Лабораторная работа №10

Задача об обедающих мудрецах

Ибатулина Дарья Эдуардовна, НФИбд-01-22

Содержание

# 1 Цель работы

Реализовать модель задачи об обедающих мудрецах в CPN Tools.

# 2 Задание

* Реализовать модель задачи об обедающих мудрецах в CPN Tools;
* Вычислить пространство состояний, сформировать отчет о нем и построить граф.

# 3 Теоретическое введение

CPN Tools — специальное программное средство, предназначенное для моделирования иерархических временных раскрашенных сетей Петри. Такие сети эквивалентны машине Тьюринга и составляют универсальную алгоритмическую систему, позволяющую описать произвольный объект. CPN Tools позволяет визуализировать модель с помощью графа сети Петри и применить язык программирования CPN ML (Colored Petri Net Markup Language) для формализованного описания модели [1,2].

**Назначение CPN Tools:**

* разработка сложных объектов и моделирование процессов в различных приклад- ных областях, в том числе:
* моделирование производственных и бизнес-процессов;
* моделирование систем управления производственными системами и роботами;
* спецификация и верификация протоколов, оценка пропускной способности сетей и качества обслуживания, проектирование телекоммуникационных устройств и сетей.

**Основные функции CPN Tools:**

* создание (редактирование) моделей;
* анализ поведения моделей с помощью имитации динамики сети Петри;
* построение и анализ пространства состояний модели.

# 4 Выполнение лабораторной работы

Пять мудрецов сидят за круглым столом и могут пребывать в двух состояниях - думать и есть. Между соседями лежит одна палочка для еды. Для приёма пищи необходимы две палочки. Палочки – пересекающийся ресурс. Необходимо синхронизировать процесс еды так, чтобы мудрецы не умерли с голода. [3–6] Рисуем граф сети [7]. Для этого с помощью контекстного меню создаём новую сеть, добавляем позиции, переходы и дуги (рис. [1](#fig:001)).

Начальные данные:

* позиции: мудрец размышляет (philosopher thinks), мудрец ест (philosopher eats), палочки находятся на столе (sticks on the table)
* переходы: взять палочки (take sticks), положить палочки (put sticks)

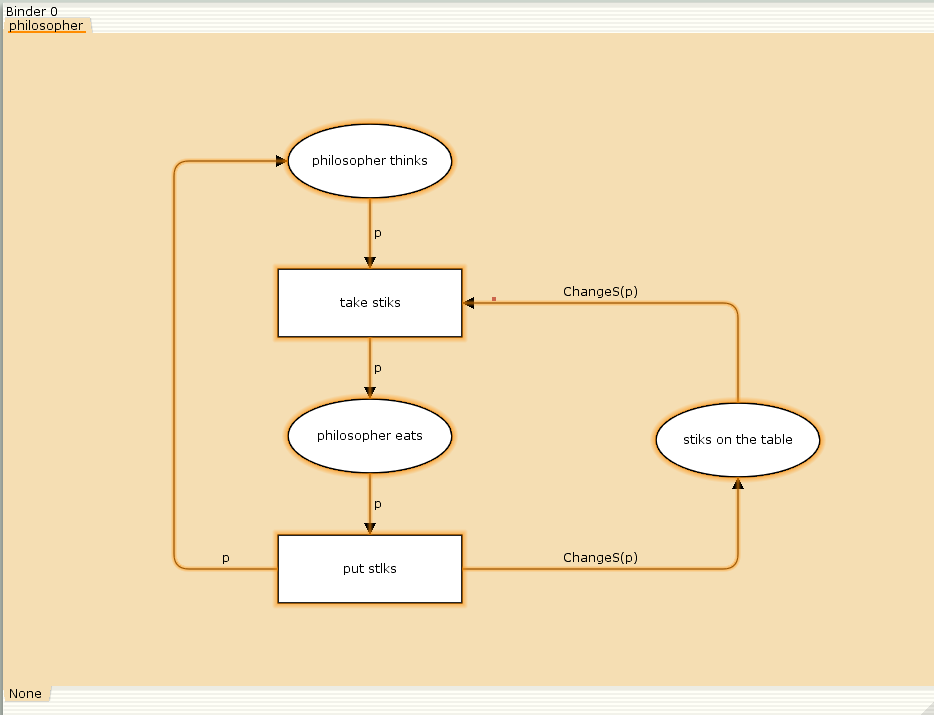


Figure 1: Граф сети задачи об обедающих мудрецах

В меню задаём новые декларации модели (рис. [2](#fig:002)): типы фишек, начальные значения позиций, выражения для дуг:

* — число мудрецов и палочек ;
* — фишки, обозначающие мудрецов, имеют перечисляемый тип PH от 1 до ;
* — фишки, обозначающие палочки, имеют перечисляемый тип ST от 1 до ;
* функция ChangeS(p) ставит в соответствие мудрецам палочки (возвращает номера палочек, используемых мудрецами); по условию задачи мудрецы сидят по кругу и мудрец может взять и палочки, поэтому функция ChangeS(p) определяется следующим образом:

fun ChangeS (ph(i))= 1`st(i)++st(if = n then 1 else i+1)

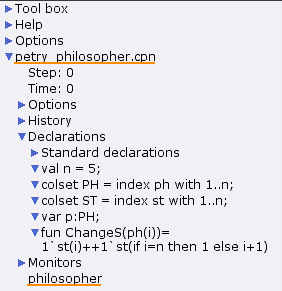


Figure 2: Задание деклараций задачи об обедающих мудрецах

В результате получаем работающую модель (рис. [3](#fig:003)).

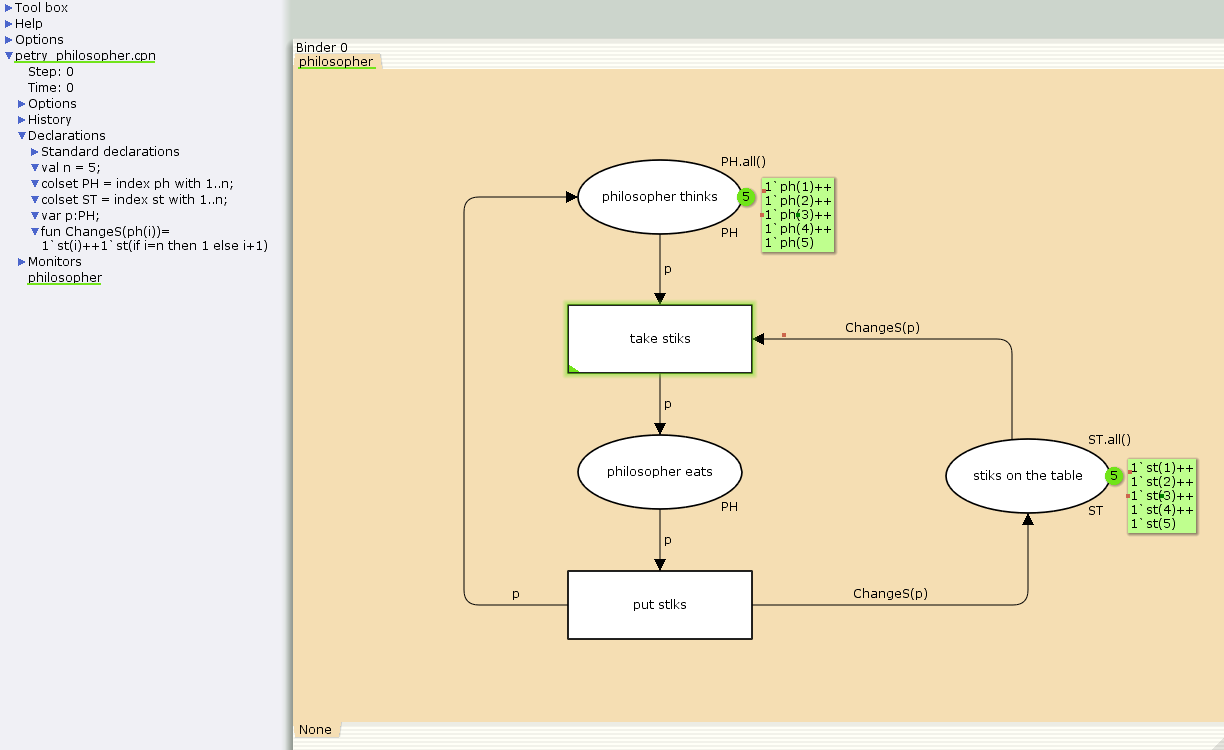


Figure 3: Модель задачи об обедающих мудрецах

После запуска модели наблюдаем, что одновременно палочками могут воспользоваться только два из пяти мудрецов (рис. [4](#fig:004)).

