НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт з комп'ютерного практикуму №7

«Розробка моделі на основі петрі-об'єктної технології.»

роботи з дисципліни: « Моделювання систем »

Студент: Мєшков Андрій Ігорович_______________

Група: ІП-15__________________________________

Викладач: асистент Дифучин А. Ю._________

Київ, 2024

Завдання

1. Розглянути алгоритм Петрі-об'єткного моделювання, реалізований в бібліотеці PetriObjModelPaint (див. github StetsenkoInna). Виконати тестування запропонованого алгоритму на моделі мережі масового обслуговування. **15 балів**.
2. За текстом завдання 2 практикуму 5 розробити відповідні Петрі-об'єктри та побудувати Петрі-об'єктну модель системи. Отримати результати імітаційного моделювання. Зробити висновки про функціонування моделі. **30 балів**.
3. За текстом завдання 3 практикуму 5 розробити відповідні Петрі-об'єктри та побудувати Петрі-об'єктну модель системи. Отримати результати імітаційного моделювання. Зробити висновки про функціонування моделі. **30 балів**.
4. Побудувати математичні рівняння, що описують побудовану за текстом завдання 1 практикуму 5 Петрі-об'єктну модель.**20 балів**.
5. Сформулювати переваги та недоліки використання технології Петрі-об'єктного моделювання. **5 балів**.

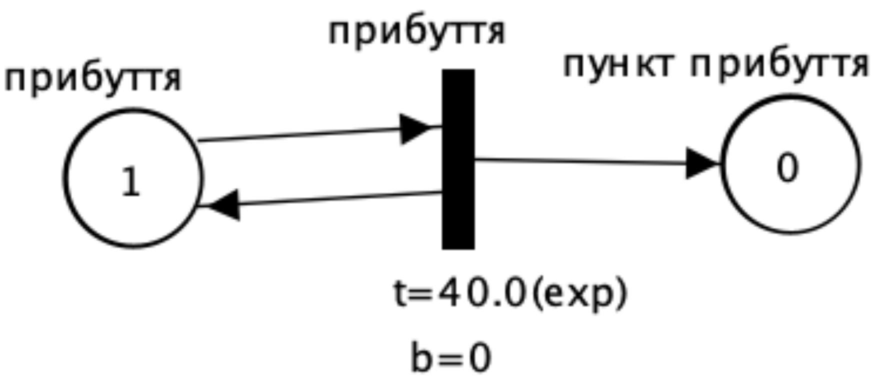
Хід роботи

Завдання 1. Результати запуску

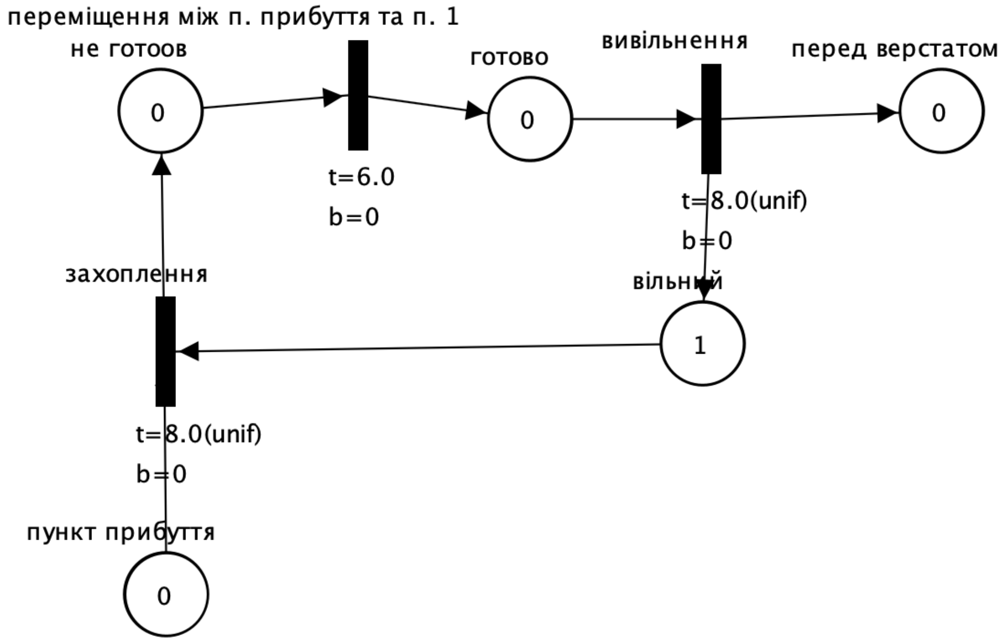
```
Mean value of queue
1.7544339269523328
0.0029378120481112764
0.004034948079828735
1.0060345600683515E-5
Mean value of channel worked
0.7124723551215196
0.05304701703885806
0.061582364814753876
0.035614476835919895
Estimation precision
 Mean value of queue  precision:
1.767417303900741 %
2.0729317296241225 %
0.8737019957183673 %
0.603456006835144 %
 Mean value of channel worked  precision:
0.2139558653333808 %
1.7647832613739665 %
0.6736051374937477 %
1.0708976780002846 %
```

