Operações da álgebra de conjuntos

Alunos: Erick Willames e João Pedro

Algoritmo 1: Uniao(A,B)

Entrada: A: Lista que contém os valores do primeiro conjunto, B: Lista que contém os valores do segundo conjunto
Saída: A: conjunto união entre os conjuntos de entrada

1 n $\leftarrow$ |A|
2 **para cada** $i \in B$ **faça**
3 | **se** $i \notin A$ **então**
4 | | $A_{n+1} \leftarrow i$
5 **fim**
6 **retorna** A

Explicação:
O algoritmo 1, recebe como parâmetro dois conjuntos, respectivamente A e B e ao fim retorna um terceiro conjunto C com a união dos elementos de A e B.

- Na linha 1 a variável *n* recebe o tamanho do conjunto A.
- Na linha 2 o algoritmo determina que para cada item pertencente ao conjunto B serão realizadas as operações do bloco.
- Na linha 3 é feita uma comparação com o elemento em questão, é verificado se ele não pertence ao conjunto A.
- Se a linha 3 for verdadeira a linha 4 atribui o elemento do conjunto B em questão no conjunto A, utilizando a variável *n* acrescentando 1 para que o elemento seja inserido no fim do conjunto.
- Finaliza na linha 5 e retorna a união dos conjuntos no conjunto A modificado.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C:  {1,2,3,4,5,6,7,9}
```

Algoritmo 2: Interseccao(A, B)

Entrada: A e B: Dois conjuntos que serão inseridos
Saída: C: conjunto intersecção entre os conjuntos de entrada
1 C $\leftarrow$ { }
2 n $\leftarrow$ |C|
3 **para cada** $i \in$ B **faça**
4 | **se** $i \in$ A **então**
5 | | $C_{n+1} \leftarrow i$
6 **fim**
7 **retorna** C

Explicação:
O algoritmo 2, recebe dois conjuntos como parâmetros e retorna um terceiro conjunto como resultado da interseção entre os dois conjuntos de entrada.

- Na linha 1 é definido o conjunto C como um conjunto vazio.
- Na segunda linha é atribuído à variável n o tamanho do conjunto C.
- Na linha 3 é definido que para cada elemento de B são realizadas as operações do bloco interno.
- Na linha 4 é verificado se o elemento de B em questão pertence ao conjunto A. Se a linha 4 for verdadeira é atribuído o elemento em questão ao final do conjunto C através da variável *n+1* na linha 5.
- Após, na linha 6 temos o fim e na 7 temos o retorno do conjunto C que tem todos os elementos da intersecção entre os conjuntos A e B.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C:  {3,4}
```

Algoritmo 3: Diferença(A, B)

Entrada: A: Lista que contém os valores do primeiro conjunto, B: Lista que contém os valores do segundo conjunto
Saída: C: conjunto contendo diferença entre os conjuntos de entrada

1 C$\leftarrow$ { }
2 n $\leftarrow$ |C|
3 para cada $i \in B$ **faça**
4 | **se** $i \notin A$ **então**
5 | **|** $C_{n+1} \leftarrow i$
6 para cada $i \in A$ **faça**
7 | **se** $i \notin B$ **então**
8 | **|** $C_{n+1} \leftarrow i$
9 fim
10 retorna C

Explicação:
O algoritmo 3, recebe dois conjuntos na entrada, respectivamente A e B, ao fim retorna o conjunto C contendo os elementos que são diferentes entre os conjuntos de entrada.

- Na linha 1 é definido o conjunto C.
- Na linha 2 a variável *n* recebe o tamanho do conjunto C.
- Na linha 3 é definido que para cada elemento pertencente ao conjunto B é executado o bloco interno de instruções.
- Na linha 4 é verificado se o elemento em questão não pertence ao conjunto A. Se for verdadeiro, o elemento é adicionado ao conjunto no final do conjunto C na linha 5.
- Na linha 6 novamente temos a definição que para cada elemento de A será executado o bloco interno.
- Na linha 7, se o elemento não pertencer ao conjunto B ele é adicionado ao conjunto C na linha 8.
- Na linha 9 temos o fim e na 10 o retorno que é o conjunto C contendo os elementos com a diferença entre os conjuntos de entrada.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C: {1,2}
```

Algoritmo 4: Complemento(A, B)

Entrada: A: Lista que contém os valores do primeiro conjunto, B: Lista que contém os valores do segundo conjunto
Saída: C: conjunto contendo complemento entre os conjuntos de entrada

1 C $\leftarrow$ { }
2 n $\leftarrow$ |C|
3 **para cada** $i \in A$ **faça**
4 | **se** $i \notin B$ **então**
5 | | $C_{n+1} \leftarrow i$
6 **fim**
7 **return** C

Explicação:
O algoritmo 4, recebe dois conjuntos A e B como parâmetro e retorna um conjunto C com o complemento entre os dois conjuntos de entrada.

- Na linha 1 temos a definição do conjunto C
- Na linha 2 temos a variável *n* que recebe o tamanho do conjunto C
- Na linha 3 é definido que para cada elemento do conjunto A, será executado o bloco interno.
- Na linha 4 é verificado se o elemento em questão não pertence ao conjunto B.
- Sendo verdadeira a linha 4, o conjunto C recebe o elemento na última posição.
- Na linha 6 temos o fim e na linha 7 o retorno com o conjunto C. Contendo os elementos que são complementos entre os conjuntos A e B.

Funcionamento do programa na linguagem Python:
Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C: {1,2}
```

