# Palavras

Muitos problemas de Combinatória envolvem seqüências de símbolos. Para facilitar o tratamento desses problemas, vamos chamar de *alfabeto* o conjunto $\Sigma$ dos possíveis símbolos e chamar as seqüências de símbolos de *palavras*, sendo $\Sigma^*$ o conjunto de todas as palavras.

A maioria das técnicas necessárias para resolver problemas de palavras são idéias clássicas da Combinatória, e aproveitamos para revisá-las.

A primeira é a que mais usamos no ensino fundamental e a que menos usamos (e deveríamos usar mais) no ensino médio.

### 1. Casos pequenos e busca de padrões

Estudar casos pequenos muitas vezes nos ajuda a encontrar padrões que nunca descobriríamos diretamente. Nessa hora, é importante listar os casos de modo ordenado (testar várias maneiras de organizar os casos também ajuda bastante) e, se for necessário, utilizar notação adequada para estudar casos não tão pequenos.

O próximo exemplo não é bem de palavras, mas envolve seqüências e é instrutivo.

**Exemplo 1.1.**

A seqüência $a_0, a_1, \ldots$ é construída da seguinte forma: $a_0 = 0$, $a_1 = 1$ e, para $n > 1$, $a_{n+1}$ é o menor inteiro maior do que $a_n$ tal que $a_0, a_1, \ldots, a_{n+1}$ não contenha progressões aritméticas de três termos. Encontre $a_{2006}$.

**Resolução**

Não tem idéia do que fazer? Teste casos pequenos! Os primeiros termos da seqüência são

| $n$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_n$ | 0 | 1 | 3 | 4 | 9 | 10 | 12 | 13 | 27 | 28 | 30 | 31 | 36 | 37 | 39 | 40 | 81 |

Veja só! Note que $a_{2^n} = 3^n$ para $n = 1, 2, 3, 4$. Talvez se escrevermos $n$ na base 2 e $a_n$ na base 3...

| $n$ (base 2) | 0 | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | 10000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_n$ (base 3) | 0 | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | 10000 |

Temos evidências suficientes para querermos provar que $a_n$ é obtido escrevendo-se $n$ na base 2 e tomando os mesmos dígitos, mas na base 3.

Vamos provar isso por indução, modificando um pouco a hipótese de indução. A nossa hipótese é

*Os $n$ primeiros valores da seqüência são os $n$ primeiros números que, na base 3, só têm dígitos 0 e 1, na ordem crescente.*

Isso é verdade para $n = 0$ e $n = 1$. Agora suponha que seja verdade para $n = k$. Provaremos que vale para $n = k + 1$.

Primeiro, $a_{k+1}$ não pode ter dígito 2, porque forma progressão aritmética com os termos obtidos trocando os dígitos 2 por 0 e por 1 e mantendo os demais.

Segundo, $a_{k+1}$ pode ser o próximo número formado por zeros e uns porque se $x < y < z$ formam uma progressão aritmética e $x$ e $y$ só têm dígitos 0 e 1, então $z - y = y - x \iff z = 2y - x$. Como $x < y$, observando seus dígitos na base 3 existe uma posição mais à esquerda em que o dígito de $y$ é 1 e o de $x$ é 0. Assim, $z$ teria que ter um dígito 2, ou seja, $a_{k+1}$ é o próximo número com dígitos 0 e 1.

Desse modo, como 2006 na base 2 é 11111010110, $a_{2006} = 11111010110$ na base 3, ou seja, $a_{2006} = 3^{10} + 3^9 + 3^8 + 3^7 + 3^6 + 3^4 + 3^2 + 3^1 = 88302$.

Você não adivinharia esse padrão de cara, adivinharia?

**Exercícios**

01. Seja $\Sigma = \{A; B\}$. Prove que toda palavra com 1995 letras de $\Sigma$ pode ser formada com menos de 800 palíndromos (palavras que não se alteram quando escritas na ordem inversa).

02. Suponha que $|\Sigma| = n$. Dizemos que uma palavra é *válida* quando no meio de quaisquer duas letras iguais não há duas letras iguais. Encontre, em função de $n$, a quantidade de palavras válidas de tamanho máximo.

03. (Banco IMO 1997) Definimos as seqüências $R_n$ da seguinte forma: $R_1 = (1)$ e se $R_n = (a_1, a_2, \ldots, a_k)$, então $R_{n+1} = (1, 2, \ldots, a_1, 1, 2, \ldots, a_2, \ldots, 1, 2, \ldots, a_k, n+1)$. Por exemplo, $R_2 = (1, 2)$, $R_3 = (1, 1, 2, 3)$, $R_4 = (1, 1, 1, 2, 1, 2, 3, 4)$. Prove que, para $n > 1$, o $m$-ésimo termo da esquerda para a direita de $R_n$ é 1 se, e somente se, o $m$-ésimo termo da direita para a esquerda de $R_n$ não é 1.

*Dica: Nesse aqui, uma boa notação é o que resolve o problema!*

## 2. Contagem; em especial, recursões

Muitos problemas de palavras consistem em provar a existência de certas palavras ou de mostrar que há uma certa quantidade de palavras que satisfazem a certas condições. Muitas vezes, essas condições é **não** ter algo. Vamos chamar as palavras que têm esse "algo" de *proibidas*.

Nesses casos, uma contagem com recursões e exclusão de casos desfavoráveis ("tudo menos o que não interessa") costuma dar conta do recado.

