Curso de Análise
(Notas)

Bernardo Paulsen

7 de Julho de 2021

Prefácio

Estas são as notas de estudo do livro ‘Curso de Análise’, de Elon Lages Lima. Nela tento implementar os conceitos do livro na linguagem de programação Python.

Conteúdo

Preliminares

```
from itertools import chain, combinations
```

Capítulo I

Conjuntos e Funções

1 Conjuntos

Um conjunto é formado pelos seus elementos. Um objeto x pertence ao conjunto A quando é um de seus elementos. Pertencimento é representado da forma

$$x \in A.$$

Um conjunto A fica definido quando se dá uma regra que permita decidir se um objeto arbitrário x pertence ou não a A.

Usa-se a notação

$$X = \{a, b, c, \dots\}$$

para representar o conjunto X cujos elementos são os objetos a, b, c, etc.

Exemplo 1.1. *Abaixo criamos o conjunto* $X = \{1, 2, 3, 4\}$.

```
X = {1,2,3,4}
print(f"X: {X}")
```

```
X: 1, 2, 3, 4
```

Exemplo 1.2. *Abaixo checamos se o objeto* $a = 1$ *pertence ao conjunto* $X = \{1, 2, 3, 4\}$.

```python
a = 1
X = {1,2,3,4}
print(f"Pertence: {a in X}")
```

```
Pertence: True
```

Quando x não é elemento do conjunto A ele não pertence a A. Não pertencimento é representado da forma

$$x \notin A.$$

Exemplo 1.3. *Abaixo checamos se o objeto $a = 1$ não pertence ao conjunto $X = \{1, 2, 3, 4\}$.*

```python
a = 1
X = {1,2,3,4}
print(f"Não pertence: {a not in X}")
```

```
Não pertence: False
```

Agora, alguns conjuntos que utilizaremos: conjunto dos números naturais: $\mathbb{N}$; conjunto dos números inteiros: $\mathbb{Z}$; conjunto dos números racionais: $\mathbb{Q}$.

É possível definir um cojunto por meio de uma propriedade comum e exclusiva dos seus elementos. Uma propriedade P define um conjunto X caso $x \in X$ se um objeto x goza da propriedade P, enquanto $x \notin X$ se x não goza de P. Escreve-se

$$X = \{x; x \text{ goza da propriedade } P\}.$$

Nos casos que a propriedade P se refere a elementos de um conjunto fundamental E escreve-se

$$X = \{x \in E; x \text{ goza da propriedade } P\}.$$

Exemplo 1.4. *Abaixo criamos o conjunto a seguir.*

$$X = \{x \in E; x \textit{ é par}\}, \quad E = \{1, 2, 3, 4\}$$

```python
E = {1,2,3,4}
```

```
X = set(x for x in E if not x%2)
print(f"X: {X}")
```

```
X: 2, 4
```

O conjunto vazio $\emptyset$ é definido assim:

$$\text{Qualquer que seja } x\text{, tem-se } x \notin \emptyset.$$

Exemplo 1.5. *Abaixo criamos um conjunto vazio.*

```
O = set()
print(f"Conjunto vazio: {O}")
```

```
Conjunto vazio: set()
```

O conjunto A é um subconjunto de B quando todo elemento de A também é elemento de B.

$$A \subset B$$

Exemplo 1.6. *Abaixo checamos se o conjunto $A = \{1, 2\}$ é subconjunto de $B = \{1, 2, 3, 4\}$.*

```
A = {1,2}
B = {1,2,3,4}
print(f"É subconjunto: {A.issubset(B)}")
```

```
É subconjunto: True
```

No caso em que $X \subset Y$ e $X \neq Y$ diz-se que X é um subconjunto próprio de Y.