Завдання 2

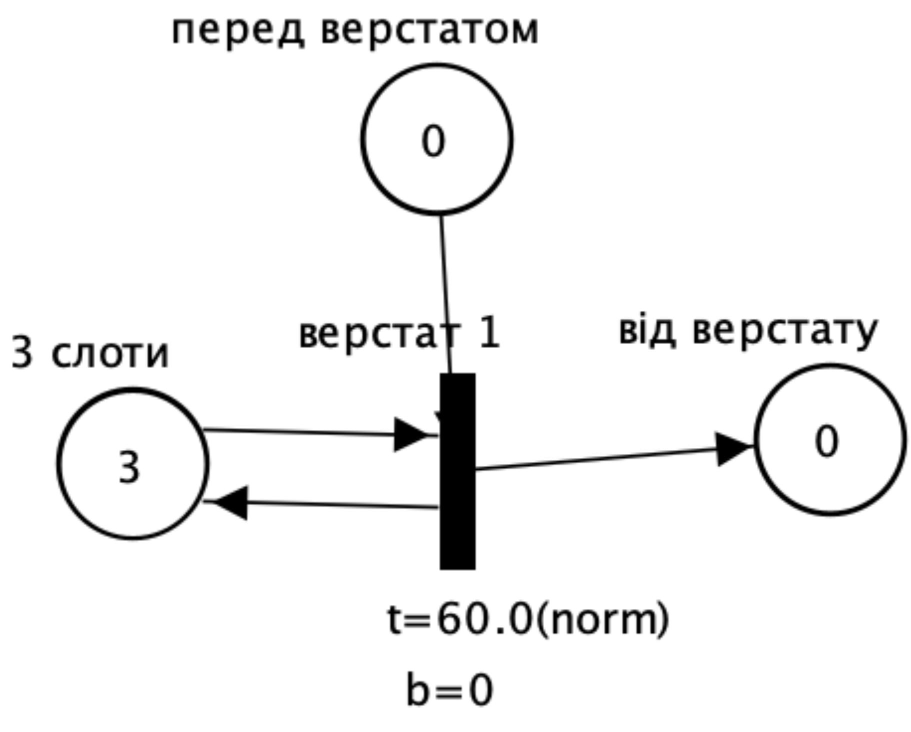
Об’єкт надсилання



Об'єкт перевезення



Об'єкт верстату



Збережемо файли як код та створимо модель.

```java
    public static PetriObjModel getModel2() throws ExceptionInvalidTimeDelay,
ExceptionInvalidNetStructure{

        ArrayList<PetriSim> list = new ArrayList<>();
        list.add(new PetriSim(NetLibrary.Create2()));
        list.add(new PetriSim(NetLibrary.Transfer2()));
        list.add(new PetriSim(NetLibrary.Machine2()));
        list.add(new PetriSim(NetLibrary.Transfer2()));
        list.add(new PetriSim(NetLibrary.Machine2()));
        list.add(new PetriSim(NetLibrary.Transfer2()));

        list.get(0).getNet().getListP()[1] = list.get(1).getNet().getListP()[0];
        list.get(1).getNet().getListP()[3] = list.get(2).getNet().getListP()[0];
        list.get(2).getNet().getListP()[2] = list.get(3).getNet().getListP()[0];
        list.get(3).getNet().getListP()[3] = list.get(4).getNet().getListP()[0];
        list.get(4).getNet().getListP()[2] = list.get(5).getNet().getListP()[0];

        PetriObjModel model = new PetriObjModel(list);
        return model;
    }
```

Отримаємо результат на симуляції в 10000

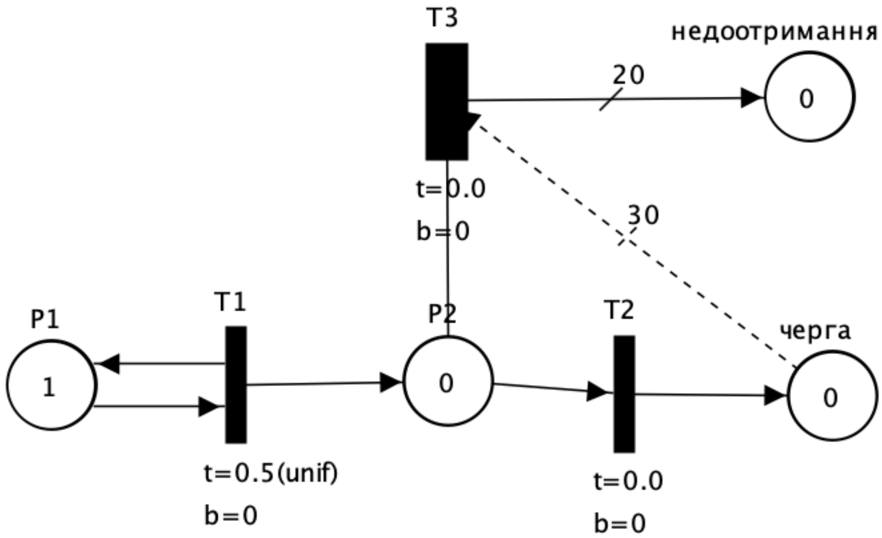
```
Mark in Net  5-2    0  0
Mark in Net  7-2-2   0  0  0  0  0
Mark in Net  7-2-3   0  3  0
Mark in Net  7-2-2   0  0  0  0  1
Mark in Net  7-2-3   0  3  0
Mark in Net  7-2-2   0  0  0  246  1
```