A recursão normalmente funciona assim:

- Seja $x_n$ a quantidade de palavras com comprimento $n$ que têm a propriedade desejada. Vamos estimar $x_{n+1}$ em função dos termos anteriores.
- Sendo $k = |\Sigma|$, então $x_{n+1} = kx_n - p$, sendo $p$ a quantidade de palavras proibidas formadas com a adição de um símbolo no final das $x_n$ palavras permitidas com comprimento $n$.
- Normalmente, $p$ é estimado (por cima) em função dos termos anteriores e montamos uma deigualdade recursiva.
- Tentamos provar por indução que $x_{n+1} \geq c \cdot x_n$ para algum $c > 1$. Para isso, usamos a desigualdade recursiva.

Vamos ver direito como isso funciona.

**Exemplo 2.1.**

Seja $\Sigma = \{a, b, c\}$. Algumas seqüências, cada uma com duas ou mais letras, são proibidas. Uma palavra é *permitida* quando não contém seqüências proibidas. Sabe-se que não há duas ou mais seqüências proibidas de mesmo comprimento. Prove que existem palavras permitidas de qualquer tamanho.

**Resolução**

Seja $x_n$ a quantidade de palavras permitidas de tamanho $n$. Temos então que provar que $x_n > 0$.

Vamos estimar $x_{n+1}$. Colocando uma letra à direita de uma palavra permitida de tamanho $n$, obtemos $3x_n$ palavras. Todavia, podemos formar uma seqüência proibida no final, e devemos excluir casos. Seja então $y_k$ o número de palavras formadas com a palavra proibida de tamanho $k$ no final. Então

$$x_{n+1} = 3x_n - (y_1 + y_2 + \cdots + y_n)$$

Vamos estimar $y_i$. Note que ao tirarmos a seqüência proibida de tamanho $i$, obtemos uma palavra permitida de $n-i$ letras. Logo $y_i \leq x_{n-i}$. Note que não necessariamente ocorre a igualdade porque pode ser que nem toda palavra permitida de $n - i$ letras seja gerada pela retirada da seqüência proibida. Portanto, fazendo $x_0 = 1$,

$$x_{n+1} \geq 3x_n - (x_{n-1} + x_{n-2} + \cdots + x_0)$$

Suponha que $x_{k+1} \geq m \cdot x_k$ para todo $k$. Vamos encontrar um valor de $m > 1$. Temos $x_n \geq m^i \cdot x_{n-i} \iff x_{n-i} \leq x_n/m^i$. Assim,

$$x_{n+1} \geq 3x_n - (x_n/m + x_n/m^2 + \cdots + x_n/m^n)$$

Aí, em vez de resolver uma inequação polinomial de grau $n$ (onde $n$ varia!), vamos fazer mais uma estimativa:

$$\frac{1}{m} + \frac{1}{m^2} + \cdots + \frac{1}{m^n} < \frac{1}{m} + \frac{1}{m^2} + \cdots = \frac{1/m}{1 - 1/m} = \frac{1}{m-1}$$

Logo

$$x_{n+1} > 3x_n - x_n/(m-1) = \frac{3m-4}{m-1} x_n$$

Basta então que

$$\frac{3m-4}{m-1} \geq m \iff m^2 - 4m + 4 \leq 0 \iff m = 2$$

Assim, $x_{n+1} \geq 2x_n$ para todo $n$ natural e, deste modo, $x_n \geq 2^n$. Logo há palavras permitidas de qualquer tamanho. ∎

**Exercícios**

04. Prove que existem pelo menos $8^n$ números de $n$ algarismos sem dois blocos consecutivos iguais de algarismos.

05. Seja $\Sigma = \{a, b, c, d\}$. Uma palavra é dita *complicada* quando contém dois blocos consecutivos iguais. Caso contrário, a palavra é dita *simples*. Por exemplo, *badcabad* é simples e *baba* é complicada. Prove que há pelo menos $2^n$ palavras simples de tamanho $n$.

06. Sejam $\Sigma = \{0; 1\}$ e $S_n$ o número de palavras de tamanho $n$ que não contêm seis blocos consecutivos iguais. Prove que $|S_n|$ tende a infinito quando $n$ vai para infinito.

07. Seja $\Sigma = \{A; B\}$ e seja $P$ um conjunto de 6 palavras, cada uma com quatro letras de $\Sigma$. Uma palavra é *permitida* se não contém uma seqüência de letras de $P$. Prove que se o conjunto de palavras permitidas é finito então o comprimento de cada palavra permitida é no máximo 10. Encontre um exemplo para $P$ que forneça um conjunto finito de palavras permitidas com comprimento máximo 10.

08. Vamos generalizar o último exemplo. Suponha agora que o alfabeto $\Sigma$ tem $n$ símbolos e que há no máximo $m$ seqüências proibidas de mesmo comprimento. Seja $p_k$ a quantidade de palavras permitidas nessas condições.

(a) Prove que se $m \leq \left(\frac{n-1}{2}\right)^2$ então existe uma constante real $\lambda > 1$ tal que $p_{n+1} \geq \lambda p_n$ para todo $n$ inteiro positivo.

(b) Um *conjunto bloqueador* é um conjunto de seqüências proibidas que tornam o conjunto de palavras permitidas finito. Prove que se $m \geq \frac{n(n-2)}{2} + 2$ então existe um conjunto bloqueador.

09. Prove que não existe conjunto bloqueador de menos de três seqüências se $\Sigma = \{a, b\}$.

### 3. Para o infinito e além! Propriedades hereditárias

Vamos formular uma outra pergunta relativa ao exemplo anterior.

