1. Conjuntos

1.1. ¿Qué es un conjunto?

Un conjunto corresponde a una colección bien definida de objetos. Estos objetos son denominados como **elementos del conjunto** y se dice que estos **pertenecen** (expresado con el simbolo $\in$) a él.
Cuando definimos un conjunto, se usan símbolos de llaves y dentro se colocan todos los objetos que pertenecen al conjunto.
Ejemplo:

$$S = \{1, 2, 3, 4\}$$

1.2. Nociones básicas de los conjuntos

1.2.1. Pertenencia ($\in$)

Si tenemos un conjunto S y un objeto a, se dice que

- $a \in S$ cuando el objeto a se encuentra dentro del conjunto S.
- $a \notin S$ cuando el objeto a **no** se encuentra dentro del conjunto S.

NOTA: Un objeto puede ser un conjunto. Esto significa que un conjunto puede pertenecer a otro conjunto.

1.2.2. Subconjunto ($\subseteq$)

Considerando a un conjunto A y un conjunto B, se dice que A es subconjunto de B si

$$\forall x. x \in A \rightarrow x \in B$$

En otras palabras, A es subconjunto de B si todo elemento presente en A está presente en B también. Cuando esto ocurre, se denota como $A \subseteq B$ (y cuando no, lógicamente se escribe como $A \nsubseteq B$)

1.2.3. Igualdad de conjuntos

Diremos que dos conjuntos A y B son iguales si se cumple que

$$A \subseteq B \land B \subseteq A \quad \text{o, escrito de otra forma} \quad \forall x. x \in A \leftrightarrow x \in B$$

En palabras simples, dos conjuntos son iguales cuando ambos conjuntos tienen exactamente los mismos objetos, sin ninguno que pertenezca a un conjunto y no a otro. Esto se expresa como $A = B$ (y cuando no, $A \neq B$).

1.2.4. Conjunto vacío

Existe un conjunto único $\emptyset$, el cual llamamos *conjunto vacío*, el cual cumple que

$$\forall x. x \notin \emptyset$$

1.3. Descripción de un conjunto

1. Por extensión: Este es el método más básico y el cual se describió anteriormente. Simplemente se listan todos los contenidos del conjunto entre llaves.
   Ejemplo:
   $$S = \{1, 2, 3, 4\}$$

2. Por comprensión: Se define una propiedad $\delta(x)$ en algún lenguaje formal que solo cumplen los elementos del conjunto.
   Ejemplo:
   $$S = \{x | \delta(x) \text{ es verdadero}\}$$
   Ejemplo un poco más creativo:
   $$P = \{x | \forall x.x \text{ es par}\}$$

1.4. Paradoja de Russell o Paradoja del Barbero (1901)

Esta es una paradoja enunciada por Bertrand Russell[1]. Para comenzar, se define el siguiente conjunto

$$S^* = \{B \mid B \notin B\}$$

S^* corresponde al "conjunto de todos los conjuntos que no se contienen a si mismos como miembros". Ahora, recordemos que por la definición de lo que es un conjunto, esto es equivalente a

$$\forall B.B \in S^* \leftrightarrow B \notin B$$

B. Russell (1872 - 1970)

O sea, "cada conjunto es elemento de B si y solo si no es elemento de si mismo". Debido a que B es un conjunto, lo podemos sustituir de la siguiente forma

$$S^* \in S^* \leftrightarrow S^* \notin S^*$$

lo cual es una contradicción.
Esto puede ser un poco complicado de entender así. Intentemos entenderlo con la historia del Barbero.

Una definición como esta es bastante problemática. El problema aquí es *"considerar definiciones que se referencian a si mismas"*. Esto nos deja de lección que no todas las definiciones son válidas en la teoría de conjuntos.[2]

[1]Esta sección contiene información extraída del siguiente artículo de Wikipedia.

[2]Todas las definiciones que se verán durante el curso son válidas, pero esto es una lección de que no siempre es así.

1.5. Operaciones sobre conjuntos

- Unión ($\cup$): $A \cup B$ son todos los elementos que se encuentran en A o en B.

$$A \cup B = \{x | x \in A \vee x \in B\}$$

- Intersección ($\cap$): $A \cap B$ son todos los elementos que se encuentran en A y B al mismo tiempo.

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

- Diferencia ($\backslash$): $A \backslash B$ son todos los elementos que se encuentran en A y no en B.

$$A \backslash B = \{x | x \in A \wedge x \notin B\}$$

- Complemento (A^C): A^C corresponde a todos los elementos que no se encuentran en A.

$$A^C = \{x | x \notin A\}$$

1.5.1. Propiedades de las operaciones sobre conjuntos

Para conjuntos A, B y C y un universo $\mathcal{U}$ tenemos las siguientes propiedades

1. Asociatividad:
$$\begin{aligned} A \cup (B \cup C) &= (A \cup B) \cup C \\ A \cap (B \cap C) &= (A \cap B) \cap C \end{aligned}$$
2. Conmutatividad:
$$\begin{aligned} A \cup B &= B \cup A \\ A \cap B &= B \cap A \end{aligned}$$
3. Idempotencia:
$$\begin{aligned} A \cup A &= A \\ A \cap A &= A \end{aligned}$$
4. Absorción:
$$\begin{aligned} A \cup (A \cap B) &= A \\ A \cap (A \cup B) &= A \end{aligned}$$
5. Distributividad:
$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned}$$
6. De Morgan
$$\begin{aligned} (A \cup B)^C &= A^C \cap B^C \\ (A \cap B)^C &= A^C \cup B^C \end{aligned}$$
7. Elemento neutro:
$$\begin{aligned} A \cup \emptyset &= A \\ A \cap \mathcal{U} &= A \end{aligned}$$
8. Dominación:
$$\begin{aligned} A \cap \emptyset &= \emptyset \\ A \cup \mathcal{U} &= \mathcal{U} \end{aligned}$$
9. Elemento inverso:
$$\begin{aligned} A \cup A^C &= \mathcal{U} \\ A \cap A^C &= \emptyset \end{aligned}$$

