5.2. Técnicas avançadas de contagem

Permutações e combinações com repetição

Podemos complicar um pouco o cálculo de permutações ou combinações se admitirmos repetição de elementos. Como o cálculo destas estruturas aparece em muitos problemas práticos será importante encontrarmos fórmulas que nos dêem a solução em cada um dos casos.

Seja então S um conjunto com n elementos. Consideremos as sequências $(a_1, a_2, \ldots, a_r)$ com elementos em S, eventualmente não todos distintos. Designemos estas sequências por *permutações com repetição de elementos de S, r a r*. Se admitirmos que cada elemento de S se pode repetir, como componente das permutações com repetição, tantas vezes quantas quisermos, temos:

Teorema 1. *O número destas permutações, que denotaremos por $\overline{P}(n, r)$, é igual a n^r.*

Prova. Ao construirmos cada permutação com repetição $(a_1, a_2, \ldots, a_r)$ temos n hipóteses de escolha do primeiro elemento a_1 e o mesmo número de hipóteses de escolha dos restantes elementos. Portanto, no total conseguimos construir

$$\underbrace{n \times n \times \cdots \times n}_{r \text{ vezes}} = n^r$$

permutações diferentes. □

Exemplos. (1) Se quisermos ter a certeza de obter 13 resultados certos no totobola teremos de preencher $\overline{P}(3, 13) = 3^{13}$ colunas.
(2) Observámos anteriormente que o número de subconjuntos de um conjunto $S = \{a_1, a_2, \ldots, a_n\}$ é igual a 2^n. Podemos concluir isso de outro modo: se a cada subconjunto S' de S fizermos corresponder uma sequência $(a'_1, a'_2, \ldots, a'_n)$ de comprimento n, definida por

$$a'_i = \begin{cases} 1 \text{ se } a_i \in S \\ \\ 0 \text{ se } a_i \notin S \end{cases}$$

concluímos que o número de subconjuntos de S é dado por $\overline{P}(2, n) = 2^n$.

Teste. *Qual é a probabilidade $p(n)$ de, entre n pessoas, existirem pelo menos duas que façam anos no mesmo dia ?*

Solução. Admitiremos só como datas possíveis de nascimento os 365 dias de um ano não bissexto. Calculemos a probabilidade do acontecimento contrário, isto é, a probabilidade de todas as pessoas fazerem anos em dias diferentes. O número de casos possíveis é igual a $\overline{P}(365, n)$, uma vez que cada caso é uma sequência de n elementos, que se podem repetir, escolhidos entre os 365 dias. O número de casos favoráveis é igual a $P(365, n)$ pois cada caso favorável é uma

sequência de n elementos, sem repetição, escolhidos entre os 365 dias. A probabilidade $p(n)$ é então dada por

$$1 - \frac{P(365,n)}{\overline{P}(365,n)} = 1 - \frac{365 \times 364 \times 363 \times \cdots \times (365-n+1)}{365^n}.$$

Alguns valores particulares de p: $p(5) = .0713557370$, $p(10) = .1169481777$, $p(15) = .2529013198$, $p(20) = .4114383836$, $p(25) = .5686997040$ e $p(30) = .7063162427$.

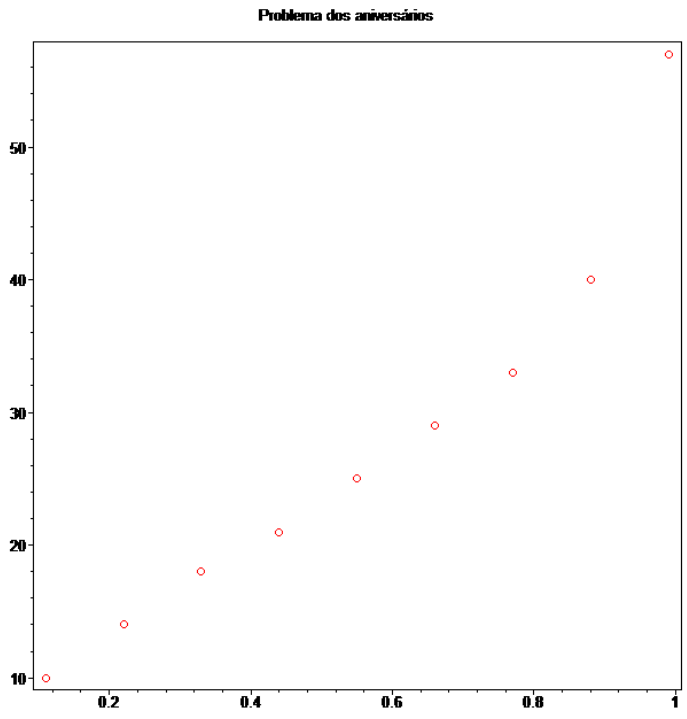
Qual é o menor valor de n para o qual $p(n) \geq 0.5$? e para o qual $p(n) \geq 0.99$? O seguinte procedimento calcula esses limites inferiores:

```
> Aniversarios := proc(percentagem)
>    local num_pessoas, prob;
>    # Inicializa
>    prob := 0; num_pessoas := 0;
>    # Efectua ciclo ate ao numero suficiente de pessoas
>    while prob < percentagem do
>       num_pessoas := num_pessoas +1;
>       prob := 1-(numbperm(365,num_pessoas) / 365^num_pessoas);
>    RETURN(num_pessoas);
>
> Aniversarios(.5); Aniversarios(.99);
```

23, 57

Este é o chamado *problema dos aniversários*, muito conhecido pois a resposta parece, à primeira vista, um pouco surpreendente: não é preciso um n muito grande para a probabilidade ser maior que 0.99, basta $n \geq 57$.

```
Aniversarios(.11)=10
Aniversarios(.22)=14
Aniversarios(.33)=18
Aniversarios(.44)=21
Aniversarios(.55)=25
Aniversarios(.66)=29
Aniversarios(.77)=33
Aniversarios(.88)=40
Aniversarios(.99)=57
```



Designemos por *multi-conjunto* uma estrutura similar à de um conjunto mas com a diferença de os seus elementos não terem forçosamente que ser distintos. Por exemplo, $M = \{a, a, b, b, b, c\}$ é um multi-conjunto com 6 elementos: 2 a's, 3 b's, 1 c. Costuma indicar-se um multi-conjunto especificando o número de ocorrências de cada elemento. Portanto o multi-conjunto M também se denota por $\{2 \cdot a, 3 \cdot b, c\}$. Chamaremos *combinação com repetição dos elementos de S, r a r*, aos multi-conjuntos de r elementos de S.

Teorema 2. *O número de combinações com repetição de elementos de S, r a r, que designaremos por $\overline{C}(n, r)$, é igual a $C(n-1+r, r) = \frac{(n-1+r)!}{r!\,(n-1)!}$.*

Prova. Podemos demonstrar este resultado utilizando somente argumentos combinatórios. De facto, cada combinação com repetição de n elementos r a r pode ser representada por uma sequência de $n-1$ barras e r asteriscos, do seguinte modo: as barras são utilizadas para demarcar em n células os n diferentes elementos de S, com a i-ésima célula contendo um asterisco sempre que o i-ésimo elemento de S ocorre na combinação. Por exemplo, para $S = \{a_1, a_2, a_3, a_4\}$:

Multi-conjunto	Representação
$\{a_1, a_1, a_2, a_4, a_4, a_4\}$	$**\,\vert\,*\,\vert\,\vert\,***$
$\{a_2, a_2, a_3, a_3, a_3, a_3\}$	$\vert\,**\vert\,****\vert$

Assim o número de combinações com repetição de n elementos r a r coincide com o número de sequências contendo $n-1$ barras e r asteriscos. O número de tais sequências é igual a $C(n-1+r, r)$, uma vez que cada sequência corresponde a uma escolha de r posições (das $n-1+r$ posições disponíveis) para colocar os r asteriscos (após a escolha das posições onde vão ficar os asteriscos, as barras ficam forçosamente nas posições restantes). □

Demonstrámos este teorema utilizando somente argumentos combinatórios. Aliás, a solução de problemas combinatórios requere geralmente o uso de métodos *ad hoc*; deve-se estudar a situação, desenvolver algum raciocínio e usar a própria intuição para encontrar a solução do problema. Isto não quer dizer que não existam princípios ou métodos que possam ser aplicados. Com efeito, já estudámos alguns. Mas todos eles requerem inteligência para se saber quando e como aplicá-los e, sobretudo, experiência (que naturalmente só se adquire resolvendo problemas).