**Exemplo 3.1.**

Seja $\Sigma = \{a, b, c\}$. Algumas seqüências, cada uma com duas ou mais letras, são proibidas. Uma palavra é *permitida* quando não contém seqüências proibidas. Sabe-se que não há duas ou mais seqüências proibidas de mesmo comprimento. Existe alguma palavra permitida *infinita*?

**Resolução**

Primeiro vamos entender por que realmente temos que resolver o problema. De fato, *provar que existe para todo n* é bem diferente de *provar que existe um infinito*. No caso de palavras, estamos pensando em *provar que existe uma palavra de tamanho n com a propriedade P para todo n* contra *provar que existe uma palavra infinita com a propriedade P*.

De fato, se a propriedade $P$ é "é finita", a resposta da primeira pergunta é sim e a resposta da segunda pergunta é não.

Nesse problema em especial, a resposta é sim e a técnica para resolver esse problema pode ser estendida para muitos outros. É essencialmente o princípio da casa dos pombos aplicado indutivamente.

Seja $P$ o conjunto das palavras permitidas. Sabemos que $P$ é infinito. Assim, como há uma quantidade finita de letras iniciais ($a$, $b$ e $c$), existe uma letra $a_1$ que é a inicial de infinitas palavras de $P$. Denote o conjunto das palavras permitidas que começam com $a_1$ por $P_1$. Agora, considere a segunda letra das palavras de $P_1$. Como $P_1$ é infinito e há uma quantidade finita de letras, existe uma letra $a_2$ que é a segunda letra de infinitas palavras de $P_1$. Denote por $P_2$ o conjunto (infinito) das palavras permitidas que iniciam com $a_1a_2$ e continue indutivamente. De fato, se $P_n$ é o conjunto infinito das palavras permitidas que começam com $a_1a_2 \ldots a_n$, se consideramos a $(n+1)$-ésima letra das palavras de $P_n$, existe uma, $a_{n+1}$, que aparece em infinitas palavras e podemos definir $P_{n+1}$.

Com isso, obtemos uma palavra infinita $a_1a_2a_3 \ldots$ que afirmamos ser permitida. De fato, suponha que ela contenha alguma seqüência proibida e considere a primeira ocorrência dela. Corte a palavra logo depois dessa primeira ocorrência, obtendo uma palavra finita. Mas essa palavra é permitida, absurdo. ∎

Sendo $P$ uma propriedade que palavras podem ter ou não, essa idéia pode ser generalizada para palavras que satisfazem os seguintes dois fatos:

- Toda palavra formada pelas primeiras letras de uma palavra com a propriedade $P$ também tem a propriedade $P$.
- Uma palavra infinita com segmentos iniciais arbitrariamente longos com a propriedade $P$ também tem a propriedade $P$.

As propriedades que satisfazem os fatos acima são ditas *hereditárias*.

**Teorema 3.1.** *Seja $P$ uma propriedade hereditária. Se existem palavras arbitrariamente longas com essa propriedade, então existe uma palavra infinita com essa propriedade.*

**Exercícios**

10. Resolva os problemas 04 a 06 indicando agora se existem palavras infinitas ou não.

11. Um número é um 2005-*diamante* se contém 2005 noves seguidos em sua representação decimal. Prove que toda seqüência $a_n$ estritamente crescente de inteiros positivos que satisfazem $a_n < c \cdot n$ para alguma constante $c > 0$ contém infinitos 2005-diamantes.

12. O número real $x$ entre 0 e 1 tem representação decimal

$$0{,}a_1a_2a_3 \ldots$$

com a seguinte propriedade: o número de blocos distintos da forma

$$a_ka_{k+1}a_{k+2}a_{k+3} \ldots a_{k+2004},$$

onde $k$ varia entre todos os inteiros positivos, é menor ou igual a 2005.

Prove que $x$ é racional.

## 4. Operações sobre palavras

Muitos problemas com palavras envolvem operações com seus símbolos. Nesse caso, assim como todo problema envolvendo operações, vale a pena tomar duas medidas:

- Procurar algum invariante;
- Verificar se alguma seqüência de operações se reduz a uma operação conhecida, como a transposição ou um *shift*.

### 4.1. Transposição? Shift?

Uma *transposição* em uma palavra é uma função de $\Sigma^*$ em $\Sigma^*$ que transforma dois símbolos seguidos $ab$ em $ba$. Uma propriedade importante da transposição é que podemos trocar a ordem dos símbolos de qualquer palavra como quisermos (prove isso, é uma indução bem simples). Ou seja, a ordem dos símbolos não importa mais.

Um *k-shift* em uma palavra é uma função de $\Sigma^*$ em $\Sigma^*$ que transforma $k$ símbolos seguidos $a_1a_2a_3\ldots a_k$ em $a_ka_1a_2\ldots a_{k-1}$. Note que um 2-shift é uma transposição.

Também podemos entender *shift* como transformar a palavra (inteira) $a = a_1a_2a_3\ldots a_n$ em $a' = a_na_1a_2\ldots a_{n-1}$. Nesses casos, é bastante útil estender a palavra infinitamente para os dois lados:

$$a^\infty = \ldots a_1a_2\ldots a_na_1a_2\ldots a_n\|a_1a_2\ldots a_na_1a_2\ldots a_n\ldots$$

Note que há um $\|$; um shift seria simplesmente mover o $\|$ uma casa para a direita.

**Exemplo 4.1.**

(Rioplatense 2005) Deseja-se colorir cada número inteiro positivo de uma cor, utilizando a maior quantidade de cores possível de modo que se verifique a seguinte condição: se, em notação decimal, o número $B$ pode ser obtido do número $A$, eliminando de $A$ dois dígitos iguais e consecutivos ($aa$) ou eliminando de $A$ quatro dígitos consecutivos que formem dois pares iguais e consecutivos ($abab$), então $A$ e $B$ são da mesma cor.

