

# Teoría de la Información

Lucas Goiriz Beltrán  
Instituto de Biología Integrativa de Sistemas  
(I<sub>2</sub>SysBio; UV-CSIC)  
Departamento de Matemática Aplicada  
Universitat Politècnica de València (UPV)

Curso 2023 - 2024

## 1 Teoría de la Información

### 1.1 Introducción

La teoría de la información es una rama de las matemáticas que estudia la cuantificación de la información. Fue propuesta por Claude Shannon en 1948 para estudiar la transmisión de mensajes en sistemas de comunicación, y desde entonces se ha convertido en una disciplina fundamental en la ingeniería de la comunicación y en la teoría de la computación.

La teoría de la información se basa en la teoría de la probabilidad y en la teoría de los sistemas de comunicación. Su objetivo es estudiar la cantidad de información que se puede transmitir a través de un canal de comunicación, y las limitaciones teóricas y prácticas que existen para la transmisión de información.

#### 1.1.1 Esquema general de los sistemas de comunicación

En general, un sistema de comunicación consta de los siguientes elementos:

- **Fuente de información:** es el origen de los mensajes que se van a transmitir. Puede ser un ser humano, un sensor, un ordenador, etc.
- **Codificador:** es el encargado de transformar los mensajes de la fuente en una forma adecuada para su transmisión a través del canal de comunicación.
- **Canal de comunicación:** es el medio físico a través del cual se transmiten los mensajes. Puede ser un cable, una fibra óptica, el aire, etc.
- **Decodificador:** es el encargado de transformar los mensajes recibidos a través del canal en una forma adecuada para su interpretación por el destinatario.
- **Destinatario:** es el receptor de los mensajes transmitidos por el sistema de comunicación.

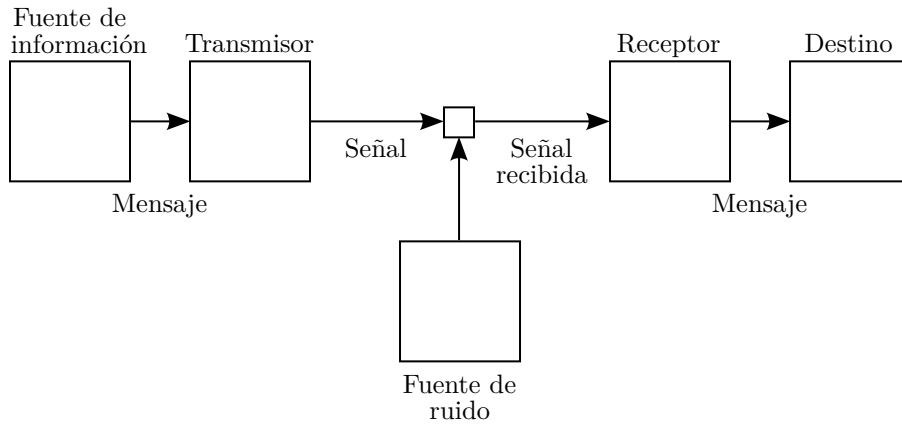


Figure 1: shannon\_comms

- **Ruido:** es cualquier perturbación que afecta a la transmisión de los mensajes a través del canal de comunicación.

Es debido a la presencia de ruido en el canal de comunicación que no es posible transmitir información de forma perfecta. Por ese motivo, es importante cuantificar la cantidad de información que se puede transmitir a través de un canal de comunicación, y estudiar las limitaciones teóricas y prácticas que existen para la transmisión de información.

## 1.2 Entropía

### 1.2.1 Preliminares

**Variable aleatoria** Una variable aleatoria es una función que asigna un valor numérico a cada resultado de un experimento aleatorio. Por ejemplo, si lanzamos un dado, la variable aleatoria  $X$  que representa el número que sale en la cara superior del dado, puede tomar los valores 1, 2, 3, 4, 5, 6. En ese caso escribiríamos que

$$X = \{1, 2, 3, 4, 5, 6\}$$

Osea,  $X$  es una variable aleatoria que toma valores en el conjunto  $\{1, 2, 3, 4, 5, 6\}$ .

En el ejemplo anterior, la variable aleatoria  $X$  es discreta, porque toma valores en un conjunto finito o numerable (recuerda, hay infinitos más grandes que otros. Un infinito numerable es aquel que se puede poner en correspondencia con los números naturales  $\mathbb{N}$ ). Pero también hay variables aleatorias continuas, que toman valores en un intervalo de números reales. Por ejemplo, si medimos la altura de una persona, la variable aleatoria que representa la altura puede tomar cualquier valor en el intervalo  $(0, \infty)$ .

**Función de probabilidad** La función de probabilidad de una variable aleatoria es una función que asigna a cada valor de la variable aleatoria la probabilidad de que ese valor ocurra. Por ejemplo, si lanzamos un dado, la función de probabilidad de la variable aleatoria  $X$  que representa el número que sale en la cara superior del dado, es una función que asigna a cada número del conjunto  $\{1, 2, 3, 4, 5, 6\}$  la probabilidad de que salga ese número. En este caso, la función de probabilidad es uniforme, porque todos los números tienen la misma probabilidad de salir.

Para una variable aleatoria discreta, la función de probabilidad se puede representar mediante una tabla o mediante una fórmula. Por ejemplo, si lanzamos un dado, la función de probabilidad de la variable aleatoria  $X$  que representa el número que sale en la cara superior del dado, se puede representar mediante la siguiente tabla:

$x$	$P(X = x)$
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

O mediante la siguiente fórmula:

$$P(X = x) = \frac{1}{6} \quad \text{para } x \in \{1, 2, 3, 4, 5, 6\}$$

Para una variable aleatoria continua, la función de probabilidad se puede representar mediante una función de densidad de probabilidad. Por ejemplo, si medimos la altura de una persona, la función de densidad de probabilidad de la variable aleatoria que representa la altura, es una función que asigna a cada valor del intervalo  $(0, \infty)$  la probabilidad de que la altura de una persona esté en ese intervalo.

**Esperanza matemática** La esperanza matemática de una variable aleatoria es el valor promedio que toma la variable aleatoria en un experimento aleatorio. Se calcula multiplicando cada valor de la variable aleatoria por su probabilidad, y sumando los resultados:

$$\mathbb{E}[X] = \sum_{i=0}^n x_i \cdot P(X = x_i)$$

Por ejemplo, si lanzamos un dado, la esperanza matemática de la variable aleatoria  $X$  que representa el número que sale en la cara superior del dado, se calcula como

$$\mathbb{E}[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = 3.5$$

Osea, en promedio, el número que sale en la cara superior del dado es 3.5.

Para una variable aleatoria continua, la esperanza matemática se calcula de forma similar, pero en lugar de sumar los valores de la variable aleatoria, se integran:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xf(x)dx$$

Donde  $f(x)$  es la función de densidad de probabilidad de la variable aleatoria  $X$ .

**LOTUS: Law of the Unconscious Statistician (Ley del Estadístico Inconsciente)** La LOTUS es una regla que permite calcular la esperanza matemática de una función de una variable aleatoria, sin necesidad de conocer la función de probabilidad de la variable aleatoria. La LOTUS establece que la esperanza matemática de una función de una variable aleatoria se calcula multiplicando la función por la función de probabilidad de la variable aleatoria, y sumando los resultados:

$$\mathbb{E}[g(X)] = \sum_{i=0}^n g(x_i) \cdot P(X = x_i)$$

Por ejemplo, si lanzamos un dado, y definimos la variable aleatoria  $Y = X^2$ , donde  $X$  es la variable aleatoria que representa el número que sale en la cara superior del dado, la esperanza matemática de la variable aleatoria  $Y$  se calcula como

$$\mathbb{E}[Y] = 1^2 \cdot \frac{1}{6} + 2^2 \cdot \frac{1}{6} + 3^2 \cdot \frac{1}{6} + 4^2 \cdot \frac{1}{6} + 5^2 \cdot \frac{1}{6} + 6^2 \cdot \frac{1}{6} = 15.1667$$

Osea, en promedio, el cuadrado del número que sale en la cara superior del dado es 15.1667.

Para una variable aleatoria continua, la LOTUS se calcula de forma similar, pero en lugar de sumar los valores de la variable aleatoria, se integran:

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

Donde  $f(x)$  es la función de densidad de probabilidad de la variable aleatoria  $X$ .

### 1.2.2 Entropía

En física, la entropía es una medida de la cantidad de desorden o caos en un sistema; más concretamente en termodinámica, la entropía es una medida de la cantidad de energía que no se puede utilizar para realizar trabajo. En teoría de la información, la entropía es una medida de la incertidumbre de una variable aleatoria. También se puede interpretar como una medida de la cantidad de información que se necesita para describir una variable aleatoria.

La entropía de una variable aleatoria  $X$  se define como la esperanza matemática de la información de la variable aleatoria. La información de una variable aleatoria es una medida de la “*sorpres*a” que produce un valor de la variable aleatoria.

La información de una variable aleatoria se calcula como el negativo del logaritmo de la probabilidad de que ocurra un valor de la variable aleatoria:

$$I(x) = -\log_n (P(X = x))$$

Nótese que la base del logaritmo determina la unidad de medida de la información. Esta base puede ser cualquier número positivo, pero las bases más comunes son:

- 2: en cuyo caso la unidad de medida de la información es el *bit* (binary digit).
- e: en cuyo caso la unidad de medida de la información es el *nat* (natural unit of information).
- 10: en cuyo caso la unidad de medida de la información es el *hartley*.

**En caso de no especificar ninguna base, asumiremos que la base del logaritmo es 2.**

Sabiendo esto, la entropía de una variable aleatoria  $X$ , que denotaremos como  $H_k(X)$ , se calcula de la siguiente manera (gracias al teorema LOTUS):

$$H_k(X) = \mathbb{E}[I(X)] = \sum_{i=0}^n -\log_k (P(X = x_i)) \cdot P(X = x_i)$$

O para una variable aleatoria continua:

$$H_k(X) = \int_{-\infty}^{\infty} -\log_k (f(x)) \cdot f(x) dx$$

donde el subíndice  $k$  indica la base del logaritmo. **Recordad que si no se especifica ninguna base, asumiremos que la base del logaritmo es 2.**

En teoría de la información, partiremos de una fuente de información definida como un par  $(S, P)$  donde  $S$  es un alfabeto predefinido y  $P$  es una distribución de probabilidad sobre  $S$ .

Dado que la fuente de información introduce una incertidumbre en la variable aleatoria definida por el alfabeto predefinido, la entropía nos será de utilidad para medir el grado de incertidumbre, el grado de aleatoriedad en la fuente de información y, en consecuencia, estimar las unidades de información necesarias

en promedio para codificar todos los valores posibles que puedan darse en la fuente de información.

