Задание для самостоятельного выполнения

Отчёт по лабораторной работе №13

Ибатулина Дарья Эдуардовна

Содержание

# 1 Вводная часть

## 1.1 Цель и 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 [1].

## 1.2 Теоретическое введение

Протоколы передачи данных - это наборы правил, определяющих порядок и способы обмена информацией между участниками вычислительных или телекоммуникационных систем. Они регулируют формат сообщений, последовательность их передачи, временные интервалы, а также методы обнаружения и исправления ошибок. В современной практике протоколы используются на всех уровнях модели OSI, начиная с физического и заканчивая прикладным [2].

Реальные протоколы передачи данных часто имеют сложную структуру, что затрудняет их анализ и отладку. Для исследования их свойств и выявления возможных ошибок широко применяется моделирование, в частности, с помощью сетей Петри и их расширений - раскрашенных сетей Петри (Coloured Petri Nets, CPN). Этот подход позволяет наглядно представить процессы передачи и подтверждения данных, а также учесть такие важные аспекты, как потеря пакетов и подтверждений, задержки и порядок доставки [3].

В данной лабораторной работе рассматривается простейший протокол передачи данных в ненадёжной сети. Передача осуществляется по следующему принципу: отправитель разбивает исходное сообщение на части (пакеты) и отправляет их получателю, ожидая подтверждения (ACK) о доставке каждого пакета перед отправкой следующего. Если подтверждение не получено (например, из-за потери пакета или ACK), пакет передаётся повторно. Такой механизм обеспечивает надёжность передачи даже в условиях, когда сеть может терять отдельные сообщения.

Модель протокола строится в среде CPN Tools с использованием раскрашенных сетей Петри, что позволяет формально описать все возможные состояния системы и переходы между ними, а также смоделировать случайные потери пакетов и подтверждений. Анализ полученной модели даёт возможность оценить корректность работы протокола, выявить возможные тупиковые состояния и оптимизировать алгоритмы передачи данных [4].

# 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 (B1 и B2). Прежде чем записать информацию на внешний накопитель, необходимо вторично обратиться к центральному процессору, определяющему состояние накопителя и выдающему необходимую управляющую информацию. Накопители (B1 и B2) могут работать в 3-х режимах:

1. B1 — занят, B2 — свободен;
2. B2 — свободен, B1 — занят;
3. B1 — занят, B2 — занят.

## 2.2 Описание модели

Сеть Петри моделируемой системы представлена на рис. [1](#fig:001).

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 расматривать 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.