Por exemplo, 8, 833 e 22811 devem ser da mesma cor; e, também, 72, 676772 e 1173329898 são da mesma cor.

Determinar qual é a maior quantidade de cores que se pode utilizar.

**Resolução**

Primeiro, um invariante: note que as operações não alteram a paridade da quantidade de cada algarismos. Assim, como há duas paridades para cada quantidade de algarismos de cada tipo e dez algarismos, precisamos de pelo menos $2^{10} = 1024$ cores.

A verdade é que 1024 cores são suficientes (se bem que não conheço alguém que possa diferenciar 1024 cores). Para ver isso, considere um número que contém, em algum momento, os dígitos $ababba$. Retirando $abab$, obtemos $ba$; retirando $bb$, obtemos $abaa$ e agora retirando $aa$, obtemos $ab$. Isto é: conseguimos fazer transposições!

Isto quer dizer que a ordem dos algarismos não nos interessa mais: assim podemos colocar todos os noves, depois todos os oitos, até os zeros. Em seguida, retiramos todos os pares de dígitos repetidos, obtendo um ou nenhum algarismo de cada tipo. Dessa forma reduzimos todo número inteiro positivo a um de $2^{10}$ casos, e isso mostra que 1024 cores são suficientes.

**Exemplo 4.2.**

Sejam $n > 1$ inteiro e $p_1, p_2, \ldots, p_n$ palavras cujas $n$ concatenações cíclicas $p_1p_2\ldots p_n$, $p_2p_3\ldots p_1$, $\ldots$, $p_np_1\ldots p_{n-1}$ são iguais. Prove que existe uma palavra $q$ tal que toda palavra $p_i$ é uma concatenação de $q$'s, ou seja, $p_i = \underbrace{qq\ldots q}_{n_i \text{ vezes}}$.

**Resolução**

Seja $p = p_1p_2 \dots p_n = p_2p_3 \dots p_1 = \dots = p_np_1 \dots p_{n-1}$. Considere agora $p^\infty$ (isto é, estenda $p$ infinitamente para os dois lados, não se esquecendo do sinal $\|$).

Vamos denotar um shift (de um símbolo) da palavra $x$ como $\sigma(x)$. Caso façamos $k$ shifts, denotamos $\sigma^k(x)$. Um shift para a esquerda é $\sigma^{-1}(x)$ e $k$ shifts para a esquerda é $\sigma^{-k}(x)$. Seja $\ell_i$ o comprimento da palavra $p_i$ e seja $s_i = \ell_1 + \ell_2 + \cdots + \ell_i$. Note que um shift de tamanho $s_i$ move o $\|$ para o final de uma palavra $p_i$ em $p^\infty$. Assim,

$$\sigma^{s_i}(p^\infty) = p^\infty$$

Compondo vários $\sigma^{s_i}$, vemos que

$$\sigma^{a_1s_1+a_2s_2+\cdots+a_ns_n}(p^\infty) = p^\infty,$$

sendo $a_1, a_2, a_3, \dots, a_n$ inteiros. Assim, pelo teorema de Bezóut, se $d$ é o mdc de todos os $s_i$'s, existem inteiros $a_1, a_2, a_3, \dots, a_n$ tais que $d = a_1s_1 + a_2s_2 + \cdots + a_ns_n$. Assim

$$\sigma^d(p^\infty) = p^\infty$$

Isto quer dizer que mover $p^\infty$ $d$ casas para a direita não o altera, ou seja, $p^\infty = q^\infty$, sendo que $q$ tem comprimento $d$. Isto é, $p^\infty$ é uma palavra $q$ de tamanho $d$ repetidas infinitas vezes.

Enfim, como $d$ divide $s_i$ e $s_{i-1}$, então divide $s_i - s_{i-1} = \ell_i$. Assim, cada palavra $p_i$ é $\ell_i/d$ vezes a palavra $q$ repetida. ∎

**Exercícios**

13. Esse último exemplo também podia ser resolvido com indução sobre $n$. Resolva-o dessa forma.

14. Suponha que $\Sigma$ é finito. Uma palavra é dita *periódica* quando é formada pela concatenação de duas ou mais palavras iguais. Duas palavras $p$ e $q$ têm o mesmo número de letras. A primeira letra de $p$ é diferente da primeira letra de $q$, mas as demais letras são respectivamente iguais. Prove que pelo menos uma das palavras $p$ e $q$ não é periódica.

15. (Banco IMO 2001) Seja $A = (a_1; a_2; \dots; a_{2001})$ uma seqüência de inteiros positivos. Seja $m$ o número de subseqüências de três elementos $(a_i; a_j; a_k)$ com $1 \leq i < j < k \leq 2001$, tais que $a_j = a_i + 1$ e $a_k = a_j + 1$. Considerando todas as seqüências $A$, encontre o valor máximo de $m$.

*Dica: defina algumas operações que não reduzam/aumentem m para facilitar sua vida; depois, o problema fica realmente fácil.*

16. Seja $n$ um inteiro positivo. Uma seqüência de zeros e uns é dita *balanceada* quando contém $n$ zeros e $n$ uns. Duas seqüências balanceadas $a$ e $b$ são *vizinhas* quando é possível mover um dos $2n$ símbolos de $a$ para outra posição para formar $b$. Por exemplo, para $n = 4$, as seqüências balanceadas 01101001 e 00110101 são vizinhas porque o terceiro (ou o quarto) zero na primeira seqüência pode ser movido para a primeira ou segunda posição para formar a segunda seqüência. Prove que existe um conjunto $S$ com no máximo $\frac{1}{n+1}\binom{2n}{n}$ seqüências balanceadas tais que toda seqüência balanceada é igual a ou é vizinho de pelo menos uma seqüência de $S$.

