

# **Лабораторная работа №13**

**Задание для самостоятельного выполнения**

Герра Гарсия Максимиано

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
2.1	Схема модели . . . . .	5
2.2	Описание модели . . . . .	5
2.3	Анализ сети Петри . . . . .	7
2.4	Реализация модели в CPN Tools . . . . .	8
2.5	Пространство состояний . . . . .	11
<b>3</b>	<b>Выводы</b>	<b>16</b>

# Список иллюстраций

2.1	Сеть для выполнения домашнего задания . . . . .	7
2.2	Дерево достижимости . . . . .	8
2.3	Модель задачи в CPN Tools . . . . .	9
2.4	Задание деклараций . . . . .	10
2.5	Запуск модели . . . . .	10
2.6	Граф пространства состояний . . . . .	11

# 1 Постановка задачи

1. Используя теоретические методы анализа сетей Петри, провести анализ сети (с помощью построения дерева достижимости). Определить, является ли сеть безопасной, ограниченной, сохраняющей, имеются ли тупики.
2. Промоделировать сеть Петри с помощью CPNTools.
3. Вычислить пространство состояний. Сформировать отчёт о пространстве состояний и проанализировать его. Построить граф пространства состояний.

## **2 Выполнение лабораторной работы**

### **2.1 Схема модели**

Заявка (команды программы, операнды) поступает в оперативную память (ОП), затем передается на прибор (центральный процессор, ЦП) для обработки. После этого заявка может равновероятно обратиться к оперативной памяти или к одному из двух внешних запоминающих устройств (В1 и В2). Прежде чем записать информацию на внешний накопитель, необходимо вторично обратиться к центральному процессору, определяющему состояние накопителя и выдающему необходимую управляющую информацию. Накопители (В1 и В2) могут работать в 3-х режимах:

- 1) В1 — занят, В2 — свободен;
- 2) В2 — свободен, В1 — занят;
- 3) В1 — занят, В2 — занят.

### **2.2 Описание модели**

Сеть Петри моделируемой системы представлена на рис. 2.1.  
Множество позиций:

P1 — состояние оперативной памяти (свободна / занята);

P2 — состояние внешнего запоминающего устройства B1 (свободно / занято);

P3 — состояние внешнего запоминающего устройства B2 (свободно / занято);

P4 — работа на ОП и B1 закончена;

P5 — работа на ОП и B2 закончена;

P6 — работа на ОП, B1 и B2 закончена;

Множество переходов:

T1 — ЦП работает только с RAM и B1;

T2 — обрабатываются данные из RAM и с B1 переходят на устройство вывода;

T3 — CPU работает только с RAM и B2;

T4 — обрабатываются данные из RAM и с B2 переходят на устройство вывода;

T5 — CPU работает только с RAM и с B1, B2;

T6 — обрабатываются данные из RAM, B1, B2 и переходят на устройство вывода.

Функционирование сети Петри можно рассматривать как срабатывание переходов, в ходе которого происходит перемещение маркеров по позициям:

- работа CPU с RAM и B1 отображается запуском перехода T1 (удаление маркеров из P1, P2 и появление в P1, P4), что влечет за собой срабатывание перехода T2, т.е. передачу данных с RAM и B1 на устройство вывода;
- работа CPU с RAM и B2 отображается запуском перехода T3 (удаление маркеров из P1 и P3 и появление в P1 и P5), что влечет за собой срабатывание перехода T4, т.е. передачу данных с RAM и B2 на устройство вывода;

- работа CPU с RAM, B1 и B2 отображается запуском перехода T5 (удаление маркеров из P4 и P5 и появление в P6), далее срабатывание перехода T6, и данные из RAM, B1 и B2 передаются на устройство вывода;
- состояние устройств восстанавливается при срабатывании: RAM — переходов T1 или T2; B1 — переходов T2 или T6; B2 — переходов T4 или T6.

