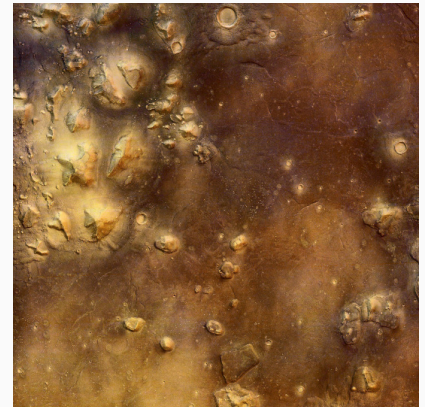


Teoria da Informação e Codificação

Teoria da informação: Codificação de fonte



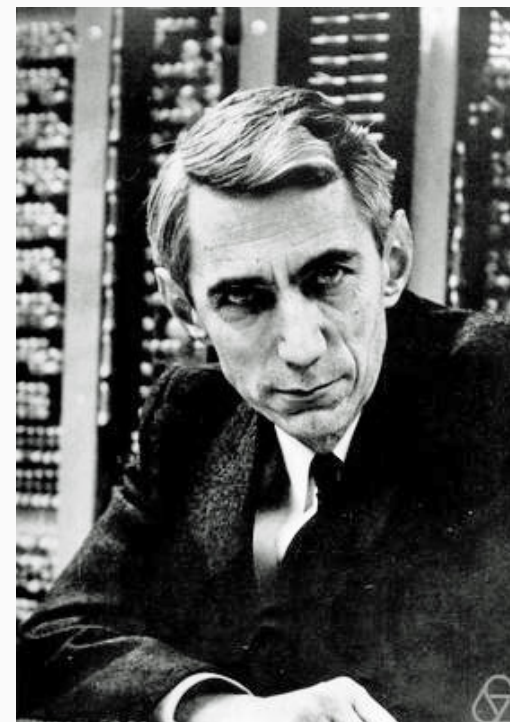
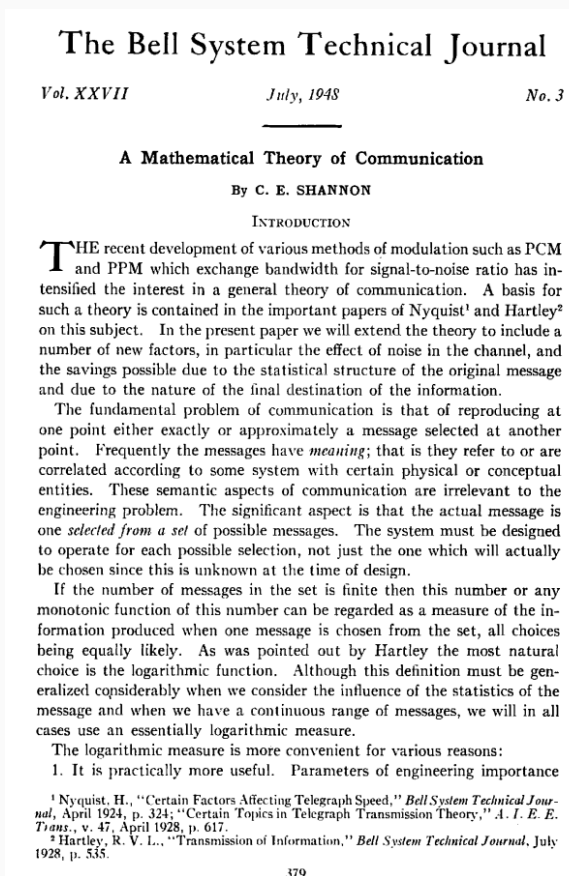
Prof. Roberto Wanderley da Nóbrega

Instituto Federal de Santa Catarina

Introdução

Uma teoria matemática da comunicação

Shannon: *A Mathematical Theory of Communication*. Bell Labs, 1948.



Claude E. Shannon (1916–2001).

Information theorist's coat of arms

Codificação de fonte

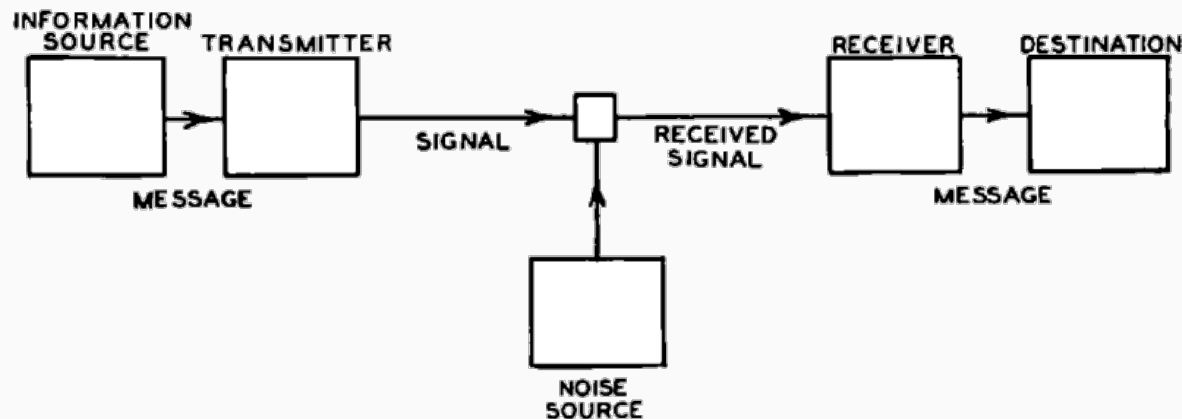
Representar com **fidelidade** uma *fonte de informação* através de uma *sequência de bits* com a **menor taxa possível**.

Conceito chave: **Entropia da fonte**.

Codificação de canal

Transmitir com **confiabilidade** uma *sequência de bits* por um *canal de comunicação* com a **maior taxa possível**.

Conceito chave: **Capacidade do canal**.

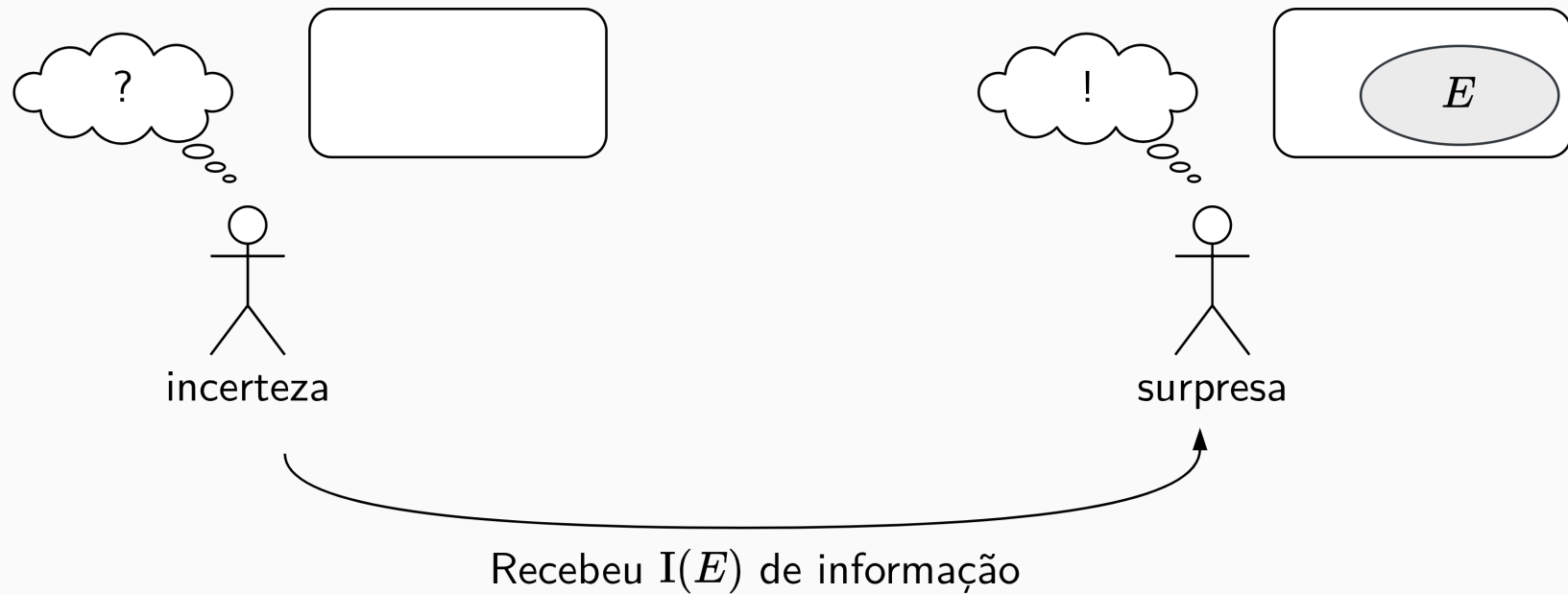


Fonte: Shannon–Weaver [1].

Medida de informação

Como quantificar a informação?

Seja E um evento de um experimento probabilístico.



Como quantificar a informação recebida ao se observar E ?

1. Um evento que ocorre com probabilidade 1 não fornece informação.

Se $\mathbb{P}[E] = 1$, então $I(E) = 0$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.
Se $\mathbb{P}[E] = 1$, então $I(E) = 0$.
2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.
Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $I(E_1) > I(E_2)$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.

Se $\mathbb{P}[E] = 1$, então $I(E) = 0$.

2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.

Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $I(E_1) > I(E_2)$.

3. A informação recebida ao observar dois eventos independentes é a soma das suas informações individuais.

Se $\mathbb{P}[E_1 \cap E_2] = \mathbb{P}[E_1] \mathbb{P}[E_2]$, então $I(E_1 \cap E_2) = I(E_1) + I(E_2)$.

Axiomas da informação

1. Um evento que ocorre com probabilidade 1 não fornece informação.

Se $\mathbb{P}[E] = 1$, então $I(E) = 0$.

2. Quanto mais improvável um evento, maior a informação recebida ao observá-lo.

Se $\mathbb{P}[E_1] < \mathbb{P}[E_2]$, então $I(E_1) > I(E_2)$.