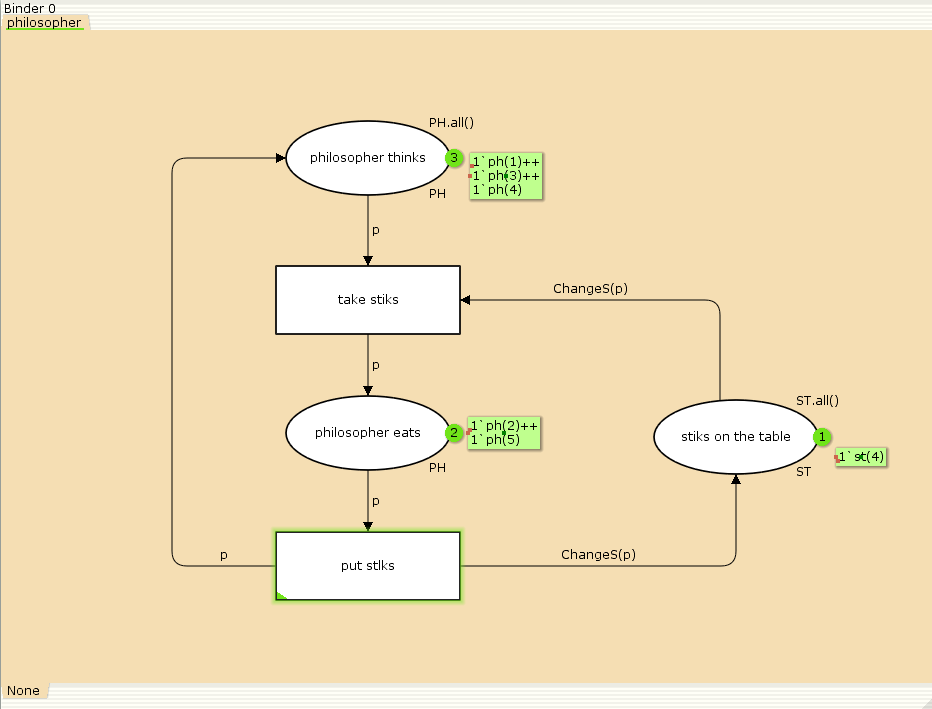


Figure 4: Запуск модели задачи об обедающих мудрецах

## 4.1 Упражнение

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.

**Анализ отчёта:**

1. Статистика (Statistics)

* Nodes: 11 — Модель имеет 11 уникальных состояний.
* Arcs: 30 — Возможны 30 переходов между состояниями.
* Status: Full — Полное исследование пространства состояний гарантирует корректность анализа.

Закономерность:  
- В каждом состоянии одновременно едят 2 мудреца, что подтверждается верхней границей philosopher\_eats: Upper=2.  
- Это возможно благодаря непересекающемуся выбору палочек (например, мудрецы 1 и 3 могут есть одновременно, так как используют палочки (1,2) и (3,4) соответственно).

1. Ограниченность ресурсов (Boundedness Properties)

* philosopher\_eats: Upper=2, Lower=0  
  Одновременно могут есть до 2 мудрецов. Минимум — 0 (все размышляют).
* philosopher\_thinks: Upper=5, Lower=3  
  Всегда 3–5 мудрецов размышляют, что исключает взаимоблокировку.
* stiks\_on\_the\_table: Upper=5, Lower=1  
  На столе остаётся 1–5 палочек, предотвращая полное исчерпание ресурсов.

Механизм синхронизации:  
- Мудрецы берут палочки только если обе свободны (функция ChangeS гарантирует атомарный захват).  
- Пример: Если мудрец 1 взял палочки (1,2), мудрец 2 не может взять (2,3), но мудрец 3 может взять (3,4).

1. Liveness-свойства (Отсутствие блокировок)

* Dead Markings: None — Нет тупиковых состояний.
* Live Transition Instances: All — Все переходы (take\_stiks, put\_stlks) активны.