Debido a las propiedades del logaritmo,  $H_k(X)$  admite las siguientes definiciones alternativas (pero equivalentes):

- $H_k(X) = -\sum_{i=0}^n \log_k(P(X = x_i)) \cdot P(X = x_i)$
- $H_k(X) = \sum_{i=0}^n \log_k\left(\frac{1}{P(X=x_i)}\right) P(X = x_i)$

Además, debido a las propiedades del logaritmo (en particular que  $\log_a(b) = \frac{\log_c(b)}{\log_c(a)}$ ), se cumple que  $H_b(X) = \log_b(a) \cdot H_a(X)$ . La demostración es sencilla:

$$\begin{aligned}
 H_b(X) &= \sum_{i=0}^n -\log_b(P(X = x_i)) \cdot P(X = x_i) \\
 &= \sum_{i=0}^n -\frac{\log_a(P(X = x_i))}{\log_a(b)} \cdot P(X = x_i) \\
 &= \frac{1}{\log_a(b)} \sum_{i=0}^n -\log_a(P(X = x_i)) \cdot P(X = x_i) \\
 &= \left(\frac{1}{\log_a(b)}\right) H_a(X) \\
 &= \left(\frac{1}{\frac{\log_b(b)}{\log_b(a)}}\right) H_a(X) \\
 &= \left(\frac{1}{\frac{1}{\log_b(a)}}\right) H_a(X) \\
 &= \log_b(a) \cdot H_a(X)
 \end{aligned}$$

---

**Ejemplo 1: Entropía de una variable aleatoria discreta** Sea el alfabeto  $S = \{a, b, c, d\}$  con una distribución de probabilidades  $P = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$ . La entropía de la variable aleatoria  $X$  definida por el alfabeto  $S$  y la distribución de probabilidad  $P$  es:

$$H(X) = -\left(\frac{1}{2} \log\left(\frac{1}{2}\right) + \frac{1}{4} \log\left(\frac{1}{4}\right) + \frac{1}{8} \log\left(\frac{1}{8}\right) + \frac{1}{8} \log\left(\frac{1}{8}\right)\right) = 1.75$$

La entropía de la variable aleatoria  $X$  es 1.75 bits.

---

Una propiedad interesante de la entropía es la siguiente: sea  $X$  una variable aleatoria sobre el alfabeto  $\{x_1, x_2, \dots, x_n\}$ , entonces:

$$0 \leq H(X) \leq \log(n)$$

La entropía de una variable aleatoria  $X$  está acotada por 0 y  $\log(n)$ , donde  $n$  es el número de elementos del alfabeto de la variable aleatoria. La entropía es máxima cuando la distribución de probabilidad es uniforme, y es mínima cuando la distribución de probabilidad es degenerada (es decir, cuando un único valor del alfabeto tiene probabilidad 1 y el resto de valores tienen probabilidad 0).

**Ejemplo 2: Entropía de una variable aleatoria distribuida bernoulli**

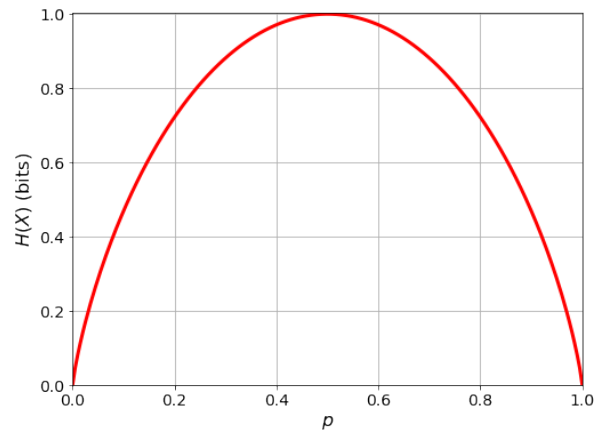
Sea  $X$  una variable aleatoria distribuida Bernoulli con parámetro  $p$  (es decir,  $X \sim \text{Bernoulli}(p)$ , lo cual quiere decir que nuestra variable aleatoria mide “el número de éxitos en un experimento de Bernoulli”). La función de probabilidad de  $X$  es:

$$P(X = x) = p^x(1 - p)^{1-x} \quad \text{para } x \in \{0, 1\}$$

La entropía de la variable aleatoria  $X$  es:

$$\begin{aligned} H(X) &= - \sum_{x=0}^1 p^x(1 - p)^{1-x} \log(p^x(1 - p)^{1-x}) \\ &= -(p \log(p) + (1 - p) \log(1 - p)) \\ &= (p - 1) \log(1 - p) - p \log(p) \end{aligned}$$

Si representamos la entropía de la variable aleatoria  $X$  en función de  $p$ , obtenemos la siguiente gráfica:



¿Qué interpretación tiene esto? La entropía de una variable aleatoria Bernoulli es máxima cuando  $p = 0.5$ , es decir, cuando la distribución de probabilidad es uniforme. Esto tiene sentido, porque en una distribución uniforme, todos los valores del alfabeto tienen la misma probabilidad de ocurrir, y por lo tanto hay más incertidumbre sobre el valor que tomará la variable aleatoria. Por otro lado, la entropía de una variable aleatoria Bernoulli es mínima cuando  $p = 0$  o  $p = 1$ , es decir, cuando la distribución de probabilidad es degenerada. Esto también tiene sentido, porque en una distribución degenerada, solo un valor del alfabeto tiene probabilidad 1 y el resto de valores tienen probabilidad 0, y por lo tanto no hay incertidumbre sobre el valor que tomará la variable aleatoria.

¿Cómo se programaría esto en *python*?

```
import numpy as np
import matplotlib.pyplot as plt

def H_Bernoulli(p):

    h = (p - 1)*np.log2(1 - p) - p*np.log2(p)

    # Fix the NaN values that occur due to the logarithm of zero
    h[np.isnan(h)] = 0

    return h

plt.figure(figsize=(8, 6))

p = np.linspace(0, 1, 200)
plt.plot(p, H_Bernoulli(p), "r", linewidth=3)

plt.grid()
plt.xlim(0, 1)
plt.ylim(0, 1.005)
plt.ylabel(r"$H(X)$ (bits)", fontsize=16)
plt.xlabel(r"$p$", fontsize=16)
plt.tick_params(axis="both", which="major", labelsize=14)
```

**Recordatorio: probabilidades conjuntas y condicionales** Dadas dos variables aleatorias  $X$  e  $Y$ , la probabilidad conjunta de  $X$  e  $Y$  es la probabilidad de que ambas variables aleatorias tomen un valor determinado. Se denota como  $P(X = x, Y = y)$ , y se calcula de la siguiente manera:

$$P(X = x, Y = y) = P(Y = y|X = x)P(X = x)$$

donde  $P(Y = y|X = x)$  es la probabilidad condicional de  $Y$  dado  $X$ , es decir, la probabilidad de que  $Y$  tome un valor determinado dado que  $X$  ha tomado un valor determinado.



De la misma manera, podemos definir la probabilidad conjunta como

$$P(X = x, Y = y) = P(X = x|Y = y)P(Y = y)$$

Igualando ambas expresiones, se obtiene el Teorema de Bayes:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

Además, si las variables aleatorias  $X$  e  $Y$  son independientes, entonces la probabilidad conjunta de  $X$  e  $Y$  es igual al producto de las probabilidades marginales de  $X$  e  $Y$ :

$$P(X = x, Y = y) = P(X = x)P(Y = y)$$

¿Por qué? Porque si  $X$  e  $Y$  son independientes, entonces la probabilidad de que  $Y$  tome un valor determinado no depende de que  $X$  haya tomado un valor determinado, y viceversa, por ende  $P(Y = y|X = x) = P(Y = y)$  y  $P(X = x|Y = y) = P(X = x)$ .

**Entropía conjunta y entropía condicional** Hasta ahora hemos definido la entropía de una única variable aleatoria. Pero en muchos problemas de teoría de la información, estamos interesados en la entropía de dos o más variables aleatorias.

La entropía conjunta de dos variables aleatorias  $X$  e  $Y$  se define como la entropía de la variable aleatoria  $(X, Y)$ , que es una variable aleatoria que toma valores en el producto cartesiano (es decir, todas las parejas posibles) de los alfabetos de  $X$  e  $Y$ . La entropía conjunta de  $X$  e  $Y$  se denota como  $H(X, Y)$ , y se calcula de la siguiente manera:

$$H(X, Y) = - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x, Y = y))$$

O para variables aleatorias continuas:

$$H(X, Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \log(f(x, y)) dx dy$$

Otra manera de escribir la entropía conjunta sería utilizando índices para los valores de las variables aleatorias:

$$H(X, Y) = - \sum_{i=0}^n \sum_{j=0}^m P(X = x_i, Y = y_j) \log(P(X = x_i, Y = y_j))$$

En algunos casos, esta notación es más clara que la notación con los alfabetos  $S_X$  y  $S_Y$ . Habitualmente  $S_X = S_Y$ , es decir, estamos usando el mismo alfabeto para ambas variables aleatorias.

De forma similar a la entropía conjunta, podemos definir la entropía condicional de una variable aleatoria  $X$  dada otra variable aleatoria  $Y$ . La entropía condicional de  $X$  dado  $Y$  se denota como  $H(X|Y)$ , y se calcula de la siguiente manera:

$$H(X|Y) = - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x|Y = y))$$

Es más, utilizando las propiedades de las probabilidades conjuntas y condicionales, podemos escribir la entropía condicional de  $X$  dado  $Y$  de la siguiente manera:

$$\begin{aligned} H(X|Y) &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x|Y = y)) \\ &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x|Y = y) P(Y = y) \log(P(X = x|Y = y)) \\ &= - \sum_{y \in S_Y} P(Y = y) \sum_{x \in S_X} P(X = x|Y = y) \log(P(X = x|Y = y)) \\ &= \sum_{y \in S_Y} P(Y = y) H(X|Y = y) \end{aligned}$$

De la misma manera que hemos hecho en la demostración anterior, las propiedades de la entropía conjunta se originan de operar con las probabilidades conjuntas y condicionales de las variables aleatorias  $X$  e  $Y$ .

Una de estas propiedades es

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

Esta propiedad se conoce como la regla de la cadena de la entropía, y se puede demostrar a partir de la definición de la entropía conjunta y condicional:

$$\begin{aligned} H(X, Y) &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x, Y = y)) \\ &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x|Y = y) P(Y = y)) \\ &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) (\log(P(X = x|Y = y)) + \log(P(Y = y))) \\ &= - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(X = x|Y = y)) - \sum_{x \in S_X} \sum_{y \in S_Y} P(X = x, Y = y) \log(P(Y = y)) \\ &= H(X|Y) - \sum_{y \in S_Y} P(Y = y) \log(P(Y = y)) \\ &= H(X|Y) + H(Y) \end{aligned}$$

De la misma manera, se puede demostrar que  $H(X, Y) = H(Y) + H(X|Y)$ .  
De este resultado se puede demostrar que dadas las variables aleatorias  $X$  e  $Y$ :

1.  $H(X, Y) \leq H(X) + H(Y)$
2.  $H(X, Y) = H(X) + H(Y)$ , (en caso de que  $X$  e  $Y$  sean independientes)

*Una forma sencilla de pensar sobre estas propiedades es pensar en las probabilidades conjuntas y condicionales, de las cuales obtenemos desigualdades similares, solamente que en el caso de la entropía, debido al logaritmo, las desigualdades no son productos si no sumas.*

Un corolario de las propiedades anteriores es que:

1.  $H(X|Y) \leq H(X)$
2.  $H(X|Y) = H(X)$ , (en caso de que  $X$  e  $Y$  sean independientes)

Volviendo a la regla de la cadena, esta la podemos extender para el caso de más de dos variables, y se define como:

$$H(X_1, X_2, \dots, X_n) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \dots + H(X_n|X_1, X_2, \dots, X_{n-1})$$

o de forma más compacta:

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i|X_1, X_2, \dots, X_{i-1})$$

**Entropía Relativa** La entropía relativa, también conocida como divergencia de Kullback-Leibler, es una medida de la diferencia entre dos distribuciones de probabilidad (la cantidad de información que se necesita para describir una distribución de probabilidad utilizando otra distribución de probabilidad). La entropía relativa de dos distribuciones de probabilidad  $P$  y  $Q$  se denota como  $D(P||Q)$ , y se calcula de la siguiente manera:

$$D(P||Q) = \sum_{x \in S} P(x) \log \left( \frac{P(x)}{Q(x)} \right)$$

O para distribuciones de probabilidad continuas:

$$D(P||Q) = \int_{-\infty}^{\infty} f(x) \log \left( \frac{f(x)}{g(x)} \right) dx$$

La entropía relativa es siempre **no negativa**, y es igual a cero si y solo si  $P$  y  $Q$  son iguales.