3. A informação recebida ao observar dois eventos independentes é a soma das suas informações individuais.

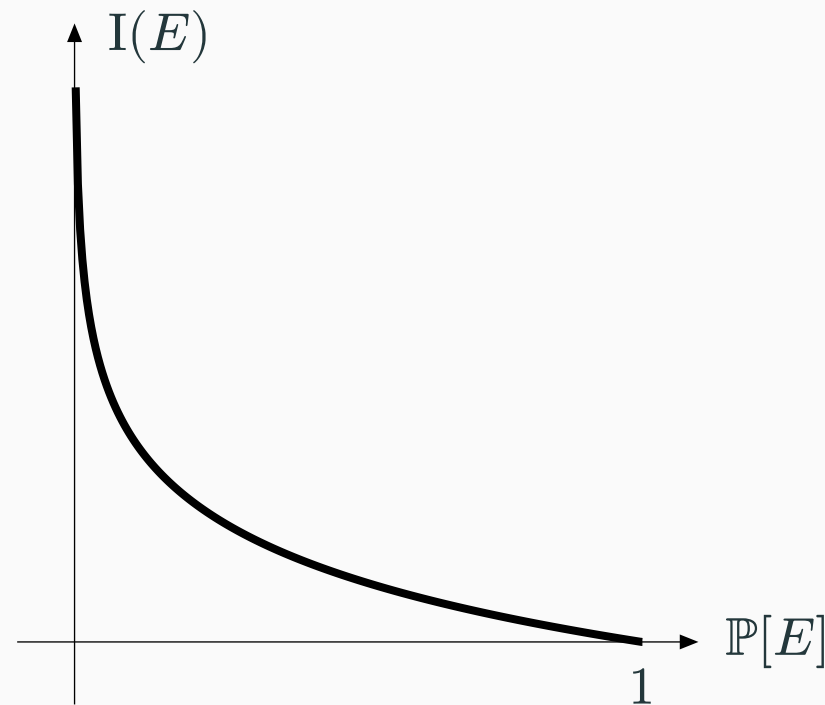
Se $\mathbb{P}[E_1 \cap E_2] = \mathbb{P}[E_1] \mathbb{P}[E_2]$, então $I(E_1 \cap E_2) = I(E_1) + I(E_2)$.

A função I que satisfaz os três axiomas é **única**, a menos de uma constante de escala.

Informação de Shannon

Medida de informação proposta por Shannon:

$$I(E) = \log \frac{1}{\mathbb{P}[E]}$$



Unidades de informação

A base do logaritmo equivale à constante de escala e determina a *unidade* da informação. Lembre-se da fórmula de mudança de base:

$$\log_b x = \frac{1}{\log_a b} \log_a x.$$

Base	Unidade
2	bit ou shannon
3	trit
$e = 2.7183\dots$	nat
10	dígito decimal ou hartley

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

$$I(\text{die}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$$

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

$$I(\text{die}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$$

4. Está  em um semáforo com $\mathbb{P}[\text{red}] = 0.80$, $\mathbb{P}[\text{yellow}] = 0.01$, $\mathbb{P}[\text{green}] = 0.19$? E ? E .

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

$$I(\text{die}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$$

4. Está  em um semáforo com $\mathbb{P}[\text{red}] = 0.80$, $\mathbb{P}[\text{yellow}] = 0.01$, $\mathbb{P}[\text{green}] = 0.19$? E ? E .

$$I(\text{yellow}) = \log(1/0.01) = 6.64 \text{ bits}. \quad I(\text{red}) = \log(1/0.80) = 0.32 \text{ bits}. \quad I(\text{green}) = \log(1/0.19) = 2.40 \text{ bits}.$$

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

$$I(\text{die}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$$

4. Está  em um semáforo com $\mathbb{P}[\text{red}] = 0.80$, $\mathbb{P}[\text{yellow}] = 0.01$, $\mathbb{P}[\text{green}] = 0.19$? E ? E .

$$I(\text{yellow}) = \log(1/0.01) = 6.64 \text{ bits}. \quad I(\text{red}) = \log(1/0.80) = 0.32 \text{ bits}. \quad I(\text{green}) = \log(1/0.19) = 2.40 \text{ bits}.$$

5. Uma carta retirada do baralho é de ? Que é uma dama? Que é a dama de .

Exemplos

Qual a quantidade de informação recebida ao se observar que...

1. Caiu cara após o lançamento de uma moeda honesta? E coroa?

$$I(\text{cara}) = \log 2 = 1 \text{ bit}, \quad I(\text{coroa}) = \log 2 = 1 \text{ bit}.$$

2. Caiu cara após o lançamento de uma moeda viciada com $\mathbb{P}[\text{cara}] = \frac{1}{4}$? E coroa?

$$I(\text{cara}) = \log 4 = 2 \text{ bits}, \quad I(\text{coroa}) = \log(4/3) = 0.42 \text{ bits}.$$

3. Caiu  após o lançamento de um dado honesto?

$$I(\text{die}) = \log 6 = 2.58 \text{ bits} = 1 \text{ dígito senário}.$$

4. Está  em um semáforo com $\mathbb{P}[\text{red}] = 0.80$, $\mathbb{P}[\text{yellow}] = 0.01$, $\mathbb{P}[\text{green}] = 0.19$? E ? E .

$$I(\text{yellow}) = \log(1/0.01) = 6.64 \text{ bits}, \quad I(\text{red}) = \log(1/0.80) = 0.32 \text{ bits}, \quad I(\text{green}) = \log(1/0.19) = 2.40 \text{ bits}.$$

5. Uma carta retirada do baralho é de ? Que é uma dama? Que é a dama de .

$$I(\text{spade}) = \log 4 = 2 \text{ bits}, \quad I(\text{Q}) = \log 13 = 3.7 \text{ bits}, \quad I(\text{Qspade}) = \log 52 = 5.7 \text{ bits}.$$

6. Os quatro últimos dígitos do número de telefone do seu amigo são 5678?

6. Os quatro últimos dígitos do número de telefone do seu amigo são 5678?

$$I(5678) = \log 10^4 = 13.29 \text{ bits} = 4 \text{ dígitos decimais.}$$

6. Os quatro últimos dígitos do número de telefone do seu amigo são 5678?

$I(5678) = \log 10^4 = 13.29 \text{ bits} = 4 \text{ dígitos decimais.}$

7. O sol nasceu hoje?

Exemplos

6. Os quatro últimos dígitos do número de telefone do seu amigo são 5678?

$I(5678) = \log 10^4 = 13.29 \text{ bits} = 4 \text{ dígitos decimais.}$

7. O sol nasceu hoje?

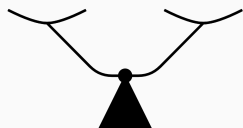
Para Laplace, $\log(1/0.9999) = 0.00014 \text{ bits.}$

Problema das 12 moedas

- Temos 12 moedas:



- Sabemos que exatamente uma é falsa (pode ser ou mais leve ou mais pesada).
- Dispomos de uma balança de braços:



- Como descobrir qual é a moeda falsa (e se é mais leve ou mais pesada) com o menor número de pesagens?

Problema das 12 moedas

- Variável aleatória que queremos descobrir: $Y \in \{1^+, 1^-, 2^+, 2^-, \dots, 12^+, 12^-\}$.
- Assumindo que Y é uniforme, temos:

$$I(Y = y) = \log 24 = 4.58 \text{ bits} = 2.89 \text{ trits}$$

para todo y .

- Resultado da pesagem i : $X_i \in \{L, B, R\}$.
- A cada pesagem i , ganhamos $I(X_i = x_i)$ de informação.
- Heurística: maximizar o pior caso (menor valor possível de $I(X_i = x_i)$).

Proposta para a primeira pesagem: 1×1

- Pesagem: (🏛️1) vs (🏛️2)

Y	1^+	1^-	2^+	2^-	3^+	3^-	...	12^+	12^-
X_1	L	R	R	L	B	B	...	B	B

- Informação recebida (em trits):

x	L	B	R
$\mathbb{P}[X_1 = x]$	2/24	20/24	2/24
$I(X_1 = x)$	2.26	0.17	2.26

- Lembrando a incerteza inicial: $I(Y = y) = 2.89$ trits.
No melhor caso ($x = L$ ou R), reduzimos a incerteza para 1 bit! (por quê?)
Mas no pior caso ($x = B$), a incerteza fica em 2.73 trits.

Proposta para a primeira pesagem: 6×6

- Pesagem: ( 1  2  3  4  5  6) vs ( 7  8  9  10  11  12)

Y	1^+	1^-	...	6^+	6^-	7^+	7^-	...	12^+	12^-
X_1	L	R	...	L	R	R	L	...	R	L

- Informação recebida (em trits):

x	L	B	R
$\mathbb{P}[X_1 = x]$	1/2	0	1/2
$I(X_1 = x)$	0.63	∞	0.63

- Lembrando a incerteza inicial: $I(Y = y) = 2.89$ trits.
Nos casos $x = L$ e $x = R$, reduzimos a incerteza a 2.26 trits.
O caso $x = B$ é impossível.

Proposta para a primeira pesagem: 4×4

- Pesagem: (🏛️1 🏛️2 🏛️3 🏛️4) vs (🏛️5 🏛️6 🏛️7 🏛️8)

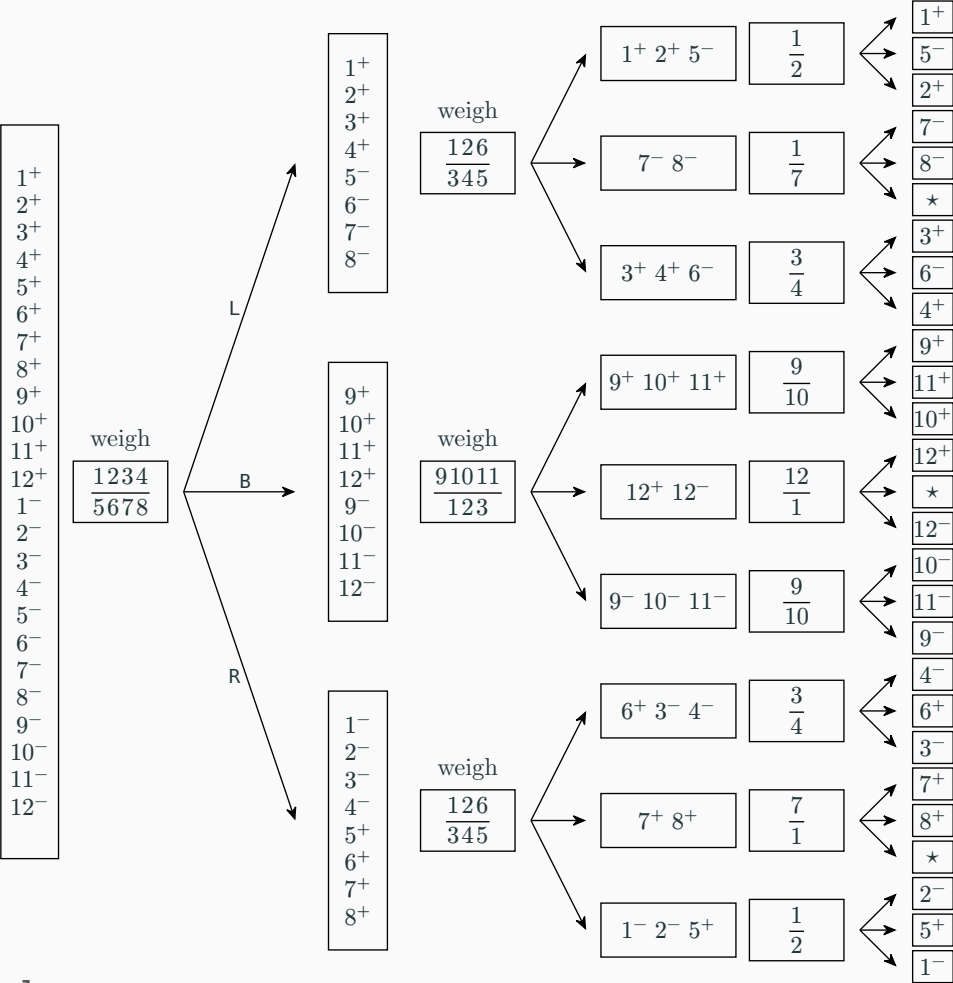
Y	1^+	1^-	...	4^+	4^-	5^+	5^-	...	8^+	8^-	9^+	9^-	...	12^+	12^-
X_1	L	R	...	L	R	R	L	...	R	L	B	B	...	B	B

- Informação recebida (em trits):

x	L	B	R
$\mathbb{P}[X_1 = x]$	1/3	1/3	1/3
$I(X_1 = x)$	1	1	1

- Lembrando a incerteza inicial: $I(Y = y) = 2.89$ trits.
Em qualquer caso ($x = L$ ou R ou B), reduzimos a incerteza para 1.89 trits.