## 5. Uma seqüência interessante e duas de suas inusitadas propriedades

Não é difícil ver que a maior palavra utilizando um alfabeto de dois símbolos sem dois blocos consecutivos iguais de qualquer tamanho tem três letras (tente ver por quê). E se tivermos três símbolos, digamos 0, 1, 2?

A resposta é que podemos ter uma palavra infinita! Vamos mostrar a construção.

### 5.1. Uma seqüência binária legal

Primeiro, construímos uma seqüência infinita somente com zeros e uns. Começamos com um zero, e a cada passo, copiamos a palavra que tem até o momento, só que trocando zero por um e um por zero. A seguir, separamos os passos com barras verticais:

$$0|1|10|1001|10010110|1001011001101001|1001011001101001011010011001011 0|\dots$$

A seqüência acima também é conhecida como *seqüência de Morse*.

Sendo a seqüência $A = a_0a_1a_2a_3\dots$ (de modo que $a_0 = 0$, $a_1 = 1$, $a_2 = 1$, $a_3 = 0\dots$), outra maneira de descrevê-la é

$$a_k = \begin{cases} 1, & \text{se o número de uns na representação binária de } k \text{ é ímpar} \\ 0, & \text{se o número de uns na representação binária de } k \text{ é par} \end{cases}$$

A demonstração é por indução sobre a quantidade de dígitos 1 na representação binária de $k$ e fica a cargo do leitor.

Assim, utilizando a definição acima, verifica-se que, para todo $n$ inteiro positivo, $a_{2n} = a_n$ ($2n$, na base binária, é "$n$ seguido de um zero") e $a_{2n} \neq a_{2n+1}$ ($2n+1$, na base binária, é "$2n$ com o último zero trocado por um").

Com isso não é difícil ver que não pode haver três uns consecutivos na seqüência e, portanto, podemos finalmente chegar a

### 5.2. A tão sonhada seqüência

**Teorema 5.1.** *A seqüência infinita $B = b_1b_2b_3\dots$, em que $b_i$ é a quantidade de uns entre o $i$-ésimo e o $(i+1)$-ésimo zeros da seqüência $A$ de Morse, não tem blocos consecutivos iguais.*

**Demonstração**

Suponha que a seqüência $B$ admita dois blocos consecutivos iguais. Então a seqüência $A$ admite um bloco da forma $FFf$, sendo $F = a_{m+1}a_{m+2}\dots a_{m+k}$ um bloco que se repete e $f$ o primeiro termo de $F$.

Vamos dividir o problema em casos. Lembre-se que em cada um deles queremos chegar a uma contradição.

*Caso 1: $k$ ímpar.* Observe que na seqüência $FF$, $a_{m+k+1} = a_{m+1}$, $a_{m+k+2} = a_{m+2}$ e, de modo mais geral, $a_{m+k+i} = a_{m+i}$. Veja que exatamente um dos números $m+k+i$ e $m+i$ é par. Isto quer dizer que $a_{m+i} \neq a_{m+i+1}$, ou seja, $F$ consiste de zeros e uns alternados ($101\dots$ ou $010\dots$). Logo, considerando ainda que $F$ tem comprimento $k$, que é ímpar, $a_{m+1} = a_{m+k} = a_{m+k+1} = a_{m+2k} = a_{m+2k+1} = f$. Mas se $m$ é ímpar, a segunda igualdade não pode ocorrer; se $m$ é par, a quarta igualdade não pode ocorrer. Contradição.

*Caso 2: $k$ par.* Aqui, dividimos em dois subcasos. Além disso, vamos supor que $FFf$ é a *primeira* ocorrência de uma repetição de blocos.

*Caso 2.1: $k$ par, $m$ ímpar.* Tome os termos de ordem par de $FFf$, ou seja,

$$\underbrace{a_{m+1}a_{m+3}\dots a_{m+k-1}}_{F'}\underbrace{a_{m+k+1}a_{m+k+3}\dots a_{m+2k-1}}_{F'}f$$

Agora vamos utilizar a propriedade $a_{2n} = a_n$. Note que sendo $m+1 = 2r$ e $k = 2s$, temos $F' = a_ra_{r+1}\dots a_{r+s-1}$ e ocorre a seqüência $F'F'f$ antes de $FFf$ em $A$, absurdo.

*Caso 2.2: $k$ par, $m$ par.* Aqui, basta notar que $a_m \neq a_{m+1}$ e $a_{m+k} \neq a_{m+k+1}$, sendo a seqüência de Morse binária e $a_{m+1} = a_{m+k+1}$, $a_m = a_{m+k}$. Assim, sendo $F'' = a_ma_{m+1}\dots a_{m+k-1}$, ocorre $F''F''f''$ antes de $FFf$ (uma posição antes!), absurdo. ∎

### 5.3. Um comentário sobre técnicas

Normalmente, quando queremos provar que algo não existe, supomos que existe e chegamos a uma contradição. Essa contradição pode vir de várias maneiras, mas vale a pena apontar duas em particular que são bastante úteis e foram ambas utilizadas nessa última demonstração:

- *Encontrar características que se contradigam ou contrariam algum fato verdadeiro.* Isso foi feito no caso 1 e é o que mais fazemos.
- *Tomar a "primeira" ocorrência e provar que tem outra "antes".* Essencialmente, isso é uma paráfrase do princípio da boa ordem (todo conjunto de inteiros positivos tem um menor elemento), e não serve só para provar que equações diofantinas não têm solução (se você não entendeu essa última frase, pense no exercício a seguir)! Essa técnica é especialmente útil em entes que não têm muita estrutura. Nesses casos, ordenar é preciso!