Algoritmo 5: ProdutoCartesiano(A,B)

Entrada: A: Lista que contém os valores do primeiro conjunto, A: Lista que contém os valores do segundo conjunto
Saída: C: conjunto contendo o produto cartesiano entre os conjuntos de entrada

1 C$\leftarrow$ { }
2 n $\leftarrow$ |C|
3 **para cada** $i \in$ A **faça**
4 | **para cada** $j \in$ B **faça**
5 | | pr $\leftarrow$ (i , j)
6 | | $C_{n+1} \leftarrow$ pr
7 **fim**
8 **retorna** C

Explicação:
O algoritmo 5, recebe dois conjuntos A e B e retorna o produto cartesiano entre dois conjuntos de entrada.

- Na linha 1 é definido o conjunto C, que representa o conjunto de saída.
- Na linha 2 a variável n recebe o tamanho do conjunto C
- Na linha 3, para cada item do conjunto A é executado o bloco interno
- Na linha 4, para cada elemento pertencente ao conjunto B, é executado o conjunto interno de instruções.
- Na linha 5, a variável *pr* recebe os elementos *i* e *j* que estão sendo executados.
- Na linha 6, o conjunto C, recebe na última posição o valor de *pr*
- Na linha 7 temos o fim
- Na linha 8 o algoritmo retorna o conjunto C.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C  {(1,5),(1,6),(1,3),(1,4),(1,7),(1,9),(2,5),(2,6),(2,3),(2,4),(2,7),(2,9),(3,
5),(3,6),(3,3),(3,4),(3,7),(3,9),(4,5),(4,6),(4,3),(4,4),(4,7),(4,9)}
```

Algoritmo 6: UniaoDisjunta(A, B)

Entrada: A: Lista que contém os valores do primeiro conjunto, B: Lista que contém os valores do segundo conjunto
Saída: C: conjunto contendo a união disjunta entre os conjuntos de entrada

```
1  C← [ ]
2  para cada i ∈ A faça
3  |     va ← i , 'A'
4  |     C← va
5  fim
6  para cada i ∈ B faça
7  |     va ← i , 'B'
8  |     C← va
9  fim
10 retorna C
```

Explicação:
O algoritmo 6 recebe dois conjuntos A e B e retorna a união disjunta entre estes conjuntos.

- Na linha 1 é definido o conjunto C
- Na linha 2, é verificado para cada item que pertença ao conjunto A será executado o bloco de instruções interno.
- Na linha 3, a variável *va* recebe o elemento em questão seguido do caractere A.
- Na linha 4, o conjunto C recebe a variável *va*
- Na linha 5 é finalizado a execução do bloco
- Na linha 6, , é verificado para cada item que pertença ao conjunto B será executado o bloco de instruções interno.
- Na linha 7, a variável *va* recebe o elemento em questão seguido do caractere B.
- Na linha 8, o conjunto C recebe a variável *va.*
- Na linha 9 temos o final do bloco
- Na linha 10, o algoritmo retorna o conjunto C.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
B= {5,6,3,4,7,9}
```

Saída:

```
C {1A,2A,3A,4A,5B,6B,3B,4B,7B,9B}
```

Algoritmo 7: ConjuntoDasPartes(A)

Entrada: A: Lista que contém os valores do conjunto
Saída: C: conjunto contendo o conjunto das partes

Não obtivemos sucesso ao tentar fazer esse algoritmo. Achamos algo pronto, mas não conseguimos entender, por isso não temos o algoritmo.

Funcionamento do programa na linguagem Python:

Entrada:

```
A= {1,2,3,4}
```

Saída:

```
 C { {},{1},{1,2},{1,2,3},{1,2,3,4},{1,2,4},{1,3},{1,3,4},{1,4},{2},{2,3},{2,3,4
},{2,4},{3},{3,4},{4},}
```