**Información Mutua** La información mutua de dos variables aleatorias  $X$  e  $Y$  es una medida de la cantidad de información que una variable aleatoria proporciona sobre la otra variable aleatoria. La información mutua de  $X$  e  $Y$  se denota como  $I(X;Y)$ , y se calcula de la siguiente manera:

$$I(X;Y) = \sum_{x \in S_X} \sum_{y \in S_Y} P(X=x, Y=y) \log \left( \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)} \right)$$

O para variables aleatorias continuas:

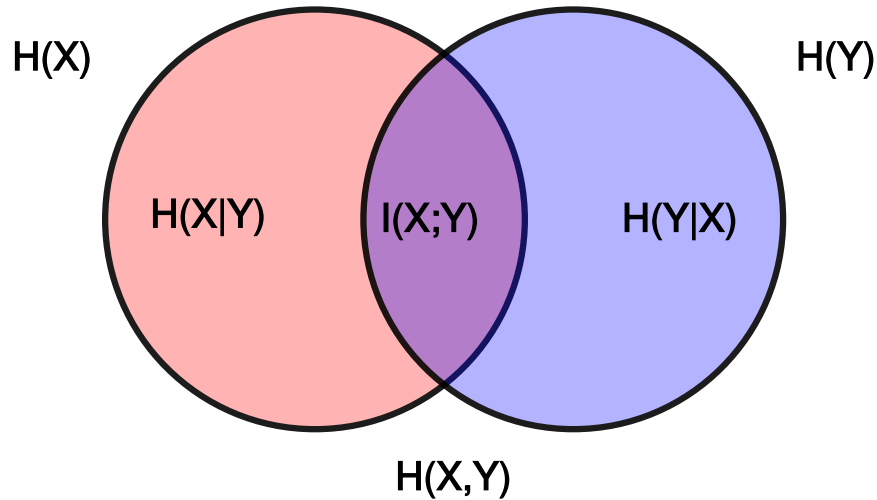
$$I(X;Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \log \left( \frac{f(x,y)}{f(x)f(y)} \right) dx dy$$

La información mutua es siempre **no negativa**, y es igual a cero si y solo si  $X$  e  $Y$  son independientes. Se puede pensar también que la información mutua es la entropía relativa entre la distribución conjunta de dos variables aleatorias y el producto de sus distribuciones marginales.

De la definición de la información mutua se pueden deducir las siguientes propiedades:

1.  $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$
2.  $I(X;Y) = H(X) + H(Y) - H(X,Y)$
3.  $I(X;Y) = I(Y;X)$
4.  $I(X;X) = H(X)$

Gráficamente, estas propiedades se pueden representar de la siguiente manera:



### El caso continuo

**Ejemplo 1: Distribución uniforme** Sea la variable aleatoria  $X$  uniformemente distribuída entre 0 y  $a$ . Su función de densidad de probabilidad es

$$f(x) = \begin{cases} \frac{1}{a} & \text{si } x \in [0, a] \\ 0 & \text{en caso contrario} \end{cases}$$

Su entropía es

$$\begin{aligned} H(X) &= - \int_{-\infty}^{\infty} f(x) \log(f(x)) dx \\ &= - \int_{-\infty}^0 0 \log(0) dx - \int_0^a \frac{1}{a} \log\left(\frac{1}{a}\right) dx - \int_a^{\infty} 0 \log(0) dx \\ &= \int_0^a \frac{1}{a} \log(a) dx \\ &= \left[ \frac{x}{a} \log(a) \right]_0^a \\ &= \log(a) \end{aligned}$$

**Ejemplo 2: Distribución normal** Sea la variable aleatoria  $X$  con función de densidad de probabilidad

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

(es decir una distribución normal con media  $\mu$  y varianza  $\sigma^2$ )

Su entropía es

$$\begin{aligned}
H(X) &= - \int_{-\infty}^{\infty} f(x) \log(f(x)) dx \\
&= - \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \log\left(\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}\right) dx \\
&= - \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \left( -\log(\sigma\sqrt{2\pi}) + \frac{-\log(e)(x-\mu)^2}{2\sigma^2} \right) dx \\
&= - \left( -\log(\sigma\sqrt{2\pi}) \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} + \int_{-\infty}^{\infty} \frac{-\log(e)(x-\mu)^2}{2\sigma^3\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \right) dx \\
&= - \left( -\log(\sigma\sqrt{2\pi}) \cdot 1 - \frac{\log(e)}{2\sigma^2} \int_{-\infty}^{\infty} \frac{(x-\mu)^2}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \right) dx \\
&= \log(\sigma\sqrt{2\pi}) + \frac{\log(e)}{2\sigma^2} (\sigma^2) \\
&= \log(\sigma\sqrt{2\pi}) + \frac{\log(e)}{2} \\
&= \log(\sigma\sqrt{2\pi}) + \log(\sqrt{e}) \\
&= \log(\sigma\sqrt{2\pi e})
\end{aligned}$$

### 1.3 Estructura del lenguaje

Antes de estudiar fuentes de información, es necesario definir un lenguaje para representar los mensajes que se van a transmitir. En general, un lenguaje es un conjunto de símbolos y reglas que permiten construir mensajes a partir de esos símbolos.

#### 1.3.1 Alfabetos y palabras

Un **alfabeto** es un conjunto finito de símbolos. Por ejemplo, nuestro alfabeto podría decirse que está formado por las letras del abecedario. Un **símbolo** es un elemento de un alfabeto. Por ejemplo, la letra “a” es un símbolo de nuestro alfabeto.

En lenguaje formal matemático, un alfabeto  $\mathcal{A}$  es un conjunto finito (salvo que se indique lo contrario de forma explícita) y no vacío cuyos elementos se denominan **símbolos**. Por ejemplo, si  $\mathcal{A} = \{a, b, c\}$ , entonces  $a$ ,  $b$  y  $c$  son símbolos de  $\mathcal{A}$ .

Sea entonces  $\mathcal{A}$  un alfabeto. Una **palabra**  $x$  es una secuencia finita de símbolos de  $\mathcal{A}$  de la forma

$$x = a_1 a_2 \dots a_n, \quad a_i \in \mathcal{A}, \quad i = 1, 2, \dots, n$$

Definimos la **longitud** de una palabra  $x$  como el número de símbolos que contiene. La longitud de una palabra  $x$  se denota por  $|x|$ .

Sea  $\mathcal{A}^n, n \geq 1$  el conjunto de todas las palabras de longitud  $n$  formadas por símbolos de  $\mathcal{A}$ .

El número de elementos (también llamado cardinal) de  $\mathcal{A}^n$  es  $|\mathcal{A}|^n$ .

---

**Ejemplo** Tenemos un alfabeto  $\mathcal{A} = \{0, 1\}$ , y queremos encontrar el conjunto de todas las palabras de longitud 3 formadas por símbolos de  $\mathcal{A}$ .

Entonces,  $\mathcal{A}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$ .

Como  $|\mathcal{A}| = 2$ , el número de elementos de  $\mathcal{A}^3$  es  $|\mathcal{A}|^3 = 2^3 = 8$ .

---

Teniendo un alfabeto  $\mathcal{A}$ , el conjunto de todas las palabras finitas (no nulas) formadas por símbolos de  $\mathcal{A}$  se denota por  $\mathcal{A}^+$  y se define como:

$$\begin{aligned}\mathcal{A}^+ &= \mathcal{A}^1 \cup \mathcal{A}^2 \cup \dots \cup \mathcal{A}^n, \quad n \geq 1 \\ &= \bigcup_{n \geq 1} \mathcal{A}^n\end{aligned}$$

Es decir,  $\mathcal{A}^+$  es la unión de todos los conjuntos de palabras de longitud  $n$  formadas por símbolos de  $\mathcal{A}$ . Se trata de un conjunto donde existen palabras de todas las longitudes posibles hasta un máximo de longitud  $n$ . Este conjunto se parece un poco a las palabras en el lenguaje natural, donde existen palabras de diferentes longitudes.

*¿Cuántos elementos tiene  $\mathcal{A}^+$ ?*

Sabemos que

$$\begin{aligned}\mathcal{A}^+ &= \bigcup_{n \geq 1} \mathcal{A}^n \\ &= \mathcal{A}^1 \cup \mathcal{A}^2 \cup \dots \cup \mathcal{A}^n, \quad n \geq 1\end{aligned}$$

Además, el número de elementos de  $\mathcal{A}^n$  es  $|\mathcal{A}|^n$ . Por lo tanto, el número de elementos de  $\mathcal{A}^+$  es

$$\begin{aligned}|\mathcal{A}^+| &= |\mathcal{A}^1| + |\mathcal{A}^2| + \dots + |\mathcal{A}^n| \\ &= |\mathcal{A}| + |\mathcal{A}|^2 + \dots + |\mathcal{A}|^n \\ &= \sum_{n \geq 1} |\mathcal{A}|^n\end{aligned}$$

Esto se parece a una serie geométrica...

$$\begin{aligned}\sum_{n \geq 1} |\mathcal{A}|^n &= \sum_{n \geq 0} |\mathcal{A}|^n - 1 \\ &= \frac{1 - |\mathcal{A}|^{n+1}}{1 - |\mathcal{A}|} - 1\end{aligned}$$

Por lo tanto, el número de elementos de  $\mathcal{A}^+$  es

$$|\mathcal{A}^+| = \frac{1 - |\mathcal{A}|^{n+1}}{1 - |\mathcal{A}|} - 1$$


---

**Ejemplo** Tenemos un alfabeto  $\mathcal{A} = \{0, 1\}$ , y queremos encontrar el conjunto de todas las palabras finitas (no nulas, de hasta un máximo de longitud 5) formadas por símbolos de  $\mathcal{A}$ . ¿Cuántos elementos tiene  $\mathcal{A}^+$ ?