Problema das 12 moedas: Solução



Fonte: Adaptado de MacKay [3].

Entropia

Entropia de uma variável aleatória

A entropia é a *quantidade de informação média* de uma variável aleatória.

Definição: Seja X uma va discreta com alfabeto $\mathcal{X}$ e pmf p . A **entropia** de X é dada por

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \mathbb{I}(X = x) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}.$$

Observação: Como a entropia depende apenas da distribuição p , utiliza-se também a notação $H(X) = H(p)$.

Exercício

Determine a entropia da variável aleatória X , considerando:

(a) $\mathcal{X} = \{\text{red circle}, \text{yellow circle}, \text{green circle}\}$ e $p = [0.80, 0.01, 0.19]$.

(b) $\mathcal{X} = \{\text{1 dot}, \text{2 dots}, \text{3 dots}, \text{4 dots}, \text{5 dots}, \text{6 dots}\}$ e $p = [\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$.

Exercício

Determine a entropia da variável aleatória X , considerando:

(a) $\mathcal{X} = \{\text{red}, \text{yellow}, \text{green}\}$ e $p = [0.80, 0.01, 0.19]$.

(b) $\mathcal{X} = \{\square, \square, \square, \square, \square, \square\}$ e $p = [\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$.

Resposta:

(a) 0.78 bits.

(b) $\log(6) = 2.58$ bits = 1 dígito senário.

Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{0, 1\}$ e $p = [1 - \alpha, \alpha]$.

Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{0, 1\}$ e $p = [1 - \alpha, \alpha]$.

- Por exemplo, $p = [\frac{1}{2}, \frac{1}{2}]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit.}$$

Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{0, 1\}$ e $p = [1 - \alpha, \alpha]$.

- Por exemplo, $p = [\frac{1}{2}, \frac{1}{2}]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit.}$$

- Por exemplo, $p = [\frac{1}{4}, \frac{3}{4}]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = \frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3} = 0.81 \text{ bits.}$$

Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{0, 1\}$ e $p = [1 - \alpha, \alpha]$.

- Por exemplo, $p = [\frac{1}{2}, \frac{1}{2}]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = \log 2 = 1 \text{ bit.}$$

- Por exemplo, $p = [\frac{1}{4}, \frac{3}{4}]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = \frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3} = 0.81 \text{ bits.}$$

- Por exemplo, $p = [0, 1]$:

$$H(X) = p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} = 0 \log \frac{1}{0} + 1 \log 1 = 0 \text{ bits.}$$

Parênteses: $0 \log(1/0) = 0$

Fato:

$$\lim_{x \rightarrow 0^+} x \log \frac{1}{x} = 0.$$

Demonstração: Considere log natural, sem perda de generalidade. Temos que

$$\begin{aligned} \lim_{x \rightarrow 0^+} x \log \frac{1}{x} &= \lim_{x \rightarrow 0} \frac{\log 1/x}{1/x} \\ &= \lim_{x \rightarrow 0} \frac{-1/x}{-1/x^2} \\ &= \lim_{x \rightarrow 0} x = 0, \end{aligned}$$

onde a segunda igualdade segue da regra de L'Hôpital. ■

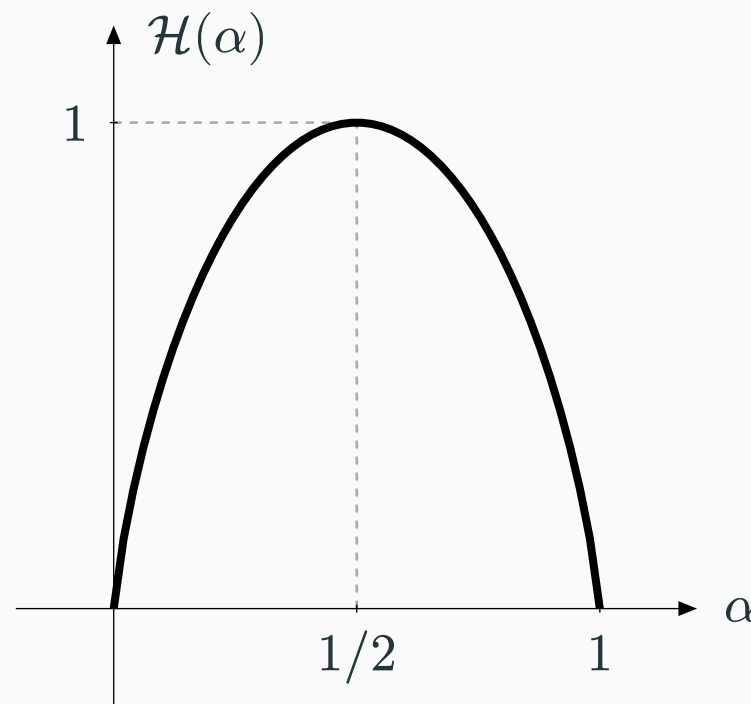
Entropia de uma Bernoulli

Seja $X \sim \text{Bernoulli}(\alpha)$, isto é, $\mathcal{X} = \{0, 1\}$ e $p = [1 - \alpha, \alpha]$.

Generalizando:

$$\begin{aligned} H(X) &= p(0) \log \frac{1}{p(0)} + p(1) \log \frac{1}{p(1)} \\ &= (1 - \alpha) \log \frac{1}{1 - \alpha} + \alpha \log \frac{1}{\alpha} \stackrel{\text{def}}{=} \mathcal{H}(\alpha), \end{aligned}$$

onde $\mathcal{H}$ é chamada de **função entropia binária**.

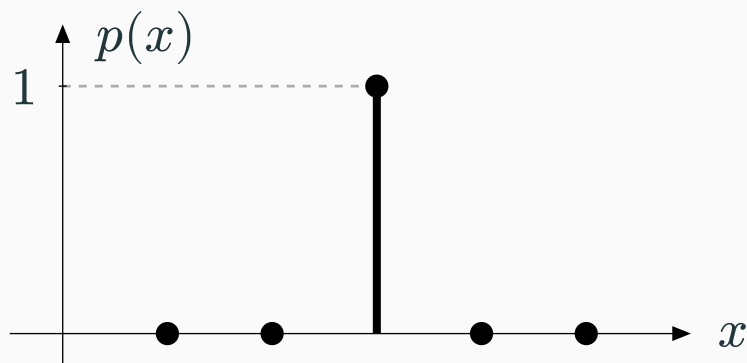


Propriedades da entropia

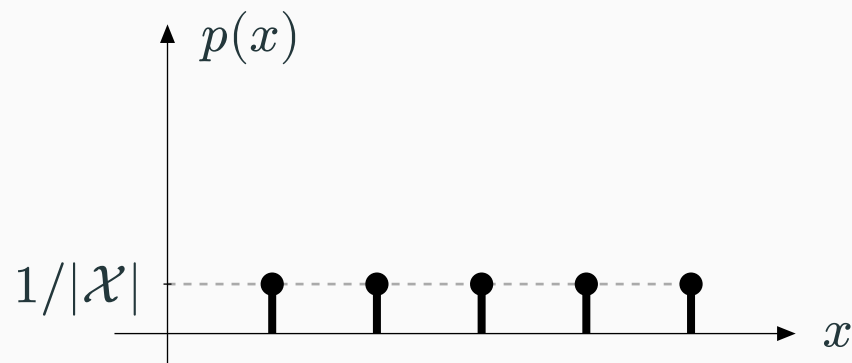
1. A entropia é uma função côncava de p .
2. A entropia satisfaz

$$0 \stackrel{(a)}{\leq} H(X) \stackrel{(b)}{\leq} \log|\mathcal{X}|,$$

com igualdade em (a) para X determinística e igualdade em (b) para X uniforme sobre $\mathcal{X}$.



Entropia zero.



Entropia máxima.

Exercício

Seja $X \sim \mathcal{N}(0, 1)$ a entrada de um quantizador escalar, uniforme, simétrico, do tipo mid-riser, com $L = 4$ níveis e passo $\Delta > 0$. Seja Y a saída do quantizador.

- (a) Determine a pmf de Y em função de Δ .
- (b) Obtenha a entropia de Y , em bits, em função de Δ . Esboce.
- (c) Qual valor de Δ maximiza a entropia de Y ?

Seja $X \sim \mathcal{N}(0, 1)$ a entrada de um quantizador escalar, uniforme, simétrico, do tipo mid-riser, com $L = 4$ níveis e passo $\Delta > 0$. Seja Y a saída do quantizador.

- (a) Determine a pmf de Y em função de Δ .
- (b) Obtenha a entropia de Y , em bits, em função de Δ . Esboce.
- (c) Qual valor de Δ maximiza a entropia de Y ?

Respostas:

- (a) $p_Y = [Q(\Delta), 1/2 - Q(\Delta), 1/2 - Q(\Delta), Q(\Delta)]$.
- (b) $H(Y) = 1 + \mathcal{H}(2Q(\Delta))$.
- (c) $\Delta^* = Q^{-1}(1/4) \approx 0.6745$ fornecendo $H(Y) = 2$ bits.

Compressão de fonte sem perdas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0X_1\cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0X_1\cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.
- Uma fonte discreta $\mathbf{X} = X_0X_1\cdots$ é dita ser **sem memória** (DMS) quando $X_n \stackrel{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas X para representar a DMS.

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0X_1\cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.
- Uma fonte discreta $\mathbf{X} = X_0X_1\cdots$ é dita ser **sem memória** (DMS) quando $X_n \stackrel{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas X para representar a DMS.

