

# **Лабораторная работа №13**

**Задание для самостоятельного выполнения**

Хватов М. Г.

# 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	Анализ сети Петри . . . . .	7
2.3	Реализация модели в CPN Tools . . . . .	8
<b>3</b>	<b>Вывод</b>	<b>15</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	Граф 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.

Сеть Петри моделируемой системы представлена на рис. 2.1.

Множество позиций:

- Р1 — состояние оперативной памяти (свободна / занята);
- Р2 — состояние внешнего запоминающего устройства В1 (свободно / занято);
- Р3 — состояние внешнего запоминающего устройства В2 (свободно / занято);
- Р4 — работа на ОП и В1 закончена;
- Р5 — работа на ОП и В2 закончена;
- Р6 — работа на ОП, В1 и В2 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.

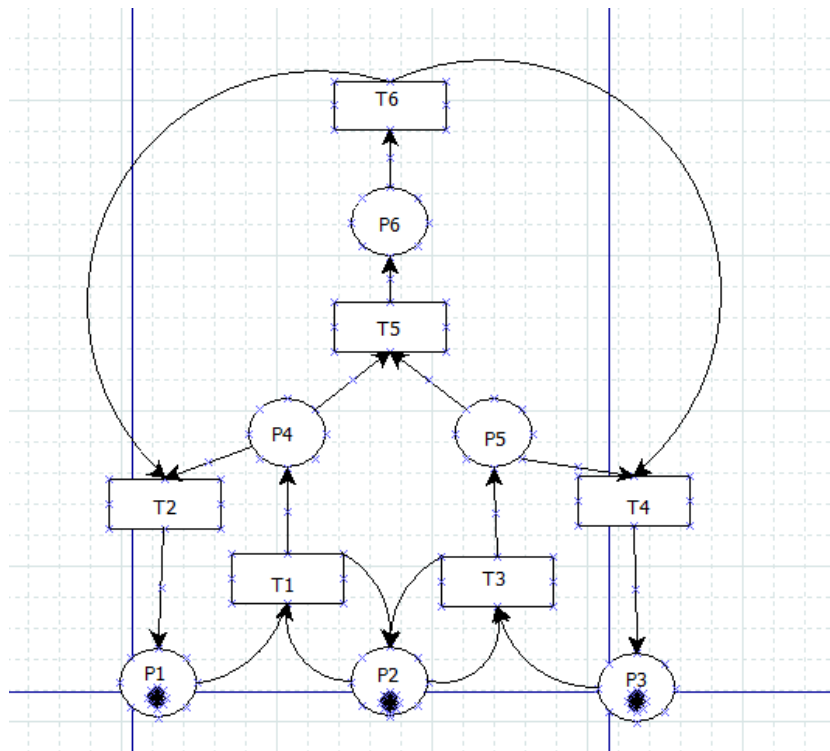


Рис. 2.1: Сеть для выполнения домашнего задания

## 2.2 Анализ сети Петри

Построим дерево достижимости (рис. 2.2).

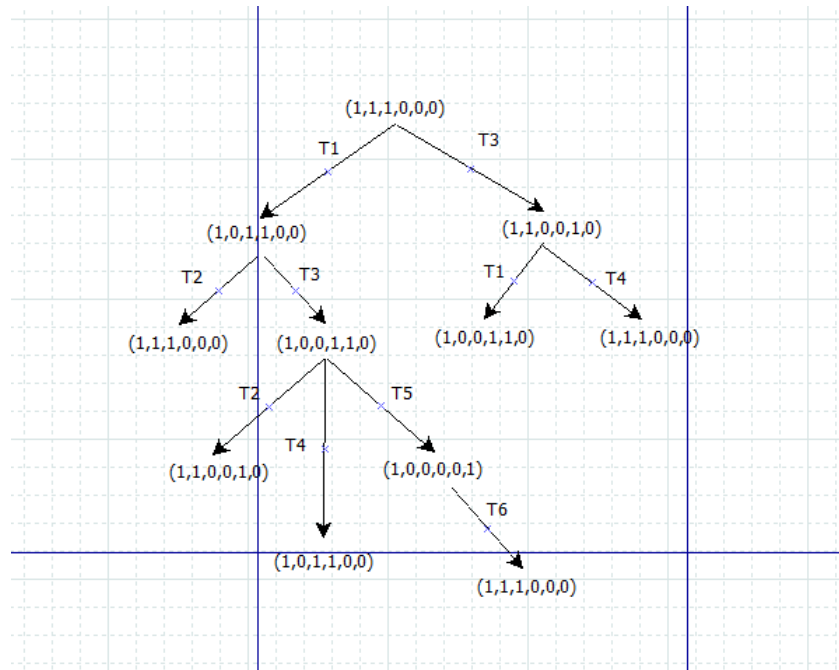


Рис. 2.2: Дерево достижимости

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  $t5$  и  $t6$  количество фишек меняется.