Em resumo:

Tipo	**Repetição permitida?**	**Fórmula**
Permutações $P(n, r)$	Não	$\frac{n!}{(n-r)!}$
Combinações $C(n, r)$	Não	$\frac{n!}{r!\,(n-r)!}$
Permutações $\overline{P}(n, r)$	Sim	n^r
Combinações $\overline{C}(n, r)$	Sim	$\frac{(n-1+r)!}{r!\,(n-1)!}$

Exemplos. (1) O Dominó tem 28 peças. De facto, cada peça é uma combinação com repetição $\{n, m\}$ onde $n, m \in \{0, 1, \ldots, 6\}$, pelo que o número de peças é igual a $\overline{C}(7, 2) = C(8, 2) = 56/2 = 28$.
(2) O número de sequências crescentes (em sentido lato) com r componentes, escolhidas no conjunto $\{1, 2, \ldots, n\}$, é igual a $C(n + r - 1, r)$.
(3) O número de soluções da equação $x_1 + x_2 + x_3 = 11$, $(x_1, x_2, x_3 \in \mathbb{N}_0)$, é igual a

$$\overline{C}(3, 11) = C(3 + 11 - 1, 11) = C(13, 11) = 78.$$

Mais geralmente, $\overline{C}(n, r)$ é igual ao número de soluções inteiras (não negativas) da equação $x_1 + x_2 + \cdots + x_n = r$. De facto, qualquer combinação com repetição de elementos de $S = \{a_1, a_2, \ldots, a_n\}$, r a r, contém, para cada i, p_i elementos iguais a a_i, e $\sum_{i=1}^{n} p_i = r$; por outro lado, é evidente que a cada conjunto $\{p_1, p_2, \ldots, p_n\}$ de inteiros positivos ou nulos, com $p_1 + p_2 + \cdots + p_n = r$, podemos fazer corresponder a combinação com repetição de elementos de S, r a r,

$$\{\underbrace{a_1, a_1, \ldots, a_1}_{p_1 \text{ vezes}}, \underbrace{a_2, a_2, \ldots, a_2}_{p_2 \text{ vezes}}, \ldots, \underbrace{a_n, a_n, \ldots, a_n}_{p_n \text{ vezes}}\}.$$

Assim, as equações $x_1 + x_2 + \cdots + x_6 = 8$ e $x_1 + x_2 + \cdots + x_9 = 5$ têm o mesmo número de soluções inteiras não negativas pois

$$\overline{C}(6, 8) = C(13, 8) = \frac{13!}{8!\,5!} = 1287$$

e

$$\overline{C}(9, 5) = C(13, 5) = C(13, 8) = 1287.$$

(4) Qual é o valor de k depois do seguinte algoritmo ter sido executado?

```
k := 0
for i_1 := 1 to n
   for i_2 := 1 to i_1
      for i_3 := 1 to i_2
         ⋮
            for i_r := 1 to i_{r-1}
               k := k + 1
```

Observemos que o valor inicial de k é 0 e que uma unidade é adicionada a k de cada vez que o *ciclo* é atravessado com um conjunto de inteiros $i_1, i_2, \ldots, i_r$ tais que

$$1 \leq i_r \leq i_{r-1} \leq \cdots \leq i_2 \leq i_1 \leq n.$$

O número de tais conjuntos de inteiros é igual ao número de maneiras de escolher r inteiros de $\{1, 2, \ldots, n\}$, ordenados por ordem crescente, com repetição permitida, ou seja, é igual a $\overline{C}(n, r) = C(n + r - 1, r)$.

Podemos impôr algumas restrições à repetição dos elementos nas combinações e permutações:

Corolário 1. *Seja S um conjunto com n elementos. O número de combinações com repetição de elementos de S, r a r ($r \geq n$), contendo todos os elementos de S (cada um pelo menos uma vez) é igual a $C(r-1, n-1)$.*

Prova. Cada multi-conjunto conterá n elementos distintos de S, podendo os $r-n$ restantes serem elementos quaisquer de S. Consequentemente, o número a contar é igual a

$$\overline{C}(n, r-n) = C(n+r-n-1, r-n) = C(r-1, r-n) = C(r-1, r-1-r+n) = C(r-1, n-1).$$

□

Mais geralmente, tem-se:

Corolário 2. *Seja $S = \{a_1, a_2, \ldots, a_n\}$. O número de combinações com repetição de elementos de S, r a r, contendo cada elemento a_i pelo menos r_i vezes ($r \geq r_1 + r_2 + \cdots + r_n$), é igual a $C(n + r - r_1 - \cdots - r_n - 1, r - r_1 - \cdots - r_n)$.*

Prova. Em cada multi-conjunto haverá r_i elementos iguais a a_i ($i = 1, 2, \ldots, n$) podendo os restantes $r - r_1 - \cdots - r_n$ serem elementos quaisquer de S. Portanto, o número de combinações requerido é igual a

$$\overline{C}(n, r - r_1 - \cdots - r_n) = C(n + r - r_1 - \cdots - r_n - 1, r - r_1 - \cdots - r_n).$$

□

E o número das respectivas permutações? Aqui aparecem os números

$$\frac{n!}{n_1!\, n_2!\, \ldots\, n_k!},$$

onde $n_1, n_2, \ldots, n_k \in \mathbb{N}_0$ são tais que $n_1 + n_2 + \cdots + n_k = n$. Estes números designam-se por *números (ou coeficientes) multinomiais*, e denotam-se habitualmente por

$$C(n; n_1, n_2, \ldots, n_k) \quad \text{ou} \quad \binom{n}{n_1, n_2, \ldots, n_k}.$$

Estes números generalizam os coeficientes binomiais:

$$\binom{n}{n_1, n_2} = \frac{n!}{n_1!\, n_2!} = \frac{n!}{n_1!\, (n-n_1)!} = \binom{n}{n_1} = \binom{n}{n_2}.$$

Teorema 3. *Seja $S = \{a_1, a_2, \ldots, a_n\}$. O número de permutações com repetição de elementos de S, r a r, contendo cada elemento a_i pelo menos r_i vezes ($r \geq r_1 + r_2 + \cdots + r_n$), é igual a*

$$\sum_{s_1 + s_2 + \cdots + s_n = r;\, s_i \geq r_i} \binom{r}{s_1, s_2, \ldots, s_n}.$$

(O somatório é tomado sobre todos os $s_i \geq r_i$ tais que $s_1 + s_2 + \cdots + s_n = r$.)

Prova. Basta observar que o número dessas permutações é, pelo Princípio da Multiplicação, igual a

$$\sum_{s_1+s_2+\cdots+s_n=r;s_i\geq r_i} C(r,s_1)\times C(r-s_1,s_2)\times C(r-s_1-s_2,s_3)\times\cdots\times C(\underbrace{r-s_1-s_2-\cdots-s_{n-1}}_{s_n},s_n).$$

Mas este número é igual a

$$\frac{r!}{s_1!\,(r-s_1)!}\times\frac{(r-s_1)!}{s_2!\,(r-s_1-s_2)!}\times\frac{(r-s_1-s_2)!}{s_3!\,(r-s_1-s_2-s_3)!}\times\cdots\times\frac{(r-s_1-s_2-\cdots-s_{n-1})!}{s_n!\,\underbrace{(r-s_1-s_2-\cdots-s_n)!}_{=0}}=$$

$$=\frac{r!}{s_1!s_2!\ldots s_n!}=\binom{r}{s_1,s_2,\ldots,s_n}.$$

□

É claro que podemos também impôr condições ao número máximo de vezes que cada elemento pode ser repetido nas combinações e permutações. Aqui as fórmulas são um pouco mais complicadas e não as apresentaremos.

Observação. Tal como os números binomiais aparecem no desenvolvimento do binómio, os números multinomiais

$$\binom{n}{n_1,n_2,\ldots,n_k}$$

são precisamente os coeficientes que aparecem no desenvolvimento de $(x_1+x_2+\cdots+x_k)^n$. Por exemplo, desenvolvendo $(x+y+z)^n$ para $n=1,2,3,4$ obtemos:

$x+y+z$
$x^2+2xy+2xz+y^2+2yz+z^2$
$x^3+3x^2y+3x^2z+3xy^2+6xyz+3xz^2+y^3+3y^2z+3yz^2+z^3$
$x^4+4x^3y+4x^3z+6x^2y^2+12x^2yz+6x^2z^2+4xy^3+12xy^2z+12xyz^2+4xz^3+y^4+4y^3z+$
$6y^2z^2+4yz^3+z^4$

Princípio da Inclusão-Exclusão

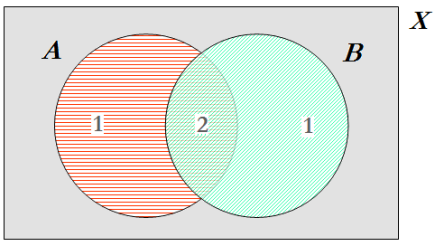
O Princípio da Inclusão-Exclusão que vamos agora apresentar é também conhecido por *fórmula do crivo* ou *fórmula de da Silva-Sylvester*[33] e generaliza o Princípio da Adição.

Se A e B forem dois subconjuntos de X qual será o número de elementos da união $A\cup B$? Se somarmos simplesmente o número de elementos de A com o número de elementos de B estaremos

[33] O Princípio da Inclusão-Exclusão foi publicado pela primeira vez em 1854, num artigo de Daniel da Silva, e mais tarde, em 1883, por Sylvester. Por isso, a fórmula do crivo e suas similares são, por vezes, apelidadas de fórmulas de da Silva ou de Sylvester. Realçamos o facto de Daniel da Silva, na opinião de Gomes Teixeira o mais notável matemático português do séc. XIX, ter sido estudante da Universidade de Coimbra; transcrevemos de [J. Silva Oliveira, *Daniel Augusto da Silva*, Boletim da SPM 2 (1979) 3-15]: "Daniel da Silva (1814-1878) foi, além de matemático eminente do seu tempo, oficial da Armada e professor da Escola Naval. Como estudante frequentou primeiro a Academia Real de Marinha e prosseguiu depois os seus estudos na Universidade de Coimbra onde, com altas classificações, se licenciou em Matemática e acabou por se doutorar".