Пример перехода:  
1. Мудрец 1 берёт палочки (1,2) -> переходит в philosopher\_eats.  
2. Мудрец 3 берёт палочки (3,4) -> переходит в philosopher\_eats.  
3. После завершения еды палочки возвращаются -> система переходит в новое состояние.  
4. Fairness-свойства (Справедливость)  
- Impartial для переходов take\_stiks и put\_stlks:  
Все мудрецы получают равный доступ к палочкам. Даже если два мудреца едят одновременно, остальные не остаются без ресурсов навсегда.

1. Home-свойства (Возврат в начальное состояние)

* All markings are home — Система может вернуться в состояние, где все мудрецы размышляют, а все палочки на столе.

**Итог:**

Модель корректно решает проблему:

1. Нет взаимоблокировок за счёт ограничения на одновременный захват палочек.
2. Два мудреца могут есть одновременно благодаря непересекающимся парам палочек.
3. Нет голодания благодаря fairness-гарантиям.

Граф состояний будет содержать циклы, где система переключается между комбинациями двух едящих мудрецов и трёх размышляющих. Пример цикла:  
(0 едят, 5 размышляют) -> (1 ест, 4 размышляют) -> (2 едят, 3 размышляют) -> (1 ест, 4 размышляют) -> (0 едят, 5 размышляют) и т.д.

Отчёт:

CPN Tools state space report for:  
/home/openmodelica/petry\_philosopher.cpn  
Report generated: Wed Apr 9 19:45:30 2025  
  
  
 Statistics  
------------------------------------------------------------------------  
  
 State Space  
 Nodes: 11  
 Arcs: 30  
 Secs: 0  
 Status: Full  
  
 Scc Graph  
 Nodes: 1  
 Arcs: 0  
 Secs: 0  
  
  
 Boundedness Properties  
------------------------------------------------------------------------  
  
 Best Integer Bounds  
 Upper Lower  
 philosopher'philosopher\_eats 1  
 2 0  
 philosopher'philosopher\_thinks 1  
 5 3  
 philosopher'stiks\_on\_the\_table 1  
 5 1  
  
 Best Upper Multi-set Bounds  
 philosopher'philosopher\_eats 1  
 1`ph(1)++  
1`ph(2)++  
1`ph(3)++  
1`ph(4)++  
1`ph(5)  
 philosopher'philosopher\_thinks 1  
 1`ph(1)++  
1`ph(2)++  
1`ph(3)++  
1`ph(4)++  
1`ph(5)  
 philosopher'stiks\_on\_the\_table 1  
 1`st(1)++  
1`st(2)++  
1`st(3)++  
1`st(4)++  
1`st(5)  
  
 Best Lower Multi-set Bounds  
 philosopher'philosopher\_eats 1  
 empty  
 philosopher'philosopher\_thinks 1  
 empty  
 philosopher'stiks\_on\_the\_table 1  
 empty  
  
  
 Home Properties  
------------------------------------------------------------------------  
  
 Home Markings  
 All  
  
  
 Liveness Properties  
------------------------------------------------------------------------  
  
 Dead Markings  
 None  
  
 Dead Transition Instances  
 None  
  
 Live Transition Instances  
 All  
  
  
 Fairness Properties  
------------------------------------------------------------------------  
 philosopher'put\_stlks 1  
 Impartial  
 philosopher'take\_stiks 1  
 Impartial

Построим граф пространства состояний (рис. [5](#fig:005)):