1.5.2. Paréntesis y precedencia

Se asumirá el siguiente orden de precedencia entre operadores

Operadores	Precedencia
$\cdot^C$	1
$\cap$	2
$\cup$	3

1.5.3. Operaciones generalizadas

- Unión generalizada: $\bigcup \mathcal{S}$ son todos los elementos que pertenecen a algún elemento de $\mathcal{S}$.

$$\bigcup \mathcal{S} = \{x \mid \exists A.\ A \in \mathcal{S} \wedge x \in A\} = \bigcup_{A \in \mathcal{S}} A = \bigcup_{i=1}^{k} A_i$$

- Intersección generalizada: $\bigcap \mathcal{S}$ son todos los elementos que pertenecen a todos los elementos de $\mathcal{S}$

$$\bigcap \mathcal{S} = \{x \mid \forall A.\ A \in \mathcal{S} \rightarrow x \in A\} = \bigcap_{A \in \mathcal{S}} A = \bigcap_{i=1}^{k} A_i$$

1.6. Conjunto Potencia

Para un conjunto A, el conjunto potencia $\mathcal{P}(A)$ de todos los subconjuntos de A es definido como:

$$\mathcal{P}(A) = \{X \mid X \subseteq A\}$$

En palabras simples, el conjunto potencia de A corresponde al conjunto que contiene a todos los subconjuntos de A.

1.7. Cardinalidad de un conjunto

La cardinalidad del conjunto corresponde a la cantidad de elementos distintos contenidos en un conjunto

$$A = \# \text{ de elementos distintos en } A$$

Teorema:

$$A = \{1, 2, 3, \ldots, n\} \rightarrow |\mathcal{P}(A)| = 2^n$$

En otras palabras, la cardinalidad del conjunto potencia de A corresponde a 2 elevado a la cardinalidad del conjunto A.

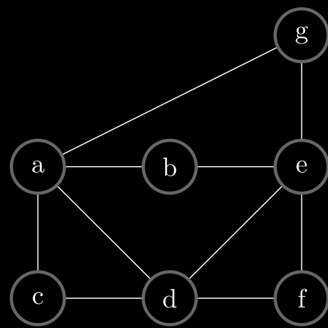
1.8. Grafos como conjuntos

Pero ... ¿qué es un grafo?[3]

Un grafo consiste en un conjunto de objetos llamados vertices y nodos, los cuales se encuentran unidos entre sí.

[3]Información extraida del siguiente artículo de Wikipedia

Definición: Un grafo G sobre el dominio V es un subconjunto $E \subseteq \mathcal{P}(V)$, tal que para todo $e \in E$ se cumple que $|e| = 2$.



$$\begin{aligned} V &= \{\text{a, b, c, d, e, f, g}\} \\ E &= \{\ \{\text{a,b}\}, \{\text{a,c}\}, \{\text{a,d}\}, \{\text{a,g}\}, \{\text{b,e}\}, \{\text{c,d}\}, \{\text{d,e}\}, \{\text{d,f}\}, \{\text{e,f}\}, \{\text{e,g}\}\ \} \end{aligned}$$

En un grafo, los elementos que están presentes en V los llamamos *vértices o nodos*, y los elementos en E los llamamos *aristas*.

1.9. Pares Ordenados

Para dos elementos a y b se define el par ordenado (a, b) como

$$(a, b) = \{\{a\}, \{a, b\}\}$$

De forma mas informal, decimos que (a, b) es un par ordenado si es que el primer elemento (a) se distingue del segúndo elemento (b). Por esto, tenemos que $(a, b) \neq (b, a)$.
Generalización: Se define una *n-tupla ordenada* como

$$(a_1, a_2, \ldots, a_n) = ((a_1, a_2, \ldots, a_{n-1}), a_n)$$

1.10. Producto Cartesiano

Sea A y B conjuntos. El producto cartesiano se define como

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

Si tenemos n conjuntos $A_1, A_2, \ldots, A_n$, el producto cartesiano generalizado es

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, \ldots, a_n) \mid a_i \in A_i\}$$

2. Relaciones

Considerando un conjunto A y B, R es una relación binaria sobre A y B si se cumple que

$$R \subseteq A \times B$$

Es posible que $B = A$, entonces se dice que R es una relación binaria sobre A

$$R \subseteq A \times A$$

Notación: Para una relación R y un par (a, b), se usará la siguiente notación

$$\left.\begin{array}{c} (a, b) \in R \\ \text{o} \\ a\ R\ b \end{array}\right\} \quad (a, b) \text{ pertenece a la relación } R$$

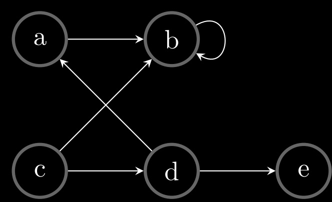
$$\left.\begin{array}{c}(a,b) \notin R \\ \text{o} \\ a \not R\, b\end{array}\right\} \quad (a,b) \text{ no pertenece a la relación } R$$

2.1. Representación de Relaciones

2.1.1. Grafos dirigidos

Toda relación binaria R sobre A se puede ver como un grafo dirigido $G_R = (A, R)$
Ejemplo:

$$A = \{a, b, c, d, e\} \quad ; \quad R = \{(a, b), (b, b), (c, b), (c, d), (d, a), (d, d), (d, e)\}$$



2.1.2. Matrices sobre bits / Representación Matricial

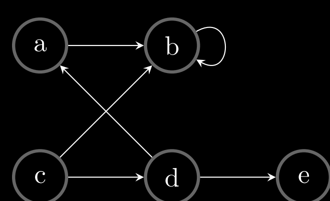
Sea $A = \{a_1, a_2, \ldots, a_n\}$ un conjunto y R una relación binaria sobre A. Definimos la matriz M_R que representa a la relación R como

$$M_R[i,j] = \begin{cases} 1 & \text{si } a_i \; R \; a_j \\ 0 & \text{si } a_i \not R \; a_j \end{cases}$$

