



Facultad: Ingeniería
Escuela: Ingeniería en Computación
Asignatura: Autómatas y Compiladores

Operaciones con conjuntos en Python

Introducción

En esta práctica se abordan las operaciones con conjuntos de elementos que requieren la aplicación de métodos específicos y librerías de lenguaje Python.

Competencias

- Conoce las operaciones fundamentales de la teoría de conjuntos.
- Aplica lenguaje de programación para simular conjuntos.
- Implementa operaciones con conjuntos en lenguaje de programación.
- Representa gráficamente operaciones con conjuntos.

Material y Equipo

- Guía de laboratorio N° 3.
- Computadora con Python 3.0 o superior.
- Memoria USB.

Introducción Teórica

¿Qué es un conjunto?

Un conjunto es una colección de objetos distintos, a menudo llamados elementos o miembros. Existen dos características que hacen a los conjuntos totalmente diferentes al resto de otras colecciones de elementos.

- Un conjunto está siempre bien definido, es decir, que si realizamos la pregunta: ¿este objeto particular se encuentra en esta colección?; siempre debe existir una respuesta clara por sí o por no basada en una regla o algunos criterios dados.
- No hay dos miembros del mismo conjunto que sean exactamente iguales, es decir, que no hay elementos repetidos.

Un conjunto puede obtener cualquier cosa imaginable, incluyendo números, letras, colores, incluso, otros conjuntos.

Además, utilizaremos el constructor `FiniteSet`, que proporciona la librería `sympy`, el cual tiene ciertas ventajas sobre la versión por defecto de Python.

Procedimiento

Continuando con las operaciones con conjuntos que se pueden implementar en lenguaje Python:

1. Calculando producto cartesiano. Para realizar esta operación debemos utilizar el constructor `FiniteSet`:

```

1 #-*- coding: utf-8 -*-
2 """
3 Created on Thu Aug 13 09:50:39 2020
4
5 @author: HSerrano
6 """
7 from sympy import FiniteSet
8
9 A = FiniteSet(1, 2, 3, 4, 5)
10 B = FiniteSet(3, 4, 8, 6, 0)
11 PCartesiano = A * B
12 print(PCartesiano)
13 #Calculando producto cartesiano:
14 for elemento in PCartesiano:
15     print(elemento)

```

Salida:

```

{1, 2, 3, 4, 5} x {0, 3, 4, 6, 8}
(1, 0)
(1, 3)
(1, 4)
(1, 6)
(1, 8)
(2, 0)
(2, 3)
(2, 4)
(2, 6)
(2, 8)
(3, 0)
(3, 3)
(3, 4)
(3, 6)
(3, 8)
(4, 0)
(4, 3)
(4, 4)
(4, 6)
(4, 8)
(5, 0)
(5, 3)
(5, 4)
(5, 6)
(5, 8)

```

2. Elevar a la n potencia un conjunto. Calcula el n producto cartesiano el mismo conjunto:

```

1 #-*- coding: utf-8 -*-
2 """
3 Created on Thu Aug 13 09:58:31 2020
4
5 @author: HSerrano
6 """
7
8 from sympy import FiniteSet
9
10 A = FiniteSet(1, 2, 3, 4)
11 Potencia = A ** 2
12 print(Potencia)
13
14 #Si queremos calcular la tercera potencia:
15 Potencia2 = A ** 3
16 print(Potencia2)
17
18 for elemento in Potencia2:
19     print(elemento)

```

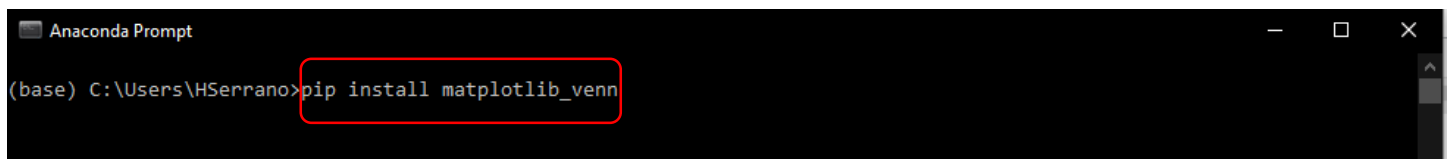
Ejecutar y visualizar el resultado.