Висновки

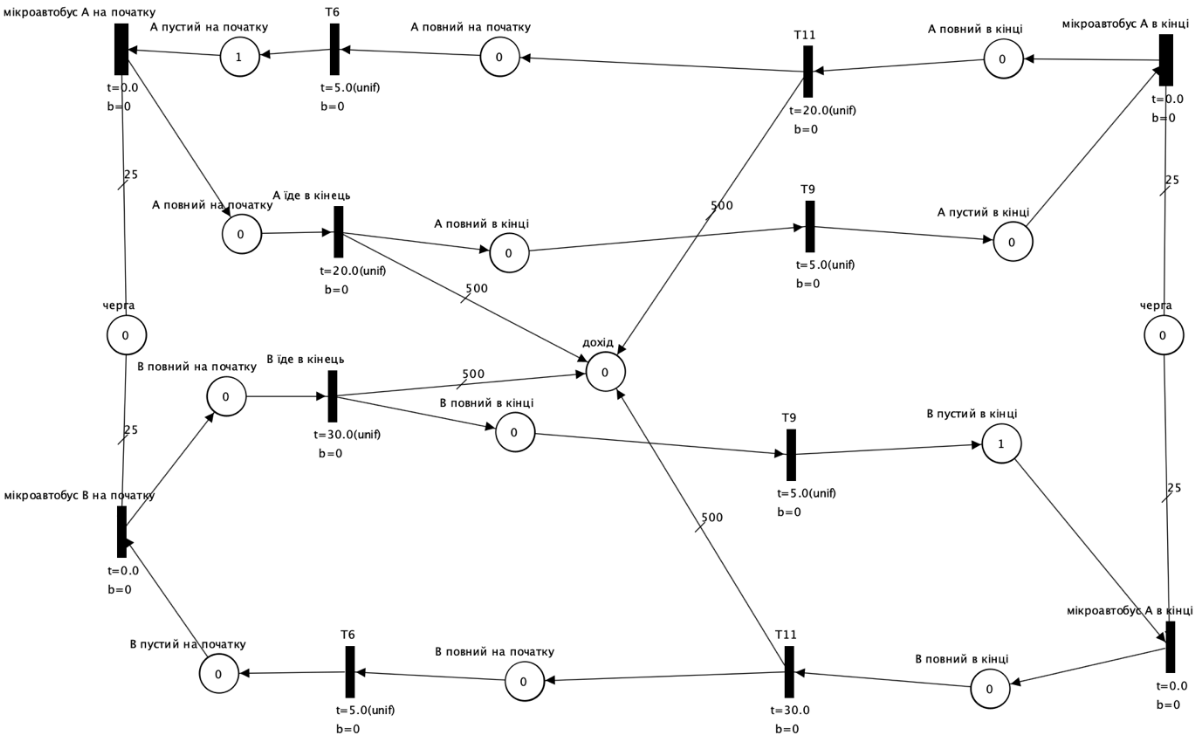
- Система демонструє нормальне функціонування з послідовним передаванням маркерів через мережу.
- Однак спостерігається затримка в останньому вузлі, що може бути спричинено низькою продуктивністю або недостатньою швидкістю обробки маркерів в одному з вузлів (Machine2).
- Слід звернути увагу на оптимізацію параметрів обробки в цих вузлах для уникнення накопичення маркерів.

Завдання 3

Об’єкт черги



Об’єкт доріг



Створимо модель

```
    public static PetriObjModel getModel3() throws ExceptionInvalidTimeDelay,
ExceptionInvalidNetStructure{

        ArrayList<PetriSim> list = new ArrayList<>();