Ejemplo:

$$A = \{a, b, c, d, e\} \quad ; \quad R = \{(a, b), (b, b), (c, b), (c, d), (d, a), (d, d), (d, e)\}$$

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



Operaciones sobre matrices: Dada dos matrices de bits M y N de tamaño n, entonces

$$\begin{array}{rcl} (M \vee N)[i,j] & = & M[i,j] \vee N[i,j] \\ (M \wedge N)[i,j] & = & M[i,j] \wedge N[i,j] \\ (\neg M)[i,j] & = & \neg M[i,j] \\ M[i,j] & \leq & N[i,j] \end{array} \qquad \text{para todo } 1 \leq i \leq n \text{ y } 1 \leq j \leq n \text{ asumiendo que } 0 \leq 1.$$

2.2. Operaciones entre relaciones

Sea A un conjunto y $R \subseteq A \times A$. Se definen las siguientes operaciones

- Proyección 1: $\pi_1(R)$ son todos los elementos que estan en la primera componente de R (No se consideran los duplicados).

$$\pi_1(R) = \{x \mid \exists y \in A.(x, y) \in R\}$$

- Proyección 2: $\pi_2(R)$ son todos los elementos que están en la segúnda componente de R (No se consideran los duplicados).

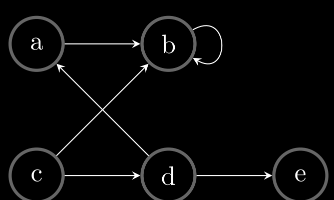
$$\pi_2(R) = \{y \mid \exists x \in A.(x, y) \in R\}$$

- Inverso: R^{-1} son todos los pares (x, y) de la relación R, pero al revés (como (y, x)). Más formalmente escrito como:
$$R^{-1} = \{(x, y) \mid (y, x) \in R\}$$
- Composición: $R_1 \circ R_2$ son todos los pares (x, y) que cumplen con que exista un z tal que $(x, z) \in R_1$ y $(z, y) \in R_2$
$$R_1 \circ R_2 = \{(x, y) \mid \exists z.(x, z) \in R_1 \land (z, y) \in R_2\}$$
- Unión: $R_1 \cup R_2$ son todos los pares que existen en R_1 o R_2. De forma más formal:
$$R_1 \cup R_2 = \{(x, y)|(x, y) \in R_1 \lor (x, y) \in R_2\}$$
- Intersección: $R_1 \cap R_2$ son todos los pares que existen en R_1 y R_2 de forma simultánea. Más formalmente expresado como:
$$R_1 \cap R_2 = \{(x, y)|(x, y) \in R_1 \land (x, y) \in R_2\}$$
- Relación identidad: I_A solamente contiene a los pares (x, x) para los x que existen en A.
$$I_A = \{(x, x)|x \in A\}$$

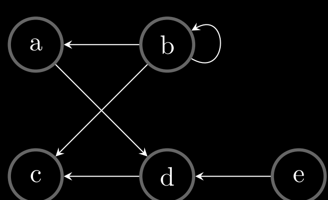
Debido a que esta parte puede llegar a ser un poco complicada de entender, vamos a mostrar una serie de ejemplos en base al conjunto y relación que tenemos de antes.

$$A = \{a, b, c, d, e\} \quad ; \quad R = \{(a, b), (b, b), (c, b), (c, d), (d, a), (d, d), (d, e)\}$$

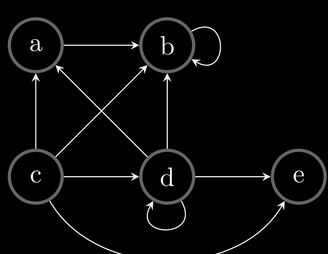
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



$$\pi_1(R) = \{a, b, c, d\} \quad ; \quad \pi_2(R) = \{a, b, d, e\}$$
$$R^{-1} = \{(b, a), (b, b), (b, c), (d, c), (a, d), (d, d), (e, d)\}$$



$$R \circ R = \{(a, b), (b, b), (c, b), (c, a), (c, d), (c, e), (d, b), (d, a), (d, d), (d, e)\}$$



2.3. Tipos de Relaciones

- Refleja: $\forall a \in A.(a, a) \in R$
- Irrefleja: $\forall a \in A.(a, a) \notin R$
- Simétrica: $\forall a, b \in A.(a, b) \in R \to (b, a) \in R$
- Asimétrica: $\forall a, b \in A.(a, b) \in R \to (b, a) \notin R$
- Antisimétrica: $\forall a, b \in A.((a, b) \in R \wedge (b, a) \in R) \to a = b$
- Transitiva: $\forall a, b, c \in A.((a, b) \in R \wedge (b, c) \in R) \to (a, c) \in R$
- Conexa: $\forall a, b \in A.(a, b) \in R \vee (b, a) \in R$

2.3.1. Tipos de Relaciones caracterizadas con operaciones

Se considera un conjunto A y una relación binaria $R \subseteq A \times A$

- R es refleja si y solo si $I_A \subseteq R$.
- R es irrefleja si y solo si $R \cap I_A = \emptyset$.
- R es simétrica si y solo si $R = R^{-1}$.
- R es asimétrica si y solo si $R \cap R^{-1} = \emptyset$.
- R es antisimétrica si y solo si $R \cap R^{-1} \subseteq I_A$.
- R es transitiva si y solo si $R \circ R \subseteq R$.
- R es conexa si y solo si $R \cup R^{-1} = A \times A$.

2.4. Ordenes Parciales

Si tenemos un conjunto A y una relación $R \subseteq A \times A$, decimos que R es un orden parcial si es que cumple que

- R es refleja: $I_A \subseteq R$
- R es antisimétrica: $R \cap R^{-1} \subseteq I_A$
- R es transitiva: $R \circ R \subseteq R$

El orden parcial se denota como $(A, \preceq)$.