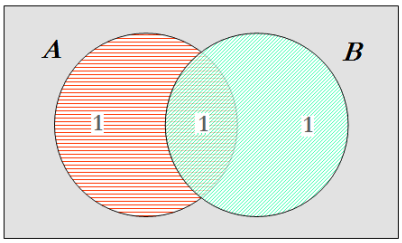
a contar, uma vez cada um, os elementos de $A - B$ e os de $B - A$, mas estaremos a contar por duas vezes os elementos da intersecção $A \cap B$ (é o que o número 2 indica na região $A \cap B$ na figura):

$|A| + |B|$:



Teremos então que descontar esses elementos:

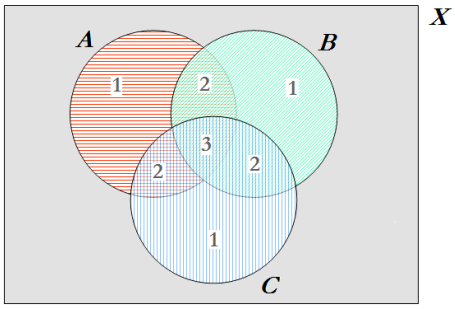
$|A| + |B| - |A \cap B|$:



Em conclusão, $|A \cup B| = |A| + |B| - |A \cap B|$ e, consequentemente, o complementar $X - (A \cup B)$ tem cardinal igual a $|X| - (|A| + |B|) + |A \cap B|$.

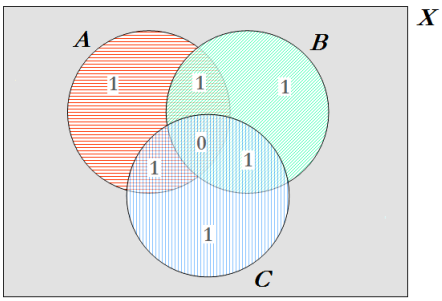
E se forem 3 subconjuntos A, B e C em vez de 2? Neste caso poderemos começar por somar os elementos em A, B e C.

$|A| + |B| + |C|$:



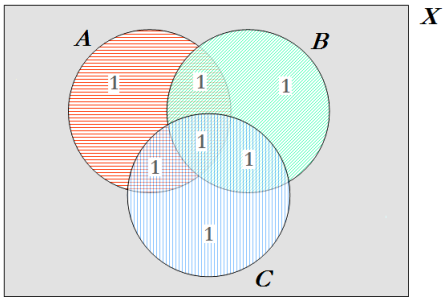
Deste modo estaremos a contar, uma vez cada um, os elementos de $A - (B \cup C)$, os de $B - (A \cup C)$ e os de $C - (A \cup B)$, mas estaremos a contar por duas vezes os elementos de $(A \cap B) - C$, $(A \cap C) - B$ e $(B \cap C) - A$, e, pior ainda, estaremos a contar por três vezes os elementos da intersecção $A \cap B \cap C$. Podemos começar por descontar os primeiros adicionando $|A \cap B|$, $|A \cap C|$ e $|B \cap C|$.

$|A| + |B| + |C| - (|A \cap B| + |A \cap C| + |B \cap C|)$:



Mas agora acabámos por descontar os elementos da intersecção $A \cap B \cap C$ mais do que devíamos (o zero na figura acima indica que os elementos dessa região ainda não foram considerados para a contagem dos elementos de $A \cup B \cup C$), tendo que os contar novamente, para que a contagem fique certa.

$|A| + |B| + |C| - (|A \cap B| + |A \cap C| + |B \cap C|) + |A \cap B \cap C|$:



Agora, finalmente, todos os elementos de todas as regiões de $A \cup B \cup C$ foram contados precisamente uma vez.

Estamos em condições de analisar o caso geral de n subconjuntos. No que se segue, assumiremos que X é um conjunto finito e $P_1, P_2, \ldots, P_n$ são n propriedades que cada elemento de X poderá ou não possuir. Denotaremos por A_i, $i \in \{1, 2, \ldots, n\}$, o conjunto dos elementos de X que possuem a propriedade P_i, e por $\overline{A_i}$ o respectivo complementar $X - A_i$.

Princípio da Inclusão-Exclusão. *O número $|A_1 \cup A_2 \cup \cdots \cup A_n|$ de elementos de X que possuem, pelo menos, uma das propriedades $P_1, P_2, \ldots, P_n$ é igual a*

$$\sum_{i=1}^{n} |A_i| - \sum_{\substack{i,j=1 \\ i<j}}^{n} |A_i \cap A_j| + \sum_{\substack{i,j,k=1 \\ i<j<k}}^{n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|,$$

onde o primeiro somatório percorre todos os inteiros $1, 2, \ldots, n$, o segundo somatório percorre todas as combinações $\{i, j\}$ dos inteiros $1, 2, \ldots, n$, dois a dois, o terceiro somatório percorre todas as combinações $\{i, j, k\}$ dos inteiros $1, 2, \ldots, n$, três a três, e assim sucessivamente.

Prova. É evidente que o conjunto dos elementos de X que possuem alguma das propriedades $P_1, P_2, \ldots, P_n$ é a união $A_1 \cup A_2 \cup \cdots \cup A_n$. Podemos verificar a validade da identidade a provar

mostrando que um objecto com alguma das propriedades $P_1, P_2, \ldots, P_n$ contribui com uma unidade para a soma do enunciado do princípio e que um objecto que não verifique nenhuma dessas propriedades contribui com um zero para essa mesma soma.

Designemos esta soma por M. Cada elemento de X que não possui nenhuma das propriedades $P_1, P_2, \ldots, P_n$ contribui com

$$0 - 0 + 0 - \cdots + (-1)^{n+1} \times 0 = 0$$

unidades para o valor M, pois não pertence a nenhum A_i ($i \in \{1, 2, \ldots, n\}$).

Por outro lado, cada elemento de X que possui m ($1 \leq m \leq n$) das n propriedades contribui com $C(m, 1) = m$ unidades para $\sum_{i=1}^{n} |A_i|$ (pois pertence a m dos conjuntos $A_1, A_2, \ldots, A_n$), com $C(m, 2)$ unidades para $\sum_{i,j=1;i<j}^{n} |A_i \cap A_j|$ (pois existem $C(m, 2)$ maneiras diferentes de escolher um par de propriedades distintas que ele satisfaça) e assim sucessivamente. Então a sua contribuição para M é igual a

$$C(m, 1) - C(m, 2) + C(m, 3) - \cdots - (-1)^m C(m, m)$$

que, por sua vez, é igual a $C(m, 0) = 1$, pois por uma fórmula deduzida na secção anterior [34],

$$C(m, 0) - C(m, 1) + C(m, 2) - C(m, 3) + \cdots + (-1)^{m+1} C(m, m) = 0. \qquad \square$$

Por vezes, a seguinte formulação alternativa do Princípio da Inclusão-Exclusão é mais útil:

Corolário. *O número $|\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n}|$ de elementos de X que não possuem qualquer das propriedades $P_1, P_2, \ldots, P_n$ é dado por*

$$|X| - \sum_{i=1}^{n} |A_i| + \sum_{\substack{i,j=1 \\ i<j}}^{n} |A_i \cap A_j| - \sum_{\substack{i,j,k=1 \\ i<j<k}}^{n} |A_i \cap A_j \cap A_k| + \cdots + (-1)^n |A_1 \cap A_2 \cap \cdots \cap A_n|.$$

Prova. É claro que o número de elementos de elementos de X que não verificam nenhuma das propriedades $P_1, P_2, \ldots, P_n$ é o cardinal de $\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n} = X - (A_1 \cup A_2 \cup \cdots \cup A_n)$. Pelo Princípio da Inclusão-Exclusão esse número é igual a

$$|X| - \Big(\sum_{i=1}^{n} |A_i| - \sum_{\substack{i,j=1 \\ i<j}}^{n} |A_i \cap A_j| + \sum_{\substack{i,j,k=1 \\ i<j<k}}^{n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n| \Big)$$

$$= |X| - \sum_{i=1}^{n} |A_i| + \sum_{\substack{i,j=1 \\ i<j}}^{n} |A_i \cap A_j| - \sum_{\substack{i,j,k=1 \\ i<j<k}}^{n} |A_i \cap A_j \cap A_k| + \cdots + (-1)^n |A_1 \cap A_2 \cap \cdots \cap A_n|.$$

$\square$

Vejamos alguns exemplos de aplicação do Princípio da Inclusão-Exclusão. Comecemos por recordar o Problema (B3) do texto introdutório do curso"O que é a Matemática Discreta?":

[34] Terceira identidade binomial na página 93.

À saída de um restaurante, de quantas maneiras podem ser devolvidos os chapéus de n pessoas de modo a que nenhuma pessoa receba o seu chapéu?[35]

Este problema é um caso particular do seguinte problema geral, designado por **problema dos desencontros**:

Estando os elementos de um conjunto finito S dispostos segundo uma certa ordem, quantas permutações de S existem nas quais nenhum elemento esteja na sua posição primitiva?