Exemplo: Considere uma DMS com alfabeto $\mathcal{X} = \{a, b, c\}$ e pmf $p = [0.80, 0.15, 0.05]$. Uma possível sequência amostra dessa fonte poderia ser

$\mathbf{x} = \text{ababaaaaacaaaaaaaaaacaabaaccaaaaaaaaaaaaaaaaaabaaaaba} \dots$

Fontes de informação discretas

- Uma **fonte discreta** é qualquer sequência aleatória $\mathbf{X} = X_0X_1\cdots$, que assume valores em um alfabeto finito $\mathcal{X}$. Os elementos de $\mathcal{X}$ são chamados de **letras** ou **símbolos**.
- Uma fonte discreta $\mathbf{X} = X_0X_1\cdots$ é dita ser **sem memória** (DMS) quando $X_n \stackrel{\text{iid}}{\sim} p$ para alguma pmf p sobre $\mathcal{X}$. Nesse caso, removeremos o negrito da notação e escreveremos apenas X para representar a DMS.

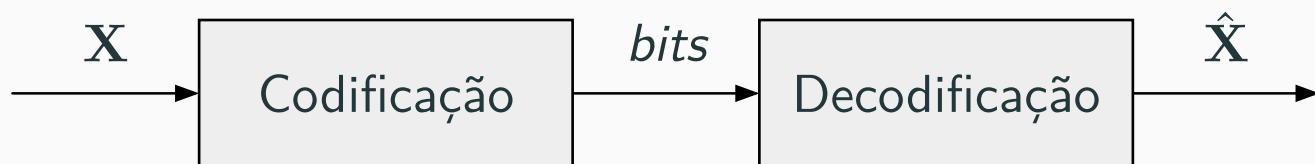
Exemplo: Considere uma DMS com alfabeto $\mathcal{X} = \{a, b, c\}$ e pmf $p = [0.80, 0.15, 0.05]$. Uma possível sequência amostra dessa fonte poderia ser

$\mathbf{x} = \text{ababaaaaacaaaaaaaaaacaabaaccaaaaaaaaaaaaaaaaaabaaaaba} \dots$

Observação: Há outros modelos de fontes discretas. Um modelo mais geral, descrito por Shannon em seu artigo seminal [1], consiste na *fonte discreta markoviana*. No entanto, nesta disciplina, estudaremos apenas as fontes discretas sem memória.

Compressão de fonte

Objetivo: Representar com *fidelidade* uma fonte de informação através de uma sequência de bits com a *menor taxa* possível.



- Compressão **sem perdas** (*lossless*): $\hat{\mathbf{X}} = \mathbf{X}$.
- Compressão **com perdas** (*lossy*): $\hat{\mathbf{X}} \approx \mathbf{X}$.

Exemplo: Minimizar $\mathbb{P}[\hat{\mathbf{X}} \neq \mathbf{X}]$ ou alguma *função de distorção* $d(\hat{\mathbf{X}}, \mathbf{X})$.

Nesta disciplina: Apenas os fundamentos de compressão sem perdas de fontes discretas.

Seja $\mathcal{X}$ um conjunto. Define-se:

- **Potência cartesiana:** $\mathcal{X}^n = \{x_0x_1\cdots x_{n-1} \mid x_i \in \mathcal{X}\}.$

Ou seja, $\mathcal{X}^n$ é o conjunto de todas as sequências com elementos em $\mathcal{X}$ de comprimento exatamente n .

Exemplo: $\{a, b, c\}^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}.$

- **Kleene star:** $\mathcal{X}^* = \mathcal{X}^0 \cup \mathcal{X}^1 \cup \mathcal{X}^2 \cup \dots.$

Ou seja, $\mathcal{X}^*$ é o conjunto de todas as sequências finitas com elementos em $\mathcal{X}$.

Exemplo: $\{a, b, c\}^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, \dots\}$

- **Kleene plus:** $\mathcal{X}^+ = \mathcal{X}^1 \cup \mathcal{X}^2 \cup \dots.$

Ou seja, $\mathcal{X}^+$ é o conjunto de todas as sequências finitas não vazias com elementos em $\mathcal{X}$.

Exemplo: $\{a, b, c\}^+ = \{a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, \dots\}$

Notação: A **sequência vazia** é denotada por ε .

Definição: Um **código de fonte símbolo a símbolo** para uma fonte discreta (qualquer) com alfabeto $\mathcal{X}$ é dado por um mapeamento

$$C : \mathcal{X} \rightarrow \{0, 1\}^+.$$

Exemplo: Os quatro códigos abaixo são todos para uma fonte com $\mathcal{X} = \{a, b, c, d\}$.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

Os elementos da imagem de C são chamados de **palavras-código**.

Extensão de um código símbolo a símbolo

Estende-se o código para qualquer sequência $\mathbf{x} \in \mathcal{X}^+$ *concatenativamente*. Abusando da notação, também chamaremos essa extensão de $C : \mathcal{X}^+ \rightarrow \{0, 1\}^+$.

Exemplo: Considere novamente os quatro códigos abaixo.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

Por exemplo, para $\mathbf{x} = \text{babaca}$:

- $C_1(\mathbf{x}) = 010001001000.$
- $C_2(\mathbf{x}) = 0110110001.$
- $C_3(\mathbf{x}) = 0100100110.$
- $C_4(\mathbf{x}) = 1010000.$

Códigos unicamente decodificáveis

- Um código é dito ser **unicamente decodificável** se o mapeamento estendido for injetivo:

$$\mathbf{x} \neq \mathbf{x}' \implies C(\mathbf{x}) \neq C(\mathbf{x}'),$$

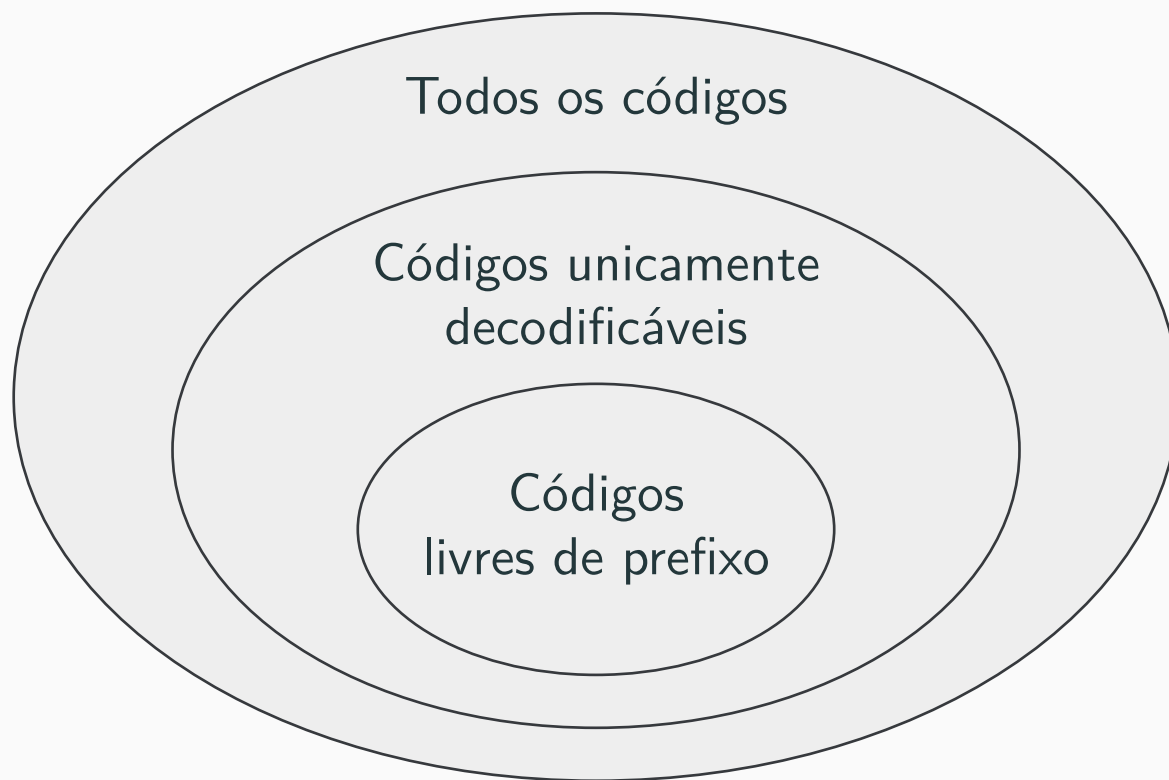
para todos $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^+$.

- No exemplo:
 - C_1 , C_2 e C_3 são unicamente decodificáveis.
 - C_4 não é unicamente decodificável.
Por exemplo, $C_4(aa) = 00 = C_4(c)$.
- Há diversos algoritmos para verificar se um dado código é unicamente decodificável. Talvez o mais famoso deles seja o *algoritmo de Sardinas–Patterson (1953)*.

- Um código é dito ser **livre de prefixo** (ou **instantâneo**) se nenhuma palavra-código é prefixo de outra.
- No exemplo:
 - C_1 e C_2 são livres de prefixo.
 - C_3 e C_4 não são livres de prefixo.
- Verificar se um código é livre de prefixo é trivial.
- Decodificar um código livre de prefixo também é trivial (e “instantâneo”).

Diagrama de Venn das classes de códigos

Fato: Livre de prefixo $\Rightarrow$ unicamente decodificável.



Desigualdade de Kraft–McMillan

Teorema. Seja $\ell(x)$ o número de bits em $C(x)$, para $x \in \mathcal{X}$. Então:

1. (*B. McMillan, 1956*) Todo código unicamente decodificável satisfaz

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{\ell(x)}} \leq 1. \quad (*)$$

2. (*L. G. Kraft, 1949*) Dados comprimentos $\ell(x) : x \in \mathcal{X}$ que satisfazem $(*)$, é possível construir um código livre de prefixo com tais comprimentos.

Verdadeiro ou falso?

- (a) Não existe um código com comprimentos 1, 2, 3, 3, 4 que seja unicamente decodificável.
- (b) Não existe um código com comprimentos 1, 2, 3, 3, 4 que seja livre de prefixo.
- (c) Todo código com comprimentos 2, 2, 3, 3, 4, 4, 5 é unicamente decodificável.
- (d) Existe um código com comprimentos 2, 2, 3, 3, 4, 4, 5 que seja livre de prefixo.
- (e) Existe um código com comprimentos 2, 2, 3, 3, 4, 4, 5 que seja unicamente decodificável.

Verdadeiro ou falso?

- (a) Não existe um código com comprimentos 1, 2, 3, 3, 4 que seja unicamente decodificável.
- (b) Não existe um código com comprimentos 1, 2, 3, 3, 4 que seja livre de prefixo.
- (c) Todo código com comprimentos 2, 2, 3, 3, 4, 4, 5 é unicamente decodificável.
- (d) Existe um código com comprimentos 2, 2, 3, 3, 4, 4, 5 que seja livre de prefixo.
- (e) Existe um código com comprimentos 2, 2, 3, 3, 4, 4, 5 que seja unicamente decodificável.

Resposta:

- (a) V. (b) V. (c) F. (d) V. (e) V.

O supermercado das palavras-código

0	00	000	0000
			0001
		001	0010
			0011
	01	010	0100
			0101
		011	0110
			0111
1	10	100	1000
			1001
		101	1010
			1011
	11	110	1100
			1101
		111	1110
			1111

Fonte: MacKay [3].

