

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Дискретна математика

Лабораторна робота №2

**«Бінарні відношення та їх основні властивості,
операції над відношеннями»**

Виконала:
студентка групи ІО-64
Бровченко А. В.
Перевірів Новотарський М. А.

Київ
2017 р.

Мета: вивчити основні властивості бінарних відношень та оволодіти операціями над бінарними відношеннями.

Загальне завдання:

1. Написати в окремому модулі функцію для формування несуперечливих бінарних відношень.
2. Написати в окремому модулі функції виконання логічних операцій над бінарними відношеннями.
3. Пояснити правило формування несуперечливих відношень відповідно до варіанту.

Теоретичні відомості:

Упорядкована пара предметів – це сукупність, що складається із двох предметів, розташованих у деякому певному порядку.

Бінарним (або **двомісним**) відношенням R називають підмножину впорядкованих пар, тобто множину, кожен елемент якої є впорядкованою парою **Декартовим добутком** $X * Y$ множин X і Y є множина $\{(x, y) \mid x \in X, y \in Y\}$.

Бінарним відношенням R називають підмножину пар $(x, y) \in R$ прямого добутку $X * Y$. У силу визначення бінарних відношень, як **спосіб їх задавання** можуть бути використані будь-які способи задавання множин. Відношення, визначені на скінченних множинах, зазвичай задають списком чи матрицею.

Операції над відношеннями:

1. **Об'єднання:** $R_1 \cup R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ або } (a, b) \in R_2\}$.

2. **Перетин:** $R_1 \cap R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ і } (a, b) \in R_2\}$.

3. **Різниця:** $R_1 \setminus R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ і } (a, b) \notin R_2\}$.

4. **Доповнення:** $\overline{R} = U \setminus R$, де $U = A \times B$.

Крім того, необхідно визначити інші операції над бінарними відношеннями.

5. **Обернене відношення** R^{-1} .

Якщо $(a, b) \in R$ – відношення, то відношення R^{-1} називають **оберненим відношенням** до даного відношення R тоді й тільки тоді, коли $R^{-1} = \{(b, a) \mid (a, b) \in R\}$.

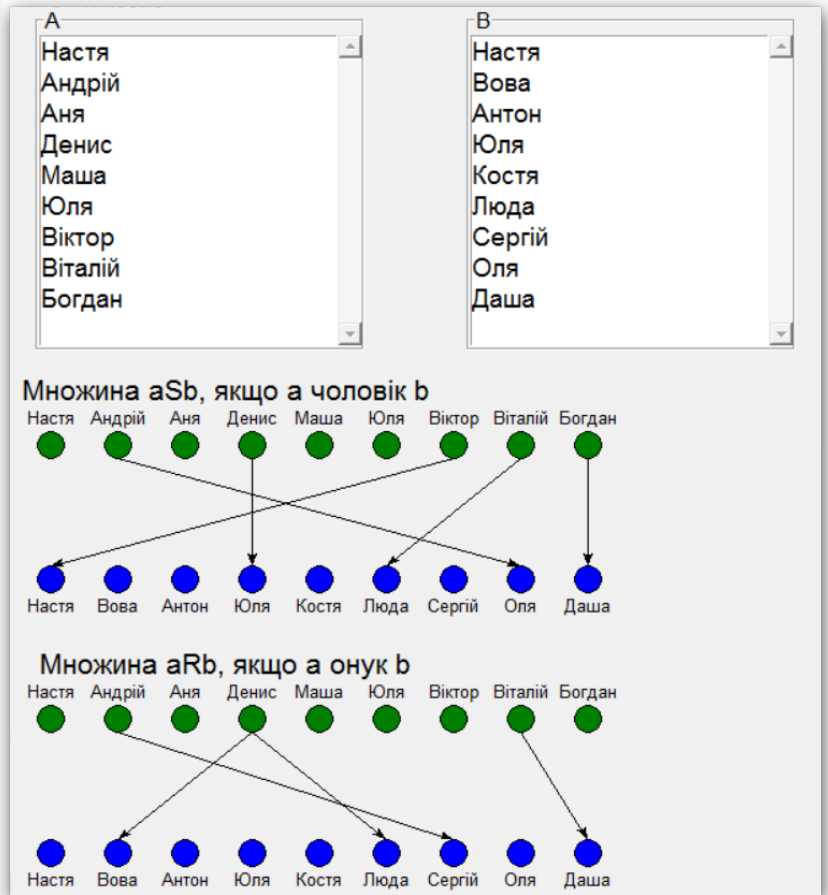
Варіант:

