

# Matemáticas Discretas ICI-427

Sergio Hernández  
shernandez@ucm.cl

Facultad de Ciencias de la Ingeniería



# Conjuntos

## Conjunto

Un conjunto es una colección no-ordenada de elementos, tal que  $a \in A$  implica que  $a$  es un elemento del conjunto  $A$ .

# Conjuntos

## Conjunto

Un conjunto es una colección no-ordenada de elementos, tal que  $a \in A$  implica que  $a$  es un elemento del conjunto  $A$ .

## Cardinalidad

Sea  $A$  un conjunto con  $n$  elementos distintos con  $n \in \mathbb{Z}^+$ . El conjunto  $A$  es finito si y solo si  $|A| = n$

# Conjuntos

## Conjunto

Un conjunto es una colección no-ordenada de elementos, tal que  $a \in A$  implica que  $a$  es un elemento del conjunto  $A$ .

## Cardinalidad

Sea  $A$  un conjunto con  $n$  elementos distintos con  $n \in \mathbb{Z}^+$ . El conjunto  $A$  es finito si y solo si  $|A| = n$

## Subconjuntos

Sean  $A, T$  dos conjuntos tal que  $A \subseteq T$ , entonces  $a \in A \Rightarrow a \in T$

# Conjuntos

- Si un conjunto es finito y tiene un número acotado de elementos, entonces podemos escribir el conjunto mediante una lista.

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

# Conjuntos

- Si un conjunto es finito y tiene un número acotado de elementos, entonces podemos escribir el conjunto mediante una lista.

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

- Una construcción alternativa para un conjunto es mediante una propiedad.

$$A = \{a | a \in \mathbb{Z}^+\}$$

# Conjuntos

## Identidad

Dos conjuntos  $A$  y  $B$  son iguales si tienen exactamente los mismos elementos, es decir  $A = B$  implica que  $\forall x(x \in A \Leftrightarrow x \in B)$

# Conjuntos

## Identidad

Dos conjuntos  $A$  y  $B$  son iguales si tienen exactamente los mismos elementos, es decir  $A = B$  implica que  $\forall x (x \in A \Leftrightarrow x \in B)$

## Conjunto nulo

Un conjunto con  $n = 0$  elementos es llamado **conjunto nulo** y se denota  $A = \emptyset$



# Conjuntos

## Identidad

Dos conjuntos  $A$  y  $B$  son iguales si tienen exactamente los mismos elementos, es decir  $A = B$  implica que  $\forall x(x \in A \Leftrightarrow x \in B)$

## Conjunto nulo

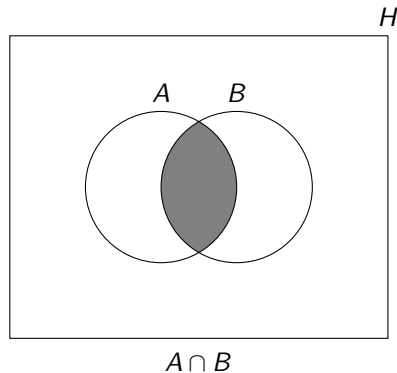
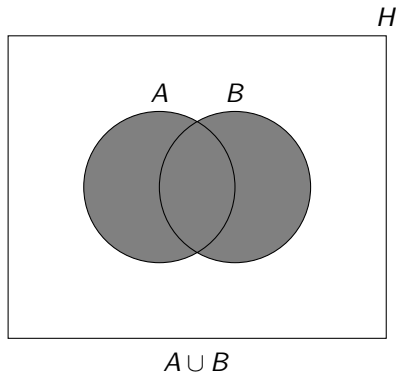
Un conjunto con  $n = 0$  elementos es llamado **conjunto nulo** y se denota  $A = \emptyset$

## Conjunto potencia

El conjunto de todos los subconjuntos de  $B$  se llama el **conjunto potencia**  $\mathcal{P}(B) = \{A | A \subseteq B\}$

$$|B| = n \implies |\mathcal{P}(B)| = 2^n$$

# Diagramas de Venn



# Funciones

## Función biyectiva

Sean  $A$  y  $B$  conjuntos no nulos. Una **función**  $f : A \mapsto B$  asigna exactamente un valor de cada elemento  $b$  de  $B$  a cada  $a$  elemento de  $A$ , tal que  $f(a) = b$ .

# Funciones

## Función biyectiva

Sean  $A$  y  $B$  conjuntos no nulos. Una **función**  $f : A \mapsto B$  asigna exactamente un valor de cada elemento  $b$  de  $B$  a cada  $a$  elemento de  $A$ , tal que  $f(a) = b$ .

## Dominio

Sea  $f$  una función tal que  $f : A \mapsto B$ , entonces  $A$  es el dominio de  $f$  y  $B$  es el co-dominio.

# Funciones

## Función biyectiva

Sean  $A$  y  $B$  conjuntos no nulos. Una **función**  $f : A \mapsto B$  asigna exactamente un valor de cada elemento  $b$  de  $B$  a cada  $a$  elemento de  $A$ , tal que  $f(a) = b$ .

## Dominio

Sea  $f$  una función tal que  $f : A \mapsto B$ , entonces  $A$  es el dominio de  $f$  y  $B$  es el co-dominio.