## 2.3 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).



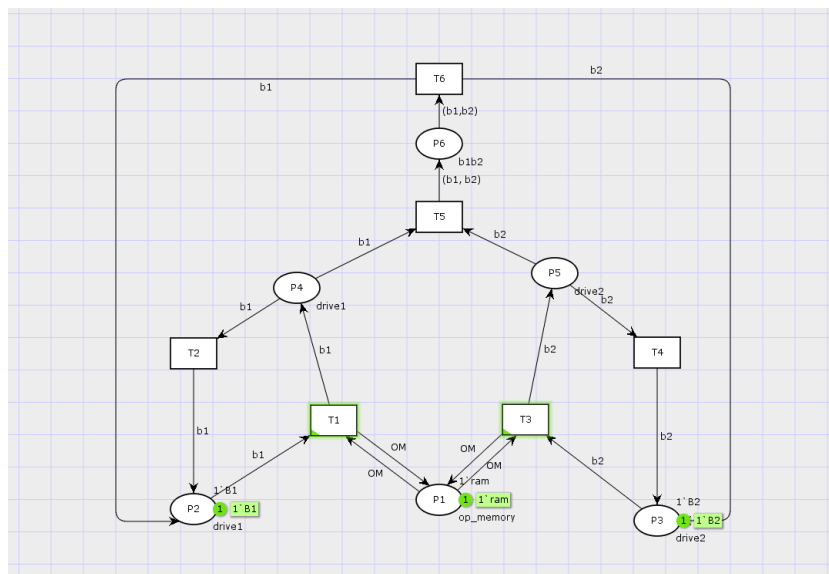
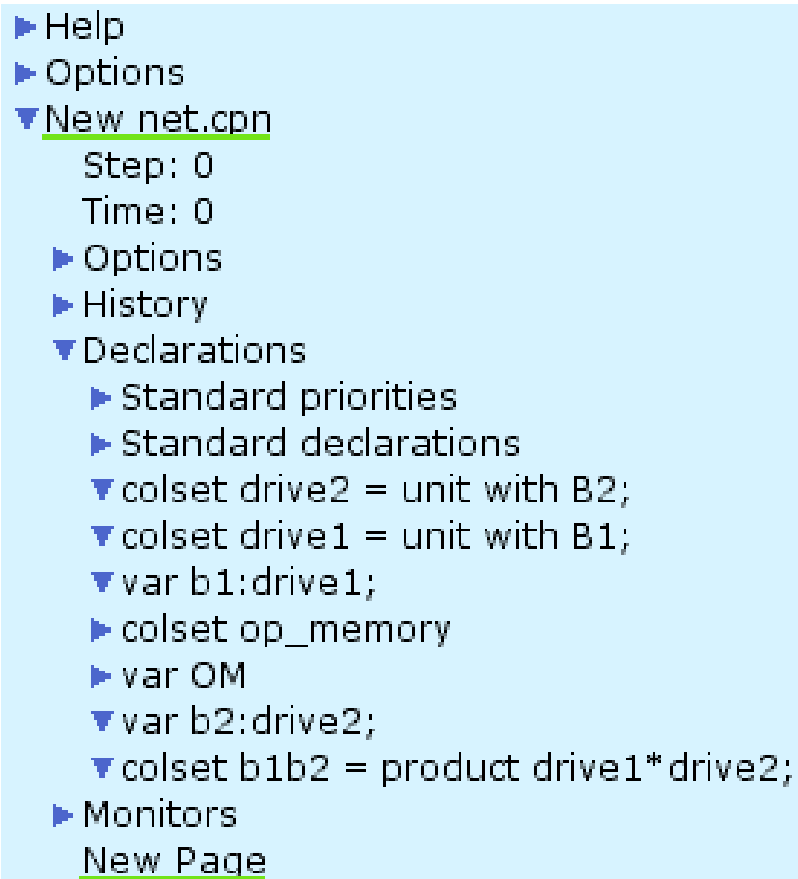


Рис. 2.3: Модель задачи в CPN Tools

Также зададим нужные декларации (рис. 2.4).