**Exercícios**

17. Prove que a equação $x^4 + y^4 = z^4$ não têm soluções inteiras positivas.

### 5.4. Ei, cadê a outra propriedade inusitada?

Uma outra propriedade legal da seqüência de Morse é a seguinte. Observe-a novamente, porém, começaremos numerando do 1:

| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | ... |

Considere os quatro primeiros números. A seqüência de Morse induz uma partição do conjunto dos números de 1 a 4 em dois subconjuntos de 4 elementos, $\{1, 4\}$ e $\{2, 3\}$. Vejam só, $1 + 4 = 2 + 3$!

Não está impressionado? OK, então considere agora os oito primeiros números. A partição do conjunto dos números de 1 a 8 em dois subconjuntos de 4 elementos é agora $\{1, 4, 6, 7\}$ e $\{2, 3, 5, 8\}$. Note agora que

$$1^1 + 4^1 + 6^1 + 7^1 = 2^1 + 3^1 + 5^1 + 8^1$$
$$1^2 + 4^2 + 6^2 + 7^2 = 2^2 + 3^2 + 5^2 + 8^2$$

Ah, tudo bem, é sorte, certo? Que tal os *dezesseis* primeiros termos? Parece mentira, mas

$$1^1 + 4^1 + 6^1 + 7^1 + 10^1 + 11^1 + 13^1 + 16^1 = 2^1 + 3^1 + 5^1 + 8^1 + 9^1 + 12^1 + 14^1 + 15^1$$
$$1^2 + 4^2 + 6^2 + 7^2 + 10^2 + 11^2 + 13^2 + 16^2 = 2^2 + 3^2 + 5^2 + 8^2 + 9^2 + 12^2 + 14^2 + 15^2$$
$$1^3 + 4^3 + 6^3 + 7^3 + 10^3 + 11^3 + 13^3 + 16^3 = 2^3 + 3^3 + 5^3 + 8^3 + 9^3 + 12^3 + 14^3 + 15^3$$

Generalizando esse fato: a partição induzida pela seqüência de Morse dos $2^n$ primeiros termos, $n > 1$, divide os números de 1 a $2^n$ em dois subconjuntos $X$ e $Y$ com a mesma quantidade cujas somas das $m$-ésimas potências, $m = 1, 2, \ldots, n - 1$, são iguais. Simbolicamente:

$$\sum_{x \in X} x^m = \sum_{y \in Y} y^m, \quad m = 1, 2, \ldots, n - 1$$

**Exercícios**

18. Prove o fato acima. Na verdade, dá para generalizar ainda um pouco mais (por exemplo, progressões aritméticas no lugar dos inteiros positivos). *Dica: indução sobre $n$.*

### 5.5. Mais propriedades

Se dois é um número muito pequeno, veja mais algumas propriedades:

- A chamada *regra alemã* em xadrez diz que se uma seqüência de movimentos ocorre três vezes consecutivamente, então o jogo termina empatado. Utilizando a seqüência de Morse, Max Euwe provou que existe uma partida de xadrez que nunca termina segundo essa regra.
- A seqüência de Morse também foi utilizada para provar um teorema de Geometria Diferencial.
- Considere o conjunto $A$ obtido da seguinte forma: 0 e 1 pertencem a $A$; os ímpares pertencem a $A$; e colocamos sempre o menor par $2n$ tal que $n \notin A$. Assim $A = \{0, 1, 3, 4, 5, 7, 9, 11, 13, 15, 16, 17, 19, 20, \ldots\}$. Seja $a_n$ o $n$-ésimo elemento, em ordem crescente, de $A$. Defina a seqüência

$$z = 0^{a_2 - a_1} 1^{a_3 - a_2} 0^{a_4 - a_3} \ldots 0^{a_{2n+1} - a_{2n}} 1^{a_{2n+2} - a_{2n+1}},$$

  em que $0^k$ é 0 repetido $k$ vezes e $1^k$ é definido analogamente. Então $z$ é a seqüência de Morse.

  Procure por [4] para ver ainda mais propriedades.

**Exercícios**

19. Prove que $A$ e $2A = \{2x : x \in A\}$ particionam os inteiros não negativos.

### 5.6. Mais sobre seqüências binárias

Entringer e Jackson provaram que

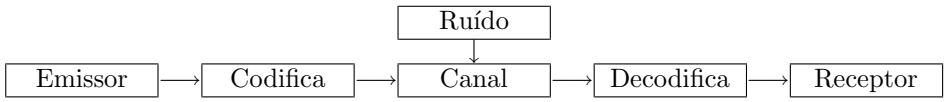
- Existe uma seqüência infinita binária sem blocos consecutivos de tamanho maior ou igual a 3.
- Toda seqüência binária de tamanho maior do que 18 tem blocos consecutivos de tamanho maior ou igual a 2.
- Toda seqüência binária infinita tem blocos arbitrariamente grandes consecutivos que são permutações um do outro.

  Dekking provou também que

- Há infinitas seqüências binárias sem repetições triplas e sem repetições de tamanho 4 ou maior.
- Seqüências binárias sem repetições triplas e sem repetições de tamanho 3 ou maior são finitas.