Бровченко Анастасія
група ІО-64
варіант 8

Алгоритм генерування відношень:

```
def A_cholovik_B():
    a = set()
    for i in self.A:
        if i in self.men:
            a.add(i)
    b = set()
    for j in self.B:
        if j in self.women:
            b.add(j)
    S = []
    for i in range(min(len(a), len(b))):
        p = random.choice(list(a))
        q = random.choice(list(b))
        S.append([p, q])
        a.remove(p)
        b.remove(q)
    return S

def A_onuk_B():
    a = set()
    for i in self.A:
        if i in self.men:
            a.add(i)
    b = self.B
    R = []
    for i in range(min(len(a), len(b))):
        p = random.choice(list(a))
        q = random.choice(list(b))
        if [p, q] not in self.S:
            if [p, q] not in R:
                R.append([p, q])
    return R
```



Алгоритми виконання операцій над відношеннями:

```
def but1():
    canv.delete("all")
    canv.create_text(150, 20, text='R \u222A S', font='Arial 16')
    dict_b1 = {}
    dict_b2 = {}
    V = self.R+self.S
    for i in range(len(self.A)):
        canv.create_text(30+i*50, 50, text=list(self.A)[i], font='Arial 10')
        canv.create_oval([20+i*50, 60], [40+i*50, 80], fill="green")
        dict_b1.update({list(self.A)[i]: [30+i*50, 80]})
    for j in range(len(self.B)):
        canv.create_text(30+j*50, 190, text=list(self.B)[j], font='Arial 10')
        canv.create_oval([20+j*50, 160], [40+j*50, 180], fill="blue")
        dict_b2.update({list(self.B)[j]: [30+j*50, 160]})
    for k in V:
        canv.create_line(dict_b1[k[0]], dict_b2[k[1]], arrow=LAST)
```

Операції над відношеннями

R ∪ S

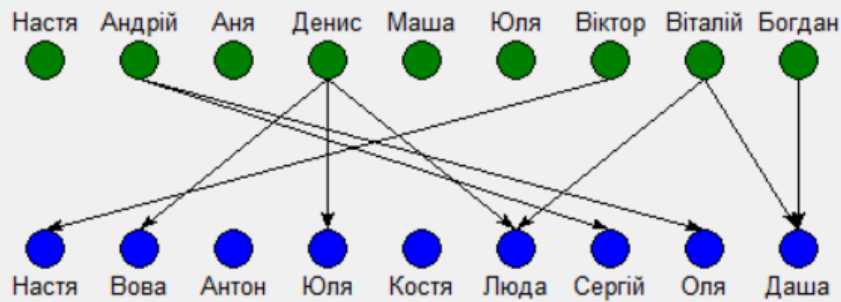
R ∩ S

R \ S

U \ R

S⁻¹

R ∪ S



```
def but2():
    canv.delete("all")
    canv.create_text(150, 20, text='R ∪ S', font='Arial 16')
    dict_b1 = {}
    dict_b2 = {}
    V = []
    for i in self.R:
        if i in self.S:
            V.append(i)

    for i in range(len(self.A)):
        canv.create_text(30+i*50, 50, text=list(self.A)[i], font='Arial 10')
        canv.create_oval([20+i*50, 60], [40+i*50, 80], fill="green")
        dict_b1.update({list(self.A)[i]: [30+i*50, 80]})

    for j in range(len(self.B)):
        canv.create_text(30+j*50, 190, text=list(self.B)[j], font='Arial 10')
        canv.create_oval([20+j*50, 160], [40+j*50, 180], fill="blue")
        dict_b2.update({list(self.B)[j]: [30+j*50, 160]})

    for k in V:
        if len(V) != 0:
            canv.create_line(dict_b1[k[0]], dict_b2[k[1]], arrow=LAST)
```