Exemplo 1.7. *Abaixo, checamos de A é subconjunto próprio de B.*

```
A = {1,2}
B = {1,2,3,4}
print(f"É subconjunto: {A.issubset(B)}")
print(f"É idêntico: {A == B}")
```

```
É subconjunto: True
É idêntico: False
```

O conjunto vazio $\emptyset$ é subconjunto próprio de qualquer conjunto.

$$\emptyset \subset X, \text{ seja qual for o conjunto } X$$

Exemplo 1.8. *Abaixo, checamos se o conjunto vazio é subconjunto de um conjunto X.*

```python
O = set()
X = {1,2,3,4}
print(f"É subconjunto: {O.issubset(X)}")
```

```
É subconjunto: True
```

A relação de inclusão $A \subset B$ é

- Reflexiva - $A \subset A$, seja qual for o conunto A;
- Anti-simétrica - se $A \subset B$ e $B \subset A$ então $A = B$;
- Transitiva - se $A \subset B$ e $B \subset C$ então $A \subset C$.

Exemplo 1.9. *Primeiramente definimos os conjuntos $A = \{1,2\}$, $B = \{1,2\}$ e $C = \{1,2,3\}$.*

```python
A = {1,2}
B = {1,2}
C = {1,2,3}
```

Abaixo, checamos abaixo se $A \subset A$.

```python
print(f"A é subconjunto de A: {A.issubset(A)}")
```

```
A é subconjunto de A: True
```

Abaixo, checamos $A \subset B$, $B \subset A$ e $A = B$.

```
print(f"A é subconjunto de B: {A.issubset(B)}")
print(f"B é subconjunto de A: {B.issubset(A)}")
print(f"A é B: {A == B}")
```

```
A é subconjunto de B: True
B é subconjunto de A: True
A é B: True
```

Abaixo, checamos se $B \subset C$, $A \subset C$.

```
print(f"B é subconjunto de C: {B.issubset(C)}")
print(f"A é subconjunto de C: {A.issubset(C)}")
```

```
B é subconjunto de C: True
A é subconjunto de C: True
```

Dado um conjunto X, indica-se com $P(X)$ o conjunto cujos elementos são as partes de X. Afirmar que $A \in P(X)$ é o mesmo que dizer $A \subset X$. O conjunto das partes de X nunca é vazio: tem-se pelo menos $\emptyset \in P(X)$ e $X \in P(X)$.

Exemplo 1.10. *Primeiramente, precisamos criar a função que retornará o conjunto de partes de um conjunto qualquer.*

```
def powerset(iterable):
    s = list(iterable)
    i = (set(combinations(s, r)) for r in range(len(s)+1))
    p = set(frozenset(e) for e in chain.from_iterable(i))
    return p
```

Agora, geramos todos o conjunto de partes de $X = \{1, 2\}$.

```
X = {1,2}
P = powerset(X)
print(P)
```

```
frozenset(), frozenset(2), frozenset(1), frozenset(1, 2)
```

Sejam P e Q propriedades que se referem a elementos de um certo conjunto E. As propriedades P e Q definem subconjutos X e Y de E, a saber:

$$X = \{x \in E; x \text{ goza de } P\} \text{ e } Y = \{y \in E; y \text{ goza de } Q\}.$$

Exemplo 1.11. *Abaixo criamos os conjuntos a seguir.*

$$E = \{1, 2, 3, 4, 5, 6, 7, 8\}$$
$$X = \{x \in E; x \text{ é divisível por } 4\}$$
$$Y = \{y \in E; y \text{ é divisível por } 2\}$$

```
E = {1,2,3,4,5,6,7,8}
X = set((x for x in E if not x%4))
Y = set((y for y in E if not y%2))
print(f"X: {X}")
print(f"Y: {Y}")
```

```
X: 8, 4
Y: 8, 2, 4, 6
```

As afirmações "P implica Q" e "se P então Q" têm o mesmo significado, de que $X \subset Y$ - que todo objeto que goza de P também goza de Q. Usa-se a notação

$$P \implies Q.$$

Exemplo 1.12. *Abaixo utilizamos os conjuntos do Exemplo 1.11.*

```
print(f"X é subconjunto de Y: {X.issubset(Y)}")
```

```
X é subconjunto de Y: True
```