Uma permutação $a_{j_1}a_{j_2}\ldots a_{j_n}$ de $S = \{a_1, a_2, \ldots, a_n\}$ diz-se um *desencontro* de S caso $j_k \neq k$ para qualquer $k \in \{1, 2, \ldots, n\}$. Denotemos por D_n o número de desencontros de S.

Solução do problema. *Para qualquer* $n \in \mathbb{N}$, $D_n = n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right)$.

Prova. Seja X o conjunto de todas as permutações de S. Claro que $|X| = n!$. Seja ainda A_i $(i = 1, 2, \ldots, n)$ o conjunto das permutações $a_{j_1}a_{j_2}\ldots a_{j_n}$ tais que $a_{j_i} = a_i$ (portanto aquelas em que a_i está na posição primitiva). Claro que $|A_i| = (n-1)!$. As permutações em $A_i \cap A_j$ têm a_i e a_j fixos, nas posições i e j respectivamente, e os restantes $n-2$ elementos permutados nas restantes $n-2$ posições, pelo que $|A_i \cap A_j| = (n-2)!$ para $i, j \in \{1, 2, \ldots, n\}$, $i < j$. Analogamente, podemos concluir que $|A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}| = (n-k)!$ para $k \in \{1, 2, \ldots, n\}$, $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}$, $i_1 < i_2 < \cdots < i_k$. Como $D_n = |\overline{A}_1 \cap \overline{A}_2 \cap \cdots \cap \overline{A}_n|$, decorre pelo Princípio da Inclusão-Exclusão que

$$\begin{aligned} D_n &= n! - \frac{n!}{1!} + \frac{n!}{2!} - \frac{n!}{3!} + \cdots + (-1)^n \frac{n!}{n!} \\ &= n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right). \end{aligned}$$

[35] Também costuma aparecer enunciado do seguinte modo, na forma de um jogo de cartas: "No chamado 'jogo dos pares', as 52 cartas de um baralho são dispostas em linha, com o seu valor à vista. As cartas de um segundo baralho são dispostas também em linha por cima das outras. A pontuação é determinada contando o número de vezes em que a carta do segundo baralho coincide com a do primeiro sobre a qual foi colocada. Qual é a probabilidade de se obterem zero pontos?"

□

Cálculo de $D_2, D_3, \ldots, D_{15}$:

```
> Des := proc(n::integer)
>    local k;
>    RETURN(sum((-1)^k  * (n!/k!), k=0..n));
```

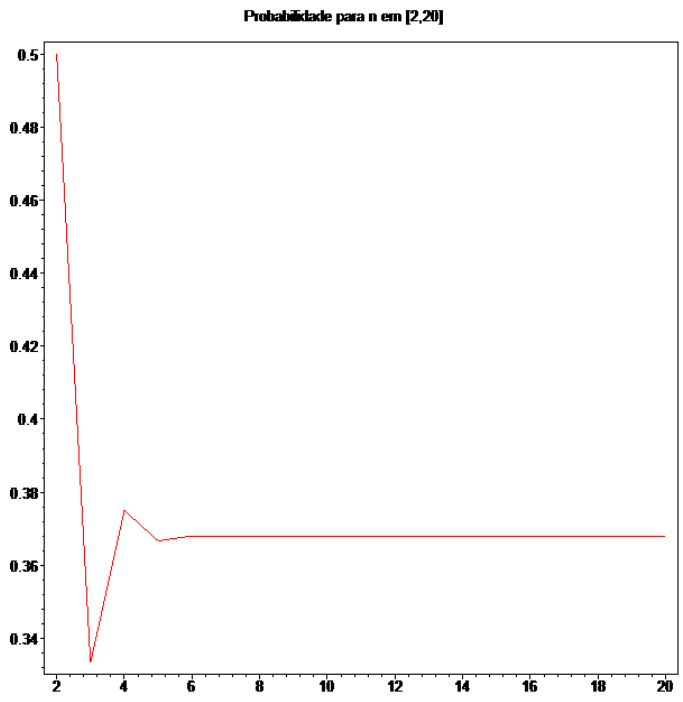
Sequência de valores Des(n) para $n = 2, 3, \ldots, 15$:
1, 2, 9, 44, 265, 1854, 14833, 133496, 1334961, 14684570, 176214841, 2290792932, 32071101049, 481066515734

Na sua forma original o problema (B3) foi formulado em termos de probabilidades, questionando a probabilidade de nenhuma pessoa receber de volta o respectivo chapéu. Evidentemente, a resposta é a probabilidade de uma permutação de n objectos, escolhida aleatoriamente, ser um desencontro, ou seja,

$$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}.$$

Cálculo desta probabilidade para alguns valores particulares de n:[36]

```
n,  probabilidade
2,  0.50000000000000000000
3,  0.33333333333333333333
4,  0.37500000000000000000
5,  0.36666666666666666667
6,  0.36805555555555555556
7,  0.36785714285714285714
8,  0.36788194444444444444
9,  0.36787918871252204586
10, 0.36787946428571428571
11, 0.36787943923360590027
12, 0.36787944132128159906
13, 0.36787944116069116069
14, 0.36787944117216190629
15, 0.36787944117139718992
16, 0.36787944117144498469
17, 0.36787944117144217323
18, 0.36787944117144232942
19, 0.36787944117144232120
20, 0.36787944117144232161
```



[36] Usando factos da Análise Matemática é possível provar que

$$\lim_{n\to+\infty} \frac{D_n}{n!} = \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!} + \cdots = e^{-1} \sim 0.3679.$$

Vejamos agora como o Princípio da Inclusão-Exclusão também serve para resolver o Problema (B2) da Introdução.

Seja $A = \{a_1, a_2, \ldots, a_t\}$ e denotemos o conjunto dos primeiros n números naturais por $[n]$. Designando o conjunto $\{x \in [n] \mid x \text{ é divisível por } a_i\}$ por A_i, o número pedido dos inteiros positivos inferiores ou iguais a n, não divisíveis por nenhum dos elementos de A é o cardinal de $\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_t}$.

Claramente $|A_i|$ é a parte inteira do número $\frac{n}{a_i}$, ou seja, $\left\lfloor \frac{n}{a_i} \right\rfloor$. Como

$$A_i \cap A_j = \{x \in [n] \mid x \text{ é divisível por } a_i \text{ e } a_j\} = \{x \in [n] \mid x \text{ é divisível por } \operatorname{mmc}(a_i, a_j)\}$$

então $|A_i \cap A_j| = \left\lfloor \frac{n}{\operatorname{mmc}(a_i,a_j)} \right\rfloor$. Mais geralmente, $|A_{i_1} \cap \cdots \cap A_{i_l}| = \left\lfloor \frac{n}{\operatorname{mmc}(a_{i_1},\ldots,a_{i_l})} \right\rfloor$, pelo que $|\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_t}|$ é igual a

$$n - \sum_{i=1}^{t} \left\lfloor \frac{n}{a_i} \right\rfloor + \sum_{\substack{i,j=1 \\ i \leq j}}^{t} \left\lfloor \frac{n}{\operatorname{mmc}(a_i, a_j)} \right\rfloor - \cdots + (-1)^t \left\lfloor \frac{n}{\operatorname{mmc}(a_1, a_2, \ldots, a_t)} \right\rfloor.$$

No caso particular em que os elementos de A são todos primos entre si, o número de inteiros positivos inferiores ou iguais a n que não são divisíveis por nenhum dos elementos de A é igual a

$$n - \sum_{i=1}^{t} \left\lfloor \frac{n}{a_i} \right\rfloor + \sum_{\substack{i,j=1 \\ i \leq j}}^{t} \left\lfloor \frac{n}{a_i a_j} \right\rfloor - \sum_{\substack{i,j,k=1 \\ i \leq j \leq k}}^{t} \left\lfloor \frac{n}{a_i a_j a_k} \right\rfloor + \cdots + (-1)^t \left\lfloor \frac{n}{a_1 a_2 \ldots a_t} \right\rfloor.$$

Por exemplo, para $a_1 = 2, a_2 = 3, a_3 = 5, a_4 = 7$ e $n = 1000$, este número é igual a 228.

Contemos agora o número $\phi(n)$ de inteiros positivos, inferiores a n, primos com n. Seja $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t}$ a factorização de n em números primos. Como os conjuntos

$$\left\{k \in \mathbb{N} \mid 1 \leq k \leq n \text{ e } \operatorname{mdc}(k, n) = 1\right\}$$

e

$$\left\{k \in \mathbb{N} \mid 1 \leq k \leq n \text{ e } p_i \not| k \text{ para } i = 1, 2, \ldots, t\right\}$$

coincidem bastará aplicar a fórmula, acima deduzida, ao conjunto $A = \{p_1, p_2, \ldots, p_t\}$. Imediatamente se conclui que o número $\phi(n)$ é igual a

$$n - \sum_{i=1}^{t} \left\lfloor \frac{n}{p_i} \right\rfloor + \sum_{\substack{i,j=1 \\ i \leq j}}^{t} \left\lfloor \frac{n}{p_i p_j} \right\rfloor - \sum_{\substack{i,j,k=1 \\ i \leq j \leq k}}^{t} \left\lfloor \frac{n}{p_i p_j p_k} \right\rfloor + \cdots + (-1)^t \left\lfloor \frac{n}{p_1 p_2 \ldots p_t} \right\rfloor.$$

Como vimos em 4.1 (pg. 97), a função

$$\begin{array}{rccl} \phi: & \mathbb{N} & \to & \mathbb{N} \\ & n & \mapsto & \phi(n) = |\{k \in \mathbb{N} \mid 1 \leq k \leq n \text{ e } \operatorname{mdc}(k, n) = 1\}| \end{array}$$

é a chamada *função de Euler*, muito importante em Teoria dos Números.