        list.add(new PetriSim(NetLibrary.Queue3()));
        list.add(new PetriSim(NetLibrary.Roads3()));
        list.add(new PetriSim(NetLibrary.Queue3()));

        list.get(0).getNet().getListP()[2] = list.get(1).getNet().getListP()[0];
        list.get(2).getNet().getListP()[2] = list.get(1).getNet().getListP()[8];

        PetriObjModel model = new PetriObjModel(list);
        return model;
    }
```

Отримаємо результат

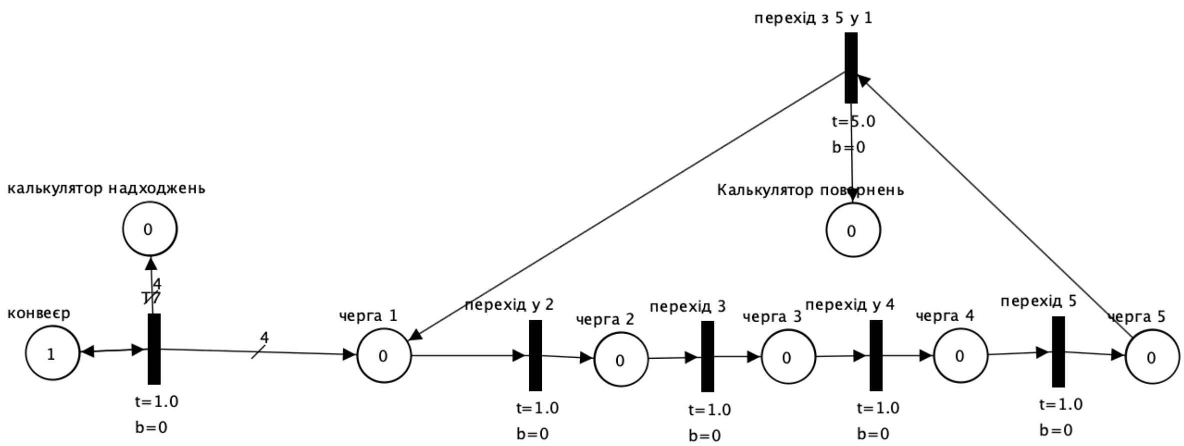
```
Mark in Net  7-3-1   0  0  29  233220
Mark in Net  7-3-2   29  0  0  0  329500  0  0  0  30  0  0  0  0  0  0
Mark in Net  7-3-1   0  0  30  233900
```