Sin necesidad de construir cada palabra, podemos calcular el número de elementos de  $\mathcal{A}^+$  usando la fórmula anterior. Como  $|\mathcal{A}| = 2$  y  $n = 5$ , el número de elementos de  $\mathcal{A}^+$  es

$$\begin{aligned}|\mathcal{A}^+| &= \frac{1 - |\mathcal{A}|^{n+1}}{1 - |\mathcal{A}|} - 1 \\ &= \frac{1 - 2^{5+1}}{1 - 2} - 1 \\ &= \frac{1 - 64}{-1} - 1 \\ &= 62\end{aligned}$$


---

Para cualquier palabra  $x \in \mathcal{A}^+$  y  $1 \leq n \leq |x|$ , sea  $x[n] = a_n$ , el  $n$ -ésimo símbolo de la secuencia. Por tanto

$$x = x[1] \dots x[|x|]$$

Dadas dos palabras  $x, y \in \mathcal{A}^+$ , decimos que  $x$  e  $y$  son **iguales** si y solo si:

$$x = y \Leftrightarrow \begin{cases} (|x| = |y|) \\ \wedge (x[i] = y[i], \forall 1 \leq i \leq |x|) \end{cases}$$

Es decir, son iguales cuando tienen la misma longitud, y además todos sus símbolos son iguales en la misma posición.

De forma similar, definimos la **concatenación** de dos palabras  $x, y \in \mathcal{A}^+$  (que denotaremos como  $xy$ ) como la palabra  $z \in \mathcal{A}^+$  definida por



1.  $|z| = |x| + |y|$
2.  $z[i] = x[i]$  con  $1 \leq i \leq |x|$
3.  $z[|x| + i] = y[i]$  con  $1 \leq i \leq |y|$

Es decir, la concatenación de dos palabras es una nueva palabra que contiene todos los símbolos de la primera palabra seguidos de todos los símbolos de la segunda palabra.

Es sencillo ver que la concatenación de palabras es una operación **asociativa**, es decir, para cualquier  $x, y, z \in \mathcal{A}^+$ , se cumple que

$$(xy)z = x(yz)$$

Sin embargo, la concatenación de palabras **no es conmutativa**, es decir, en general no se cumple que

$$xy = yx$$

*Podéis pensar en esto como si se tratase de “sumar strings” en **python**... Tened cuidado con el sistema de indexación, ya que en **python** los índices empiezan en 0, mientras que aquí los hemos hecho empezar en 1.*

En muchas ocasiones es conveniente definir un elemento identidad para una operación (en este caso, la concatenación de palabras). La identidad de la concatenación de palabras es la palabra vacía, que denotaremos como  $\lambda$  (pensad en el string vacío "" en **python**).  $\lambda$  tiene las siguientes propiedades:

1.  $|\lambda| = 0$
2.  $\mathcal{A}^0 = \{\lambda\}$
3.  $\forall x \in \mathcal{A}^+, \quad x \neq \lambda$
4.  $|x| = 0 \Leftrightarrow x = \lambda$

Es decir, que la palabra vacía es la única palabra de longitud 0, y es la única palabra que no tiene símbolos. Si una palabra tiene longitud 0, entonces es la palabra vacía, y si una palabra no es la palabra vacía, entonces tiene longitud mayor que 0.

De esta manera podemos generar otro conjunto, el cual incluye a la palabra vacía, que denotaremos como  $\mathcal{A}^*$  y se define como

$$\begin{aligned} \mathcal{A}^* &= \mathcal{A}^+ \cup \{\lambda\} \\ &= \mathcal{A}^+ \cup \mathcal{A}^0 \\ &= \left( \bigcup_{n \geq 1} \mathcal{A}^n \right) \cup \mathcal{A}^0 \\ &= \bigcup_{n \geq 0} \mathcal{A}^n \end{aligned}$$

Podemos extender la operación de concatenación de palabras a  $\mathcal{A}^*$ , de manera que para cualquier  $x, y \in \mathcal{A}^*$ , se cumplen las propiedades anteriores de la concatenación, y además

$$\lambda x = x\lambda = x$$

Convirtiéndose de esta manera la concatenación de palabras sobre  $\mathcal{A}^*$  en una operación asociativa y con elemento neutro.

---

A partir de la concatenación de palabras es posible definir la **potencia** de una palabra. Dada una palabra  $x \in \mathcal{A}^+$  y un número entero no negativo  $k$ , la potencia de  $x$  elevada a  $k$ , denotada como  $x^k$ , se define como

1.  $x^0 = \lambda$
2.  $x^k = x^{k-1}x$

Es decir, la potencia de una palabra es la concatenación de la palabra consigo misma  $k$  veces.

---

**Ejemplo** Dada la palabra  $x = 010$ , calculamos  $x^3$  como

$$\begin{aligned} x^3 &= x^2x \\ &= (xx)x \\ &= (010010)010 \\ &= 010010010 \end{aligned}$$

---

Una propiedad que se observa inmediatamente es que  $|x^k| = k|x|$ . Es decir, la longitud de la potencia de una palabra es igual al producto de la longitud de la palabra por el exponente de la potencia.

Además, otra propiedad es que  $\forall n > 1, \quad x^n = x \Leftrightarrow x = \lambda$ . Es decir, que una palabra elevada a un exponente mayor que 1 es igual a la palabra si y solo si la palabra es la palabra vacía.

---

También es posible “*extraer*” trozos de una palabra. Si tenemos una palabra  $x \in \mathcal{A}^+$ , definimos su segmento  $x[i : j]$  como

$$x[i : j] = \begin{cases} x[i] \dots x[j], & 1 \leq i \leq j \leq |x| \\ x[i], & 1 \leq i = j \leq |x| \\ \lambda, & \text{en otro caso} \end{cases}$$

Es decir, que el segmento de una palabra es la subsecuencia de la palabra que va desde el símbolo  $i$ -ésimo hasta el símbolo  $j$ -ésimo, ambos incluidos. Si  $i = j$ , entonces el segmento es simplemente el símbolo  $i$ -ésimo, y si  $i > j$ , entonces el segmento es la palabra vacía.

Rápidamente se puede deducir que  $x = x[1 : |x|]$ . Es decir, que una palabra es igual a su segmento que va desde el primer símbolo hasta el último símbolo.

### 1.3.2 Órdenes Alfabéticos

Dado un alfabeto  $\mathcal{A}$ , es posible definir un **orden alfabético** sobre los símbolos de  $\mathcal{A}$ . Un orden alfabético es una relación de orden dada mediante enumeración (sin repetición) de los símbolos de  $\mathcal{A}$ . Si  $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ , es una enumeración sin repetición, entonces

$$a_i < a_j \Leftrightarrow i < j$$

Es decir, que el orden alfabético de los símbolos de  $\mathcal{A}$  es el orden de dicha enumeración.

---

Este orden alfabético se puede extender a las palabras de  $\mathcal{A}^*$  de la siguiente manera. Dadas dos palabras  $x, y \in \mathcal{A}^+$ , decimos que  $x$  es **menor** que  $y$  en el orden alfabético si y solo si:

$$x < y \Leftrightarrow \begin{cases} (\exists u \in \mathcal{A}^+ : y = xu) \\ \vee (\exists u, v, w \in \mathcal{A}^*, \exists a, b \in \mathcal{A} : (x = uav) \wedge (y = ubw) \wedge (a < b)) \end{cases}$$

Es decir, que una palabra es menor que otra en el orden alfabético si y solo si es un prefijo de la otra palabra, o si ambas palabras tienen un prefijo común y el siguiente símbolo de una es menor que el siguiente símbolo de la otra. Nótese que el prefijo común puede ser la palabra vacía ya que  $\lambda$  es prefijo de cualquier palabra y  $u, v, w \in \mathcal{A}^*$ .

Nótese que trivialmente se tienen las siguientes propiedades:

1.  $\forall x \in \mathcal{A}^+ : \lambda < x$
2.  $\forall x, y \in \mathcal{A}^* : x \neq y \Rightarrow ((x < y) \vee (y < x))$
3. Sobre cada alfabeto  $\mathcal{A}$ , pueden definirse  $|\mathcal{A}|!$  órdenes alfabéticos distintos.

---

**Ejemplo 1** Sean las palabras  $x = 010$  e  $y = 011$ . ¿Cuál es la relación de orden alfabético entre  $x$  e  $y$ ?

Para determinar la relación de orden alfabético entre  $x$  e  $y$ , primero comprobamos si  $x$  es prefijo de  $y$  o viceversa. En este caso,  $x$  no es prefijo de  $y$  y  $y$  no es prefijo de  $x$ . Por tanto, comprobamos si tienen un prefijo común. En este

caso, el prefijo común es la palabra vacía  $\lambda$ . Por consiguiente, comprobamos si el siguiente símbolo de  $x$  es menor que el siguiente símbolo de  $y$ . En este caso, el siguiente símbolo de  $x$  es 1 y el siguiente símbolo de  $y$  es 1. Como 0 es menor que 1, entonces  $x < y$  en el orden alfabético.

---

### 1.3.3 Prefijos, Sufijos y Segmentos

Anteriormente hemos definido el segmento de una palabra de forma breve. Ahora vamos a definir de forma más precisa los conceptos de prefijo, sufijo y segmento, dado que de forma indirecta hemos utilizado estos conceptos en la definición de órdenes alfabéticos.

**Prefijos** Diremos que para  $x, y \in \mathcal{A}^*$  que  $x$  es un **prefijo** de  $y$  siempre que

$$\exists u \in \mathcal{A}^* : y = xu$$

Es decir, que una palabra es prefijo de otra si y solo si la otra palabra es la concatenación de la primera palabra con otra palabra cualquiera.

Trivialmente, se cumplen las siguientes propiedades:

1.  $\forall x \in \mathcal{A}^* : \lambda$  es prefijo de  $x$  (la palabra vacía es prefijo de cualquier palabra).
2.  $\forall x \in \mathcal{A}^* : x$  es prefijo de  $x$  (una palabra es prefijo de sí misma).
3.  $\forall x, y \in \mathcal{A}^* : x$  es prefijo de  $y \Leftrightarrow |x| \leq |y|$  en el orden alfabético (si una palabra es prefijo de otra, entonces su longitud es menor o igual que la longitud de la palabra de la cual es sufijo).
4.  $\forall x, y \in \mathcal{A}^* : x$  es prefijo de  $y \Leftrightarrow \exists i : 1 \leq i \leq |y|/x = y[1 : i]$  (una palabra es prefijo de otra si y solo si es el segmento de la otra palabra que va desde el primer símbolo hasta un símbolo cualquiera).
5.  $\forall x, y \in \mathcal{A}^* : x$  es prefijo de  $y \Rightarrow x \leq y$  para cualquier orden alfabético de  $\mathcal{A}$  (si una palabra es prefijo de otra, entonces es menor o igual que la otra palabra en el orden alfabético).

**Sufijos** Los sufijos se definen de forma similar a los prefijos. Diremos que para  $x, y \in \mathcal{A}^*$  que  $x$  es un **sufijo** de  $y$  siempre que

$$\exists u \in \mathcal{A}^* : y = ux$$

Nuevamente, se cumplen una serie de propiedades (muy similares a las de los prefijos):

1.  $\forall x \in \mathcal{A}^* : \lambda$  es sufijo de  $x$  (la palabra vacía es sufijo de cualquier palabra).
2.  $\forall x \in \mathcal{A}^* : x$  es sufijo de  $x$  (una palabra es sufijo de sí misma).
3.  $\forall x, y \in \mathcal{A}^* : x$  es sufijo de  $y \Leftrightarrow |x| \leq |y|$  en el orden alfabético (si una palabra es sufijo de otra, entonces su longitud es menor o igual que la longitud de la palabra de la cual es sufijo).

4.  $\forall x, y \in \mathcal{A}^* : x \text{ es sufijo de } y \Leftrightarrow \exists i : 1 \leq i \leq |y|/x = y[i : |y|]$  (una palabra es sufijo de otra si y solo si es el segmento de la otra palabra que va desde un símbolo cualquiera hasta el último símbolo).