Dados comprimentos *em ordem não decrescente* que satisfazem a desigualdade de Kraft–McMillan, há uma maneira sistemática de “comprar as palavras-código no supermercado” de modo a construir um código livre de prefixo:

Compre da esquerda para a direita, pegando em cada prateleira o item mais acima ainda não riscado; cada compra risca o próprio item e toda a faixa à direita dele.

Equivalentemente: Para cada comprimento, escolhe-se a primeira sequência, em ordem lexicográfica, que não tem palavra-código anterior como prefixo.

O código obtido é chamado de **código canônico** associado aos comprimentos.

Existe um código livre de prefixo com comprimentos 2, 3, 3, 3, 3, 4, 6? Em caso afirmativo, construa um; em caso negativo, explique por que não existe.

Exercício

Existe um código livre de prefixo com comprimentos 2, 3, 3, 3, 3, 4, 6? Em caso afirmativo, construa um; em caso negativo, explique por que não existe.

Resposta:

Existe sim. Segue o canônico: $C = [00, 010, 011, 100, 101, 1100, 110100]$.

Limite da compressibilidade de Shannon

Comprimento médio de um código

O **comprimento médio** de um código C em uma DMS X com alfabeto $\mathcal{X}$ e pmf p é definido por

$$\bar{\ell}(C, X) = \mathbb{E}[\ell(X)] = \sum_{x \in \mathcal{X}} p(x) \ell(x),$$

onde $\ell(x)$ denota o número de bits em $C(x)$, para $x \in \mathcal{X}$.

Comprimento médio de um código

O **comprimento médio** de um código C em uma DMS X com alfabeto $\mathcal{X}$ e pmf p é definido por

$$\bar{\ell}(C, X) = \mathbb{E}[\ell(X)] = \sum_{x \in \mathcal{X}} p(x) \ell(x),$$

onde $\ell(x)$ denota o número de bits em $C(x)$, para $x \in \mathcal{X}$.

Naturalmente, queremos o menor comprimento médio possível.

Exercício

Considere as seguintes fontes discretas sem memória, ambas com alfabeto $\mathcal{X} = \{a, b, c, d\}$:

- X_1 , com pmf $p_1 = [0.6, 0.2, 0.1, 0.1]$ e
- X_2 , com pmf $p_2 = [0.3, 0.3, 0.2, 0.2]$.

Determine o comprimento médio de cada código abaixo em cada DMS.

(a) $C_1 = [00, 01, 10, 11]$.

(b) $C_2 = [0, 10, 110, 111]$.

Exercício

Considere as seguintes fontes discretas sem memória, ambas com alfabeto $\mathcal{X} = \{a, b, c, d\}$:

- X_1 , com pmf $p_1 = [0.6, 0.2, 0.1, 0.1]$ e
- X_2 , com pmf $p_2 = [0.3, 0.3, 0.2, 0.2]$.

Determine o comprimento médio de cada código abaixo em cada DMS.

(a) $C_1 = [00, 01, 10, 11]$.

(b) $C_2 = [0, 10, 110, 111]$.

Resposta:

(a) $\bar{\ell}(C_1, X_1) = 2.0$ bits/letra e $\bar{\ell}(C_1, X_2) = 2.0$ bits/letra.

(b) $\bar{\ell}(C_2, X_1) = 1.6$ bits/letra e $\bar{\ell}(C_2, X_2) = 2.1$ bits/letra.

Exemplo

Exemplo: DMS X com alfabeto $\mathcal{X} = \{a, b, c, d\}$ e pmf $p = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}]$.

x	$C_1(x)$	$C_2(x)$	$C_3(x)$	$C_4(x)$
a	00	1	0	0
b	01	01	01	1
c	10	000	011	00
d	11	001	0111	11

$$\bar{\ell}(C_1, X) = \frac{1}{2}(2) + \frac{1}{4}(2) + \frac{1}{8}(2) + \frac{1}{8}(2) = 2 \text{ bits/letra.}$$

$$\bar{\ell}(C_2, X) = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(3) = 1.75 \text{ bits/letra.}$$

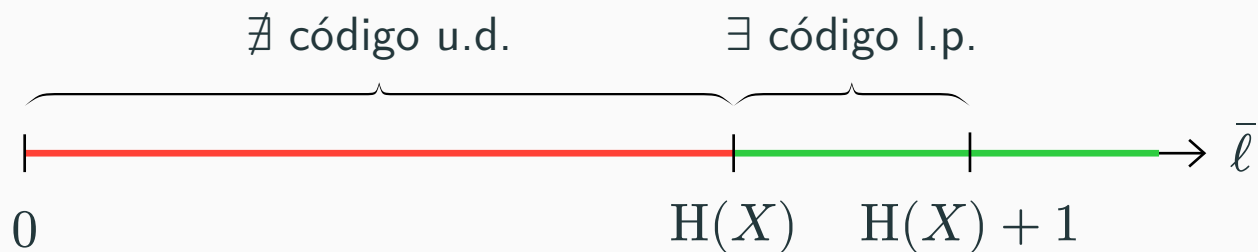
$$\bar{\ell}(C_3, X) = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(4) = 1.875 \text{ bits/letra.}$$

$$\bar{\ell}(C_4, X) = \frac{1}{2}(1) + \frac{1}{4}(1) + \frac{1}{8}(2) + \frac{1}{8}(2) = 1.25 \text{ bits/letra.}$$

Teorema da codificação de fonte sem perdas de Shannon

Teorema. (C. E. Shannon, 1948) Seja X uma DMS com entropia $H(X) = H(p)$.

1. Todo código C unicamente decodificável tem comprimento médio $\bar{\ell}(C, X) \geq H(X)$.
2. Existe um código C livre de prefixo com comprimento médio $\bar{\ell}(C, X) < H(X) + 1$.



O problema do “bit extra” do teorema de Shannon

Exemplo: Considere uma DMS X com alfabeto $\mathcal{X} = \{a, b, c\}$ e pmf $p = [0.98, 0.01, 0.01]$.

A entropia da fonte é

$$H(X) = 0.98 \log\left(\frac{1}{0.98}\right) + 0.01 \log\left(\frac{1}{0.01}\right) + 0.01 \log\left(\frac{1}{0.01}\right) = 0.161 \text{ bits/letra.}$$

Já o código ótimo ($a \mapsto 0$, $b \mapsto 10$, $c \mapsto 11$) tem comprimento médio

$$\bar{\ell}(C, X) = 0.98(1) + 0.01(2) + 0.01(2) = 1.02 \text{ bits/letra.}$$

- Comprimir 1000 letras da fonte com o código ótimo resulta em 1020 bits (em média).
- Mas temos esperanças de chegar a 161 bits (em média)!

Ideia: Agrupar as letras emitidas pela fonte em “superletras” (bloco de k letras).

$$x^{(k)} = (x_0, x_1, \dots, x_{k-1}).$$

Definição. Seja X uma DMS com alfabeto $\mathcal{X}$ e pmf p . A **extensão de k -ésima ordem** de X , denotada por $X^{(k)}$, tem alfabeto $\mathcal{X}^k$ e pmf dada por

$$p^{(k)}(x_0, x_1, \dots, x_{k-1}) = \prod_{i=0}^{k-1} p(x_i).$$

Observação: Um código para $X^{(k)}$ de comprimento médio $\bar{\ell}$ equivale a um código para X de comprimento médio $\bar{\ell}/k$.

Considere uma DMS X com alfabeto $\mathcal{X} = \{a, b, c\}$ e pmf $p = [0.98, 0.01, 0.01]$.

- (a) Determine a entropia de X .
- (b) Determine $X^{(2)}$, isto é, a extensão de segunda ordem de X .
- (c) Determine a entropia de $X^{(2)}$, em bits/superletra.

Exercício

Considere uma DMS X com alfabeto $\mathcal{X} = \{a, b, c\}$ e pmf $p = [0.98, 0.01, 0.01]$.

- (a) Determine a entropia de X .
- (b) Determine $X^{(2)}$, isto é, a extensão de segunda ordem de X .
- (c) Determine a entropia de $X^{(2)}$, em bits/superletra.

Resposta:

- (a) $H(X) \approx 0.16$ bits/letra.

(b)	$x^{(2)}$	aa	ab	ac	ba	bb	bc	ca	cb	cc
	$p^{(2)}(x^{(2)})$	0.9604	0.0098	0.0098	0.0098	0.0001	0.0001	0.0098	0.0001	0.0001

- (c) $H(X^{(2)}) \approx 0.32$ bits/superletra.

“Solução” do problema do bit extra

De fato, é possível provar que

$$H(X^{(k)}) = kH(X).$$

Aplicando o teorema de Shannon à fonte estendida:

$$H(X^{(k)}) \stackrel{(\forall)}{\leq} \bar{\ell} \stackrel{(\exists)}{<} H(X^{(k)}) + 1.$$

Substituindo $H(X^{(k)}) = kH(X)$ e dividindo por k :

$$H(X) \stackrel{(\forall)}{\leq} \frac{\bar{\ell}}{k} \stackrel{(\exists)}{<} H(X) + \frac{1}{k}.$$

No exemplo anterior: Para $k = 1000$, o limite inferior continua $H(X) = 0.161$ bits/letra, mas o limite superior diminui de 1.161 para 0.162 bits/letra.

“Solução” do problema do bit extra

De fato, é possível provar que

$$H(X^{(k)}) = kH(X).$$

Aplicando o teorema de Shannon à fonte estendida:

$$H(X^{(k)}) \stackrel{(\forall)}{\leq} \bar{\ell} \stackrel{(\exists)}{<} H(X^{(k)}) + 1.$$

Substituindo $H(X^{(k)}) = kH(X)$ e dividindo por k :

$$H(X) \stackrel{(\forall)}{\leq} \frac{\bar{\ell}}{k} \stackrel{(\exists)}{<} H(X) + \frac{1}{k}.$$

Observação: O problema do conceito de extensão de fonte é a *complexidade computacional*, que cresce exponencialmente com k .

Código de Huffman

São códigos símbolo a símbolo **ótimos** no sentido de minimizar o comprimento médio. São livres de prefixo e satisfazem a desigualdade de Kraft–McMillan com *igualdade*.