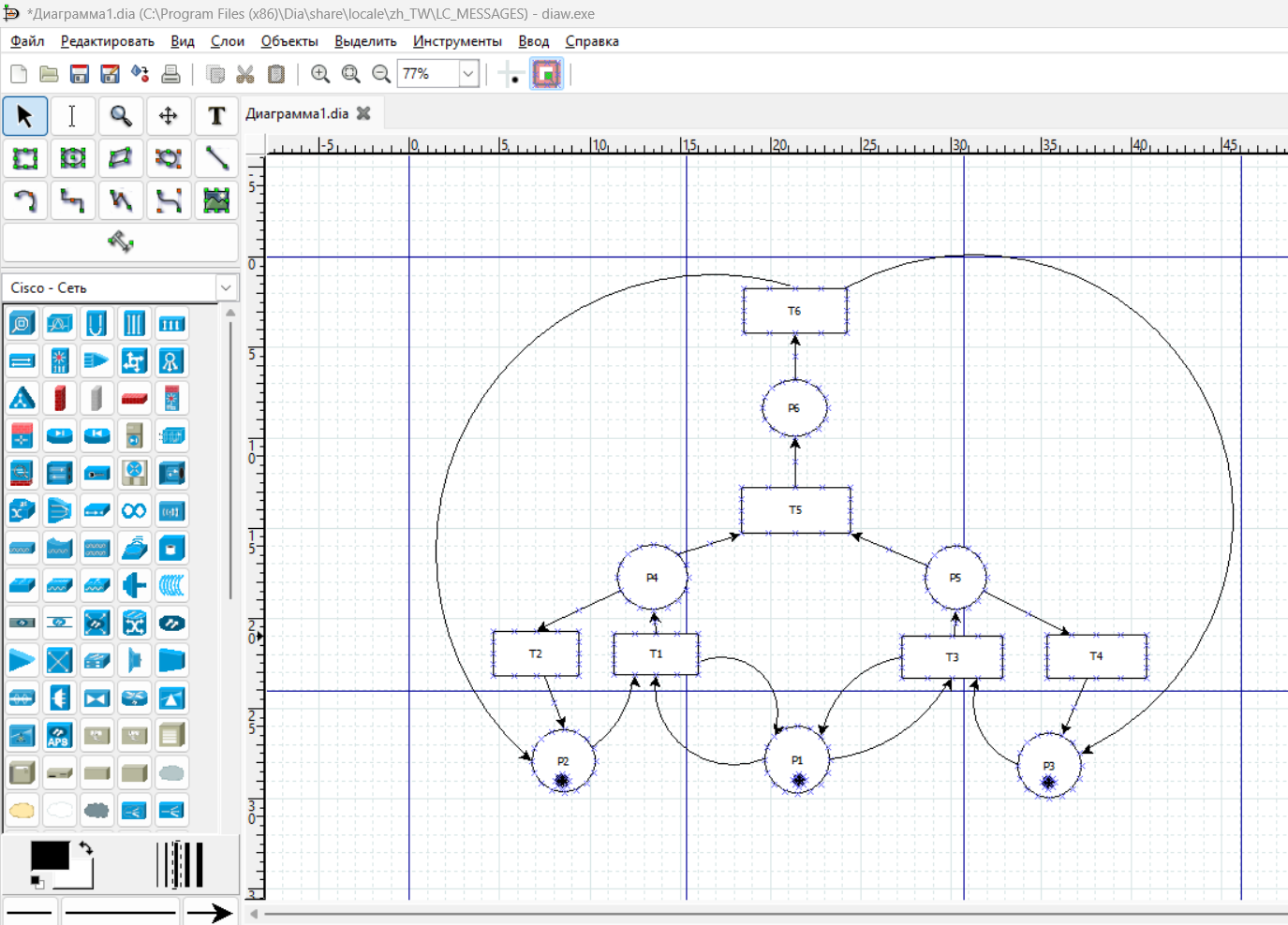


Figure 1: Сеть для выполнения домашнего задания

## 2.3 Анализ сети Петри

Построим дерево достижимости (рис. [2](#fig:002)).

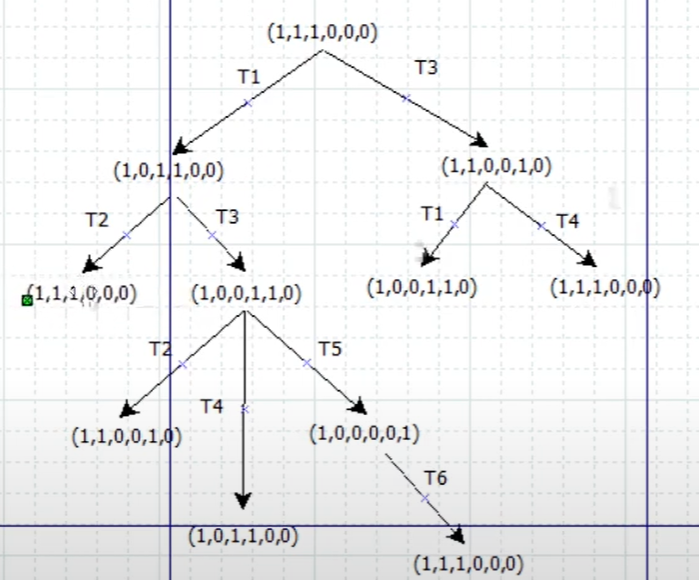


Figure 2: Дерево достижимости

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.