Graficando conjuntos y operaciones:

matplotlib.pyplot es una colección de funciones que hacen que matplotlib funcione como MATLAB. Cada función de pyplot realiza algún cambio en una figura: por ejemplo, crea una figura, crea un área de trazado en una figura, traza algunas líneas en un área de trazado, decora la trama con etiquetas, etc.

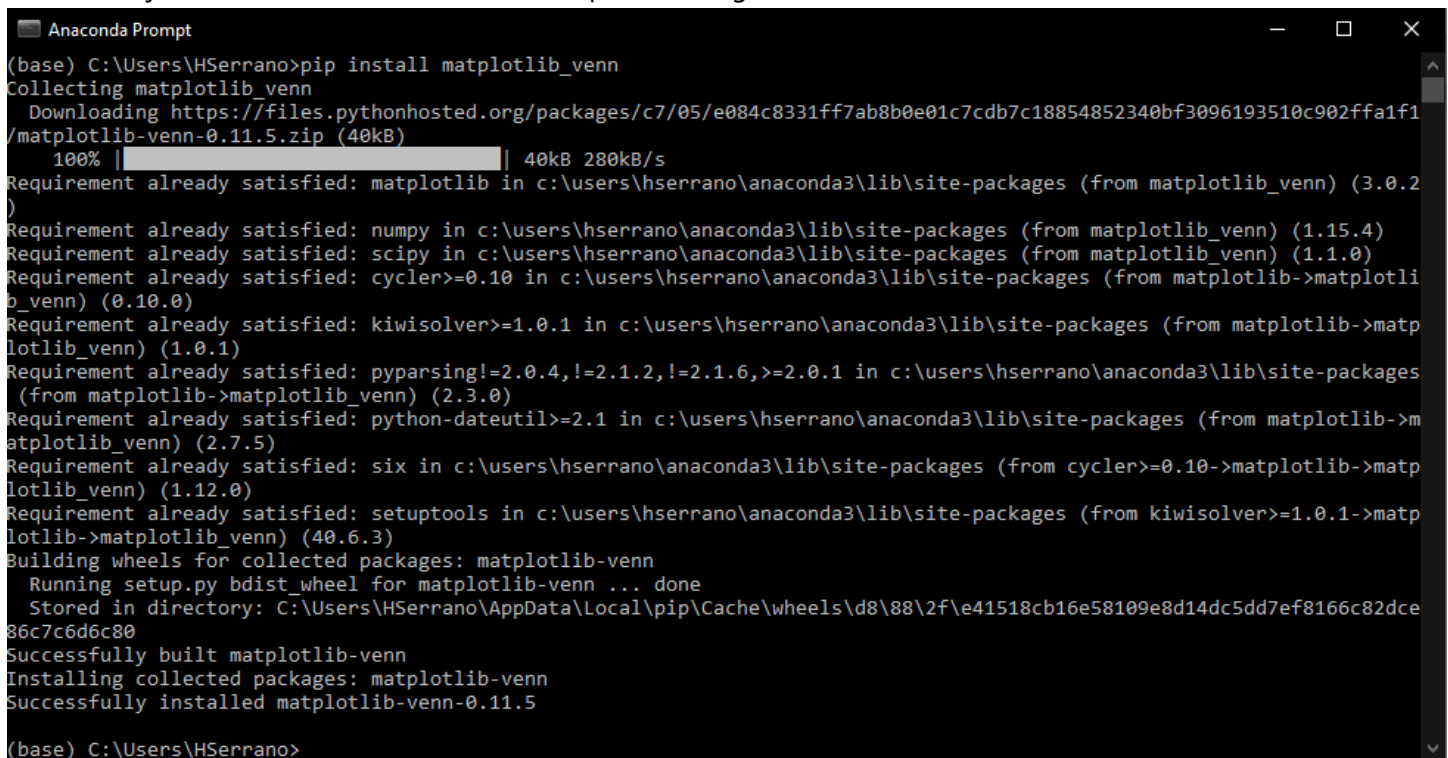
En **matplotlib.pyplot** se conservan varios estados en las llamadas a funciones, de modo que realiza un seguimiento de cosas como la figura actual y el área de trazado, y las funciones de trazado se dirigen a los ejes actuales (tenga en cuenta que "ejes" aquí y en la mayoría de los lugares de la documentación se refiere a la parte de los ejes de una figura y no al término matemático estricto para más de un eje).