Algoritmo:

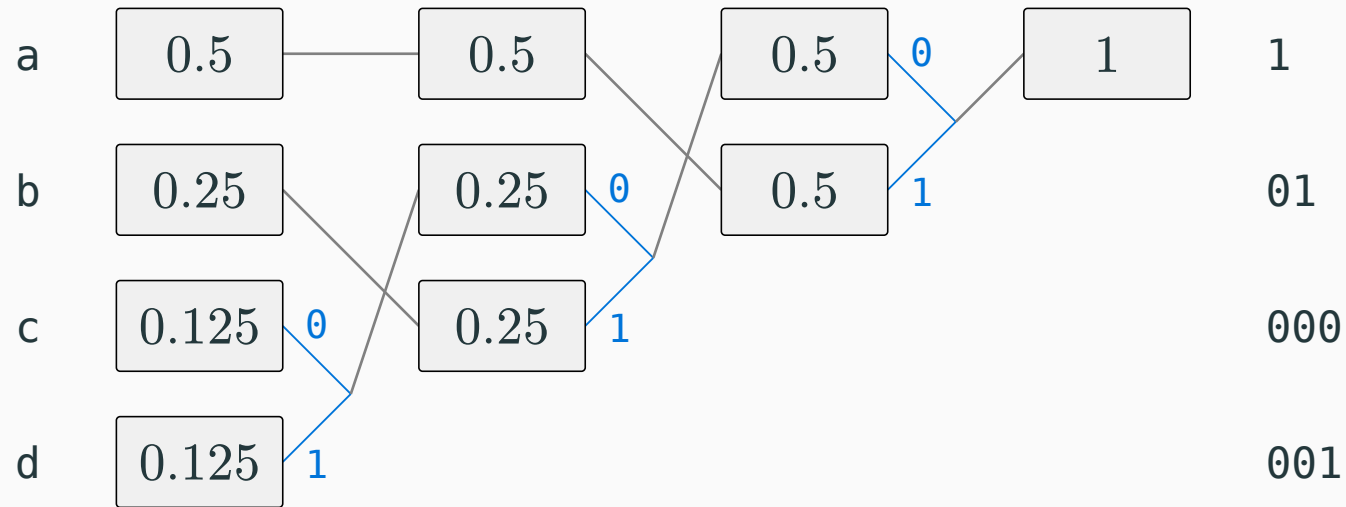
Consiste em iterar os passos abaixo até restar uma única letra, com probabilidade 1.

1. Ordene as letras em ordem decrescente de probabilidade.
2. Funda as duas letras menos prováveis em uma única, cuja probabilidade é a soma das duas.
3. Marque um dos ramos da fusão com 0 e o outro com 1.

Ao final, a palavra-código associada a cada letra original é a sequência de bits lida no caminho *inverso* (da raiz até a letra).

Exemplo 1

DMS com $\mathcal{X} = \{a, b, c, d\}$ e $p = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}]$, com $H(X) = 1.75$ bits/letra.



$$\bar{\ell} = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(3) = 1.75 \text{ bits/letra.}$$

- **A distribuição da fonte deve ser conhecida.**

Solução possível (nem sempre viável): Duas leituras dos dados: na primeira, estima-se a distribuição e constrói-se o código ótimo; na segunda, comprime-se a informação de fato.

Novo problema: O código deve ser anexado como metadado, o que gera *overhead*.

- **A distribuição da fonte deve ser conhecida.**

Solução possível (nem sempre viável): Duas leituras dos dados: na primeira, estima-se a distribuição e constrói-se o código ótimo; na segunda, comprime-se a informação de fato.

Novo problema: O código deve ser anexado como metadado, o que gera *overhead*.

- **A otimalidade vale apenas para o caso sem memória (DMS).**

Solução ingênua: Utilizar a extensão da fonte para modelar a dependência.

Novo problema: Complexidade computacional exponencial na ordem k da extensão.

Desafios do código de Huffman

- **A distribuição da fonte deve ser conhecida.**

Solução possível (nem sempre viável): Duas leituras dos dados: na primeira, estima-se a distribuição e constrói-se o código ótimo; na segunda, comprime-se a informação de fato.

Novo problema: O código deve ser anexado como metadado, o que gera *overhead*.

- **A otimalidade vale apenas para o caso sem memória (DMS).**

Solução ingênua: Utilizar a extensão da fonte para modelar a dependência.

Novo problema: Complexidade computacional exponencial na ordem k da extensão.

Os códigos de **Lempel–Ziv**, a seguir, atacam os dois desafios de uma só vez.

Códigos de Lempel–Ziv

Os **códigos de Lempel–Ziv** não necessitam do modelo probabilístico da fonte, nem assumem fonte sem memória (DMS). De fato, são *assintoticamente ótimos* para *fontes ergódicas*.

Ideia: Substituir subsequências por *referências* a ocorrências prévias da mesma subsequência.

- **LZ77** (*Ziv & Lempel, 1977*): Janela deslizante.
- **LZ78** (*Ziv & Lempel, 1978*): Dicionário explícito.

Ambos os esquemas têm como entrada uma sequência de letras de um alfabeto $\mathcal{X}$ e saída uma sequência de letras de um alfabeto $\mathcal{Y}$.

Por simplicidade, assumiremos alfabeto de saída binário: $\mathcal{Y} = \{0, 1\}$. A generalização para $\mathcal{Y}$ qualquer é imediata.

Considera uma **janela deslizante** sobre a mensagem. Esta janela é dividida em 2 partes:

- **Search buffer**, de comprimento S , e
- **Lookahead buffer**, de comprimento L .

Observação: O tamanho total da janela é $W = S + L$.

A cada passo do algoritmo, um token no formato (p, ℓ, x) é emitido, onde:

- $p \in [0 : S)$ é o **ponteiro** do match,
- $\ell \in [0 : L)$ é o **comprimento** do match, e
- $x \in \mathcal{X}$ é a letra da fonte após o match.

Ao final de cada passo, avança-se a janela $\ell + 1$ posições.

Observação: O conteúdo inicial do search buffer é arbitrário; aqui, consideraremos “zerado”.

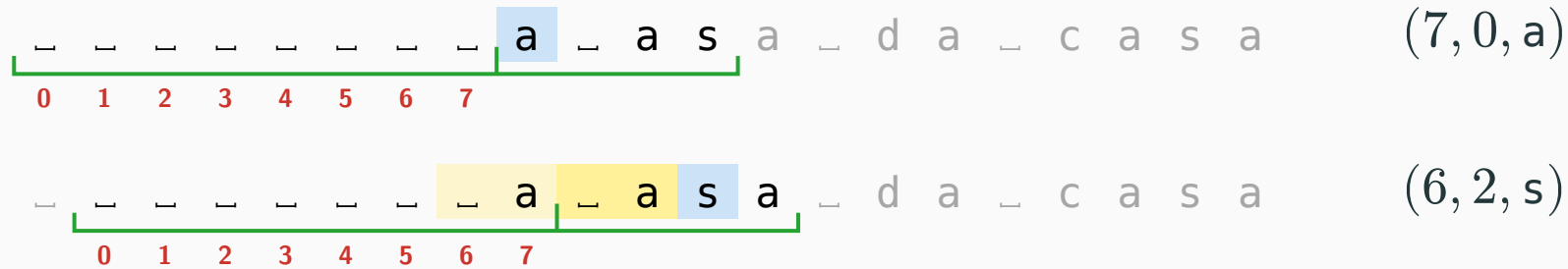
Exemplo LZ77: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Mensagem*: $x = a_asa_da_casa$.



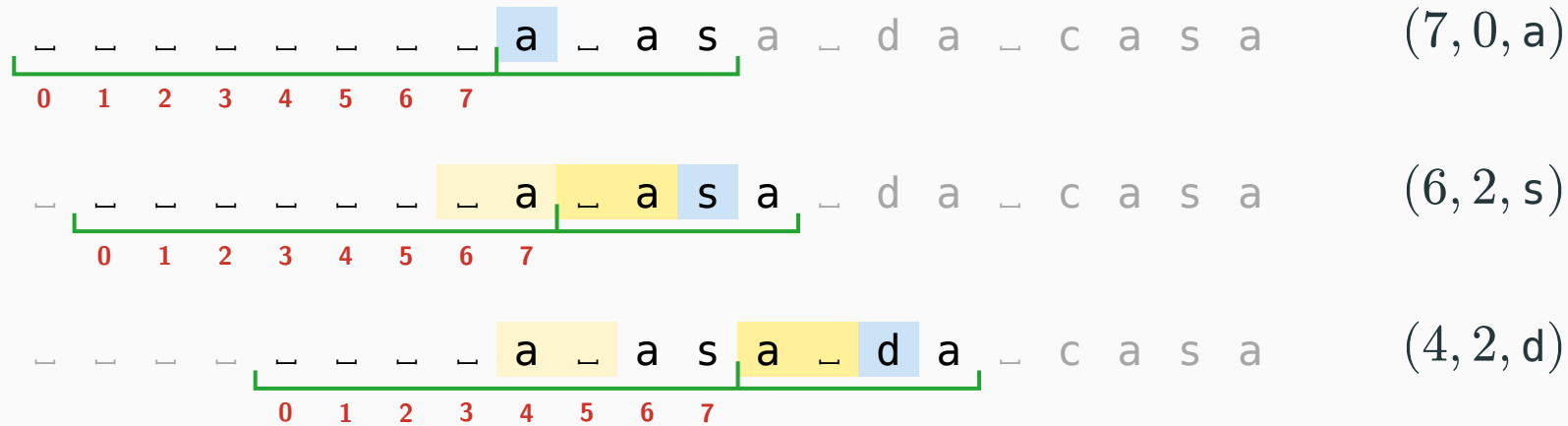
Exemplo LZ77: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Mensagem*: $x = a_asa_da_casa$.



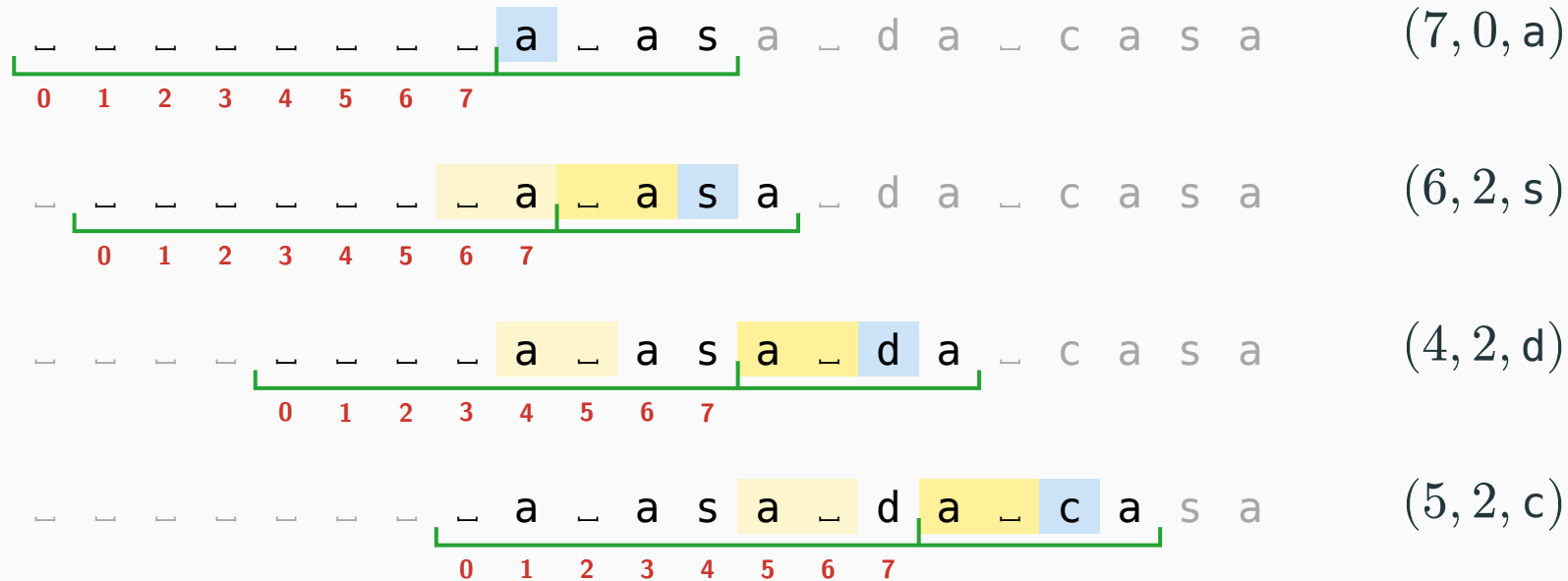
Exemplo LZ77: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Mensagem*: $x = a_asa_da_casa$.