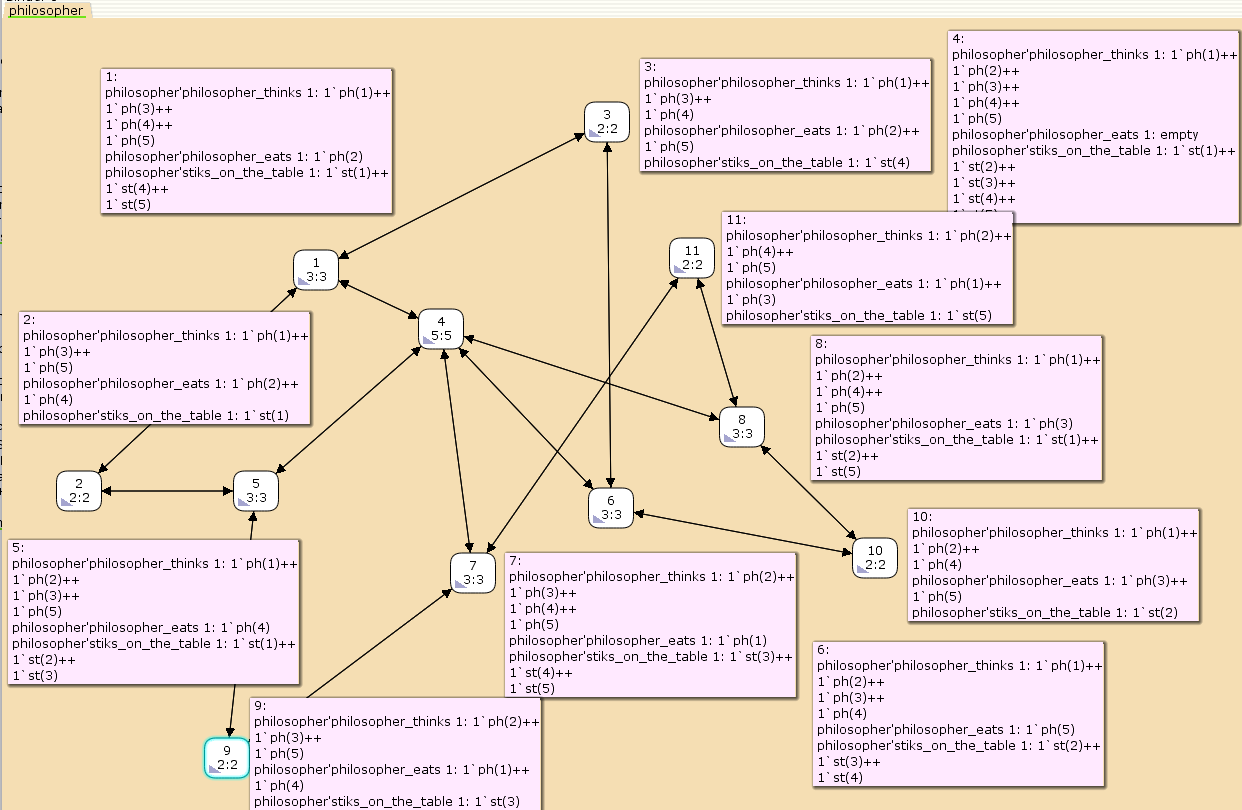


Figure 5: Пространство состояний для модели

На представленном изображении показан граф пространства состояний для задачи об обедающих мудрецах.

Общая структура пространства состояний: - Количество состояний: 11 узлов (пронумерованы от 1 до 11); - У нас всего 15 стрелок, но так как они двунаправленные, получается в итоге 30 переходов. Они представляют собой переходы между состояниями, вызванные срабатыванием переходов take\_stiks и put\_stiks.

# 5 Выводы

В ходе 10 лабораторной работы была реализована модель задачи об обедающих мудрецах в CPN Tools, вычислено пространство состояний, сформирован отчет о нем и построен граф.

# Список литературы

1. Зайцев Д.А. [Сети Петри и моделирование систем](https://f.eruditor.link/file/158457/). 2007.

2. Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. 1997.

3. [Задача об обедающих философах](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE%D0%B1_%D0%BE%D0%B1%D0%B5%D0%B4%D0%B0%D1%8E%D1%89%D0%B8%D1%85_%D1%84%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B0%D1%85).

4. [«Современные» обедающие философы на C++ посредством акторов и CSP](https://habr.com/ru/articles/437998/). 2020.

5. Алмакаев Д.Р. [Задача об обедающих мудрецах](http://lib.uvauga.ru/disk/2022/Almakaev_DR_OLR_VKR_ingener_ofo_2022.pdf). 2022.

6. Peterson G.L. Myths About the Mutual Exclusion Problem. 1981.

7. Королькова А.В., Кулябов Д.С. Руководство к лабораторной работе №10. Моделирование информационных процессов. Задача об обедающих мудрецах. 2025. С. 3.