## 2.4 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. [3](#fig:003)).

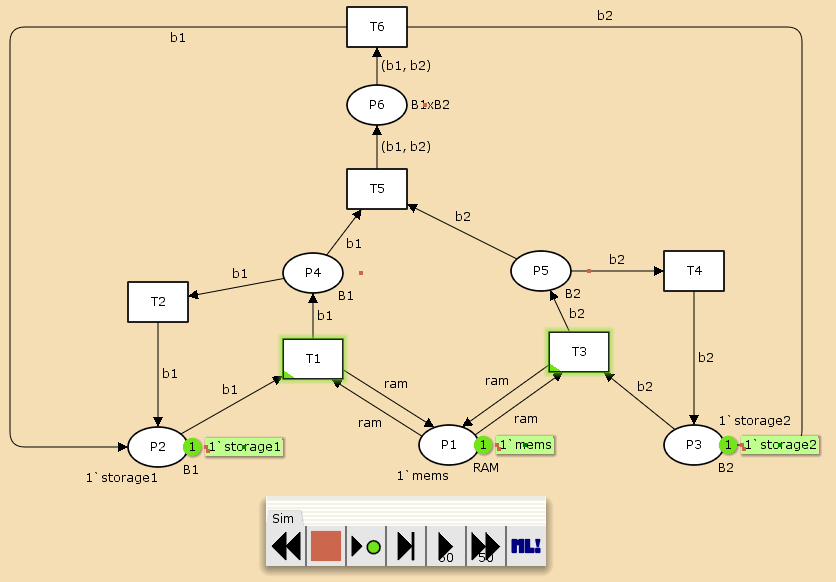


Figure 3: Модель задачи в CPN Tools

Также зададим нужные декларации (рис. [4](#fig:004)).

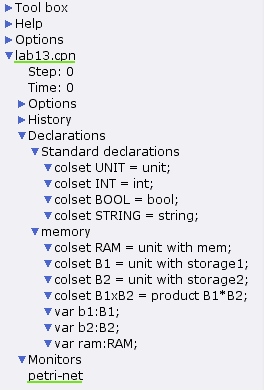


Figure 4: Задание деклараций

Запустив модель, можно посмотреть, как она работает (рис. [5](#fig:005)).

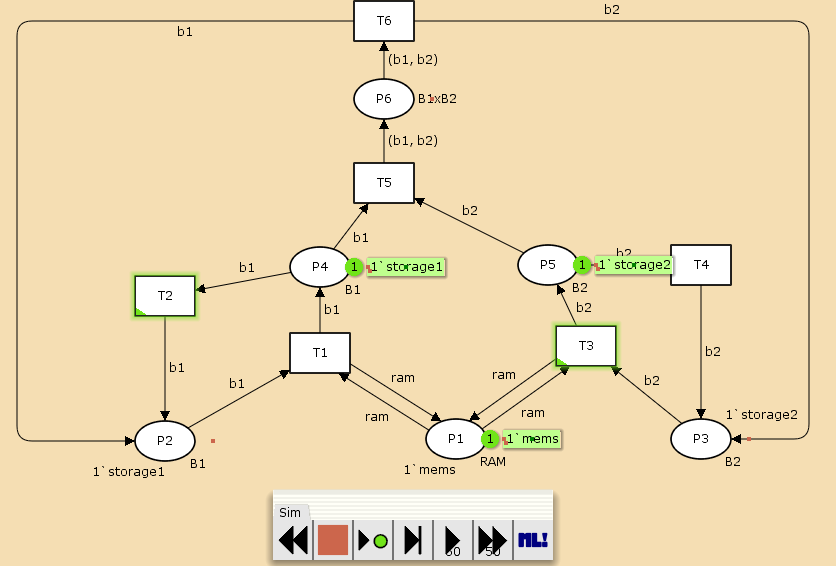


Figure 5: Запуск модели

## 2.5 Пространство состояний

Изучим пространство состояний. Сформируем граф пространства состояний, их всего 5 (рис. [6](#fig:006)).