2.4.1. Ordenes Totales

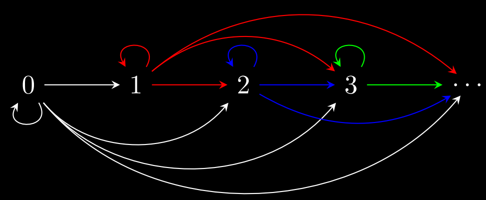
Si tenemos un conjunto A y un orden parcial $(A, \preceq)$, se dice que este orden parcial es un orden total si es que se cumple que la relación $R \subseteq A \times A$ es conexa. En otras palabras, un orden total debe cumplir con

- R es refleja: $I_A \subseteq R$
- R es antisimétrica: $R \cap R^{-1} \subseteq I_A$
- R es transitiva: $R \circ R \subseteq R$
- R es conexa: $R \cup R^{-1} = A \times A$

2.4.2. Representación de un Orden Parcial

¿Cómo se ve un orden parcial representado con grafos?

Consideremos como ejemplo, el orden $\leq$ sobre $\mathbb{N}$



Este grafo queda demasiado enredado. Normalmente, para representar ordenes parciales, se usa el **Diagrama de Hasse**, el cual consiste en un diagrama de grafos, tal como el anterior, pero omitiendo los loops y las aristas transitivas[4].

Considerando la descripción anterior, el diagrama de Hasse del orden $\leq$ sobre $\mathbb{N}$ se vería así

$$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow \cdots$$

2.5. Elementos extremos

- Cotas superiores: $c \in A$ es una cota superior de S si y solo si $\forall y \in S . y \preceq c$
- Maximales: $\hat{x} \in S$ es un maximal si y solo si $\forall y \in S . \hat{x} \preceq y \rightarrow \hat{x} = y$
- Máximo: $x^{\uparrow} \in S$ es un máximo si y solo si $\forall y \in S . y \preceq x^{\uparrow}$
- Cotas inferiores: $c \in A$ es una cota inferior de S si y solo si $\forall y \in S . c \preceq y$
- Minimales: $\breve{x} \in S$ es un minimal si y solo si $\forall y \in S . y \preceq \breve{x} \rightarrow \breve{x} = y$
- Mínimo: $x^{\downarrow} \in S$ es un máximo si y solo si $\forall y \in S . x^{\downarrow} \preceq y$

2.5.1. Sobre minimales y minimos

Sea $(a, \preceq)$ un orden parcial y $S \subseteq A$ distinto de $\emptyset$.

- Si S tiene un elemento mínimo, entonces ¿es único?
  Si. Es único.[5]
- ¿Tiene S siempre un mínimo?
  No. Es posible tener conjuntos que no tienen mínimos
- Si x es mínimo, entonces ¿es x minimal?
  Si. Un mínimo es siempre minimal también.[6]
- Si x es minimal, entonces ¿es x es mínimo?
  No necesariamente.
- ¿Tiene S siempre un elemento minimal?
  No necesariamente.

Con respecto a esto, también es verdadero respecto a maximales y máximos.

[4] Con esto nos referimos a que se omite $(a, b) \in \preceq$ si es que existe c de forma que $(a, c) \in \preceq$ y $(b, c) \in \preceq$

[5] Se puede consultar la demostración en el anexo, subsección 2

[6] Se puede consultar la demostración en el anexo, subsección 3

2.6. Ínfimos

Decimos que c^* es infimo si es una cota inferior de S y de todas las cotas inferiores de S, c^* es la cota inferior mayor.

2.7. Supremos

Decimos que c^* es supremo si es una cota superior de S y de todas las cotas superiores de S, c^* es la cota superior menor.

2.8. Clausuras

Sea A un conjunto y $R \subseteq A \times A$ una relación

2.8.1. Clausura Refleja

Una relación $R^r \subseteq A \times A$ es la clausura refleja de R si

- $R \subseteq R^r$
- R^r es refleja.
- Para toda relación refleja R' con $R \subseteq R'$ se cumple $R^r \subseteq R'$.

R^r es la menor relación refleja que contiene a R.

2.8.2. Clausura Transitiva

Una relación $R \subseteq A \times A$ es la clausura transitiva de R si

- $R \subseteq R^t$
- R^t es transitiva.
- Para toda relación refleja R' con $R \subseteq R'$ se cumple $R^t \subseteq R'$.

R^t es la menor relación transitiva que contiene a R.

2.8.3. Cálculo de clausuras

$$R^r = R \cup I_A \quad ; \quad R^t = \bigcup_{i=1}^{\infty} R^i$$

donde

Símbolo	Significa
I_A	Relación identidad
R^i	$R^{i-1} \circ R$
R^2	$R \circ R$

3. Relaciones de Equivalencia

Si se tiene que A un conjunto y $R \subseteq A \times A$ una relación binaria, se dice que R es una relación de equivalencia si cumple con ser una relación

- Refleja: $\forall a \in A.(a, a) \in R$
- Simétrica: $\forall a, b \in A.(a, b) \in R \rightarrow (b, a) \in R$
- Transitiva: $\forall a, b, c \in A.((a, b) \in R \wedge (b, c) \in R) \rightarrow (a, c) \in R$

Ejemplo práctico: Personas y cumpleaños

$$P = \{p | p \text{ es una persona}\}$$

$$C \subseteq P \times P \text{ tal que } (p_1, p_2) \in C \text{ ssi } p_1 \text{ está de cumpleaños el mismo día que } p_2$$

3.1. Particiones

Sea A un conjunto y $\mathcal{S} \subseteq 2^A$. Se dice que $\mathcal{S}$ es una partición de A si

- Todos los elementos de $\mathcal{S}$ son distintos de vacío: $\forall X \in \mathcal{S}.X \neq \emptyset$
- La unión de todos los elementos de $\mathcal{S}$ es igual a A: $\cup\mathcal{S} = A$
- Todos los elementos de $\mathcal{S}$ son distintos de a pares: $X, Y \in \mathcal{S}.X \neq Y \rightarrow X \cap Y = \emptyset$

3.2. Clases de equivalencia

Sea A un conjunto, $\simeq \subseteq A \times A$ una relación de equivalencia y $x \in A$. La clase de equivalencia de x según $\simeq$ corresponde a

$$[x]_{\simeq} = \{y \in A | x \simeq y\}$$

En palabras simples, $[x]_{\simeq}$ corresponde a todos los elementos presentes en A que son iguales a x.