Операції над відношеннями

R ∪ S

R ∩ S

R \ S

U \ R

S⁻¹

R ∩ S



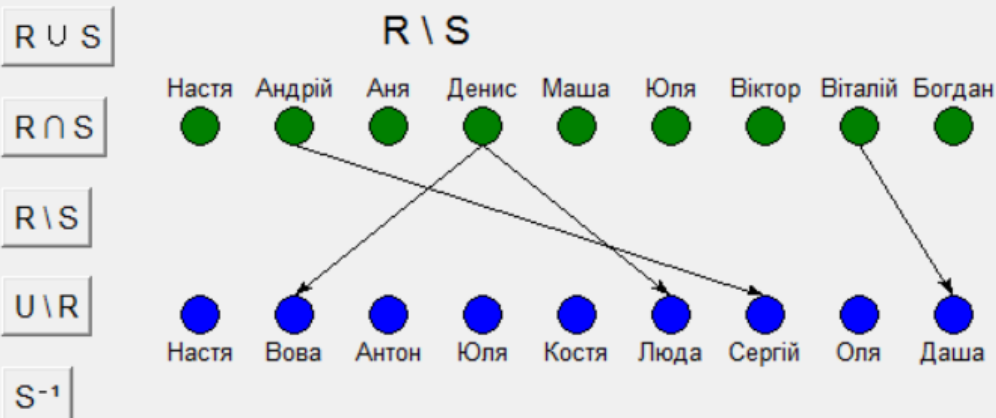
```

def but3():
    canv.delete("all")
    canv.create_text(150, 20, text='R \ S', font='Arial 16')
    dict_b1 = {}
    dict_b2 = {}
    V = copy.deepcopy(self.R)
    for i in V:
        if i in self.S:
            V.remove(i)

    for i in range(len(self.A)):
        canv.create_text(30+i*50, 50, text=list(self.A)[i], font='Arial 10')
        canv.create_oval([20+i*50, 60], [40+i*50, 80], fill="green")
        dict_b1.update({list(self.A)[i]: [30+i*50, 80]})
    for j in range(len(self.B)):
        canv.create_text(30+j*50, 190, text=list(self.B)[j], font='Arial 10')
        canv.create_oval([20+j*50, 160], [40+j*50, 180], fill="blue")
        dict_b2.update({list(self.B)[j]: [30+j*50, 160]})
    for k in V:
        canv.create_line(dict_b1[k[0]], dict_b2[k[1]], arrow=LAST)

```

Операції над відношеннями



```

def but4():
    canv.delete("all")
    canv.create_text(150, 20, text='U \ R', font='Arial 16')
    dict_b1 = {}
    dict_b2 = {}

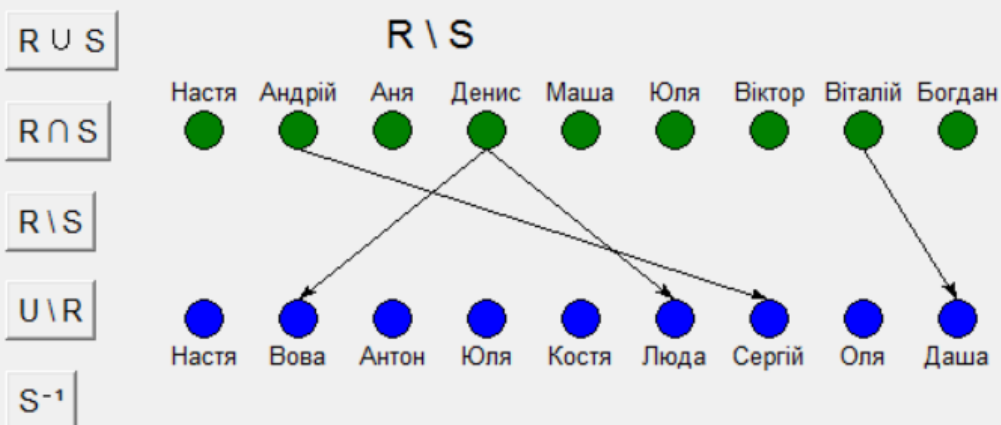
    v = []
    a = set()
    for i in self.A:
        if i in self.men:
            a.add(i)
    b = copy.deepcopy(self.B)
    for i in a:
        for j in b:
            v.append([i,j])

    for i in v:
        if i in self.R:
            v.remove(i)

    for i in range(len(self.A)):
        canv.create_text(30+i*50, 50, text=list(self.A)[i], font='Arial 10')
        canv.create_oval([20+i*50, 60], [40+i*50, 80], fill="green")
        dict_b1.update({list(self.A)[i]: [30+i*50, 80]})
    for j in range(len(self.B)):
        canv.create_text(30+j*50, 190, text=list(self.B)[j], font='Arial 10')
        canv.create_oval([20+j*50, 160], [40+j*50, 180], fill="blue")
        dict_b2.update({list(self.B)[j]: [30+j*50, 160]})
    for k in v:
        if len(v) != 0:
            canv.create_line(dict_b1[k[0]], dict_b2[k[1]], arrow=LAST)