Também as afirmações "P se, e somente se, Q", "P é condição necessária e suficiente para Q" querem dizer que $P \implies Q$ e $Q \implies P$. A notação é

$$P \iff Q.$$

Exemplo 1.13. *Utilizando os conjuntos do Exemplo 1.11, checamos abaixo se* $P \iff Q$.

```
iff = X.issubset(Y) and Y.issubset(X)
print(f"X é subcojunto de Y e vice-versa: {iff}")
```

```
X é subcojunto de Y e vice-versa: False
```

2 Operações entre conjuntos

A reunião dos conjuntos A e B é o conunto $A \cup B$, formado pelos elementos de A mais os elementos de B.

$$A \cup B = \{x; x \in A \text{ ou } x \in B\}$$

Exemplo 2.1. *Definimos os conjuntos $A = 1, 2$ e $B = 2, 3$ e calculamos $A \cup B$.*

```python
A     = {1,2}
B     = {2,3}
union = A.union(B)
print(f"União de A com B: {union}")
```

```
União de A com B: 1, 2, 3
```

Sejam quais forem os conjuntos A e B, tem-se que

- $A \subset A \cup B$;
- $B \subset A \cup B$.

Exemplo 2.2. *Utilizando os conjuntos do exemplo 2.1, checamos se $A \subset A \cup B$ e se $B \subset A \cup B$.*

```python
a = A.issubset(A.union(B))
b = B.issubset(A.union(B))
print(f"A é subconjunto da união de A com B: {a}")
print(f"B é subconjunto da união de A com B: {b}")
```

```
A é subconjunto da união de A com B: True
B é subconjunto da união de A com B: True
```

A intersecção dos conjuntos A e B é o conjunto $A \cap B$, formado pelos elementos comuns a A e B.

$$A \cap B = \{x; x \in A \text{ e } x \in B\}$$

Exemplo 2.3. *Utilizando os conjuntos do exemplo 2.1, calculamos $A \cap B$.*

```
A = {1,2}
B = {2,3}
i = A.intersection(B)
print(f"Intersecção de A e B: {i}")
```

```
Intersecção de A e B: 2
```

No caso de $A \cap B = \emptyset$, os conjuntos dizem-se disjuntos.

```
A = {1,2}
B = {3,4}
print(not A.intersection(B))
```

```
True
```

Sejam quais forem os conjuntos A e B, tem-se que

- $A \cap B \subset A$;
- $A \cap B \subset B$.

```
print(A.intersection(B).issubset(A))
print(A.intersection(B).issubset(B))
```

```
True
True
```

A diferença entre os conjuntos AeB é o conjunto $A - B$, formado pelos elementos de A que não pertencem a B.

Definição 2.1 (Diferença). $A - B = \{x; x \in A \; e \; x \notin B\}$

```
A = {1,2}
B = {2,3}
print(A.difference(B))
```

```
1
```

Quando A e B são disjuntos, $A - B = A$.

```
A = {1,2}
B = {3,4}
C = A.difference(B)
print(A.issubset(C), C.issubset(A))
```

```
True True
```

Quando se tem $B \subset A$, a diferença $A - B$ chama-se o complementar de B em relação a A e escreve-se

$$A - B = \complement_A B.$$

Relacionamos abaixo propriedades formais da operação de tomar complementares. Os conjuntos A e B são partes de um conjunto fundamental E, em relação ao qual estamos tomando os complementares.

Propriedade 2.1. $\complement\,(\complement A) = A$

Demonstração. $x \in \complement(\complement A) \iff x \notin \complement A \iff x \in A$ □

Propriedade 2.2. $A \subset B \iff \complement B \subset \complement A$

Demonstração. Se $A \subset B$ então, $x \in \complement B \implies x \in \mathsf{A}$, o que significa que $\complement B \subset \complement A$ □