3.2.1. Propiedades de las clases de equivalencia

- $\forall x \in A.x \in [x]_{\simeq}$
- $x \simeq z \leftrightarrow [x]_{\simeq} = [z]_{\simeq}$
- $x \not\simeq z \rightarrow [x]_{\simeq} \cap [z]_{\simeq} = \emptyset$

3.3. Conjunto cuociente

Se tiene A un conjunto y $\simeq \subseteq A \times A$ una relación de equivalencia. El conjunto cuociente de A, expresado como $A/\simeq$ significa

$$A/\simeq = \{x \in A\}$$

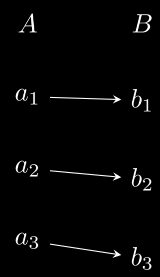
El conjunto cuociente de A corresponde a una partición de A.

4. Funciones

Las funciones son un tipo de relación. Más especificamente, $f \subseteq A \times B$ (con A y B conjuntos no vacios) es una función si es que

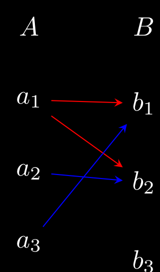
- $\forall a \in A.\exists b \in B.(a, b) \in f$ o que para todo elemento de A existe un elemento en B, tal que f los relacione.
- $\forall a \in A.\forall b_1, b_2 \in B.((a, b_1) \in f \wedge (a, b_2) \in f) \rightarrow b_1 = b_2$ o que todo elemento de A se relacione con solamente un elemento de B.

De forma más gráfica, una función se vería así

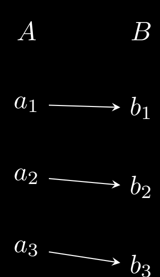


$f_1 : A \rightarrow B$

Por otro lado, estas cosas no serían funciones



$f_2 : A \rightarrow B$: Esto no es función, debido a que un elemento de A se relaciona con más de un elemento de B (También no relaciona todos los elementos de A)
$f_3 : A \rightarrow B$: Esto no es función, debido a que no todos los elementos de A se relacionan con algún elemento de B.
Una función siempre se puede ver como tabla, cosa que en la mayoría de casos hace que sea más facil visualizarla y entenderla que los diagramas de antes. Por ejemplo, para la f_1 definida antes



A	B
a_1	b_1
a_2	b_2
a_3	b_3

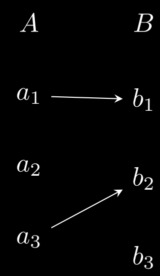
4.1. Función parcial

Una función parcial es similar a una función, sin embargo, podemos decir que es un poco más "permisiva", ya que a diferencia de la función, no exige que todos los elementos de A tengan relación con elementos de B.

De manera más formal, $f \subseteq A \times B$ es una función parcial si

$$\forall a \in A. \forall b_1, b_2 \in B.((a, b_1) \in f \land (a, b_2) \in f) \to b_1 = b_2$$

Un ejemplo sencillo de una función parcial que no es función



Todo elemento de A que se relaciona con un elemento de B, cumple con tener una unica relación, y no varias relaciones, como especificamos anteriormente. Sin embargo, hay un elemento de A que no se relaciona. Por esto, no es función, pero si es una función parcial.

4.2. Notación

Considerando $f \subseteq A \times B$, se usará la siguiente notación

$$\begin{array}{rl} f : A \to B & \text{para } f \text{ es una función de } A \text{ a } B. \\ f : A \rightharpoonup B & \text{para } f \text{ es una función parcial de } A \text{ a } B. \\ f(a) = b & \text{para decir que } (a, b) \in f. \end{array}$$

4.3. Dominio e imagen de una función

$$\begin{array}{rccccc} \text{Dominio:} & \mathrm{dom}(f) & = & \pi_1(f) & = & \{a \in A | \exists b \in B.(a, b) \in f\} \\ \text{Imagen:} & \mathrm{img}(f) & = & \pi_2(f) & = & \{b \in B | \exists a \in A.(a, b) \in f\} \end{array}$$

Proposición: Si se tiene una función parcial $f : A \rightharpoonup B$, entonces f será una función si $\mathrm{dom}(f) = A$.

4.4. Tipos de funciones

Asumiendo una función $f : A \to B$, las funciones se clasifican como:

- Inyectiva: Una función es inyectiva si es que no existen dos elementos de A con la misma imagen.

$$\forall a_1, a_2 \in A. f(a_1) = f(a_2) \to a_1 = a_2$$

- Sobreyectiva: Una función es sobreyectiva si es que todo elemento de B tiene una preimagen.

$$\forall b \in B. \exists a \in A.(a, b) \in f$$

- Biyectiva: Inyectiva y sobreyectiva simultaneamente.

4.5. Función inversa, composición y caracterización

Similar a las relaciones, las funciones tienen una forma inversa y se puede componer.