Para terminar, vejamos um processo simples de contar os números primos entre 2 e $n \geq 2$. O crivo de Eratóstenes

é um processo que permite enumerar todos os primos entre 1 e qualquer inteiro positivo k:

- Calcula-se $c = \left\lfloor \sqrt{k} \right\rfloor$;
- Apagam-se, na sucessão $2, 3, 4, \ldots, k$, todos os múltiplos de $2, 3, 4, \ldots, c$ (com excepção dos próprios números $2, 3, 4, \ldots, c$);

Passo 2: Eliminando os múltiplos de **2**

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Primos: 2

- Os números que restam são os primos entre 1 e k.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Primos: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

Então, para determinar o número de primos entre 1 e k, bastará:

- determinar os primos $p_1, p_2, \ldots, p_t$ entre 1 e $\left\lfloor \sqrt{n} \right\rfloor$, usando o crivo de Eratóstenes;

- em seguida, determinar, com a ajuda da fórmula acima deduzida, quantos inteiros positivos inferiores ou iguais a n não são divisíveis por nenhum dos elementos de $A = \{p_1, p_2, \ldots, p_t\}$. Como os primos entre $\left\lfloor \sqrt{n} + 1 \right\rfloor$ e n são exactamente os inteiros positivos inferiores ou iguais a n (com excepção do 1) que não são divisíveis por nenhum dos elementos de A, o seu número é igual a

$$M(n) = n - 1 - \sum_{i=1}^{t} \left\lfloor \frac{n}{p_i} \right\rfloor + \sum_{\substack{i,j=1 \\ i \leq j}}^{t} \left\lfloor \frac{n}{p_i p_j} \right\rfloor - \sum_{\substack{i,j,k=1 \\ i \leq j \leq k}}^{t} \left\lfloor \frac{n}{p_i p_j p_k} \right\rfloor + \cdots + (-1)^t \left\lfloor \frac{n}{p_1 p_2 \ldots p_t} \right\rfloor.$$

  Concluindo, o número de primos entre 1 e n será igual a $t + M(n)$.

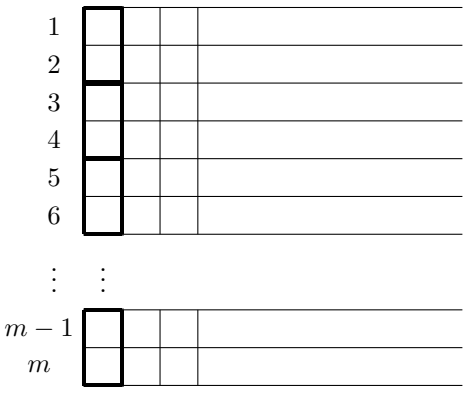
Relações de recorrência

Recordemos o Problema (A6) do capítulo "Que é a Matemática Discreta?":

> *Consideremos um tabuleiro de xadrez e algumas peças (idênticas) de dominó tais que cada uma cobre precisamente 2 quadrados adjacentes do tabuleiro. Será possível dispor 32 dessas peças no tabuleiro de modo a cobri-lo, sem sobreposição de peças?* (Tal arranjo diz-se uma *cobertura perfeita* do tabuleiro por dominós.)

Não é difícil concluir que, em geral, um tabuleiro $m \times n$ possui uma cobertura perfeita se e só se pelo menos um dos números m ou n é par:

Se o tabuleiro possui uma cobertura perfeita então o dobro do número de peças na configuração deverá ser igual a mn. Portanto $2|mn$ pelo que $2|m$ ou $2|n$. Reciprocamente suponhamos, sem perda de generalidade, que m é par. Nesse caso é evidente que cada coluna pode ser perfeitamente coberta (basta alinhar sucessivamente $m/2$ peças)

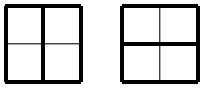


pelo que qualquer número n de colunas pode também ser coberto de modo perfeito.

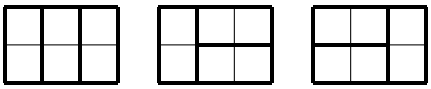
Mais difícil é contar o número de coberturas perfeitas. Façamo-lo no caso mais simples de um tabuleiro $2 \times n$. Para cada $n \in \mathbb{N}$, seja $f(n)$ o número de coberturas perfeitas de um tabuleiro $2 \times n$. Comecemos por calcular $f(1), f(2), f(3), f(4)$ e $f(5)$:

$f(1) = 1$:

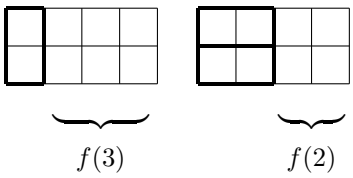
$f(2) = 2$:



$f(3) = 3$:



$f(4) = 5 = f(3) + f(2)$:

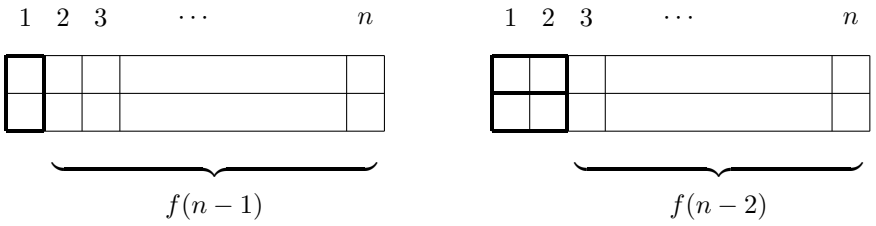


Analogamente, $f(5) = 8 = f(4) + f(3)$. Isto leva-nos a conjecturar que

$$f(n) = f(n-1) + f(n-2) \qquad (n \geq 3).$$

Esta conjectura pode ser provada, por exemplo, do seguinte modo:

Para construir uma cobertura perfeita de um tabuleiro $2 \times n$ podemos colocar uma peça na vertical, a ocupar a coluna 1, e teremos depois $f(n-1)$ maneiras diferentes de cobrir o resto do tabuleiro, ou podemos colocar duas peças na horizontal, a ocupar as colunas 1 e 2, e cobrir depois o resto do tabuleiro, o que pode ser feito de $f(n-2)$ modos distintos:



Esta relação, conjuntamente com os valores iniciais $f(1) = 1$ e $f(2) = 2$, determina univocamente a sequência dos números de coberturas perfeitas $f(1), f(2), f(3), \ldots$.

Por exemplo, $f(12)$ é igual a

$$f(11) + f(10) = 2f(10) + f(9) = 3f(9) + f(8) = \cdots = 21f(5) + 13f(4) = 233.$$

É claro que para valores muito grandes de n este método de cálculo de $f(n)$ não será praticável sem a ajuda de um computador, porque não temos aqui uma fórmula fechada para o valor de

$f(n)$ mas sim uma *relação de recorrência* que estabelece o valor de f em n a partir de valores de f em inteiros anteriores a n.

Como podemos resolver relações de recorrência destas, isto é, como podemos obter, a partir da relação de recorrência, a respectiva fórmula fechada? É o que veremos agora.

Consideremos uma *sucessão* (infinita) de elementos de um conjunto S,

$$\begin{array}{rccc} u: & \mathbb{N}_0 & \to & S \\ & n & \mapsto & u(n). \end{array}$$

O valor $u(n)$ costuma representar-se simplesmente por u_n e é frequente apresentar uma sucessão dispondo sucessivamente as imagens da aplicação u:

$$u_0, u_1, u_2, \ldots.$$

Muitas vezes uma sucessão é dada mediante a indicação do que se chama o seu *termo geral*, ou *termo de ordem* n (por exemplo, $u_n = n^2$, $u_n = \sin 2^n/(n+1)^2$, etc.). É uma situação cómoda pois, além de nesse caso ser possível calcular sem grandes problemas qualquer termo da sucessão, o estudo de várias propriedades (como a monotonia, convergência, etc.) fica muito facilitado. Usaremos a notação (u_n) para nos referirmos à sucessão $u_0, u_1, u_2, \ldots$.

Como vimos nos exemplos acima, nem sempre uma sucessão é definida por indicação do seu termo geral, mas sim por uma *relação de recorrência*: são dados uns tantos termos iniciais da sucessão, $u_0, u_1, \ldots, u_{k-1}$, e cada um dos seguintes determina-se a partir dos k anteriores por intermédio de uma relação que permanece invariável, $u_k = f(u_0, u_1, \ldots, u_{k-1})$, $u_{k+1} = f(u_1, u_2, \ldots, u_k)$, etc. Estas são as chamadas relações de recorrência para a sucessão (u_n). Ao número k chama-se *ordem* da relação de recorrência.