Висновки:

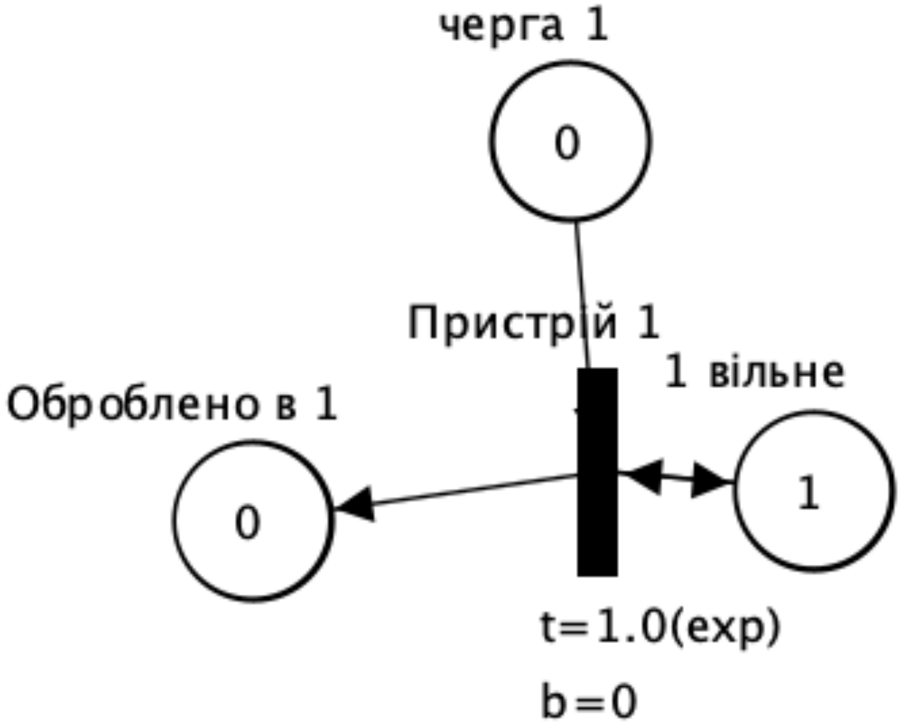
- Основна проблема цієї моделі — перевантаження черг і доріг, що може викликати затримки в обробці маркерів.
- Зменшення кількості маркерів на наступних ітераціях свідчить про те, що система поступово стабілізується, але це займає багато часу.

Завдання 4

Об’єкт конвеєру



Об'єкт верстату



Побудуємо математичне рівняння

2 типи: конвеєр та [illegible]

1) Конвеєр

$T = \{T_1, T_2, T_3, T_4, T_5, T_6\}$

$P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$

$A = \{(P_1, T_1), (T_1, P_1), (T_1, P_2), (P_2, T_2),$
$(T_2, P_3), (P_3, T_3), (T_3, P_4), (P_4, T_4),$
$(T_4, P_5), (P_5, T_5), (T_5; P_6), (P_6; T_6), (T_6; P_2)\}$