- ▶ Help
- ▶ Options
- ▼ New net.cpn
  - Step: 0
  - Time: 0
  - ▶ Options
  - ▶ History
  - ▼ Declarations
    - ▶ Standard priorities
    - ▶ Standard declarations
    - ▼ colset drive2 = unit with B2;
    - ▼ colset drive1 = unit with B1;
    - ▼ var b1:drive1;
    - ▶ colset op\_memory
    - ▶ var OM
    - ▼ var b2:drive2;
    - ▼ colset b1b2 = product drive1\*drive2;
  - ▶ Monitors
- New Page

Рис. 2.4: Задание деклараций

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

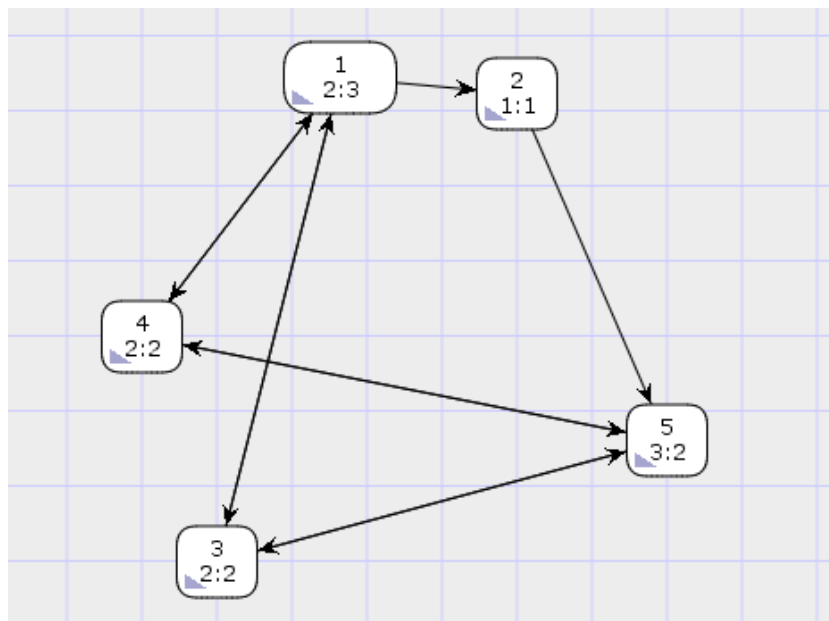


Рис. 2.5: Граф состояний

CPN Tools state space report for:

<unsaved net>

Report generated: Thu May 1 11:12:14 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
New_Page' P1 1	1	1
New_Page' P2 1	1	0
New_Page' P3 1	1	0
New_Page' P4 1	1	0
New_Page' P5 1	1	0
New_Page' P6 1	1	0

### Best Upper Multi-set Bounds

New_Page' P1 1	1`ram
New_Page' P2 1	1`B1
New_Page' P3 1	1`B2
New_Page' P4 1	1`B1
New_Page' P5 1	1`B2
New_Page' P6 1	1`(B1,B2)

### Best Lower Multi-set Bounds

New_Page' P1 1	1`ram
New_Page' P2 1	empty

New_Page'P3 1	empty
New_Page'P4 1	empty
New_Page'P5 1	empty
New_Page'P6 1	empty

#### Home Properties

---

---

#### Home Markings

All

#### Liveness Properties

---

---

#### Dead Markings

None

#### Dead Transition Instances

None

#### Live Transition Instances

All

#### Fairness Properties

-----  
---

Impartial Transition Instances

None

Fair Transition Instances

New\_Page'T6 1

Just Transition Instances

New\_Page'T5 1

Transition Instances with No Fairness

New\_Page'T1 1

New\_Page'T2 1

New\_Page'T3 1

New\_Page'T4 1

### 3 Вывод

Сеть Петри, 1-дезопасна (ограничена по одному маркеру в местах), живая(все переходы могут быть выполнены), обратима (любое состояние достижимо из любого другого), сильно связана. Отсутствуют тупиковые состояния и мертвые переходы, однако только переход  $T_6$  является справедливым,  $T_5$  - обоснованный, а остальные не обладают свойствами справедливости, что может повлиять на их регулярное исполнение без внешнего управления.