Uma sucessão diz-se uma *solução* de uma relação de recorrência se os seus termos satisfizerem a relação. De entre todas as relações de recorrência destacam-se, não só pela sua simplicidade mas também pela frequência com que ocorrem, as chamadas *relações de recorrência lineares homogéneas com coeficientes constantes*. São as do tipo

$$\boxed{u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k} \quad (n \geq k)}$$

com $a_1, a_2, \ldots, a_k$ constantes.

O adjectivo "linear"refere-se ao facto de todos os valores de u ocorrerem como potências de expoente 1, enquanto que o adjectivo "homogéneo"refere-se ao facto de não existir termo independente (constante).

Por exemplo, $u_n = u_{n-1}^2 + 2u_{n-2}$ $(n = 2, 3, \ldots)$ não é linear, enquanto que $u_n = 3u_{n-1} + 2$ $(n = 1, 2, \ldots)$ não é homogénea. Por outro lado, a relação $u_n = (n+2)u_{n-1} + 2u_{n-2}$ $(n = 2, 3, \ldots)$ é linear e homogénea mas não tem coeficientes constantes (o primeiro coeficiente $n+2$ varia com n).

Exemplos. (1) As progressões geométricas de razão r satisfazem uma relação de recorrência homogénea linear de primeira ordem:

$$u_n = r u_{n-1} \quad (n \geq 1).$$

(2) As progressões aritméticas de razão r podem ser vistas como sucessões satisfazendo relações de recorrência homogéneas lineares de segunda ordem: de $u_{n-1} = u_{n-2} + r$ e $u_n = u_{n-1} + r$ obtém-se, subtraindo a primeira identidade da segunda,

$$u_n = 2u_{n-1} - u_{n-2}.$$

(3) A sucessão do número de coberturas perfeitas satisfaz uma relação de recorrência homogénea linear de segunda ordem.

Como em muitos problemas combinatoriais a solução aparece formulada em termos de uma relação de recorrência, torna-se imperativo saber manipulá-las e conhecer métodos que permitam obter uma fórmula explícita para o termo geral da respectiva sucessão.

Convirá desde já avisar que não existem métodos gerais que nos permitam resolver todas as relações de recorrência. Uma estratégia possível ("ingénua") será calcular um número razoável de termos e tentar intuir a lei de formação do termo geral, que pode depois ser confirmada pelo método de indução matemática. Com esta estratégia, algumas tentativas, mesmo em casos simples, mostrarão que não se trata de tarefa fácil.

Por exemplo, no caso das relações de recorrência lineares de primeira ordem, temos $u_1 = au_0$, $u_2 = au_1 = a^2u_0$, etc., sendo fácil ver que, para qualquer $n \geq 1$, $u_n = a^nu_0$. Está assim encontrado o termo geral neste caso. No entanto, para as de segunda ordem, dados u_1 e u_2 e duas constantes a e b, temos:

$$\begin{aligned} u_2 &= au_1 + bu_0 \\ u_3 &= au_2 + bu_1 = (a^2 + b)u_1 + abu_0 \\ u_4 &= au_3 + bu_2 = (a^3 + 2ab)u_1 + (a^2b + b^2)u_0 \\ &\vdots \end{aligned}$$

Não é fácil descortinar aqui uma lei de formação que permita conjecturar o que deverá ser u_n em função de n, a, b, u_0 e u_1. É claro que para as sucessões recorrentes lineares de ordem superior a situação será ainda pior.

Curiosamente, como veremos, o caso das relações de recorrência lineares homogéneas com coeficientes constantes é tratável de uma forma sistemática, embora as técnicas existentes se possam revelar muito trabalhosas na prática. Apesar de ser um método indirecto e pouco natural, é elegante e engenhoso.

Restringemo-nos então à classe das relações de recorrência lineares homogéneas com coeficientes constantes, isto é, das relações de recorrência da forma

$$\boxed{u_n = a_1u_{n-1} + a_2u_{n-2} + \cdots + a_ku_{n-k} \quad (n = k, k+1, \ldots)} \qquad (*)$$

onde $a_1, a_2, \ldots, a_k$ são constantes. Podemos sempre supor que $a_k \neq 0$ pois, caso contrário, a relação reduz-se a uma de ordem inferior.

Associemos à relação de recorrência $(*)$, a equação

$$x^k - a_1x^{k-1} - a_2x^{k-2} - \cdots - a_k = 0,$$

chamada *equação característica* de $(*)$. Esta equação tem k raízes $\alpha_1, \alpha_2, \ldots, \alpha_k$, chamadas *raízes características* de $(*)$. Claro que poderão ser números complexos, não todos distintos. Como $a_k \neq 0$, são todas não nulas.

Teorema 1. *Seja α um número complexo não nulo. A sucessão*

$$1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^n, \ldots$$

é solução da relação de recorrência $(*)$ se e só se α é uma raiz característica.

Prova. A sucessão (u_n), onde $u_n = \alpha^n$, é uma solução de $(*)$ se e só se, para $n \geq k$,

$$\alpha^n = a_1\alpha^{n-1} + a_2\alpha^{n-2} + \cdots + a_k\alpha^{n-k}$$

ou, equivalentemente,

$$\alpha^{n-k}(\alpha^k - a_1\alpha^{k-1} - a_2\alpha^{k-2} - \cdots - a_k) = 0.$$

Como $\alpha \neq 0$, esta equação é ainda equivalente a

$$\alpha^k - a_1\alpha^{k-1} - a_2\alpha^{k-2} - \cdots - a_k = 0.$$

Portanto (α^n) é uma solução de $(*)$ se e só se α é uma raiz característica. □

Corolário. *Sejam $\alpha_1, \alpha_2, \ldots, \alpha_k$ as raízes características de $(*)$. Para quaisquer constantes $c_1, c_2, \ldots, c_k$ a sucessão de termo geral*

$$u_n = c_1\alpha_1^n + c_2\alpha_2^n + \cdots + c_k\alpha_k^n$$

é uma solução de $(*)$.
Prova. É um exercício simples verificar que sempre que $(u_n^1), (u_n^2), \ldots, (u_n^t)$ são soluções de $(*)$ e $c_1, c_2, \ldots, c_t$ são constantes então a sucessão de termo geral

$$u_n = c_1u_n^1 + c_2u_n^2 + \cdots + c_tu_n^t$$

ainda é solução de $(*)$. Combinando este facto com o Teorema 1 obtemos imediatamente o Corolário. □

No caso das raízes características serem todas distintas podemos obter todas as soluções de $(*)$:

Teorema 2. *Suponhamos que as raízes características $\alpha_1, \alpha_2, \ldots, \alpha_k$ da relação de recorrência $(*)$ são distintas duas a duas. Neste caso, se uma sucessão de termo geral u_n é solução de $(*)$, existem constantes $c_1, c_2, \ldots, c_k$ tais que*

$$u_n = c_1\alpha_1^n + c_2\alpha_2^n + \cdots + c_k\alpha_k^n.$$

Prova. Seja (u_n) uma solução da relação de recorrência $(*)$. Uma vez que $(*)$, conjuntamente com os k valores iniciais $u_0, u_1, \ldots, u_{k-1}$, determinam completamente a sucessão (u_n), bastará provar que existem constantes $c_1, c_2, \ldots, c_k$ tais que a sucessão de termo geral $c_1\alpha_1^n + c_2\alpha_2^n \cdots + c_k\alpha_k^n$ satisfaz $(*)$ e tem como primeiros k elementos os valores $u_0, u_1, \ldots, u_{k-1}$. Pelo Corolário, bastará provar que existem constantes $c_1, c_2, \ldots, c_k$ tais que

$$\begin{cases} c_1 + c_2 + \cdots + c_k = u_0 \\ c_1\alpha_1 + c_2\alpha_2 + \cdots + c_k\alpha_k = u_1 \\ \quad \vdots \\ c_1\alpha_1^{k-1} + c_2\alpha_2^{k-1} + \cdots + c_k\alpha_k^{k-1} = u_{k-1}. \end{cases}$$

Trata-se de um sistema de k equações lineares com k incógnitas. A matriz

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_k \\ \vdots & \vdots & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \ldots & \alpha_k^{k-1} \end{bmatrix}$$

deste sistema é uma matriz muito especial, chamada *matriz de Vandermonde*. O seu determinante é dado por

$$\prod_{\substack{i,j=1 \\ i<j}}^{k} (\alpha_j - \alpha_i)$$

(a prova deste facto encontra-se em muitos livros de Álgebra Linear). Como as raízes $\alpha_1, \alpha_2, \ldots, \alpha_k$ são todas distintas, este determinante é diferente de zero. Isto quer dizer que o sistema possui exactamente uma solução. □