- Función inversa: f^{-1} son todos los pares (a, b) tal que (b, a) existe en f.

$$f^{-1} = \{(a, b) | (b, a) \in f\}$$

- Composición de funciones: $f_1 \circ f_2$, con $f_1 : A \to B$ y $f_2 : B \to C$, son todos los elementos (a, c) tal que exista un c que permita crear un "puente" entre f_1 y f_2.

$$f_1 \circ f_2 = \{(a, c) | \exists b \in B.(a, b) \in f_1 \wedge (b, c) \in f_2\}$$

$$f_2(f_1(a)) = c$$

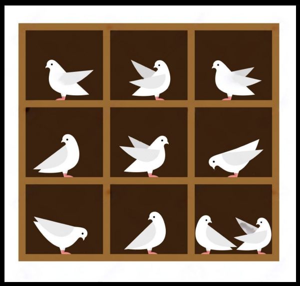
Esto nos entrega algunas propiedades interesantes. Si se tiene la función $f : A \to B$, entonces

$$\begin{array}{rll} f \text{ es inyectiva} & \text{si, y solo si,} & f^{-1} \text{ es una función parcial.} \\ f \text{ es sobreyectiva} & \text{si, y solo si,} & \mathrm{img}(f) = B. \\ f \text{ es biyectiva} & \text{si, y solo si,} & f^{-1} \text{ es una función.} \end{array}$$

$$\begin{array}{rl} \text{Si } f_1 \text{ y } f_2 \text{ son inyectivas,} & \text{entonces } f_1 \circ f_2 \text{ es inyectiva también.} \\ \text{Si } f_1 \text{ y } f_2 \text{ son sobreyectivas,} & \text{entonces } f_1 \circ f_2 \text{ es sobreyectiva también.} \end{array}$$

4.6. Principio del Palomar

"Si N palomas se distribuyen en M palomares, y tengo más palomas que palomares ($M > N$), entonces al menos habrá un palomar con más de una paloma."



Para las cosas con las que tendremos que lidiar en el curso, el principio del palomar dice que, considerando un $f : A \to B$ y $|B| < |A|$, entonces f **NO** es inyectiva.

$$\exists a_1, a_2 \in A. a_1 \neq a_2 \wedge f(a_1) = f(a_2)$$

5. Cardinalidad

Primero que nada, debemos recordar que el tamaño de un conjunto es definido como la cantidad de elementos presentes en el conjunto.

5.1. Equinumerosidad (Conjuntos equinumerosos)

Dos conjuntos A y B son equinumerosos si tienen la misma cantidad de elementos. Visto de otra forma, A y B son equinumerosos si es que existe una biyección $f : A \to B$. Si son equinumerosos, se denota como $|A| = |B|$.

5.2. Equinumerosidad como Relación de Equivalencia

La relación $| \cdot | = | \cdot |$ es una relación de equivalencia. Esto significa que es una relación refleja, simétrica y transitiva.
Entonces, para un conjunto A, se denotará su clase de equivalencia como $|A|$, según la relación $| \cdot | = | \cdot |$.

5.3. Cardinalidad de conjuntos finitos

Se dice que un conjunto es finito si es que existe un n tal que $|A| = |\{0, 1, 2, 3, \ldots, n-1\}|$. Cuando esto ocurre, se dice que su cardinalidad es n.

$$\exists n.|A| = |\{0, 1, 2, 3, \ldots, n-1\}| \rightarrow A \text{ es finito y } |A| = n$$

5.4. Cardinalidad de conjuntos infinitos

Hagamos un analisis de un ejemplo: Supongamos un conjunto $\mathbb{P}$ compuesto por todos los numeros pares

$$\mathbb{P} = \{0, 2, 4, 6, 8, \ldots\}$$

Este conjunto es infinito. Si lo pensamos, $\mathbb{N}$ (el conjunto de los naturales) también es infinito. ¿Tendrán $\mathbb{P}$ y $\mathbb{N}$ la misma cardinalidad? Nuestra definición de equinumerosidad nos será útil. Hay una biyección entre ambos conjuntos

$$\mathbb{N} = \{ \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \ldots\}$$

$$\mathbb{P} = \{ \quad 0 \quad 2 \quad 4 \quad 6 \quad 8 \quad \ldots\}$$

Con la biyección $f : \mathbb{N} \rightarrow \mathbb{P}$, donde $f(n) = 2n$, se tiene que $|\mathbb{N}| = |\mathbb{P}|$

5.5. Conjuntos numerables

Se dice que un subconjunto A es numerable si tiene la misma cardinalidad que un subconjunto de $\mathbb{N}$. Más formalmente, un conjunto A es numerable si

$$\exists S \subseteq \mathbb{N}.|A| = |S|$$

También se puede decir que un conjunto A es numerable si y solo si existe una secuencia ordenada del tipo $a_1, a_2, a_3, \ldots$, en donde $\forall i \neq j.a_i \neq a_j$ y $\forall a \in A.\exists i \in \mathbb{N}.a = a_i$. En otras palabras, todos los elementos de la secuencia ordenada deben de ser diferentes entre si y todo elemento de la secuencia debe existir en A (y estar "indexado" con numeros naturales).
Destacar que **NO TODOS** los conjuntos son numerables. Por ejemplo, el conjunto $\mathbb{R}$ no es numerable.

5.6. Paradoja del Gran Hotel de Hilbert - Paradoja del Hotel Infinito

Esta paradoja fue enunciada por *David Hilbert (1862-1943)*, la cual habla de un hotel ficticio con una cantidad infinita de habitaciones, el cual se encuentra lleno. Llega un nuevo huésped al hotel y es su tarea entregarle una habitación. ¿Cómo puede asignarle una habitación? ¡Desplazando a cada uno de los huéspedes actuales a la habitación siguiente! El huésped de la habitación 1 se deberá ir a la habitación 2 y así.
Si está interesado en comprender mejor esta paradoja, aconsejo mirar este video de Veritasium[7] en donde se habla de algunas de las implicancias que tiene esta paradoja. Este video se encuentra en inglés, pero una versión en español[8] se encuentra disponible también.