## 6. Um teorema que tem aplicação em transmissão de dados

A transmissão de dados pode ser modelada da seguinte forma: há um *emissor de dados* que envia *palavras* de um conjunto predefinido $W$; o *canal de transmissão*, que é quem transmite os dados, na maioria das vezes não envia as palavras do modo como aparecem; ele *codifica* as palavras em algo e as envia. Em seguida, as *decodifica* para que o *receptor* de dados possa lê-las.




### 6.1. Formalidades

Vamos tornar tudo isso mais formal. Seja $W$ o conjunto de palavras que um emissor pode enviar.

Um *alfabeto* um simplesmente um conjunto finito de símbolos. Exemplos são $\{0, 1\}$ (bits) e o alfabeto usual $\{a, b, c, \ldots, z\}$. $\Sigma^*$ é o conjunto de strings finitas de $\Sigma$, ou seja, o conjunto de $n$-uplas ordenadas com entradas de $\Sigma$. Na verdade, o conteúdo de $\Sigma$ não importa; o que importa é o seu tamanho.

Um *código de um emissor de dados* é uma função $f\colon W \to \Sigma^*$. Estendemos a definição da função $f$ para concatenação de palavras com a propriedade bastante razoável

$$f(w_1 w_2 \ldots w_n) = f(w_1) f(w_2) \ldots f(w_n),$$

ou seja, $f$ codifica cada palavra e as envia na ordem em que chegam.

Para que possamos decodificar uma mensagem, não podemos ter $f(w) = f(w')$ para palavras $w$ e $w'$ diferentes, senão não podemos recuperar $w$ ou $w'$ a partir de $f(w)$. Posto de outra maneira, a função $f$ deve ser *injetora*.

Enfim, dadas duas strings $s = s_1 s_2 \dots s_m$ e $t = t_1 t_2 \dots t_n$, de $\Sigma^*$, dizemos que $s$ *é prefixo de* $t$ quando $m \leq n$ e $s_1 = t_1$, $s_2 = t_2$, $\dots$, $s_m = t_m$.

Quando transmitimos uma string grande de palavras do código. Juntando as palavras (que podem ser compostas de mais de um elemento de $\Sigma$), obtemos uma string maior de $\Sigma^*$. É bastante conveniente que, ao decodificarmos essa string, as palavras do código sejam imediatamente reconhecidas sem precisarmos ler o que vem adiante, ao contrário de, por exemplo, transmitir `'carro...'` sem termos dúvida se a palavra que vem é `'carro'` ou `'carrossel'`. Ou seja, duas palavras distintas do código $f(w)$ e $f(w')$ não podem ser sufixo uma da outra. Códigos desse tipo são chamados *códigos instantâneos*.

### 6.2. Códigos instantâneos podem existir tanto como códigos!

Nosso primeiro resultado é uma surpresa: fixado o alfabeto $\Sigma$, as condições de existência de um código qualquer e de um código instantâneo são as mesmas!

**Desigualdade de Kraft.** *Seja $W$ o conjunto de palavras de um emissor de dados, com $|W| = m$, e seja $\Sigma$ um alfabeto. Então existe um* **código instantâneo** *$f\colon W \to \Sigma^*$, sendo $c_i$ o comprimento de $f(w_i)$, se e somente se,*

$$\sum_{i=1}^{m} \frac{1}{n^{c_i}} \leq 1$$

**Desigualdade de McMillan.** *Seja $W$ o conjunto de palavras de um emissor de dados, com $|W| = m$, e seja $\Sigma$ um alfabeto. Então existe um* **código** *$f\colon W \to \Sigma^*$, sendo $c_i$ o comprimento de $f(w_i)$, se e somente se,*

$$\sum_{i=1}^{m} \frac{1}{n^{c_i}} \leq 1$$

Vamos demonstrar os dois teoremas simultaneamente.

**Demonstração**

Provaremos a metade "difícil" de cada teorema. Isto é, provaremos que

(i) Se $c_1, c_2, \dots, c_m$ são tais que

$$\sum_{i=1}^{m} \frac{1}{n^{c_i}} \leq 1$$

então existe um código instantâneo tal que $c_i$ é o comprimento de $f(w_i)$ e

(ii) Se existe um código $f\colon W \to \Sigma^*$, sendo $c_i$ o comprimento de $f(w_i)$ então

$$\sum_{i=1}^{m} \frac{1}{n^{c_i}} \leq 1$$

Não é difícil ver que se provarmos (i) e (ii) provamos ambos os teoremas.

**Demonstração de (i)**

Suponha que $c_1, c_2, \dots, c_m$ são tais que

$$\sum_{i=1}^{m} \frac{1}{n^{c_i}} \leq 1$$

e seja $c$ o maior dos $c_i$'s.

Sendo $t_i$ a quantidade de palavras do código com tamanho $i$, podemos reescrever a desigualdade acima como

$$\sum_{i=1}^{c} \frac{t_i}{n^i} \leq 1$$

Multiplicando tudo por $n^c$, obtemos

$$\begin{aligned}
& n^c \geq t_1 n^{c-1} + t_2 n^{c-2} + \cdots + t_{c-1} n + t_c \\
\implies & n^c \geq t_1 n^{c-1} + t_2 n^{c-2} + \cdots + t_{c-1} n \\
\iff & n^{c-1} \geq t_1 n^{c-2} + t_2 n^{c-3} + \cdots + t_{c-1} \\
\implies & n^{c-1} \geq t_1 n^{c-2} + t_2 n^{c-3} + \cdots + t_{c-2} n \\
\iff & n^{c-2} \geq t_1 n^{c-3} + t_2 n^{c-4} + \cdots + t_{c-2} \\
& \vdots \\
& n \geq t_1
\end{aligned}$$