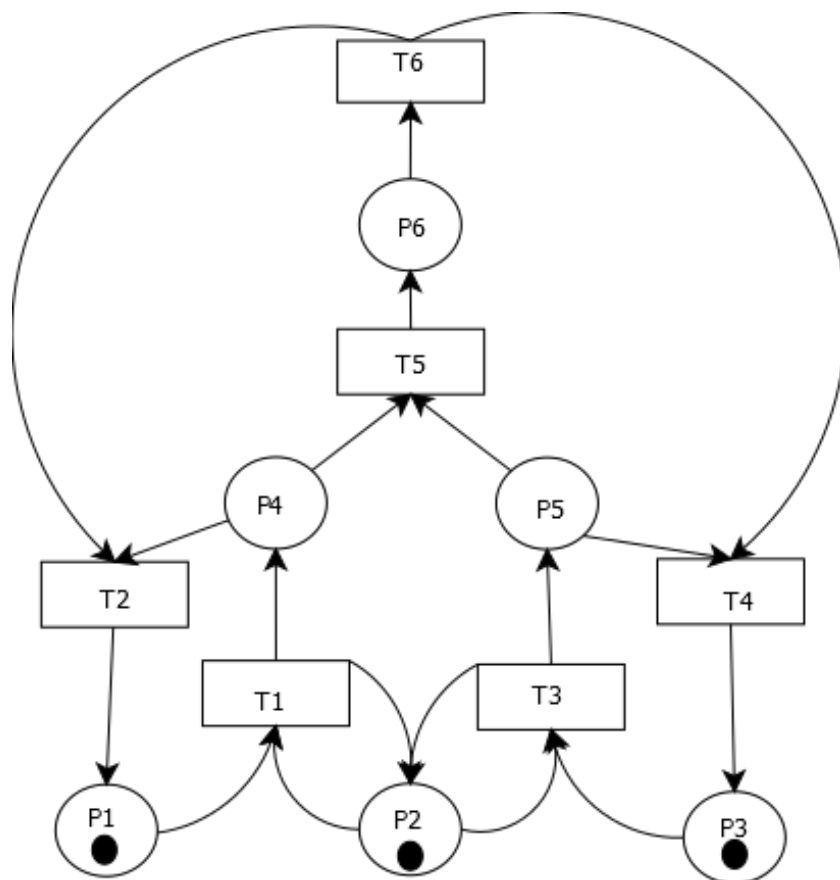


Рис. 2.1: Сеть для выполнения домашнего задания

## 2.3 Анализ сети Петри

Построим дерево достижимости (рис. 2.2).

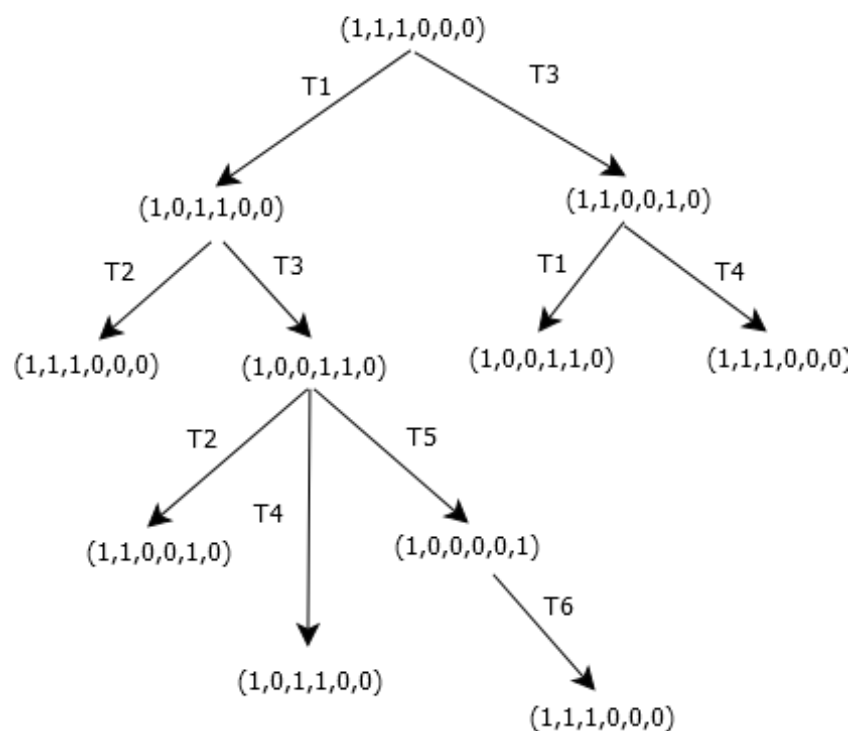


Рис. 2.2: Дерево достижимости

Можем увидеть, что представленная сеть:

- безопасна, поскольку в каждой позиции количество фишек не превышает 1;
- ограничена, так как существует такое целое  $k$ , что число фишек в каждой позиции не может превысить  $k$  (в данном случае  $k=1$ );
- сеть не имеет тупиков;
- сеть не является сохраняющей, так как при переходах  $t_5$  и  $t_6$  количество фишек меняется.