De los sufijos no se puede deducir ninguna propiedad relacionada con el orden alfabético, ya que no existe una relación directa entre los sufijos y los órdenes alfabéticos.

**Segmentos** A continuación daremos una definición más precisa de segmento. Dadas dos palabras  $x, y \in \mathcal{A}^*$ , diremos que  $x$  es un **segmento** de  $y$  siempre que

$$\exists u, v \in \mathcal{A}^* : y = uxv$$

Podemos ver que si  $u = \lambda$ , entonces  $x$  es un prefijo de  $y$ , y si  $v = \lambda$ , entonces  $x$  es un sufijo de  $y$ . Por tanto, los segmentos son una generalización de los prefijos y sufijos. Trivialmente, si tanto  $u$  como  $v$  son la palabra vacía, entonces  $x = y$ . De forma similar, se tienen las siguientes propiedades:

1.  $\forall x \in \mathcal{A}^* : \lambda \text{ es segmento de } x$  (la palabra vacía es segmento de cualquier palabra).
2.  $\forall x \in \mathcal{A}^* : x \text{ es segmento de } x$  (una palabra es segmento de sí misma).
3.  $\forall x, y \in \mathcal{A}^* : x \text{ es segmento de } y \Leftrightarrow |x| \leq |y|$  (si una palabra es segmento de otra, entonces su longitud es menor o igual que la longitud de la palabra de la cual es sufijo).
4.  $\forall x, y \in \mathcal{A}^* : x \text{ es segmento de } y \Leftrightarrow \exists i, j : 1 \leq i \leq j \leq |y|/x = y[i : j]$  (una palabra es segmento de otra si y solo si es el segmento de la otra palabra que va desde un símbolo cualquiera hasta otro símbolo cualquiera). Esta es la definición que hemos utilizado anteriormente.

### 1.3.4 Homomorfismos

Sean dos alfabetos  $\mathcal{A}$  y  $\mathcal{B}$  (no necesariamente distintos), entonces una aplicación

$$h : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

es un homomorfismo siempre que

$$\forall x, y \in \mathcal{A}^* : h(xy) = h(x)h(y)$$

Es decir, los homomorfismos preservan las operaciones de concatenación de palabras; y por ende preservan la estructura de las palabras. En consecuencia,  $\mathcal{B}^*$  queda totalmente determinado por  $\mathcal{A}^*$  y el homomorfismo  $h$  sobre cada elemento de  $\mathcal{A}$ .

Una propiedad importante es que  $h(\lambda) = \lambda$ , es decir, que el homomorfismo de la palabra vacía es la palabra vacía.

Los homomorfismos son una herramienta muy útil para estudiar la estructura de las palabras, ya que permiten transformar palabras de un alfabeto a otro manteniendo la estructura de las palabras. Por ejemplo, si tenemos un

homomorfismo  $h$  que transforma las vocales de un alfabeto a las consonantes de otro alfabeto, entonces podemos transformar palabras de un alfabeto a otro manteniendo la estructura de las palabras.

### 1.3.5 Permutaciones circulares por la izquierda

Sean  $a \in \mathcal{A}$ ,  $x, y \in \mathcal{A}^*$  y  $x = ay$ , entonces se define la operación de **permutación circular por la izquierda**, denotada por  $\hookrightarrow$ , sobre  $x$  como

$$\hookrightarrow(x) = ya$$

Entonces,

1.  $\forall x \in \mathcal{A}^+ : \hookrightarrow(x) = x[2 : |x|]x[1]$
2. En particular, si  $|x| = 1$ , entonces  $\hookrightarrow(x) = x$ .

Es más, la aplicación reiterada de la permutación circular por la izquierda sobre una palabra  $x$  genera un ciclo, de tal forma que aplicada  $|x|$  veces, se obtiene la palabra original  $x$ . Esto se denota como  $\hookrightarrow^{|x|}(x) = x$ .

### 1.3.6 Secuencias de palabras y ordenaciones

Sea  $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$  una secuencia de palabras  $x_i \in \mathcal{A}^*$ ,  $i = 1, \dots, n$ , y supóngase una ordenación alfabética sobre  $\mathcal{A}^*$  (como hemos explicado anteriormente). Entonces se define la secuencia ordenada  $\mathbf{z}$

$$\mathbf{z} = \text{ord}(\mathbf{x})$$

es una permutación de  $\mathbf{x}$  tal que

$$\mathbf{z} = \langle x'_1, x'_2, \dots, x'_n \rangle$$

donde se cumple que  $x'_i \leq x'_j$  para cualquier  $i, j = 1, \dots, n$ .

Para  $\mathbf{x}$ , denotamos su componente  $i$ -ésima como  $x(i)$ , con  $1 \leq i \leq n$ .

## 1.4 Fuentes de Información

Una fuente de información se define como una tupla  $(\mathcal{S}, P)$  donde

- $\mathcal{S}$  es un conjunto de símbolos que pueden ser emitidos por la fuente (un alfabeto).
- $P$  es una distribución de probabilidades asociada (a la emisión) de los símbolos.

Las fuentes de información pueden ser continuas (con un número de mensajes no numerable) o discretas (con un número de mensajes numerable o finito). En este curso nos centraremos en el caso discreto, por ende  $\mathcal{S}$  será un conjunto finito.

Claramente,  $\forall s \in \mathcal{S}, 0 \leq P(s) \leq 1$ , y  $\sum_{s \in \mathcal{S}} P(s) = 1$  (debido a los Axiomas de Kolmogorov).

#### 1.4.1 Fuentes de Información de Memoria Nula

En una fuente de información de memoria nula, la emisión de cada símbolo  $s_i$  es independiente de los símbolos emitidos anteriormente. En este caso, la probabilidad de emitir un símbolo  $s_i$  es constante e igual a  $P(s_i)$ .

La entropía de una fuente de información de memoria nula se define como

$$H(\mathcal{S}) = - \sum_{s \in \mathcal{S}} P(s) \log(P(s))$$

#### 1.4.2 Extensión de grado $m$

Podemos imaginarnos la fuente de información de memoria nula como una máquina que emite secuencias de  $x$  símbolos, palabras,  $x \in \mathcal{S}^+$ , de modo que los símbolos sucesivamente emitidos se eligen de acuerdo con las probabilidades  $P(s)$ .

Sea entonces  $|x| = m$ , definimos la extensión de grado  $m$  de la fuente de información de memoria nula como el par  $(\mathcal{S}^m, P^m)$ , donde

$$\forall x \in \mathcal{S}^m : P(x) = \prod_{1 \leq i \leq m} P(x[i])$$

Y trivialmente, la entropía

$$\begin{aligned} H(\mathcal{S}^m) &= - \sum_{x \in \mathcal{S}^m} P(x) \log(P(x)) \\ &= - \sum_{x \in \mathcal{S}^m} \left( \prod_{1 \leq i \leq m} P(x[i]) \right) \log \left( P \left( \prod_{1 \leq i \leq m} P(x[i]) \right) \right) \end{aligned}$$

En lugar de calcular los productos de probabilidades de los símbolos para cada una de las palabras de la extensión de grado  $m$ , se puede demostrar el siguiente útil resultado:

$$H(\mathcal{S}^m) = mH(\mathcal{S})$$

#### 1.4.3 Fuentes de Markov (fuente de información con memoria, de orden $m$ )

En una fuente de Markov, la emisión de cada símbolo  $s_i$  depende de los  $m$  símbolos emitidos anteriormente (en muchas aplicaciones). En este caso, la probabilidad de emitir un símbolo  $s_i$  depende de los  $m$  símbolos anteriores, y se denota como la probabilidad condicional  $P(s_i | s_{i-1}, \dots, s_{i-m})$ , con

$$\sum_{i=1}^m P(s_i | s_{i_1}, \dots, s_{i_m}) = 1, \quad i_k = 1, 2, \dots, n$$

A la secuencia de símbolos emitidos se les llama **cadena de Markov**. Los estados de una fuente de Markov (de orden  $m$ ) serán todas las posibles combinaciones de  $m$  símbolos  $n^m$ .

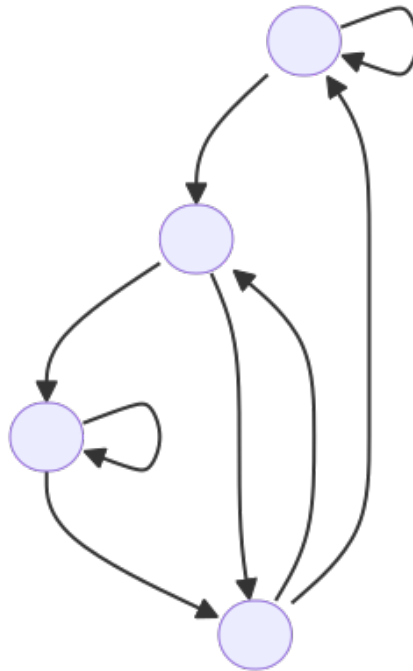
Dada una fuente de Markov de orden  $m$ , se puede establecer su diagrama de estados que muestra los estados y las probabilidades entre ellos.

---

**Ejemplos** A partir de un alfabeto binario  $\{0, 1\}$ , se puede definir una fuente de Markov de orden 2 con los siguientes estados y probabilidades condicionales:

- $P(0|00) = P(1|11) = 0.7$
- $P(1|00) = P(0|11) = 0.3$
- $P(0|01) = P(1|10) = P(0|10) = P(1|01) = 0.5$

El diagrama de estados es el siguiente:




---

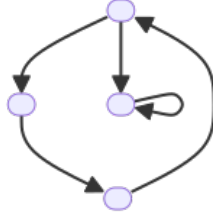
Se dice que una fuente de Markov es ergódica si es posible pasar de cualquier estado a cualquier otro estado en un número finito de pasos.

---



**Ejemplo** Una fuente ergódica podría ser la anterior (podemos alcanzar cualquier estado en un número finito de pasos a partir de otro estado).

Ejemplo de diagrama de estados de una fuente ergódica:



Nótese que el nodo interior es un estado absorbente. Es decir, una vez que se llega a ese estado, no se puede salir de él.

---

Una fuente de Markov es homogénea si las probabilidades condicionales no cambian con el tiempo.

Se dice que una fuente está en estado estacionario si la probabilidad de observación de sus estados no cambia con el tiempo. Las probabilidades se pueden obtener mediante la ecuación  $v = \pi \Pi$ , donde  $\pi$  es el vector de probabilidad de los estados y  $\Pi$  es la matriz de transición.

---

**Ejemplo** Volviendo al ejemplo inicial, la matriz de probabilidades tiene la siguiente forma:

$$\Pi = \begin{matrix} & \begin{matrix} 00 & 01 & 10 & 11 \end{matrix} \\ \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} & \begin{pmatrix} 0.7 & 0.5 & 0.5 & 0.3 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0.3 & 0.5 & 0.5 & 0.7 \end{pmatrix} \end{matrix}$$

Véase que cada fila suma 1 (debido a los axiomas de Kolmogorov). Si cada columna sumase 1, entonces la matriz es doblemente estocástica y todos los estados son equiprobables.