[7] El link del video es: https://www.youtube.com/watch?v=OxGsU8oIWjY
[8] El link del video es: https://www.youtube.com/watch?v=4c8vG-mxuao

5.7. Teorema de Cantor

Teniendo un conjunto A no vacio, el Teorema de Cantor nos dice
que no existe una biyección entre A y su conjunto potencia 2^A.
Para demostrar este teorema, *Georg Cantor (1845 - 1918)* creó una técnica la cual se conoce como el método de diagonalización de Cantor.[9]

5.8. ¿Hay algún infinito entremedio?

Se tiene la hipótesis siguiente
No existe ningún conjunto A tal que: $|\mathbb{N}| < |A| < |\mathbb{R}| \sim$ David Hilbert Actualmente no se ha logrado demostrar ni desmentir esta hipótesis, aunque cabe destacar que con los axiomas de teoria de conjuntos:

- 1940 - Kurt Gödel: No se puede demostrar que la hipótesis es falsa.
- 1963 - Paul Cohen: No se puede demostrar que la hipótesis es verdadera.

5.9. Solución a los problemas de decisión

Consideremos programas escritos en algunos lenguajes de programación, como *Python* o *C++*
Sea $\mathcal{I}$ un conjunto de inputs y $P : \mathcal{I} \to \{0, 1\}$ un problema de decisión.
Nos referimos a una solución `Program`, correspondiente a un programa computacional de Python o C++, el cual recibe inputs en $\mathcal{I}$ y retorna un output 0 o 1. Una solución `Program` es una solución a un problema de decisión P si para todo input $X \in \mathcal{I}$ se cumple que

$$P(X) = 1 \leftrightarrow \text{al ejecutar } \texttt{Program} \text{ con } X\text{, este retorna } 1$$

Ejemplo: ¿Es este número primo?

$$\text{Primo} : \mathbb{N} \to \{0, 1\} \quad ; \quad \text{Primo}(n) = 1 \text{ si y solo si n es un número primo}$$

Se puede modelar el siguiente programa de Python para resolver el problema

```python
def is_prime(n):
    if n % 2 == 0 and n > 2:
        return 0
    for i in range(3,n):
        if n % i == 0:
            return 0
    return 1
```

Recordemos que el codigo que vemos acá arriba corresponde a una abstracción solamente. El computador solamente entiende una secuencia de 1s y 0s, por lo que podemos representar todo programa computacional que es capaz de resolver un problema, usando solamente una palabra de ceros y unos. El conjunto de todas las palabras $\{0, 1\}^*$ es numerable, por lo que la cantidad de programas computacionales que son posibles también es numerable.
Ahora que sabemos que la cantidad de programas computacionales posibles es una cantidad numerable, *¿cuántos problemas de decisión existen?* Podemos decir que un problema de decisión se puede representar como una función $P : \{0, 1\}^* \to \{0, 1\}$. Se define $\mathcal{P}$ como el conjunto de todos los problemas de decisión, de forma que $\mathcal{P} = \{P : \{0, 1\}^* \to \{0, 1\}\}$. Se tiene que los conjuntos $\mathcal{P}$ y $2^{\{0,1\}^*}$ son equinumerosos, por lo que la cantidad de problemas de decisión no es numerable.
Es por esto que hay problemas de decisión los cuales no pueden ser modelados en un computador. Con esto, nos referimos a que hay problemas que no tienen algoritmo.

[9] Puede encontrar una demostración del Teorema de Cantor usando su método en el anexo de este resumen.

6. Notación Asintótica

Con $g : \mathbb{N} \to \mathbb{R}$ una función cualquiera, se pueden definir las siguientes notaciones...

6.1. Notación $\mathcal{O}$

El conjunto $\mathcal{O}(g)$ corresponde al conjunto de todas las funciones tales que

$$\exists c > 0.\exists n_0.\forall n \geq n_0.f(n) \leq c \cdot g(n)$$

En otras palabras, corresponde a todas las funciones tales que sean menores que $g(n)$ ponderado con un algún valor positivo, desde algún punto en adelante

6.1.1. Propiedades de la notación

1. $\forall a, b > 1. \log_a(n) \in \mathcal{O}(log_b(n))$
2. $\forall a < b \in \mathbb{N}.a^n \in \mathcal{O}(b^n) \wedge b^n \notin \mathcal{O}(a^n)$
3. $\forall a \in N.a^n \in \mathcal{O}(n!) \wedge n! \notin \mathcal{O}(a^n)$
4. $n! \in \mathcal{O}(2^{n \cdot \log(n)})$

6.1.2. Jerarquía de la notación

Notación	Nombre
$\mathcal{O}(1)$	Constante
$\mathcal{O}(\log(n))$	Logarítmico
$\mathcal{O}(n)$	Lineal
$\mathcal{O}(n \log(n))$	n log(n)
$\mathcal{O}(n^2)$	Cuadrático
$\mathcal{O}(n^3)$	Cúbico
$\mathcal{O}(n^m)$	Polinomial
$\mathcal{O}(k^n)$	Exponencial
$\mathcal{O}(n!)$	Factorial

6.2. Notación Ω

El conjunto $\Omega(g)$ corresponde al conjunto de todas las funciones tales que

$$\exists c > 0.\exists n_0.\forall n \geq n_0.f(n) \geq c \cdot g(n)$$

En palabras simples, es lo mismo que la notación $\mathcal{O}$, pero ahora, f debe ser mayor que g ponderado con un valor positivo.

6.3. Notación Θ

El conjunto $\Theta(g)$ corresponde al conjunto de todas las funciones tales que

$$\exists c_1, c_2 > 0.\exists n_0.\forall n \geq n_0.c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

En palabras simples, es la intersección entre $\mathcal{O}(g)$ y $\Omega(g)$.

7. Análisis de Algoritmos

7.1. ¿Qué es un algoritmo?

Un algoritmo corresponde a una secuencia finita de instrucciones para realizar una computación o resolver un problema. Los algoritmos se pueden dar en cualquier lenguaje o de cualquier forma, y estan presentes en un montón de lugares, incluso fuera de la matemática y la computación. (Por ejemplo: Una receta para cocinar brownies es un algoritmo.)
Cuando trabajamos con algoritmos, una cosa que nos interesa es saber la eficiencia de este. Hay muchos factores que pueden determinar que tan eficiente es nuestro algoritmo, pero el que estudiaremos aquí es el tiempo.