## 2.4 Реализация модели в CPN Tools

Реализуем описанную ранее модель в CPN Tools. С помощью контекстного меню создаем новую сеть, далее нам понадобятся 6



позиций и 6 блоков переходов, затем их нужно соединить, а также задать параметры и начальные значения. Получаем готовую модель (рис. 2.3).

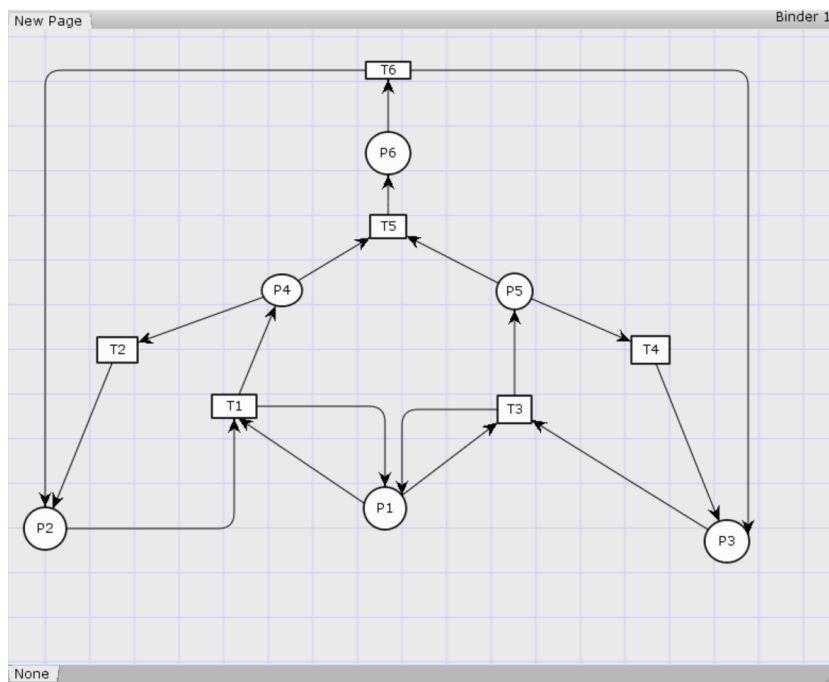


Рис. 2.3: Модель задачи в CPN Tools

Также зададим нужные декларации (рис. 2.4).

- ▼ Declarations
  - ▶ Standard declarations
  - ▼ memory
    - ▼ colset RAM = unit with mem;
    - ▼ colset B1 = unit with storage1;
    - ▼ colset B2 = unit with storage2;
    - ▼ colset B1xB2 = product B1\*B2;
    - ▼ var ram:RAM;
    - ▼ var b1:B1;
    - ▼ var b2:B2;
  - ▶ Monitors
- petri-net

Рис. 2.4: Задание деклараций

Запустив модель, можно посмотреть, как она работает (рис. 2.5).

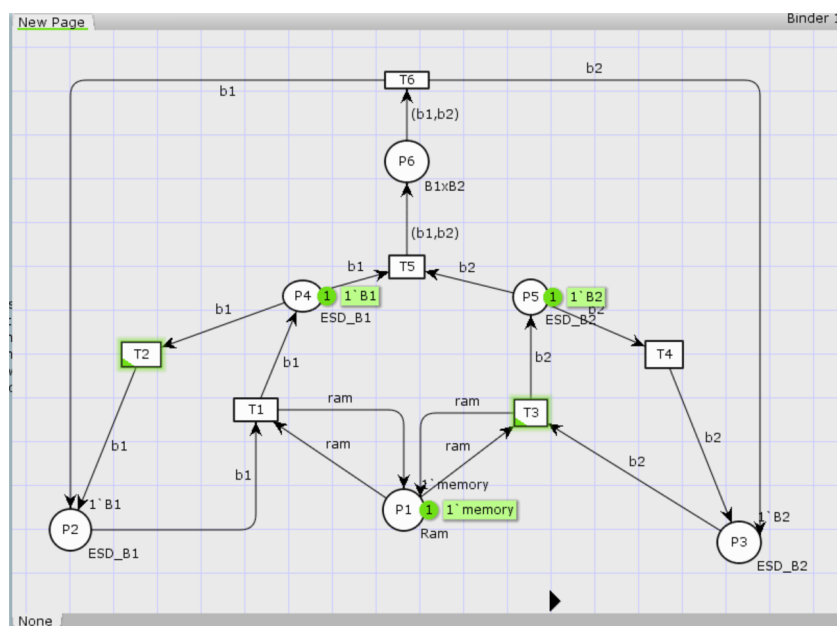


Рис. 2.5: Запуск модели

## 2.5 Пространство состояний

Изучим пространство состояний. Сформируем граф пространства состояний, их всего 5 (2.6).