Vayamos al cálculo de probabilidades de estados en régimen estacionario. Para ello, nos definimos las siguientes ecuaciones:

$$\begin{aligned} P(00) &= P(0|00) \cdot P(00) + P(0|10) \cdot P(10) \\ P(01) &= P(1|00) \cdot P(00) + P(1|10) \cdot P(10) \\ P(11) &= P(1|01) \cdot P(01) + P(1|11) \cdot P(11) \\ P(00) + P(01) + P(10) + P(11) &= 1 \end{aligned}$$

Nótese que la última ecuación surge por uno de los axiomas de Kolmogorov, ya que si incluyésemos el estado 01 en la primera ecuación, tendríamos una ecuación redundante (y por ende un sistema singular; con infinitas soluciones). Conocemos las probabilidades condicionales, por lo que podemos resolver el sistema de ecuaciones:

$$\begin{aligned}P(00) &= 0.7 \cdot P(00) + 0.5 \cdot P(10) \\P(01) &= 0.3 \cdot P(00) + 0.5 \cdot P(10) \\P(11) &= 0.5 \cdot P(01) + 0.7 \cdot P(11) \\P(00) + P(01) + P(10) + P(11) &= 1\end{aligned}$$

Este sistema tiene la siguiente solución:

$$\pi = \begin{pmatrix} P(00) \\ P(01) \\ P(10) \\ P(11) \end{pmatrix} = \begin{pmatrix} \frac{5}{16} \\ \frac{3}{16} \\ \frac{3}{16} \\ \frac{5}{16} \end{pmatrix}$$

Ahora, con esta información podemos calcular las probabilidades de aparición de cada símbolo:

$$\begin{aligned}P(0) &= P(00) + P(10) = \frac{5}{16} + \frac{3}{16} = \frac{8}{16} = \frac{1}{2} \\P(1) &= P(01) + P(11) = \frac{3}{16} + \frac{5}{16} = \frac{8}{16} = \frac{1}{2}\end{aligned}$$

---

#### 1.4.4 Fuentes afines (adjuntas) de una fuente de Markov

Dada una fuente de Markov de orden  $m$  con un alfabeto  $\mathcal{S}$  y una distribución de probabilidad  $P$ , denominaremos a la fuente afín como a la fuente de memoria nula  $(\mathcal{S}, P)$ . Es decir, la fuente afín es la fuente de memoria nula que emite los mismos símbolos que la fuente de Markov.

---

**Ejemplo** Para el ejemplo de la fuente de Markov de orden 2, la fuente afín sería la fuente de memoria nula con  $\mathcal{S} = \{0, 1\}$  y  $P = \{\frac{1}{2}, \frac{1}{2}\}$ .

---

Para el cálculo de entropía de una fuente de Markov de orden  $m$ , sobre un alfabeto  $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$  y distribución de probabilidad condicional  $P(s_i | s_{i_1}, \dots, s_{i_m})$ , se cumple que  $p(s_i, s_{i_1}, \dots, s_{i_m}) = p(s_i | s_{i_1}, \dots, s_{i_m}) \cdot p(s_{i_1}, \dots, s_{i_m})$ , lo cual nos permite calcular la entropía condicional:

$$H(\mathcal{S}|s_{i_1}, \dots, s_{i_m}) = - \sum_{i=1}^n p(s_i|s_{i_1}, \dots, s_{i_m}) \log(p(s_i|s_{i_1}, \dots, s_{i_m}))$$

Por lo tanto, la entropía de la fuente de Markov de orden  $m$  es:

$$\begin{aligned} H(\mathcal{S}) &= \sum_{\mathcal{S}^m} p(s_{i_1}, \dots, s_{i_m}) H(\mathcal{S}|s_{i_1}, \dots, s_{i_m}) \\ &= - \sum_{\mathcal{S}^m} p(s_{i_1}, \dots, s_{i_m}) \left( - \sum_{s_i \in \mathcal{S}} p(s_i|s_{i_1}, \dots, s_{i_m}) \log(p(s_i|s_{i_1}, \dots, s_{i_m})) \right) \\ &= - \sum_{\mathcal{S}^{m+1}} p(s_i, s_{i_1}, \dots, s_{i_m}) p(s_i|s_{i_1}, \dots, s_{i_m}) \log(p(s_i|s_{i_1}, \dots, s_{i_m})) \\ &= - \sum_{\mathcal{S}^{m+1}} p(s_{i_1}, \dots, s_{i_m}, s_i) \log(p(s_i|s_{i_1}, \dots, s_{i_m})) \end{aligned}$$

## 1.5 Códigos

### 1.5.1 Introducción

Como hemos visto al principio del temario, existe un componente en el esquema de transmisión de información llamado **codificador**, el cual se encarga de transformar los mensajes de la fuente (**MF**) en una forma adecuada para su transmisión a través del canal de comunicación y recepción (mensajes codificados, **MC**), donde se lleva a cabo el proceso correspondiente de decodificación para la recuperación del mensaje fuente original **MF**.

En una primera aproximación simplificada a este escenario, definimos el proceso de codificación mediante una función  $f$  tal que

$$f : \mathbf{MF} \rightarrow \mathcal{P}(\mathbf{MC})$$

donde  $\forall m \in \mathbf{MC} : |f(m)| < \infty$  (es decir que los resultados de la codificación del mensaje tienen una longitud finita). Nótese que  $\mathcal{P}(C)$  significa el conjunto potencia de  $C$ .

Por lo tanto, en este contexto de transmisión de información, cuando se emite un mensaje aleatorio  $m$ , el receptor recibe un mensaje codificado  $m' \in \mathcal{P}(C)$ . Entonces, existe en el receptor una función  $f^*$  de decodificación:

$$f^* : \mathbf{MC} \rightarrow \mathbf{MF}$$

tal que

$$\forall m \in \mathbf{MF}, \quad \exists m' : f^*(m') = m$$

Por ahora asumiremos que  $\forall m \in \mathbf{MF}, |f(m)| = 1$  (pero veremos que nos podremos extender a más casos).

### 1.5.2 Codificaciones y códigos

Llevando la introducción anterior a un caso más particular (y familiar); sea  $\mathcal{A}$  un alfabeto (correspondiente a la fuente de información de memoria nula  $(\mathcal{A}, P)$ , de la que se originan los mensajes fuente  $\mathcal{A}^+$ ) y sea  $\mathcal{C}$  el conjunto de los mensajes codificados  $\mathcal{MC}$  (también conocido como el conjunto de códigos), entonces una codificación  $f$  es una aplicación **biyectiva**:

$$f : \mathcal{A}^+ \rightarrow \mathcal{C}$$

Para que esta codificación sea admisible para un proceso eficiente de codificación-decodificación, deberá tener unas características determinadas que se irán exponiendo más adelante.

Sea  $\mathcal{B}$  un alfabeto (código), entonces  $\mathcal{C} \subseteq \mathcal{B}^+$ . Entonces una codificación admisible es una aplicación inyectiva

$$f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$$

en la que tenemos que  $f(\mathcal{A}^+) = \mathcal{C}$ . Equivalentemente, si tomamos  $f(\lambda) = \lambda$  (y también  $f^{-1}(\lambda) = \{\lambda\}$ ), entonces podemos extender esta aplicación inyectiva a

$$f : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

En este escenario  $\mathcal{A}$  recibe el nombre de “*alfabeto fuente*” y  $\mathcal{B}$  “*alfabeto código*”.

### 1.5.3 Codificación Bloque

Una codificación bloque  $f$  es una aplicación que le asigna a cada uno de los símbolos  $a \in \mathcal{A}$  una palabra de  $\mathcal{B}^+$ , y se comporta como un homomorfismo, lo cual quiere decir que

$$\forall x \in \mathcal{A}^+, \forall a \in \mathcal{A} : f(ax) = f(a)f(x)$$

y de forma equivalente

$$\forall x \in \mathcal{A}^+, \forall a \in \mathcal{A} : f(xa) = f(x)f(a)$$

y de forma general

$$\forall x, y \in \mathcal{A}^+ : f(xy) = f(x)f(y)$$

Por tanto

$$\forall x \in \mathcal{A}^+ : f(x) = f(x[1]) \dots f(x[|x|])$$

(todo esto es equivalente, y lo hemos visto anteriormente en la definición de homomorfismo en el apartado de estructura del language)

El conjunto  $f(\mathcal{A}) = \{f(a), \forall a \in \mathcal{A}\}$  recibe el nombre de “*código bloque*” (o simplemente “*código*”).

**Ejemplo 1** Consideremos los siguientes alfabetos fuente ( $\mathcal{A}$ ) y código ( $\mathcal{B}$ ):

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\mathcal{B} = \{0, 1\}$$

y además consideremos la codificación de bloque definida por  $f$ :

$$f(a) = \left( \left\lfloor \frac{a_1}{2^0} \right\rfloor \right) \left( \left\lfloor \frac{a_2}{2^1} \right\rfloor \right) \left( \left\lfloor \frac{a_3}{2^2} \right\rfloor \right) \left( \left\lfloor \frac{a_4}{2^3} \right\rfloor \right)$$

claramente tenemos que

$$f(\mathcal{A}) = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001\}$$

Se puede ver que  $f(\mathcal{A}) \subset \mathcal{B}^4$  (porque hay palabras en  $\mathcal{B}^4$  que no aparecen en  $f(\mathcal{A})$ , como por ejemplo 1111).  $f$  es una codificación de bloque ya que le asigna a cada caracter del alfabeto fuente, una palabra formada por el alfabeto código.

**Ejemplo 2: Morse** En el morse, el alfabeto código consta de los elementos  $\{\cdot, -, \text{pausa}\}$ :

Alfabeto fuente	Código
A	.-
B	-...
C	-.-.
D	-..
E	.
F	..-.
G	-.
H	....
I	..
J	. —
K	-.-
L	.-..
M	—
N	-.
P	.-.
Q	—
S	...
T	-
U	..-
V	...-
W	.- —
X	-.-

Alfabeto fuente	Código
Y	-.—
Z	—..
1	.—
2	..—
3	...—
4	....—
5	.....
6	—....
7	—...
8	—..
9	—.
0	—

Al codificar un mensaje en morse, después del código asociado a cada simbolo ha de introducirse una pausa de separación.

**Ejemplo 3: ASCII** El ASCII (*American Standard Code for Information Interchange*) es una codificación bloque de longitud constante (no como el morse, que era de longitud variable) con alfabeto código  $\mathcal{B} = \{0, 1\}$  y  $f(\mathcal{A}) = \{0, 1\}^7$ .

**Ejemplo 4: ASCII Extendido** El ASCII extendido es diferencia del ASCII por  $f(\mathcal{A}) = \{0, 1\}^8$  (la longitud de las palabras es mayor).

**Unicode** Unicode es un alfabeto universal junto con la codificación de sus símbolos (más de un millón) mediante un código bloque de longitud fija e incorpora todos los símbolos del ASCII extendido. A partir de Unicode se encuentran, entre otras, las codificaciones bloques de longitud variable UTF (Unicode Transformation Format): UTF-8 (la más común), UTF-16 y UTF-32.