7.2. Eficiencia de un algoritmo con respecto al tiempo

Para un algoritmo A sobre un conjunto de inputs $\mathcal{I}$ se define la función

$$\text{tiempo}_A(I) : \mathcal{I} \to \mathbb{N} \quad ; \quad \text{tiempo}_A(I) = \text{Número de pasos realizados por } A \text{ con input } I$$

Considerando que ahora tenemos una función que nos dice cuanto tiempo se demora un algoritmo en resolver un problema, podemos definir un poco más a la eficiencia. Decimos que un algoritmo A es el más eficiente si se cumple que para todo algoritmo B que realiza el mismo cómputo o resuelve el mismo problema, logra cumplir que

$$\forall I \in \mathcal{I} \;.\; \text{tiempo}_A(I) \leq \text{tiempo}_B(I)$$

Destacar que esta definición no esta exenta de problemas, principalmente por el hecho de que no hemos definido como se puede medir el tiempo. El tiempo que demora un algoritmo puede verse definido por muchas cosas, como la cantidad de pasos a resolver y que tan complejo puede llegar a ser cada paso del algoritmo. Por ahora, para poder simplificar las cosas, para nosotros el tiempo va a consistir en la cantidad de pasos que tiene el algoritmo para lograr resolver un problema.

7.3. Uso de notación asintótica en algoritmos

La notación asintótica nos puede resultar útil para definir el tiempo de un algoritmo.
Por ejemplo, considere el siguiente algoritmo matemático escrito en Python.

```python
for i in range(1, n):
    for j in range(1, i):
        x = x + 1
```

¿Cuántas veces se ejecuta la linea `x = x + 1` al cambiar `n`?
Analizando lo que ocurre en este código, el loop `for j in range(1, i)` es ejecutado `i` veces. Este se encuentra inmerso en un loop `for i in range(1, n)`, lo que significa que en la primera iteración se ejecutará 1 vez, en la siguiente 2 veces, en la siguiente 3 veces y así, hasta llegar a la iteración numero `n`, en donde se ejecutará `n` veces. Sumando todo...

$$\text{Tiempo}_{\text{for\_loop}}(n) = 1 + 2 + 3 + \ldots + n - 1 + n = \frac{n \cdot (n+1)}{2}$$

Resulta que

$$\frac{n \cdot (n+1)}{2} \in \Theta(n^2)$$

Entonces, la cantidad de veces que se ejecuta `x = x + 1` es $\Theta(n^2)$

7.4. Tamaño del input

Para un conjunto de inputs $\mathcal{I}$, se define su función tamaño como

$$|\cdot| : \mathcal{I} \to \mathbb{N} \quad ; \quad |I| = \text{Tamaño de } I \text{ según su representación}$$

Ejemplo: Para una palabra de bits $w \in \{0,1\}^*$; $|w|$ = largo de la palabra w o número de bits.

La definición más general que tenemos de $|I|$ es

$$|I| = \text{Número de bits de una codificación “razonable” de } I$$

7.5. Tipos de complejidad

Para todo algoritmo A y su conjunto de inputs I se puede tener un mejor caso y un peor caso.

$$\text{peor-caso}_A(n) = \max_{I \in \mathcal{I}} \{\text{tiempo}_A(I) \mid |I| = n\} \quad ; \quad \text{mejor-caso}_A(n) = \min_{I \in \mathcal{I}} \{\text{tiempo}_A(I) \mid |I| = n\}$$

8. Principio de Inducción

8.1. Principio de inducción simple

Para una afirmación $P(n)$ sobre los naturales, si $P(n)$ cumple que

- $P(0)$, $P(1)$, ..., $P(k)$ son verdaderos
- Para todo $n \in \mathbb{N}$, si $P(n)$ es verdadero, entonces $P(n+1)$ también es verdadero.

entonces, para todo $n \in \mathbb{N}$ se tiene que $P(n)$ debe ser verdadero.
En lógica de predicados, esto se expresa como (forma general)

$$(\ \underbrace{P(0) \land P(1) \land P(2) \ldots \land P(k)}_{\text{Caso(s) base}} \land \underbrace{(\forall n \geq k . P(n) \to P(n+1))}_{\text{Hipótesis}}\ ) \to \forall n\ .\ P(n)$$

8.1.1. Caso base único

$$(\ P(0) \land (\forall n \geq k . P(n) \to P(n+1))\ ) \to \forall n\ .\ P(n)$$

8.1.2. Caso base extendido

$$(\ P(k) \land (\forall n \geq k . P(n) \to P(n+1))\ ) \to \forall n \geq k\ .\ P(n)$$

8.2. Principio de inducción fuerte

Para una afirmación $P(n)$ sobre los naturales, si $P(n)$ cumple que para todo $n \in \mathbb{N}$

$$\text{Si } P(k) \text{ es verdadero para todo } k < n \text{, entonces } P(n) \text{ es verdadero}$$

Entonces, para todo $n \in \mathbb{N}$ se tiene que $P(n)$ es verdadero.
En lógica de predicados, esto se ve como

$$(\ \forall n\ .\ (\forall k < n\ .\ P(k)) \to P(n)\ ) \to \forall n . P(n)$$

8.3. Axiomas de $\mathbb{N}$ - Axiomas de Peano

1. El número $0 \in \mathbb{N}$
2. Si $n \in \mathbb{N}$, entonces $(n + 1) \in \mathbb{N}$ donde $n + 1$ es sucesor de n.
3. Todo $n \in \mathbb{N}$, a excepción de $n \neq 0$, tiene un antecesor en $\mathbb{N}$
4. Principio del Buen Orden: Todo subconjunto $A \subseteq \mathbb{N}$ tiene un elemento mínimo.