$K = \{(0; 1), (0; 1), (0; 1), (0; 1), (0; 1), (0; 1)\}$

$R = \{\cancel{1}\text{кр}; 1, 1, 1, 1, 5\}$

$J = \varnothing$

$W = \{1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$

$U = \{(T_1, P_2), (T_2, P_3), (T_3, P_4), (T_4, P_5)$
$(T_5, P_6)\}$

2) Примусові

$T = \{T_1\}$

$P = \{P_1, P_2, P_3\}$

$A = \{(P_1, T_1), (T_1, P_3), (P_2, T_1), (T_1, P_2)\}$

$K = \{(0;1)\}$

$R = \{1\ exp\}$

$\varphi = \varnothing$

$W = \{1, 1, 1, 1\}$

$U = \{(P_1, T_1)\}$

Загальна модель

$T = T_K \cup T_{n1} \cup T_{n2} \cup T_{p3} \cup T_{n4} \cup T_{n5}$

$K = K_K \cup K_{n1} \cup K_{n2} \cup K_{n3} \cup K_{n4} \cup K_{n5}$

Аналогічно інші фрагменти.

1) Конвеєр

$$S^+(0) = \left\{ \begin{pmatrix}1\\0\\0\\0\\0\\0\end{pmatrix}, \begin{pmatrix}\{\infty\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\end{pmatrix} \right\}$$

$P_1 \geq 1 \Rightarrow Z(T_1, 0) = 1 \quad \Psi(0) = \{T_1\} \Rightarrow$

$\Rightarrow X(T_1) = 1$

$D^-: M_{P_1}(0) = 1 - 1 = 0 \quad E_{T_1}(0) = \{0+1\}$

$M_{P_2}(0) = 0 - 0 = 0 \quad E_{T_2}(0) = \{\infty\}$

$M_{P_3}(0) = 0 - 0 = 0 \quad E_{T_3}(0) = \{\infty\}$

$M_{P_4}(0) = 0 - 0 = 0 \quad E_{T_4}(0) = \{\infty\}$

$M_{P_5}(0) = 0 - 0 = 0 \quad E_{T_5}(0) = \{\infty\}$

$M_{P_6}(0) = 0 - 0 = 0 \quad E_{T_6}(0) = \{\infty\}$

$$S(0) = \left\{ \begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}, \begin{pmatrix}\{1\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\\\{\infty\}\end{pmatrix} \right\} \qquad \begin{matrix} Z(T_1, 0) = 0 \\ Z(T_6, 0) = 0 \end{matrix}$$