#### 1.5.4 Codificación Bloque Unívocamente Decodificable

Una codificación de bloque  $f$  se llama **no singular** siempre que

$$\forall a, b \in \mathcal{A} : a \neq b \Rightarrow f(a) \neq f(b)$$

**Ejemplo 1** Sea  $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ ,  $\mathcal{B} = \{0, 1\}$  y  $f$  definida por:

$$f = \begin{cases} f(a_1) &= 0 \\ f(a_2) &= 11 \\ f(a_3) &= 00 \\ f(a_4) &= 01 \end{cases}$$

Es sencillo ver que  $f$  es no singular. Sin embargo, si nos extendemos a palabras, esto ya no se cumple ya que ciertas palabras de  $\mathcal{A}^+$  no son unívocamente decodificables. Por ejemplo:

- $a_1a_1a_1 \neq a_1a_3 \neq a_3a_1$  pero  $000 = f(a_1a_1a_1) = f(a_1a_3) = f(a_3a_1)$
- $a_1a_1a_2 \neq a_3a_2$  pero  $0011 = f(a_1a_1a_2) = f(a_3a_2)$

---

La extensión de orden  $n \geq 1$  de una codificación bloque

$$f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$$

se define como

$$f^{(n)} : (\mathcal{A}^n)^+ \rightarrow \mathcal{B}^+$$

tomando como símbolos las palabras de  $\mathcal{A}^n$  de forma que

$$\forall x \in \mathcal{A}^n : f^{(n)}(x) = f(x)$$

La extensión de orden  $n$  de una codificación bloque también lo es tomando, para cada caso, como alfabeto fuente el correspondiente  $\mathcal{A}$ .

Una codificación bloque  $f$  se dice unívocamente decodificable sí y sólo si su extensión de orden  $n$  es **no singular** para cada  $n \geq 1$ .

Surge una propiedad importante: una codificación bloque es unívocamente decodificable si y sólo si es inyectiva.

Diremos que un código bloque es unívocamente decodificable si proviene de una codificación bloque unívocamente decodificable, lo cual nos permite definir  $f^{-1} : \mathcal{B}^+ \rightarrow \mathcal{A}^+$ .

**Propiedad de factorización única:**

1. Si  $f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$  es una codificación bloque unívocamente decodificable, entonces se tiene que  $\forall x_1, \dots, x_n, y_1, \dots, y_m \in f(\mathcal{A})$ :

$$x_1 \dots x_n = y_1 \dots y_m \Rightarrow (n = m) \wedge (x_i = y_i, \quad i = 1, \dots, n)$$

2.  $f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$  (que es una codificación bloque) es unívocamente decodificable si y solamente si cumple la propiedad mencionada anteriormente, y además es no singular.

La cuestión de si una codificación bloque  $f$  es unívocamente decodificable o no es algorítmicamente decidible (*es decir, que hay un algoritmo para averiguar si es unívocamente decodificable o no*).

El siguiente algoritmo devuelve **true** cuando  $f$  es unívocamente decodificable y **false** en caso contrario:

```

if  $(\exists a, b \in \mathcal{A} : (a \neq b) \wedge (f(a) = f(b)))$  then:
    return false
 $A = \{u : (\exists x, y \in f(\mathcal{A}) : (x \neq y) \wedge (xu = y))\}$ 
if  $(A \cap f(\mathcal{A}) \neq \emptyset)$  then:
    return false
 $A' = \emptyset$ 
while  $A' \neq \emptyset$  :
     $A' = A \cup A'$ 
     $B = \{u : ((\exists x \in f(\mathcal{A}) : xu \in A) \vee (\exists x \in A : xu \in f(\mathcal{A})))\}$ 
     $A = B - A'$ 
    if  $(A \cap f(\mathcal{A}) \neq \emptyset)$  then:
        return false
return true

```

Sea  $f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$  una codificación bloque no singular. Si todas las palabras del código bloque tienen exactamente la misma longitud, entonces el código es unívocamente decodificable.

*Piensa en el Ejemplo 1 de esta sección. ¿De dónde salían los problemas a la hora de decodificar? Surgían del hecho de que no todas las palabras del código bloque tenían la misma longitud*

### 1.5.5 Desigualdad de McMillan

Si  $f : \mathcal{A}^+ \rightarrow \mathcal{B}^+$  es una codificación bloque unívocamente decodificable y  $|\mathcal{B}| = k$ , entonces

$$\sum_{x \in f(\mathcal{A})} k^{-|x|} \leq 1$$

**Ejemplo 1** Recuperando el ejemplo anterior de los siguientes alfabetos fuente ( $\mathcal{A}$ ) y código ( $\mathcal{B}$ ):

$$\begin{aligned} \mathcal{A} &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ \mathcal{B} &= \{0, 1\} \end{aligned}$$

con

$$f(\mathcal{A}) = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001\}$$



La desigualdad de McMillan debería de cumplirse ya que es una codificación bloque unívocamente decodificable:

$$\sum_{x \in f(\mathcal{A})} k^{-|x|} = \sum_{x \in f(\mathcal{A})} 2^{-4} = \frac{10}{16} < 1$$

Vemos rápidamente que se cumple.

¿Qué condición debería cumplirse para tener la igualdad estricta?

Imaginémonos ahora que tenemos

$$\begin{aligned}\mathcal{A} &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f\} \\ \mathcal{B} &= \{0, 1\}\end{aligned}$$

con

$$f(\mathcal{A}) = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$$

(¿no resulta esto familiar? Es la representación en binario del hexadecimal)

Para este caso:

$$\sum_{x \in f(\mathcal{A})} k^{-|x|} = \sum_{x \in f(\mathcal{A})} 2^{-4} = \frac{16}{16} = 1$$

Claramente cuando  $f(\mathcal{A}) \subset \mathcal{B}^+$ , entonces la desigualdad es estricta. Pero cuando  $f(\mathcal{A}) = \mathcal{B}^+$ , entonces tenemos la igualdad.

### 1.5.6 Códigos instantáneos

Empecemos observando un ejemplo. Consideremos la codificación unívocamente decodificable  $f(a) = 0$  y  $f(b) = 01$ . Para decodificar  $x = 01$ , tendríamos que conocer de la existencia del 1, ya que únicamente con el 0 tenemos una ambigüedad y no sabemos si se trata de  $a$  o de  $b$ . Al leer el 1, resolvemos esto y podemos decodificar  $x$  como  $b$ . Es decir, hasta no leer la palabra entera, no hemos podido saber cómo decodificarla.

Esta situación empeora cuanto más larga es la palabra, y puede llevar a que la decodificación sea impracticable.

Consideremos ahora la codificación unívocamente decodificable  $f(a) = 0$ ,  $f(b) = 01$  y  $f(c) = 11$ . Sea  $x = 01^n$  con  $n \geq 1$ . Tenemos que

$$f^{-1}(x) = \begin{cases} ac^m & \text{si } n \text{ es par, con } m = \frac{n}{2} \\ bc^m & \text{si } n \text{ es impar, con } m = \frac{n-1}{2} \end{cases}$$

Veamos una serie de ejemplos:

- $f^{-1}(x) = f^{-1}(01^6) = f^{-1}(01111111) = accc$
- $f^{-1}(x) = f^{-1}(01^{11}) = f^{-1}(011111111111) = bccccc$

Para decodificar, es esencial conocer cuantos 1 tenemos en la palabra  $x$ . Esto no lo podemos averiguar si no leemos toda la palabra primero (independientemente de lo larga que sea esta), antes de decodificar.

Se dice que un código unívocamente decodificable es **instantáneo** cuando es posible decodificar cada símbolo de  $\mathcal{A}$  de cada mensaje sin necesitar más símbolos de  $\mathcal{B}$  de los estrictamente necesarios. Es decir, si

$$h : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

es una codificación bloque unívocamente decodificable, entonces es instantánea siempre que

$$\forall x \in \mathcal{A}^*, \exists u, v \in \mathcal{A}^*, \exists a \in \mathcal{A} : x = uav$$

tal que

$$y = h(x) = h(u)h(a)h(v)$$

Entonces para decodificar  $y$  a  $x$ , para decodificar el segmento  $h(a)$  a  $a$  de forma inmediata, independientemente del sufijo  $h(v)$ , solamente hay que procesar el segmento de  $y$  asociado a  $h(a)$ .

Entonces para descodificar el segmento  $y[|h(u)| + 1 : |h(u)| + n]$  con, en este caso,  $n = |h(a)|$ :

$$\begin{aligned} y[|h(u)| + 1 : |h(u)| + n] &= y[|h(u)| + 1 : |h(u)| + |h(a)|] \\ &= y[|h(u)| + 1 : |h(ua)|] \end{aligned}$$

es suficiente, en cualquier caso, para la correcta decodificación. Eso quiere decir que

$$\forall a, b \in \mathcal{A} : h(a) \text{ es prefijo de } h(b) \Rightarrow a = b$$

Una vez más, diremos que un código bloque es instantáneo cuando proviene de una codificación bloque instantánea. También podemos llamarlo código bloque prefijo o simplemente código prefijo.

Sea  $f : \mathcal{A}^* \rightarrow \mathcal{B}^*$  una codificación de bloque. Diremos que  $f$  es **estable** siempre que:

1. Sea unívocamente decodificable.
2.  $\forall x \in \mathcal{A}^*, f(x) = u$  se cumple que  $\forall v \in \mathcal{B}^*, \text{ si } \exists y \in \mathcal{A}^*, f(y) = uv \Rightarrow y = xz$  para algún  $z \in \mathcal{A}^*$

Tenemos entonces la siguiente propiedad: sea  $f : \mathcal{A}^* \rightarrow \mathcal{B}^*$  una codificación bloque. Entonces,  $f$  es estable si y sólo si es instantánea.

### 1.5.7 Desigualdad de Kraft

Veamos cómo estas características cualitativas se traducen cuantitativamente. Sea la codificación bloque:

$$f : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

con  $\mathcal{A} = \{a_1, \dots, a_n\}$ ,  $\mathcal{B} = \{b_1, \dots, b_k\}$  y  $f(\mathcal{A}) = \{x_1, \dots, x_n\}$ , con  $l_i = |x_i|$ . La desigualdad de Kraft nos da la condición **necesaria** y **suficiente** para que exista un código bloque instantáneo con palabras de su código bloque de longitud  $l_1, \dots, l_n$  sobre un alfabeto con  $k$  símbolos:

$$\sum_{1 \leq i \leq n} k^{l_i} \leq 1$$

Cuando esto es una igualdad estricta, se conoce como la igualdad de Kraft.

Como consecuencia inmediata, a partir de la desigualdad de McMillan, se tiene que para cada código unívocamente decodificable se tiene un código instantáneo con las mismas longitudes de las palabras de su código bloque.

### 1.5.8 Códigos Completos

Sea la codificación de bloque:

$$f : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

diremos que es completa si:

1. Es instantánea.
2. Cumple que  $\forall x \in \mathcal{B}^*, \exists a \in \mathcal{A}^* \Rightarrow ((h(a) \text{ es prefijo de } x) \vee (x \text{ es prefijo de } h(a)))$ .

Tenemos entonces la siguiente propiedad: si  $|\mathcal{A}| \geq 2$  y  $|\mathcal{B}| = 2$ , entonces sólo existe una codificación de bloque completa

$$f' : \mathcal{A}^* \rightarrow \mathcal{B}^*$$

tal que  $\forall a \in \mathcal{A} : |f'(a)| \leq |f(a)|$ .

Nuevamente, un código bloque es completo si proviene de una codificación de bloque completa.