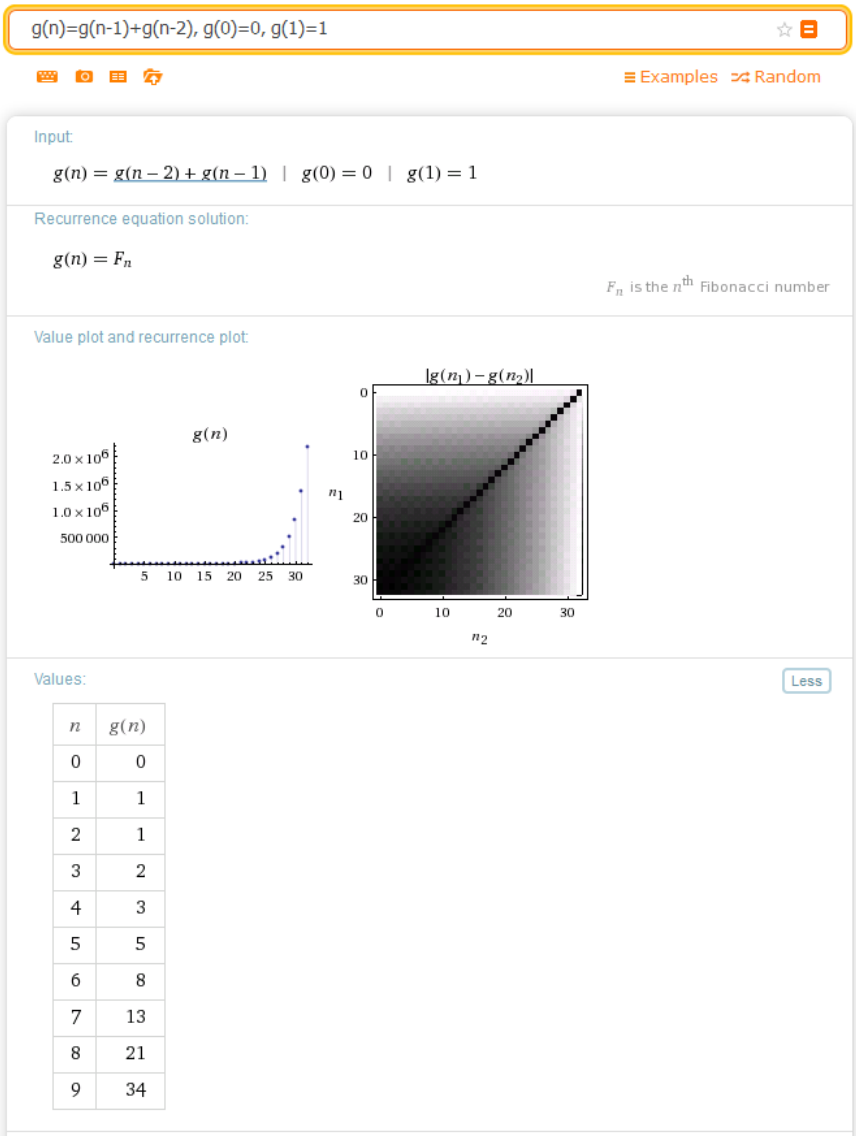
Exemplos. A sucessão de Fibonacci $F(1), F(2), F(3), \ldots$ do problema (B4) da Introdução também é definida pela relação $F(n) = F(n-1) + F(n-2)$ $(n = 3, 4, 5, \ldots)$, mas desta vez sujeita às condições iniciais $F(0) = 0$ e $F(1) = 1$.

Procedimento que calcula, por recursão, os termos da sucessão de Fibonacci:

```
> Fibonacci := proc(n)
>    if n=1 or n=2 then RETURN( 1 ) else
>    Fibonacci(n-1) + Fibonacci(n-2);
```

Valores de Fibonacci(n) para $n = 1, 2, \ldots, 20$:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765



n	$g(n)$
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34

As raízes da equação característica $x^2 - x - 1 = 0$ desta relação de recorrência são o *número de ouro* $\frac{1+\sqrt{5}}{2}$ e o seu conjugado $\frac{1-\sqrt{5}}{2}$. Então, pelo Teorema 2, os números de Fibonacci são dados por

$$F(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n,$$

para algum par de constantes c_1 e c_2. As condições iniciais $F(0) = 0$ e $F(1) = 1$ permitem-nos determinar tais constantes. Com efeito,

$$\begin{cases} c_1 + c_2 = F(0) = 0 \\ c_1\left(\frac{1+\sqrt{5}}{2}\right) + c_2\left(\frac{1-\sqrt{5}}{2}\right) = F(1) = 1 \end{cases}$$

cuja solução é $c_1 = \frac{\sqrt{5}}{5}$ e $c_2 = -\frac{\sqrt{5}}{5}$.

Concluindo, os números de Fibonacci satisfazem a fórmula

$$\boxed{F(n) = \tfrac{\sqrt{5}}{5}\Big[\Big(\tfrac{1+\sqrt{5}}{2}\Big)^n - \Big(\tfrac{1-\sqrt{5}}{2}\Big)^n\Big].}$$

Fica assim resolvido o Problema (B4) da Introdução: o número de pares de coelhos existentes na ilha ao fim de n meses será igual a

$$\frac{1}{\sqrt{5}}\Big[\Big(\frac{1+\sqrt{5}}{2}\Big)^n - \Big(\frac{1-\sqrt{5}}{2}\Big)^n\Big].$$

Consequentemente, o número $f(n)$ de coberturas perfeitas de um tabuleiro $2 \times n$ é igual a

$$\frac{1}{\sqrt{5}}\Big[\Big(\frac{1+\sqrt{5}}{2}\Big)^{n+1} - \Big(\frac{1-\sqrt{5}}{2}\Big)^{n+1}\Big],$$

pois $f(n) = F(n+1)$.

Teste. *Pretendemos transmitir mensagens codificadas através de um determinado canal de comunicação. Essas mensagens são formadas por palavras de comprimento n, construídas com os símbolos '0', ''1 e '2' e sujeitas à condição "não podem aparecer palavras com dois símbolos '2' consecutivos". Seja $T(n)$ o* tamanho *deste código, isto é, o número de palavras que podemos transmitir com ele. Determine uma relação de recorrência que $T(n)$ satisfaça e determine explicitamente esse número, resolvendo a relação de recorrência.*

Solução. De acordo com a definição do código, $T(1) = 3$ (as únicas palavras de comprimento 1 são: '0', ''1 e '2') e $T(2) = 8$:

$$00, 01, 02, 10, 11, 12, 20, 21.$$

Para $n \geq 3$ tem-se $T(n) = 2\,T(n-1) + 2\,T(n-2)$. De facto, as palavras de comprimento n que terminam em 0, bem como as que terminam em 1, são em número igual a $T(n-1)$:

No entanto, nas palavras de comprimento n que terminam em 2, a penúltima posição $n-1$ já só pode conter os números 0 ou 1, pelo que as palavras são, no total, em número igual a $2\,T(n-2)$:

A equação característica desta relação de recorrência é igual a $x^2 - 2x - 2 = 0$, que tem raízes $x = \frac{2 \pm \sqrt{12}}{2} = 1 \pm \sqrt{3}$. Portanto,

$$T(n) = c_1(1+\sqrt{3})^{n-1} + c_2(1-\sqrt{3})^{n-1}.$$

Das condições iniciais tiramos

$$\begin{cases} c_1(1+\sqrt{3})^0 + c_2(1-\sqrt{3})^0 = 3 \\ \\ c_1(1+\sqrt{3}) + c_2(1-\sqrt{3}) = 8 \end{cases} \Leftrightarrow \begin{cases} c_1 + c_2 = 3 \\ \\ c_1(1+\sqrt{3}) + c_2(1-\sqrt{3}) = 8 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \cdots \Leftrightarrow \begin{cases} c_1 = \dfrac{5+3\sqrt{3}}{2\sqrt{3}} \\ \\ c_2 = \dfrac{-5+3\sqrt{3}}{2\sqrt{3}}. \end{cases}$$

Finalmente,

$$\boxed{T(n) = \tfrac{5+3\sqrt{3}}{2\sqrt{3}}(1+\sqrt{3})^{n-1} + \tfrac{-5+3\sqrt{3}}{2\sqrt{3}}(1-\sqrt{3})^{n-1}.}$$

Se as raízes características $\alpha_1, \alpha_2, \ldots, \alpha_k$ não forem todas distintas então

$$u_n = c_1\alpha_1^n + c_2\alpha_2^n + \cdots + c_k\alpha_k^n \tag{1}$$

não é uma solução geral da relação de recorrência. Por exemplo, a relação de recorrência $u_n = 4u_{n-1} - 4u_{n-2}$ tem como equação característica $x^2 - 4x + 4 = (x-2)^2 = 0$. Neste caso (1) é igual a

$$u_n = c_1 2^n + c_2 2^n = (c_1 + c_2)2^n = c2^n$$

onde $c = c_1 + c_2$ é uma constante. Temos então uma só constante c e não será sempre possível escolhê-la de modo a que os dois valores iniciais u_1 e u_2 sejam satisfeitos. Por exemplo, se $u_0 = 1$ e $u_1 = 3$ teria que ser $c = 1$ e $2c = 3$, o que é manifestamente impossível. Portanto, $u_n = c2^n$ não é uma solução geral daquela relação (isto é, nem toda a solução da relação de recorrência pode ser expressa na forma $c2^n$ para alguma constante c).