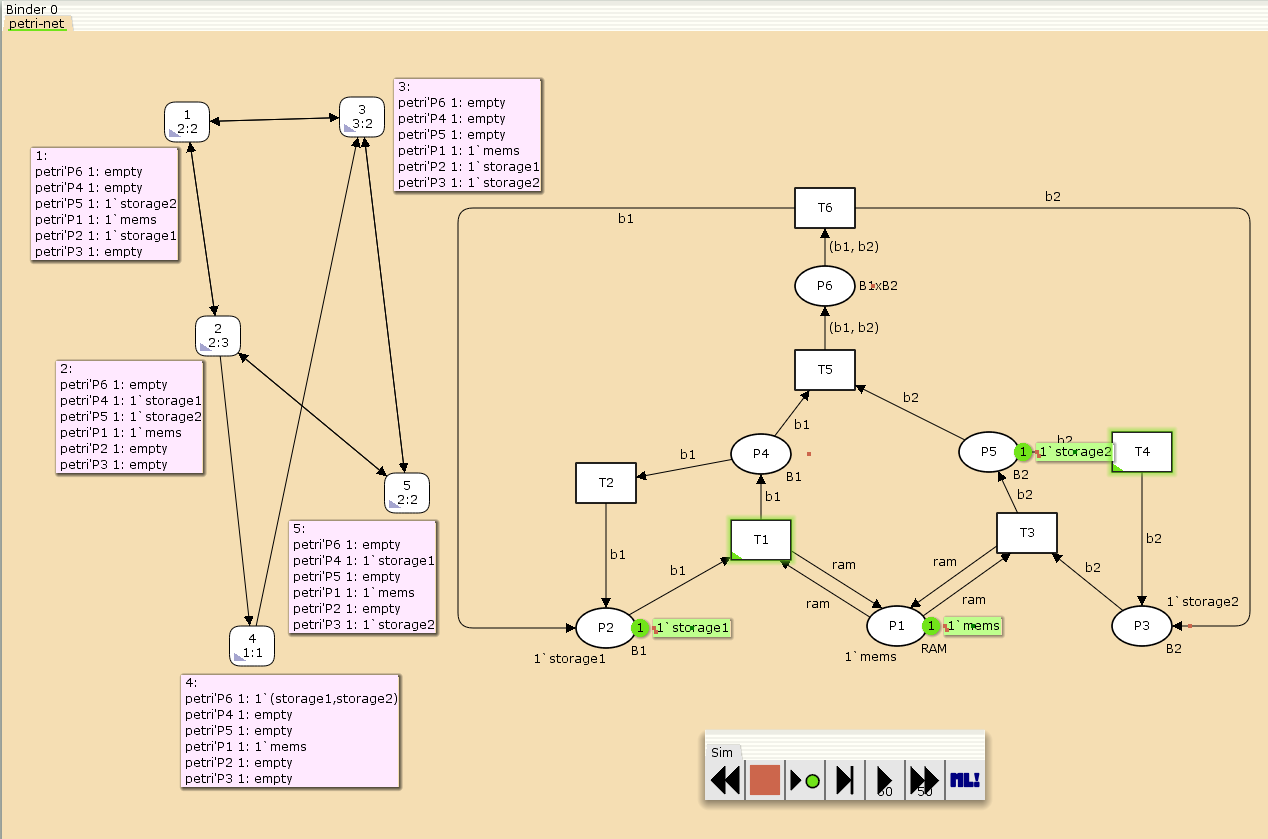


Figure 6: Граф пространства состояний

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/lab13.cpn  
Report generated: Mon Apr 28 02:01:37 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`mems  
 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`mems  
 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 выполнила задание для самостоятельного выполнения, а именно провела анализ сети Петри, построила сеть в CPN Tools, построила граф состояний и провела его анализ.

# Список литературы

1. Королькова А.В., Кулябов Д.С. Моделирование информационных процессов: Лабораторная работа №12. РУДН, 2025.

2. Jensen K., Kristensen L.M., Wells L. [Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems](https://doi.org/10.1007/s10009-007-0038-x) // International Journal on Software Tools for Technology Transfer. 2007. Т. 9. С. 213–254.

3. CPN Tools – A tool for editing, simulating, and analyzing Colored Petri nets [Электронный ресурс]. 2023. URL: <https://cpntools.org> (дата обращения: 28.04.2025).

4. Дворянский Л.В., Ломазова И.А. Имитационное моделирование и верификация вложенных сетей Петри с использованием CPN Tools // Труды НИУ ВШЭ. Издательский дом ВШЭ, 2012. С. 1–15.