## 1.6 Codificaciones de Fuentes de Información

### 1.6.1 Longitud media de un código

Para un  $\mathcal{A}^*$  y  $\mathcal{B}^*$ , es posible definir más de una codificación instantánea o unívocamente decodificable. Esto requiere entonces que intentemos elegir las más eficientes con el objetivo de tener una transmisión de información óptima.

Un criterio natural de selección (aún cuando no es el único posible) es el de la mínima longitud media.

Sea un código bloque que asocia los símbolos de una **fente de información de memoria nula**  $FI = (\mathcal{A}, P)$  donde  $\mathcal{A} = \{a_1, \dots, a_n\}$  y  $|\mathcal{B}| = k$  mediante la codificación  $f : \mathcal{A}^* \rightarrow \mathcal{B}^*$  con las palabras  $f(a_i) = x_i$ , con  $l_i = |x_i|$ ,  $i = 1, \dots, n$ . Definimos la longitud media como la **esperanza matemática de la longitud de los códigos bloque**:

$$\mathcal{L}_f = \mathbb{E}[l] = \mathbb{E}[|x|] = \mathbb{E}[|f(a)|] = \sum_{i=1}^n |f(a_i)| p(a_i)$$

### 1.6.2 Códigos óptimos

Sea un código bloque unívocamente decodificable que asocia los símbolos de la fuente de memoria nula  $FI = (\mathcal{A}, P)$  con palabras formadas por un alfabeto  $|\mathcal{B}| = k$ . Diremos que es **compacto** u **óptimo** con respecto a la fuente de información si su longitud media es menor o igual que la longitud media de cada uno de los códigos bloque unívocamente decodificables que pueden definirse entre la fuente y alfabetos códigos con  $k$  símbolos.

Surge de aquí la observación de que, puesto que esto incide únicamente sobre las longitudes, la búsqueda puede reducirse por la desigualdad de McMillan a códigos instantáneos.

Además, si se tiene que  $|\mathcal{A}| \geq 2$  y  $|\mathcal{B}| = 2$ , si un código bloque es óptimo, entonces es completo.

También se tiene la propiedad de si  $f$  es un código óptimo, entonces

$$\forall a, b \in \mathcal{A} : p(a) < p(b) \Rightarrow |f(a)| \geq |f(b)|$$

### 1.6.3 Primer Teorema de Shannon (*teorema de la codificación sin ruido*)

Consideremos una fuente de memoria nula  $FI$  cuyos símbolos  $a_1, \dots, a_n$  con probabilidades  $p_1, \dots, p_n$  se codifican cada uno en una palabra de longitud  $l_i$  en un alfabeto con  $k$  símbolos mediante la codificación  $f$ . Entonces se tiene que  $H_k(FI) \leq \mathcal{L}_f$ .

Si suponemos que nos encontramos en el caso de la igualdad, es decir,  $H_k(FI) = \mathcal{L}_f$ :

$$\begin{aligned} H_k(FI) &= \mathcal{L}_f \\ \sum_{i=1}^n p_i \log_k \left( \frac{1}{p_i} \right) &= \sum_{i=1}^n p_i l_i \end{aligned}$$

inferimos que si tomamos longitudes de código  $l_i = |x_i| = |f(a_i)|$ , tendremos que los códigos obtenidos para cada  $a_i$  serán instantáneos, completos y óptimos con una longitud media que coincide con la entropía en base  $k$  de la fuente de información (asumiendo claramente que  $\log_k \left( \frac{1}{p_i} \right)$  es un número entero  $\forall i \leq n$ ).

Veamos un ejemplo: supón que tenemos  $FI = (\{a_1, a_2, a_3\}, \{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\})$ , entonces la codificación de bloque:

$$h : \{a_1, a_2, a_3\}^* \rightarrow \{0, 1\}^*$$

definida tomando las longitudes de los códigos como  $\log\left(\frac{1}{p_i}\right)$ :

$$h = \begin{cases} h(a_1) &= 1 \\ h(a_2) &= 00 \\ h(a_3) &= 01 \end{cases}$$

es instantánea, completa y óptima con  $\mathcal{L}_h = H(FI)$ .

Pero, ¿qué sucede si resulta que  $\log_k\left(\frac{1}{p_i}\right)$  no resultan ser números enteros? Parece intuitivo y apropiado en este contexto redondear el valor obtenido hacia arriba, para elegir  $l_i$  (aunque claramente en este caso, el código obtenido no tiene por qué ser óptimo). Entonces si  $l_i = \left\lceil \log_k\left(\frac{1}{p_i}\right) \right\rceil$ , por definición se tiene que

$$\log_k\left(\frac{1}{p_i}\right) \leq l_i \leq \log_k\left(\frac{1}{p_i}\right) + 1$$

Se puede demostrar que se cumple la desigualdad de Kraft. Tomemos  $\log_k\left(\frac{1}{p_i}\right) \leq l_i$ :

$$\begin{aligned} \log_k\left(\frac{1}{p_i}\right) &\leq l_i \\ k^{\log_k\left(\frac{1}{p_i}\right)} &\leq k^{l_i} \\ \frac{1}{p_i} &\leq k^{l_i} \\ \frac{1}{k^{l_i}} &\leq p_i \\ k^{-l_i} &\leq p_i, \quad \forall 1 \leq i \leq n \\ \sum_{i=1}^n k^{-l_i} &\leq \sum_{i=1}^n p_i \\ \sum_{i=1}^n k^{-l_i} &\leq 1 \end{aligned}$$

En consecuencia, pueden asociárseles códigos bloque instantáneos. Volviendo a la siguiente desigualdad:

$$\log_k\left(\frac{1}{p_i}\right) \leq l_i \leq \log_k\left(\frac{1}{p_i}\right) + 1$$

Multiplicando todos los términos por  $p_i$ :

$$p_i \log_k \left( \frac{1}{p_i} \right) \leq p_i l_i \leq p_i \log_k \left( \frac{1}{p_i} \right) + p_i$$

y sumando ya que la desigualdad se cumple  $\forall 1 \leq i \leq n$ :

$$\begin{aligned} \sum_{i=1}^n p_i \log_k \left( \frac{1}{p_i} \right) &\leq \sum_{i=1}^n p_i l_i \leq \sum_{i=1}^n p_i \log_k \left( \frac{1}{p_i} \right) + \sum_{i=1}^n p_i \\ H_k(FI) &\leq \mathcal{L}_f \leq H_k(FI) + \sum_{i=1}^n p_i \\ H_k(FI) &\leq \mathcal{L}_f \leq H_k(FI) + 1 \end{aligned}$$

Cumpléndose esto también para cualquier código óptimo, lo que constituye la formulación inicial del primer teorema de Shannon.

Puesto que esto puede aplicarse además a cualquier extensión de grado  $m$  de una fuente de memoria nula...

$$\begin{aligned} H_k(FI^{(m)}) &\leq \mathcal{L}_f^m \leq H_k(FI^{(m)}) + 1 \\ mH_k(FI) &\leq \mathcal{L}_f^m \leq mH_k(FI) + 1 \\ H_k(FI) &\leq \frac{\mathcal{L}_f^m}{m} \leq H_k(FI) + \frac{1}{m} \end{aligned}$$

Obteniendo así **el primer teorema de Shannon**, que es uno de los teoremas fundamentales de la teoría de la información.

Una propiedad interesante surge cuando intentamos ver qué pasa cuando la extensión de grado  $m$  de nuestra fuente de información de memoria nula emite palabras largas. Para ello tenemos que ver qué sucede cuando  $m \rightarrow \infty$ :

$$\begin{aligned} \lim_{m \rightarrow \infty} H_k(FI) &\leq \lim_{m \rightarrow \infty} \frac{\mathcal{L}_f^m}{m} \leq \lim_{m \rightarrow \infty} H_k(FI) + \lim_{m \rightarrow \infty} \frac{1}{m} \\ H_k(FI) &\leq \lim_{m \rightarrow \infty} \frac{\mathcal{L}_f^m}{m} \leq H_k(FI) + 0 \end{aligned}$$

Como  $\lim_{m \rightarrow \infty} \frac{\mathcal{L}_f^m}{m}$  queda acotado por arriba y por abajo por  $H_k(FI)$ , hemos demostrado mediante “*el teorema del sándwich*” que

$$\lim_{m \rightarrow \infty} \frac{\mathcal{L}_f^m}{m} = H_k(FI)$$

Nótese que  $\frac{\mathcal{L}_f^m}{m}$  es el **número medio de símbolos del alfabeto código** ( $\mathcal{B}$ ) **empleados en la codificación de un símbolo del alfabeto fuente** ( $\mathcal{A}$ ) cuando se emiten secuencias de longitud  $m$ , es decir, como símbolos del alfabeto  $\mathcal{A}^{(m)}$ .

---

**Ejemplo Aplicado** Sea una fuente de  $FI$  de memoria nula definida por  $\mathcal{A} = \{a_1, a_2, a_3\}$  con  $P = \{\frac{3}{4}, \frac{1}{12}, \frac{2}{12}\}$ , y además  $\mathcal{B} = \{0, 1\}$ . Construyámonos una tabla con python, pandas y numpy:

```
import pandas as pd
import numpy as np
df = pd.DataFrame(
    np.array([[3/4, 1/12, 2/12]]).T,
    columns=["p"],
    index=["a_1", "a_2", "a_3"]
)
```

obteniéndose la tabla:

	p
a1	0.75
a2	0.0833333
a3	0.166667

(si queréis generar tablas para markdown o latex a partir de un `DataFrame` de `pandas`, podéis emplear los métodos `DataFrame.to_markdown()` y `DataFrame.to_latex()`)

Nuestro siguiente paso es calcular la información  $I(a_i) = \log\left(\frac{1}{p_i}\right)$  para cada símbolo:

```
df["log2(1/p)"] = np.log2(1/df["p"])
```

obteniéndose la tabla

	p	log2(1/p)
a1	0.75	0.415037
a2	0.0833333	3.58496
a3	0.166667	2.58496

y a continuación calculamos las longitudes  $l_i = \left\lceil \log\left(\frac{1}{p_i}\right) \right\rceil$ :

```
df["l"] = np.ceil(df["log2(1/p)"])
```

obteniéndose la tabla

	p	log2(1/p)	l
a1	0.75	0.415037	1
a2	0.0833333	3.58496	4
a3	0.166667	2.58496	3

Por lo tanto podemos proponer un código instantáneo dado por estas longitudes y empleando el alfabeto código mencionado anteriormente. Por ejemplo:

	p	log2(1/p)	l	codigo
a1	0.75	0.415037	1	1
a2	0.0833333	3.58496	4	0001
a3	0.166667	2.58496	3	001

Si quisiéramos calcular la longitud media

```
(df["p"]*df["l"]).sum()
```

obteniéndose  $L_f \approx 1.58$ . Si quisiéramos calcular la entropía

```
(df["p"]*df["log2(1/p)"]).sum()
```

obteniéndose  $H(FI) \approx 1.04$ .

#### 1.6.4 Rendimiento y redundancia de un código unívocamente decodificable

Se define en el contexto en el que hablábamos, el rendimiento  $\eta$  de un código como

$$\eta = \frac{H_k(FI)}{L_f}$$

y la redundancia como

$$1 - \eta = 1 - \frac{H_k(FI)}{L_f} = \frac{L_f - H_k(FI)}{L_f}$$

Un código óptimo tiene máximo rendimiento y mínima redundancia.