```

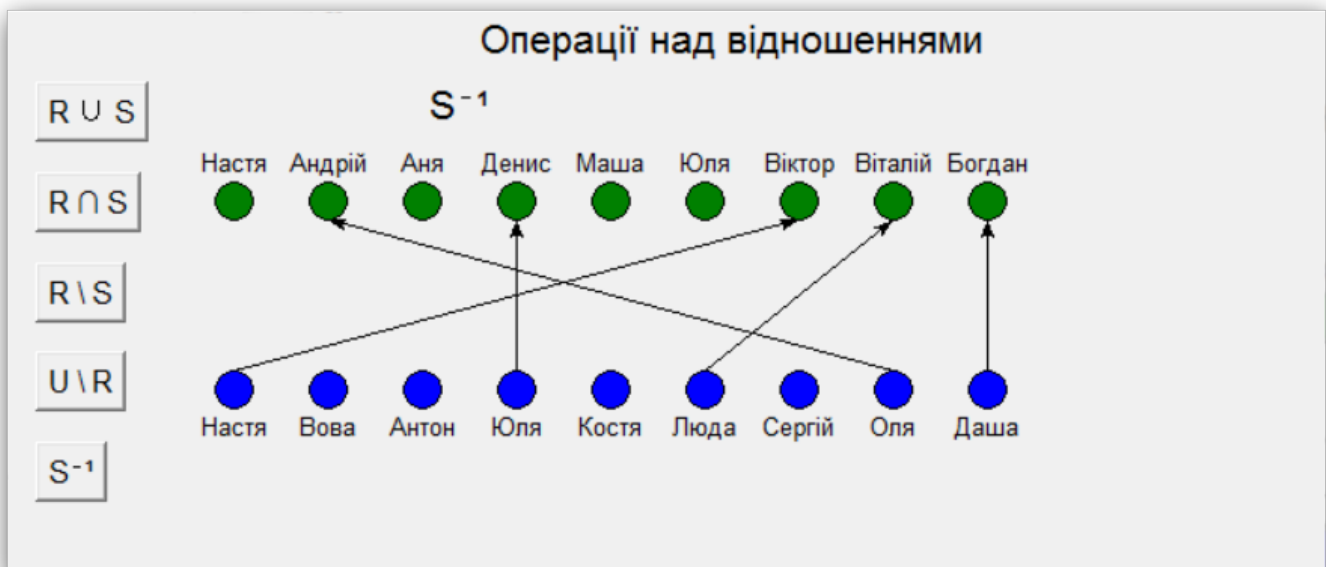
Операції над відношеннями



```

def but5():
    canv.delete("all")
    canv.create_text(150, 20, text='S-1', font='Arial 16')
    dict_b1 = {}
    dict_b2 = {}
    V = copy.deepcopy(self.S)
    for i in V:
        i[0], i[1] = i[1], i[0]
    for i in range(len(self.A)):
        canv.create_text(30+i*50, 50, text=list(self.A)[i], font='Arial 10')
        canv.create_oval([20+i*50, 60], [40+i*50, 80], fill="green")
        dict_b1.update({list(self.A)[i]: [30+i*50, 80]})
    for j in range(len(self.B)):
        canv.create_text(30+j*50, 190, text=list(self.B)[j], font='Arial 10')
        canv.create_oval([20+j*50, 160], [40+j*50, 180], fill="blue")
        dict_b2.update({list(self.B)[j]: [30+j*50, 160]})
    for k in V:
        canv.create_line(dict_b2[k[0]], dict_b1[k[1]], arrow=LAST)

```



Висновок:

У ході виконання лабораторної роботи я закріпила свої знання про операції над бінарними відношеннями, навчилась реалізувати їх в програмному коді.