O teorema seguinte, que não demonstraremos, diz-nos como determinar uma solução geral nestes casos. A ideia da demonstração é a mesma da do Teorema 2, mas naturalmente mais técnica e trabalhosa.

Teorema 3. *Sejam $\alpha_1, \alpha_2, \ldots, \alpha_t$ as raízes distintas da equação característica da relação de recorrência $(*)$, com multiplicidades, respectivamente, $e_1, e_2, \ldots, e_t$. Uma sucessão de termo geral u_n é solução de $(*)$ se e só se existem constantes*

$$c_{11}, c_{12}, \ldots, c_{1e_1}, c_{21}, c_{22}, \ldots, c_{2e_2}, \ldots, c_{t1}, c_{t2}, \ldots, c_{te_t}$$

tais que

$$\begin{aligned} u_n \;=\; & \Big(c_{11} + c_{12}n + \cdots + c_{1e_1}n^{e_1-1}\Big)\alpha_1^n + \Big(c_{21} + c_{22}n + \cdots + c_{2e_2}n^{e_2-1}\Big)\alpha_2^n + \cdots + \\ & + \Big(c_{t1} + c_{t2}n + \cdots + c_{te_t}n^{e_t-1}\Big)\alpha_t^n. \end{aligned}$$

Exemplo. Determinemos a solução da relação de recorrência

$$u_n = -u_{n-1} + 3u_{n-2} + 5u_{n-3} + 2u_{n-4} \quad (n = 4, 5, \dots)$$

sujeita às condições iniciais $u_0 = 1, u_1 = 0, u_2 = 1, u_3 = 2$. A equação característica $x^4 + x^3 - 3x^2 - 5x - 2 = 0$ tem raízes -1 e 2, sendo -1 raiz de multiplicidade 3. Portanto, a parte da solução geral correspondente à raiz -1 é

$$(c_{11} + c_{12}n + c_{13}n^2)(-1)^n,$$

enquanto que a parte correspondente à raiz 2 é $c_{21}2^n$. As constantes estão sujeitas às condições iniciais

$$\begin{cases} c_{11} + c_{21} & = & 1 \\ -c_{11} - c_{12} - c_{13} + 2c_{21} & = & 0 \\ c_{11} + 2c_{12} + 4c_{13} + 4c_{21} & = & 1 \\ -c_{11} - 3c_{12} - 9c_{13} + 8c_{21} & = & 2, \end{cases}$$

pelo que, resolvendo o sistema, obtemos $c_{11} = 7/9$, $c_{12} = -1/3$, $c_{13} = 0$ e $c_{21} = 2/9$. Em conclusão, a solução é

$$u_n = \left(\frac{7}{9} - \frac{1}{3}n\right)(-1)^n + \frac{2^{n+1}}{9} \quad (n \in \mathbb{N}_0).$$

O sucesso deste método depende da nossa capacidade em determinar as raízes da equação característica, o que poderá por vezes não ser possível. No caso de tal ser possível, será ainda necessário resolver um sistema de equações lineares. Se a ordem da relação de recorrência for k, este sistema tem k equações com k incógnitas. Portanto a aplicação deste método, na prática, poderá ser muito problemática.

Se a relação de recorrência não for homogénea ou linear, com coeficientes constantes, não se conhecem métodos para a resolver de uma forma sistemática (a não ser para alguns tipos de relações não homogéneas nas quais o termo independente tem uma forma muito especial — é um polinómio ou uma exponencial). Cada caso terá que ser analisado individualmente. Por exemplo, para resolver a relação de recorrência não homogénea $u_n = u_{n-1} + n^3$, para $n = 1, 2, \dots$, sujeita à condição $u_0 = 0$, podemos, por sucessivas iterações, obter

$$\begin{aligned} u_n &= u_{n-1} + n^3 \\ &= u_{n-2} + (n-1)^3 + n^3 \\ &= \cdots \\ &= u_1 + 2^3 + \cdots + (n-1)^3 + n^3 \\ &= 1^3 + 2^3 + \cdots + n^3. \end{aligned}$$

Assim, u_n é a soma dos primeiros n cubos. Podemos determinar uma expressão simples para esta soma? Usando a relação de recorrência podemos determinar os primeiros valores de u_n e

tentar encontrar um padrão:

$$\begin{aligned} u_1 &= 1 \\ u_2 &= 1 + 2^3 = 9 = 3^2 = (1+2)^2 \\ u_3 &= 9 + 3^3 = 36 = 6^2 = (1+2+3)^2 \\ u_4 &= 36 + 4^3 = 100 = 10^2 = (1+2+3+4)^2 \\ u_5 &= 100 + 5^3 = 225 = 15^2 = (1+2+3+4+5)^2. \end{aligned}$$

Como

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2},$$

podemos conjecturar que

$$u_n = \frac{n^2(n+1)^2}{4},$$

o que pode ser confirmado pelo método de indução matemática.

Apêndice: caso não homogéneo

No caso não homogéneo, é possível em alguns casos uma abordagem sistemática que nos conduza à solução. Uma recorrência linear, não necessariamente homogénea, de coeficientes constantes é dada por uma equação do tipo

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k} + g(n), \tag{1}$$

onde o termo independente $g(n)$ é uma função de n que toma valores reais. A uma recorrência deste tipo podemos, esquecendo a função g, associar a recorrência homogénea

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k}. \tag{2}$$

Será de esperar que as soluções de (1) estejam relacionadas com as soluções de (2). De facto, é fácil provar que:

Teorema 4. *Seja*

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k} + g(n)$$

uma relação de recorrência linear com coeficientes constantes e seja (α_n) uma solução desta relação de recorrência. Se (β_n) é também uma solução dessa relação de recorrência, então a sucessão $(\gamma_n) = (\beta_n - \alpha_n)$ é uma solução da relação de recorrência homogénea

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k}.$$

Reciprocamente, se (γ_n) é uma solução desta relação de recorrência homogénea, então a sucessão $(\beta_n) = (\alpha_n + \gamma_n)$ é uma solução da relação de recorrência inicial.

Assim, para determinar a expressão geral das soluções de uma dada relação de recorrência linear com coeficientes constantes, bastará:

(1) Obter a expressão geral das soluções (γ_n) da relação de recorrência homogénea associada;

(2) Identificar uma solução particular (β_n) da relação de recorrência dada;

(3) A expressão geral das soluções (α_n) da relação de recorrência é dada pela soma $(\alpha_n) = (\beta_n + \gamma_n)$.

O passo (1) pode realizar-se pelo método apresentado no caso homogéneo, mas a realização de (2) depende da função g envolvida. Em geral, não há nenhuma garantia que (2) se possa efectuar de modo fácil; os casos mais simples são aqueles em que g é polinomial ou exponencial. A tabela seguinte fornece-nos soluções particulares para esses casos:

Soluções particulares para a relação de recorrência linear com coeficientes constantes
$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_k u_{n-k} + g(n)$
onde $x^k - a_1 x^{k-1} - \cdots - a_{k-1} x - a_k \qquad (*)$
é a equação característica da relação de recorrência homogénea associada

Função f	**Condições**	**Solução particular**
$f(n) = b\lambda^n$	λ não é raíz de $(*)$	$(\beta\lambda^n)$
$(b, \lambda \in \mathbb{R} - \{0\})$	λ é raíz de $(*)$, com multiplicidade m	$(\beta n^m \lambda^n)$
$f(n) = b_0 + b_1 n + \cdots + b_r n^r$	1 não é raíz de $(*)$	$(\beta_0 + \beta_1 n + \cdots + \beta_r n^r)$
$(r \in \mathbb{N}, b_0, \ldots, b_r \in \mathbb{R}, b_r \neq 0)$	1 é raíz de $(*)$, com multiplicidade m	$(n^m(\beta_0 + \beta_1 n + \cdots + \beta_r n^r))$
$f(n) = bn^r\lambda^n$	λ não é raíz de $(*)$	$((\beta_0 + \beta_1 n + \cdots + \beta_r n^r)\lambda^n)$
$(r \in \mathbb{N}, b, \lambda \in \mathbb{R} - \{0\})$	λ é raíz de $(*)$, com multiplicidade m	$(n^m(\beta_0 + \beta_1 n + \cdots + \beta_r n^r)\lambda^n)$

Bibliografia

[1] Carlos André e Fernando Ferreira, *Matemática Finita*, Universidade Aberta, 2000.

[2] Stephen Barnett, *Discrete Mathematics: Numbers and Beyond*, Prentice Hall, 1998.

[3] Jon Barwise e John Etchemendy, *Language, Proof and Logic*, CSLI Publications, 1999.

[4] James Hein, *Discrete Structures, Logic and Computability*, Portland State University, 2002.

[5] James Hein, *Maple Experiments in Discrete Mathematics*, Portland State University, Janeiro 2005.

[6] Kenneth H. Rosen, *Discrete Mathematics and its Applications*, McGraw-Hill, 1995.

[7] Kenneth H. Rosen, *Exploring Discrete Mathematics with Maple*, McGraw-Hill, 1997.

[8] R. J. Wilson, *Introduction to Graph Theory*, Longman, 1972.