Note que a última desigualdade é óbvia porque, como há $n$ elementos no alfabeto $\Sigma$, a quantidade de palavras com um elemento não pode passar de $n$, pela injetividade de $f$. Mas todas as outras não são óbvias! Vamos reescrevê-las como

$$\begin{aligned}
t_c &\leq n^c - t_1 n^{c-1} - t_2 n^{c-2} - \cdots - t_{c-1} n \\
t_{c-1} &\leq n^{c-1} - t_1 n^{c-2} - t_2 n^{c-3} - \cdots + t_{c-2} n \\
t_{c-2} &\leq n^{c-2} - t_1 n^{c-3} - t_2 n^{c-4} - \cdots + t_{c-3} n \\
&\vdots \\
t_2 &\leq n^2 - t_1 n
\end{aligned}$$

Agora, utilizando as desigualdades acima, vamos construir um código instantâneo com $t_i$ palavras de tamanho $i$. Escolha $t_1$ elementos quaisquer de $W$ e $t_1$ palavras de tamanho 1, e as associe de qualquer maneira.

A condição necessária e suficiente para que o código seja instantâneo é as $t_2$ palavras de tamanho 2 não poderem começar com essas $t_1$ palavras. Mas basta então escolher quaisquer $t_2$ palavras dentre as $n(n-t_1)$ que sobraram ($n-t_1$ escolhas para o primeiro símbolo e $n$ para o segundo), e isso é possível porque $t_2 \leq n^2 - t_1 n = n(n-t_1)$.

Continuando, podemos escolher as $t_3$ palavras de tamanho 3 entre as $n(n^2 - t_1 n - t_2)$ que sobram (tiramos $t_2$ e $t_1 n$ das palavras de tamanho 2 que não podem ser sufixos das palavras de tamanho 3), o que é possível porque $t_3 \leq n^3 - t_1 n^2 - t_2 n = n(n^2 - t_1 n - t_2)$. Prosseguimos da mesma maneira, e é possível construir, por indução, o código instantâneo. ∎

**Demonstração de (ii)**

Utilizaremos idéias de funções geratrizes para provar (ii). Como antes, sejam $c_i$ o tamanho de $f(w_i)$ e $c$ o maior dos $c_i$'s. Devemos, então, provar que

$$n^{-c_1} + n^{-c_2} + \cdots + n^{-c_m} \leq 1$$

Seja

$$F(t) = (n^{-c_1} + n^{-c_2} + \cdots + n^{-c_m})^t$$

Ao abrirmos essa expressão, obtemos

$$F(t) = \sum_{s=1}^{tc} C_s n^{-s}$$

O que é esse coeficiente $C_s$? Ou melhor, como obtemos $n^{-s}$? Multiplicando termos cujos expoentes somam $s$. Mas cada termo representa uma palavra $f(w_{c_i})$ de tamanho $c_i$, ou seja, cada $n^{-s}$ associa-se a uma seqüência de $s$ elementos de $\Sigma$ que corresponde a uma seqüência de $f(w)$'s. Portanto $C_s$ é a quantidade de seqüências de $s$ elementos de $\Sigma$ geradas concatenando $f(w)$'s. Pela unicidade de códigos, duas seqüências geradas por $f(w)$'s diferentes não podem ser iguais, logo $C_s$ deve ser no máximo igual à quantidade de palavras de $s$ elementos de $\Sigma$, ou seja, $C_s \leq n^s$. Portanto

$$F(t) \leq \sum_{s=1}^{tc} n^s \cdot n^{-s} = tc$$

Tirando a raiz $t$-ésima nos dois membros, obtemos

$$n^{-c_1} + n^{-c_2} + \cdots + n^{-c_m} \leq (tc)^{1/t}$$

Fazendo $t$ tender a infinito, e observando que $\lim_{t\to+\infty}(tc)^{1/t} = \lim_{t\to+\infty} e^{\ln(tc)/t} = 1$, obtemos a desigualdade desejada. ∎

Portanto podemos supor sem perdas que todos os códigos estudados são instantâneos!

### 6.3. A pergunta que todo professor de Matemática teme: "para que serve isso?"

Nesse caso, os teoremas acima são extremamente úteis para encontrar um limitante teórico superior para a compressão média de dados; tal limitante é demonstrado pelo *teorema da codificação sem ruído*, provado pelo pai da Teoria da Informação, Claude Shannon.

## 7. Referências bibliográficas


[1] Se você quer encontrar problemas, vá ao Mathlinks. `http://www.mathlinks.ro/`

[2] O Edmilson disse uma vez em Matemática, quando o negócio está russo, está bom! A revista Kvant é um grande exemplo disso. O exemplo de contagem e a parte sobre seqüências hereditárias foram retirados da coletânea *Kvant Selecta: Combinatorics, I*, editado por Serge Tabachnikov.

[3] A seção sobre a seqüência de Morse foi retirada de *The Lure of the Integers*, de Joe Roberts. Esse livro tem um capítulos separados por número inteiro positivo, mostrando diversas propriedades dele. Essa é uma propriedade do número 3.

[4] Mais propriedades da seqüência de Morse? Veja o artigo *The ubiquitous Prouhet-Thue-Morse sequence*, de Jena-Paul Allouche e Jeffrey Shallit. Eu achei na Wikipedia!

[5] A parte de Teoria da Informação foi retirada de *The Mathematics of Coding Theory*, de Paul Garrett, uma ótima introdução ao assunto e também a códigos corretores de erros. O texto é bem didático, com muitos exemplos, e toca nos pontos mais relevantes desse rico ramo da Matemática Aplicada.