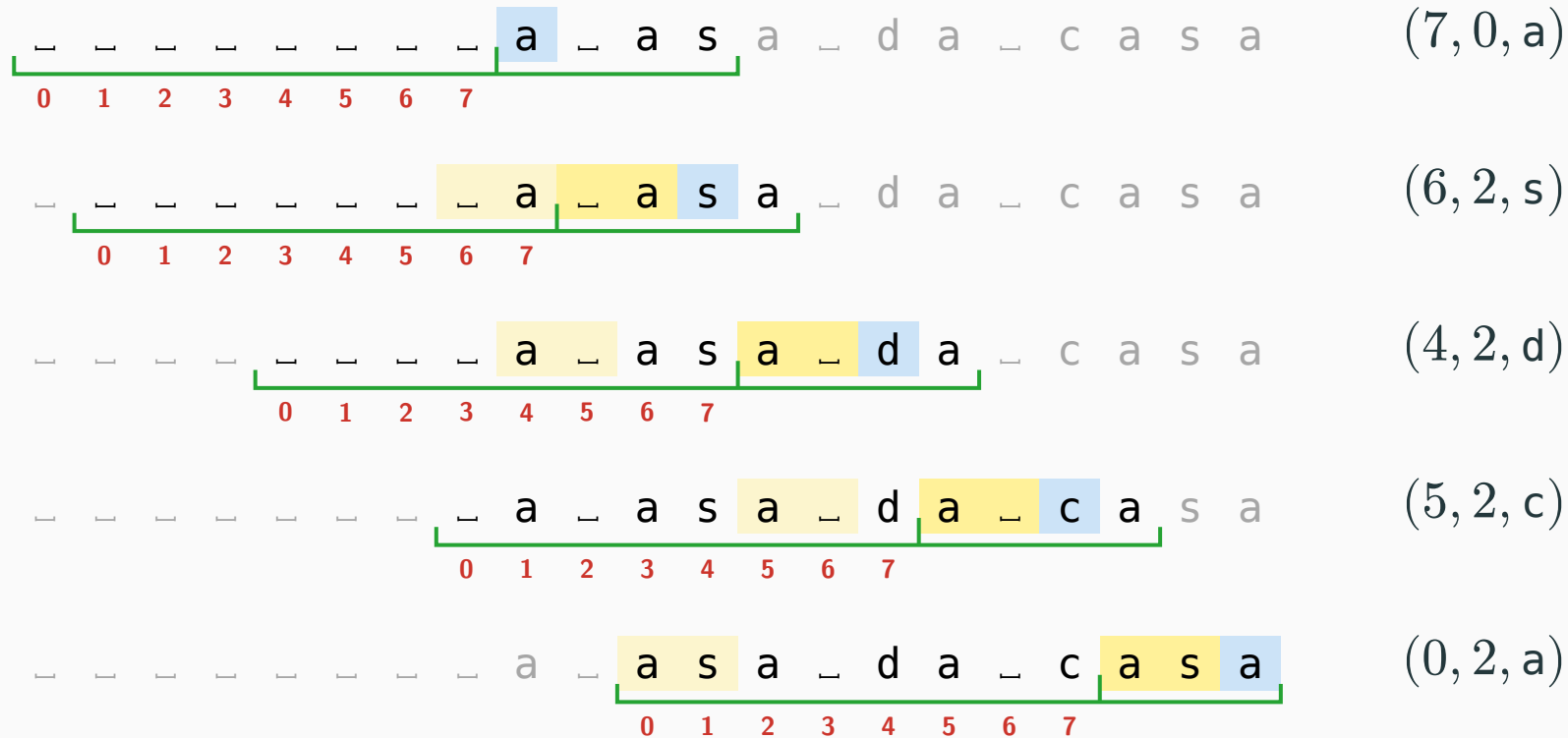
Exemplo LZ77: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Mensagem*: $x = a_asa_da_casa$.



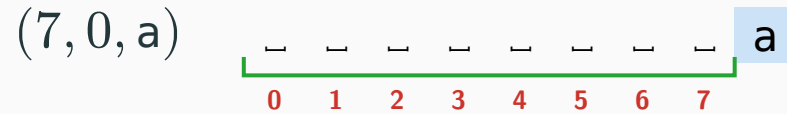
Exemplo LZ77: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Mensagem*: $x = a_asa_da_casa$.



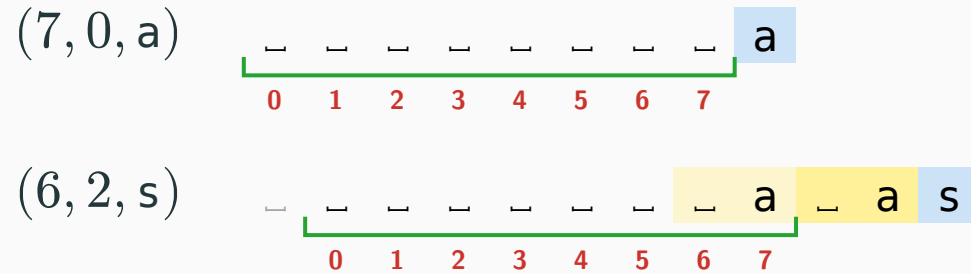
Exemplo LZ77: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Tokens*: $(7, 0, a)$, $(6, 2, s)$, $(4, 2, d)$, $(5, 2, c)$, $(0, 2, a)$.



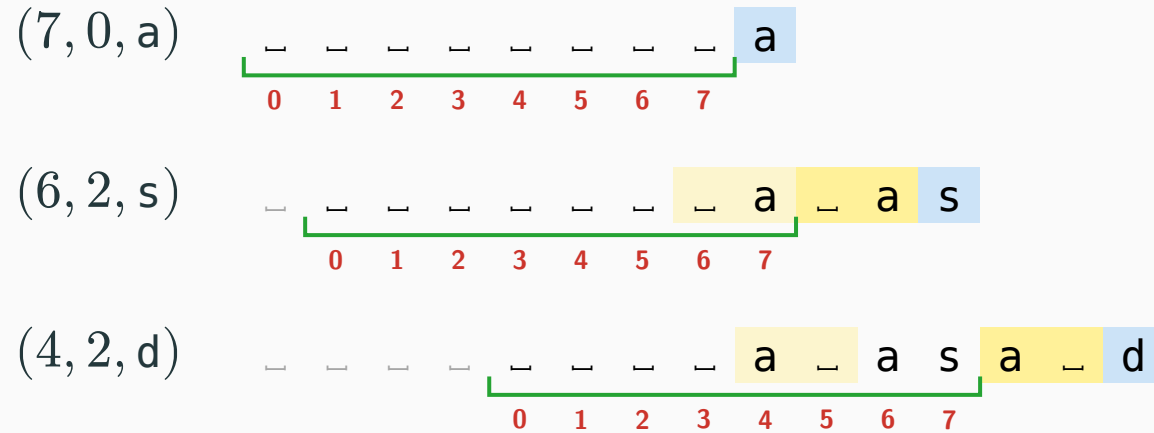
Exemplo LZ77: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Tokens*: $(7, 0, a)$, $(6, 2, s)$, $(4, 2, d)$, $(5, 2, c)$, $(0, 2, a)$.



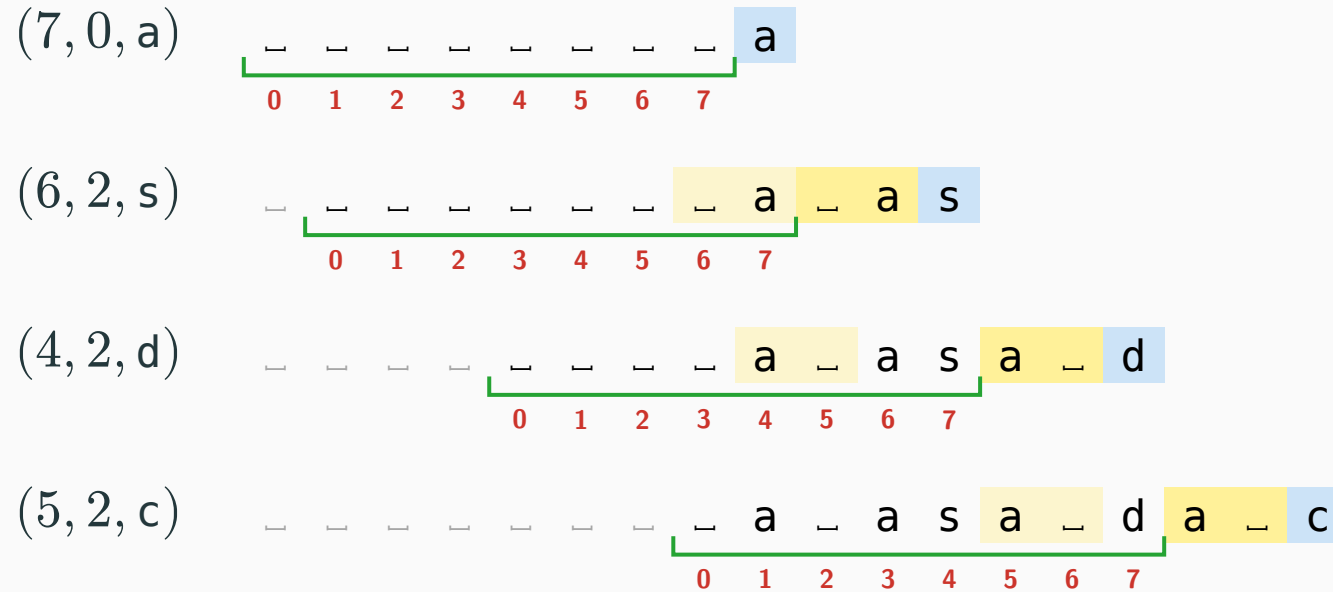
Exemplo LZ77: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Tokens*: $(7, 0, a)$, $(6, 2, s)$, $(4, 2, d)$, $(5, 2, c)$, $(0, 2, a)$.