3. Vamos a instalar el componente matplotlib_venn desde anaconda prompt:



```
Anaconda Prompt
(base) C:\Users\HSerrano>pip install matplotlib_venn
```

4. Al ejecutar el comando anterior, deberá aparecer lo siguiente:



```
Anaconda Prompt
(base) C:\Users\HSerrano>pip install matplotlib_venn
Collecting matplotlib_venn
  Downloading https://files.pythonhosted.org/packages/c7/05/e084c8331ff7ab8b0e01c7cdb7c18854852340bf3096193510c902ffa1f1/matplotlib-venn-0.11.5.zip (40kB)
    100% |#####| 40kB 280kB/s
Requirement already satisfied: matplotlib in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib_venn) (3.0.2)
Requirement already satisfied: numpy in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib_venn) (1.15.4)
Requirement already satisfied: scipy in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib_venn) (1.1.0)
Requirement already satisfied: cycycler>=0.10 in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib->matplotlib_venn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib->matplotlib_venn) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!2.1.2,!2.1.6,>=2.0.1 in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib->matplotlib_venn) (2.3.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\hserrano\anaconda3\lib\site-packages (from matplotlib->matplotlib_venn) (2.7.5)
Requirement already satisfied: six in c:\users\hserrano\anaconda3\lib\site-packages (from cycycler>=0.10->matplotlib->matplotlib_venn) (1.12.0)
Requirement already satisfied: setuptools in c:\users\hserrano\anaconda3\lib\site-packages (from kiwisolver>=1.0.1->matplotlib->matplotlib_venn) (40.6.3)
Building wheels for collected packages: matplotlib_venn
  Running setup.py bdist_wheel for matplotlib_venn ... done
  Stored in directory: C:\Users\HSerrano\AppData\Local\pip\Cache\wheels\d8\88\2f\e41518cb16e58109e8d14dc5dd7ef8166c82dce86c7c6d6c80
Successfully built matplotlib_venn
Installing collected packages: matplotlib_venn
Successfully installed matplotlib_venn-0.11.5
(base) C:\Users\HSerrano>
```

5. Comprobando la instalación:

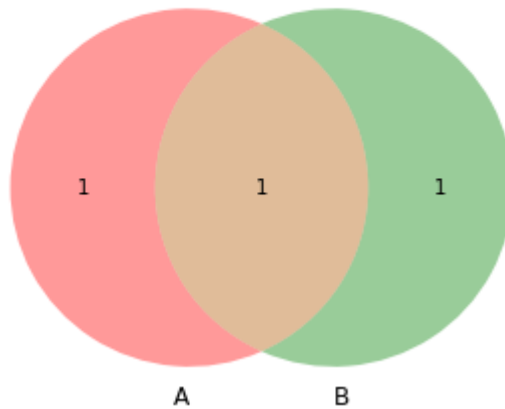
```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Aug 13 10:04:58 2020
4
5 @author: HSerrano
6 """
7 from sympy import FiniteSet
8 # Dibujando el diagrama de venn de 2 conjuntos
9 from matplotlib import pyplot as plt
10 from matplotlib_venn import venn2
11 venn2((1, 1, 1))
12 plt.show()

```

Salida:

```
In [14]: runfile('C:/Users/HSerrano/G3_E3.py', wdir='C:/Users/HSerrano')
```



Ejercicios propuestos

1. Investigar cómo representar las operaciones con conjunto utilizando matplotlib.

Investigación Complementaria

1. Cómo dibujar las diferentes operaciones y representaciones de operaciones con conjuntos.

Bibliografía

- Python, Guido Van Rossum