$t_1 = 1$

$D^+: \gamma(T_1, 1) = 1 \quad E_{T_1}(1) = \{\varnothing\}$

$\gamma(T_6, 1) = 1 \quad E_{T_2}(1) = \{\varnothing\}$

$\dots$

$E_{T_6}(1) = \{\varnothing\}$

$M_{P_1}(1) = 0 + 1 = 1$

$M_{P_2}(1) = 0 + 4 = 4$

$M_{P_3}(1) = 0$

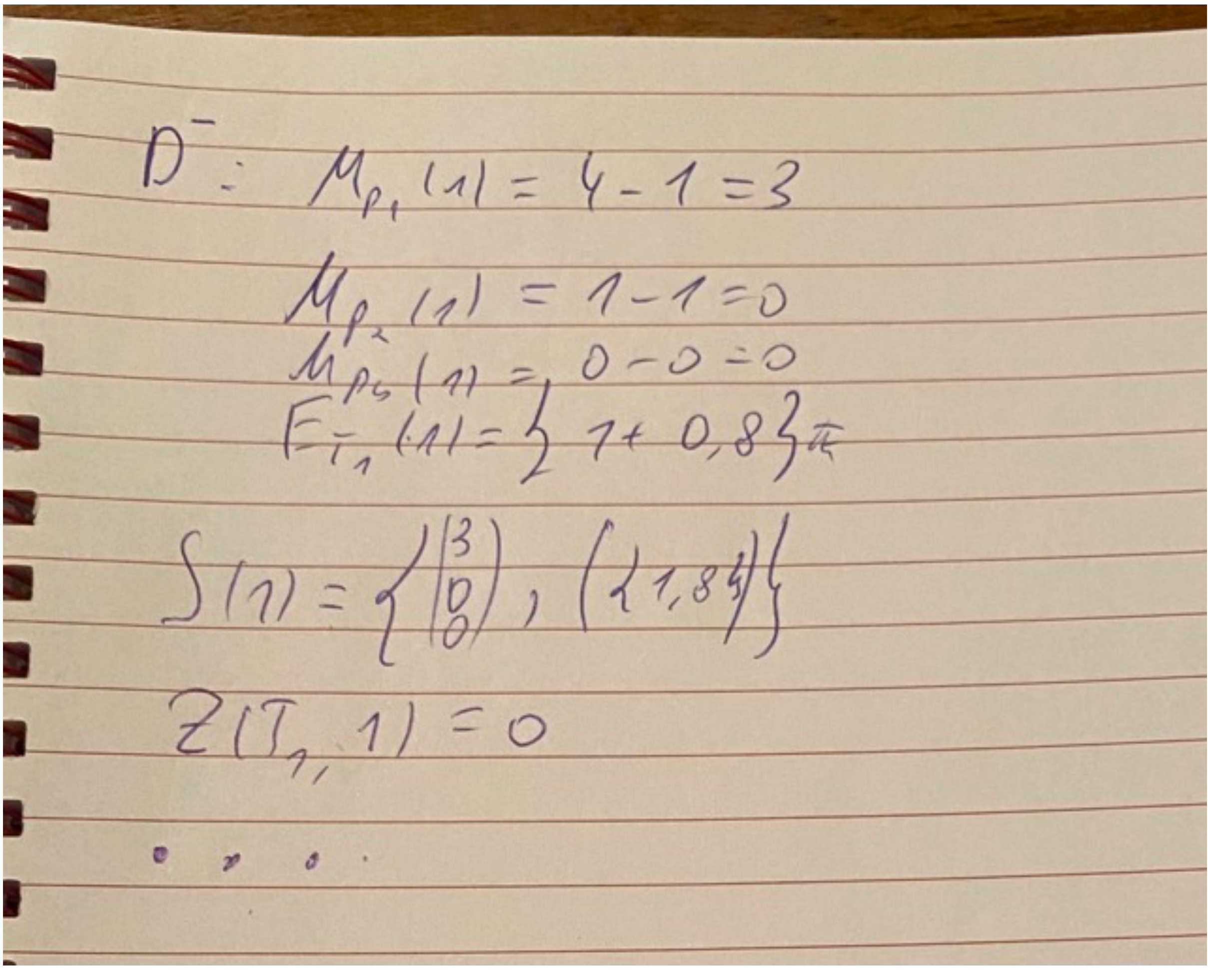
$\dots$

$$S^+(1) = \left\{ \begin{pmatrix} 1 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \{\varnothing\} \\ \{\varnothing\} \\ \{\varnothing\} \\ \{\varnothing\} \\ \{\varnothing\} \\ \{\varnothing\} \end{pmatrix} \right\}$$

Запускаємо перехід 1

$P_1 \geq 1,\ P_2 \geq 1 \Rightarrow Z(T_1, 1) = 1$

$\Psi(1) = \{T_1\} \Rightarrow x(T_1) = 1$



D^-: $M_{p_1}(1) = 4 - 1 = 3$

$M_{p_2}(1) = 1 - 1 = 0$

$M_{p_6}(1) = 0 - 0 = 0$

$E_{T_1}(1) = \{1 + 0,8\}_{\pi}$

$S(1) = \left\{ \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}, (\{1,8\}) \right\}$

$Z(T_1, 1) = 0$

. . . .

Перевагами Петрі-об’єктного моделювання є універсальність. Оскільки, об’єкти можна перевикористовувати. Також, легше вносити зміни у модель, оскільки не треба повторювати 1 зміну декілька разів.

Недоліками Петрі-об’єктного моделювання є складність. Для малої системи без схожих елементів використання недоцільне.

ВИСНОВКИ

У результаті виконання практичної роботи було перероблено 3 задачі за допомогою Петрі-об’єктного моделювання. Було побудовано математичне рівняння для однієї моделі.