## Conjuntos contables

El conjunto de números naturales  $\mathbb{N}$  se define como **contable**. Cualquier conjunto (finito o infinito) que posea una función biyectiva sobre  $\mathbb{N}$  es también contable.

# Listas y tuplas en Python

## Listas

Las listas son arreglos dinámicos y por lo tanto son mutables (permiten cambiar tanto el contenido de los elementos como la cantidad de elementos).

# Listas y tuplas en Python

## Listas

Las listas son arreglos dinámicos y por lo tanto son mutables (permiten cambiar tanto el contenido de los elementos como la cantidad de elementos).

## Tuplas

Las tuplas son arreglos estáticos y por lo tanto no son mutables (no permiten cambiar el contenido de los elementos ni la cantidad de elementos).

# Listas y tuplas en Python <sup>a</sup>

<sup>a</sup>High Performance Python by Micha Gorelick and Ian Ozsvald (O'Reilly). Copyright 2014 Micha Gorelick and Ian Ozsvald. 978-1-4493-6159-4

## Listing 1: Redimensionado de Listas

```
>>> l=[1,2]
>>> for i in range(3,7):
...     l.append(i)
...
>>> l
[1, 2, 3, 4, 5, 6]
```



# Listas y tuplas en Python

## Listing 2: Tuplas

```
>>> l=(1,2)
>>> l[1]
2
>>> l[1]=3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item
assignment
```

# Listas y tuplas en Python

## Listing 3: Tiempo de instanciación

```
>>> import math
>>> import timeit
>>> sum(timeit.repeat("l=[1,2,3,4,5,6,7,8,9,10]"))/3
0.18932326634724936
>>> sum(timeit.repeat("l=(1,2,3,4,5,6,7,8,9,10)"))/3
0.02806266148885091
```

# Diccionarios y conjuntos en Python

## Diccionarios

Los diccionarios utilizan *tablas hash* para almacenar elementos sin orden pre-establecido.

# Diccionarios y conjuntos en Python

## Diccionarios

Los diccionarios utilizan *tablas hash* para almacenar elementos sin orden pre-establecido.

## Conjuntos

A diferencia de los diccionarios, los conjuntos no permiten duplicados (no existen colisiones en la funcion de hash).

# Diccionarios y conjuntos en Python

## Listing 4: Búsqueda en listas

```
>>> def find_phonenumber(phonebook, name):  
    for n, p in phonebook:  
        if n == name: return p  
    return None  
  
>>> phonebook = [  
    ("John Doe", "555-555-5555"),  
    ("Albert Einstein", "212-555-5555")]  
  
>>> print "John Doe's phone number is",  
find_phonenumber(phonebook, "John Doe")
```

# Conjuntos en Python

## Listing 5: Conjuntos

```
>>> p={'a','b','c'}
>>> p
set(['a', 'c', 'b'])
>>> p.add('a')
>>> p
set(['a', 'c', 'b'])
>>> p.remove('c')
>>> p
set(['a', 'b'])
```

# Operaciones de conjuntos en Python

## Listing 6: Conjuntos

```
>>> p={'a','b','c'}
>>> 'a' in p
True
>>> p | {'c','b','d'}
set(['a', 'c', 'b', 'd'])
>>> p & {'c','b','d'}
set(['c', 'b'])
```

# Tablas de Hash

|    |   |          |
|----|---|----------|
| 0  | → | 110      |
| 1  | → | 100      |
| 2  | → | 90       |
| 3  | → | 80       |
| 4  | → | 70       |
| 5  | → | 93 → 60  |
| 6  | → | 50       |
| 7  | → | 40       |
| 8  | → | 96 → 30  |
| 9  | → | 97 → 20  |
| 10 | → | 120 → 10 |



# Diccionarios y conjuntos en Python

## Listing 7: Búsqueda en diccionarios

```
>>> phonebook = {  
    "John Doe": "555-555-5555",  
    "Albert Einstein" : "212-555-5555",  
}  
>>> print "John Doe's phone number is",  
phonebook["John Doe"]
```

# Funciones de Hash

## Listing 8: Funciones de Hash

```
>>> class Point(object):  
...     def __init__(self,x,y):  
...         self.x,self.y=x,y  
...  
>>> p1=Point(1,1)  
>>> p2=Point(1,1)  
>>> p1==p2  
False  
>>> s={p1,p2}  
>>> s  
set([<__main__.Point object at 0x10381bc10>,  
<__main__.Point object at 0x10381bc50>])
```

# Funciones de Hash

## Listing 9: Funciones de Hash

```
>>> class Point(object):
...     def __init__(self,x,y):
...         self.x,self.y=x,y
...     def __hash__(self):
...         return hash((self.x,self.y))
...     def __eq__(self,other):
...         return self.x==other.x
...             and self.y==other.y
...
>>> p1=Point(1,1)
>>> p2=Point(1,1)
>>> p1==p2
True
>>> s={p1,p2}
>>> s
set([<__main__.Point object at 0x10381be90>])
```

# Funciones de Hash

## Listing 10: Funciones de Hash

```
>>>help('hash')
Help on built-in function hash in module __builtin__:

hash(...)
    hash(object) -> integer
```

Return a **hash** value **for** the **object**.  
Two objects with the same value have  
the same **hash** value. The reverse **is not**  
necessarily true, but likely.