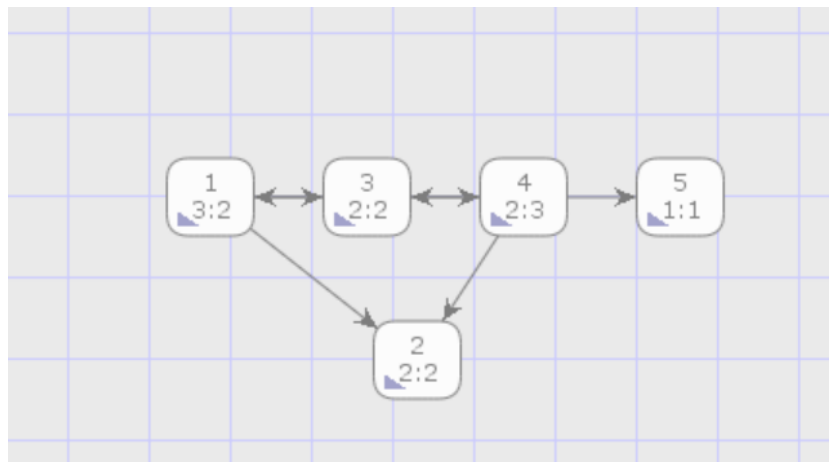


Рис. 2.6: Граф пространства состояний

Вычислим пространство состояний. Прежде, чем пространство состояний может быть вычислено и проанализировано, необходимо сформировать код пространства состояний. Этот код создается, когда используется инструмент Войти в пространство состояний. Вход в пространство состояний занимает некоторое время. Затем, если ожидается, что пространство состояний будет небольшим, можно просто применить инструмент Вычислить пространство состояний к листу, содержащему страницу сети. Сформируем отчёт о пространстве состояний и проанализируем его. Чтобы сохранить отчет, необходимо применить инструмент Сохранить отчет о пространстве состояний к листу, содержащему страницу сети и ввести имя файла отчета.

Из отчета можно увидеть:

- есть 5 состояний и 10 переходов между ними, strongly connected components (SCC) graph содержит 1 вершину и 0 переходов.

- Затем указаны границы значений для каждого элемента: состояние P1 всегда заполнено 1 элементом, а остальные содержат максимум 1 элемент, минимум – 0.
- Также указаны границы в виде мультимножеств.
- Маркировка home для всех состояний, так как в любую позицию мы можем попасть из любой другой маркировки.
- Маркировка dead равная None, так как нет состояний, из которых переходов быть не может.
- В конце указано, что бесконечно часто могут происходить переходы T1, T2, T3, T4, но не обязательно, также состояние T5 необходимо для того, чтобы система не попадала в тупик, а состояние T6 происходит всегда, если доступно.

CPN Tools state space report for:

/home/openmodelica/petri\_net.cpn

Report generated: Sat May 3 00:38:28 2025

## Statistics

-----  
 ----

## State Space

Nodes: 5  
 Arcs: 10  
 Secs: 0  
 Status: Full

## Scc Graph

Nodes: 1  
 Arcs: 0

Secs: 0

## Boundedness Properties

-----

----

### Best Integer Bounds

	Upper	Lower
petri'P1 1	1	1
petri'P2 1	1	0
petri'P3 1	1	0
petri'P4 1	1	0
petri'P5 1	1	0
petri'P6 1	1	0

### Best Upper Multi-set Bounds

petri'P1 1	1`memory
petri'P2 1	1`storage1
petri'P3 1	1`storage2
petri'P4 1	1`storage1
petri'P5 1	1`storage2
petri'P6 1	1`(storage1,storage2)

### Best Lower Multi-set Bounds

petri'P1 1	1`memory
petri'P2 1	empty
petri'P3 1	empty
petri'P4 1	empty

petri'P5 1	empty
petri'P6 1	empty

#### Home Properties

---

----

Home Markings  
All

#### Liveness Properties

---

----

Dead Markings  
None

Dead Transition Instances  
None

Live Transition Instances  
All

#### Fairness Properties

---

----

petri'T1 1	No Fairness
petri'T2 1	No Fairness
petri'T3 1	No Fairness
petri'T4 1	No Fairness
petri'T5 1	Just
petri'T6 1	Fair

## **3 Выводы**

В результате выполнения данной лабораторной работы я выполнил задание для самостоятельного выполнения.