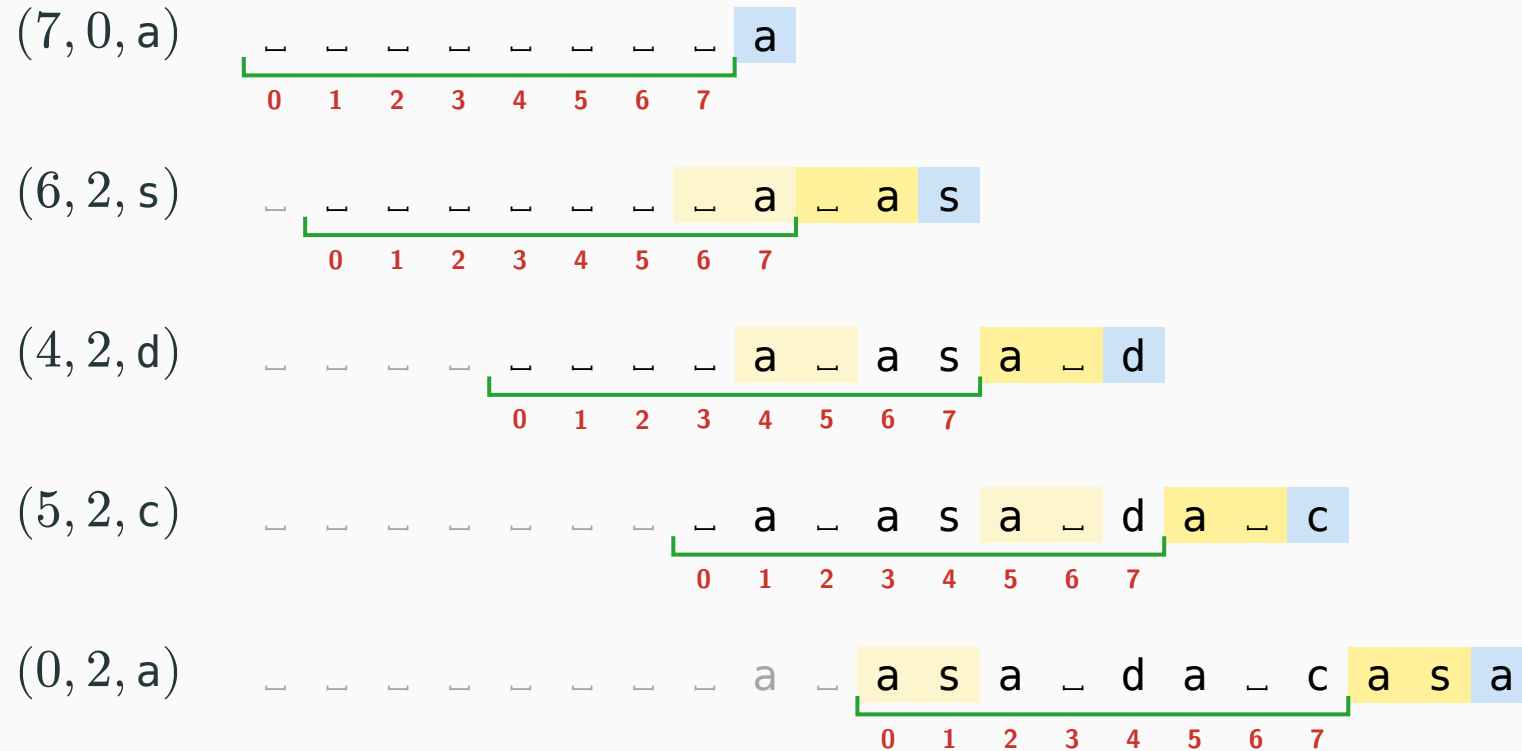
Exemplo LZ77: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela*: $S = 8$ e $L = 4$.
- *Tokens*: $(7, 0, a)$, $(6, 2, s)$, $(4, 2, d)$, $(5, 2, c)$, $(0, 2, a)$.



Exemplo LZ77: Decodificação

- *Alfabeto:* $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$. *Janela:* $S = 8$ e $L = 4$.
- *Tokens:* $(7, 0, a)$, $(6, 2, s)$, $(4, 2, d)$, $(5, 2, c)$, $(0, 2, a)$.



LZ77: Conversão tokens $\leftrightarrow$ bits

Cada token (p, ℓ, x) deve ser convertido em uma sequência de bits:

- $p \in [0 : S)$: requer $\lceil \log_2 S \rceil$ bits.

No exemplo, $S = 8$, de modo que p requer 3 bits.

- $\ell \in [0 : L)$: requer $\lceil \log_2 L \rceil$ bits.

No exemplo, $L = 4$, de modo que ℓ requer 2 bits.

- $x \in \mathcal{X}$: requer $\lceil \log_2 |\mathcal{X}| \rceil$ bits.

No exemplo, $|\mathcal{X}| = 27$, de modo que x requer 5 bits.

Exemplo LZ77: tokens $\leftrightarrow$ bits

token (p, ℓ, x)	bits
$(7, 0, a)$	$(111, 00, 00001)$
$(6, 2, s)$	$(110, 10, 10011)$
$(4, 2, d)$	$(100, 10, 00100)$
$(5, 2, c)$	$(101, 10, 00011)$
$(0, 2, a)$	$(000, 10, 00001)$

- *Bits:* $y = 111\ 00\ 00001\ 110\ 10\ 10011\ 100\ 10\ 00100\ 101\ 10\ 00011\ 000\ 10\ 00001.$

Comparação:

- *Comprimido:* $10\ \text{bits/token} \times 5\ \text{tokens} = 50\ \text{bits}.$
- *Sem compressão:* $5\ \text{bits/letra} \times 13\ \text{letras} = 65\ \text{bits}.$

Neste algoritmo:

- A mensagem é segmentada em *subsequências que ainda não ocorreram*.
- Um **dicionário** é construído à medida que os segmentos são processados.

A cada passo do algoritmo, um token no formato (p, x) é emitido, onde:

- $p \in \mathbb{N}$ é o **ponteiro** (índice) do dicionário.
- $x \in \mathcal{X}$ é a letra da fonte após o match no dicionário.

Observação: O dicionário inicia com $\{0 : \varepsilon\}$.

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	<code>a_asa_da_casa</code>		ε

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_
3	a_asa_da_casa	$(1, s)$	as

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_
3	a_asa_da_casa	$(1, s)$	as
4	a_asa_da_casa	$(1, _)$	a_

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_
3	a_asa_da_casa	$(1, s)$	as
4	a_asa_da_casa	$(1, _)$	a_
5	a_asa_da_casa	$(0, d)$	d

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_
3	a_asa_da_casa	$(1, s)$	as
4	a_asa_da_casa	$(1, _)$	a_
5	a_asa_da_casa	$(0, d)$	d
6	a_asa_da_casa	$(4, c)$	a_c

Exemplo LZ78: Codificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Mensagem*: $x = a_asa_da_casa$.

i	mensagem	token (p, x)	dicionário
0	a_asa_da_casa		ε
1	a_asa_da_casa	$(0, a)$	a
2	a_asa_da_casa	$(0, _)$	_
3	a_asa_da_casa	$(1, s)$	as
4	a_asa_da_casa	$(1, _)$	a_
5	a_asa_da_casa	$(0, d)$	d
6	a_asa_da_casa	$(4, c)$	a_c
7	a_asa_da_casa	$(3, a)$	asa

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_
3	$(1, s)$	a_as	as

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_
3	$(1, s)$	a_as	as
4	$(1, _)$	a_asa_	a_

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_
3	$(1, s)$	a_as	as
4	$(1, _)$	a_asa_	a_
5	$(0, d)$	a_asa_d	d

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_
3	$(1, s)$	a_as	as
4	$(1, _)$	a_asa_	a_
5	$(0, d)$	a_asa_d	d
6	$(4, c)$	a_asa_da_c	a_c

Exemplo LZ78: Decodificação

- *Alfabeto*: $\mathcal{X} = \{_, a, b, c, \dots, z\}$, com $|\mathcal{X}| = 27$.
- *Tokens*: $(0, a)$, $(0, _)$, $(1, s)$, $(1, _)$, $(0, d)$, $(4, c)$, $(3, a)$.

i	token (p, x)	mensagem	dicionário
0			ε
1	$(0, a)$	a	a
2	$(0, _)$	a_	_
3	$(1, s)$	a_as	as
4	$(1, _)$	a_asa_	a_
5	$(0, d)$	a_asa_d	d
6	$(4, c)$	a_asa_da_c	a_c
7	$(3, a)$	a_asa_da_casa	asa

Cada token (p, x) deve ser convertido em uma sequência de bits:

- A letra $x \in \mathcal{X}$ requer $\lceil \log_2 |\mathcal{X}| \rceil$ bits.

No exemplo, $|\mathcal{X}| = 27$, de modo que x requer 5 bits.

- O ponteiro $p \in \mathbb{N}$ poderia ser representado com um número fixo de $\lceil \log_2 N \rceil$ bits, onde N é o número total de tokens.

No exemplo, $N = 7$, de modo que p iria requerer 3 bits.

- No entanto, há uma maneira alternativa: como o ponteiro p do passo i só assume valores em $[0 : i)$, é possível representá-lo com $\lceil \log_2 i \rceil$ bits.

Isso economiza alguns bits às custas de um tamanho variável da representação de cada token.

Exemplo LZ78: tokens $\leftrightarrow$ bits

i	$\lceil \log_2 i \rceil$	token (p, x)	bits
1	0	(0, a)	(, 00001)
2	1	(0, -)	(0, 00000)
3	2	(1, s)	(01, 10011)
4	2	(1, -)	(01, 00000)
5	3	(0, d)	(000, 00100)
6	3	(4, c)	(100, 00011)
7	3	(3, a)	(011, 00001)

- *Bits:* $y = 00001\ 0\ 00000\ 01\ 10011\ 01\ 00000\ 000\ 00100\ 100\ 00011\ 011\ 00001.$
- *Comparação:* *Comprimido:* 49 bits. *Sem compressão:* 65 bits.

To infinity and beyond

- **Códigos de Huffman adaptativos** (*Faller, 1973; Gallager, 1978*): Estimam a distribuição em uma única passada, atualizando o código a cada letra.
- **Codificação aritmética** (*Rissanen, 1976; Pasco, 1976*): Atinge $H(X) + 2/k$ sem o custo exponencial da extensão de fonte.
- **ANS** (*Duda, 2009*): “*Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding*”.
- **Compressão com perdas**: Aceita-se $\hat{X} \approx X$ em troca de taxas muito menores. Fundamentada na *teoria taxa-distorção* (*Shannon, 1959*).
É amplamente utilizado em compressão de imagem, áudio e vídeo.

Referências

- [1] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [2] S. Haykin, *Communication Systems*, 4th ed. John Wiley & Sons, 2001.
- [3] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.