

Лабораторная работа №13

Имитационное моделирование

Серёгина Ирина Андреевна

Содержание

1	Цель работы	3
2	Задание	4
3	Теоретическое введение	5
4	Выполнение лабораторной работы	7
5	Выводы	13

1 Цель работы

Выполнить задание для самостоятельного выполнения

2 Задание

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.

3 Теоретическое введение

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.

Описание модели

Множество позиций:

P1 — состояние оперативной памяти (свободна / занята);

P2 — состояние внешнего запоминающего устройства В1 (свободно / занято);

P3 — состояние внешнего запоминающего устройства В2 (свободно / занято);

P4 — работа на ОП и В1 закончена;

P5 — работа на ОП и В2 закончена;

P6 — 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

4 Выполнение лабораторной работы

Для теоретического анализа сети Петри строю дерево доступности (рис. 4.1).

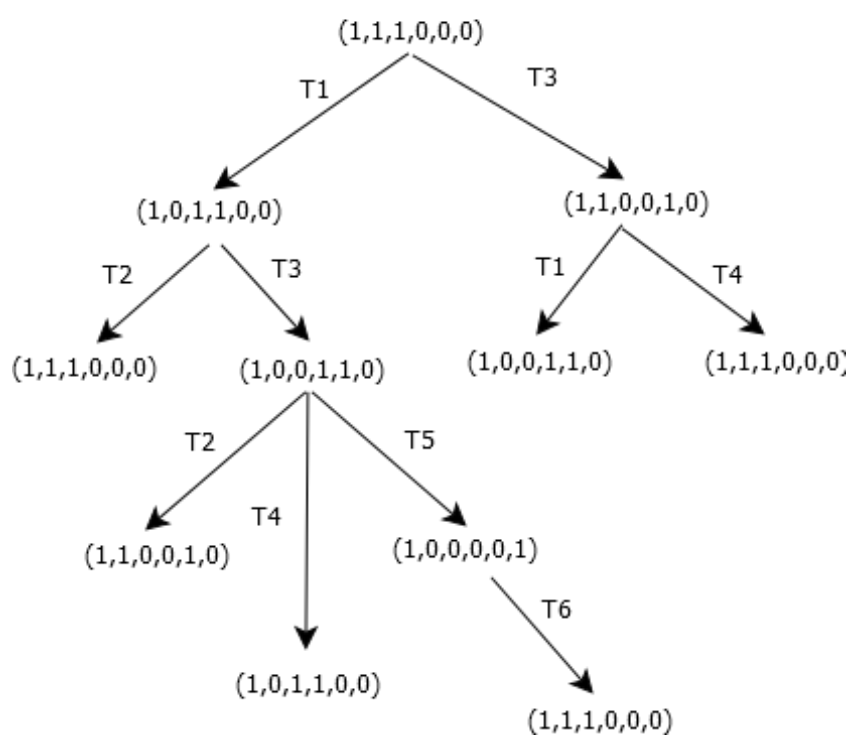


Рис. 4.1: Дерево доступности

Перед построением сети задаю декларации модели (рис. 4.2).

```

▼ Declarations
  ► memory
  ▼ colset B1 = unit with storage1;
  ▼ colset B2 = unit with storage2;
  ▼ colset RAM = unit with ramem;
  ▼ colset B1xB2 = product B1 * B2;
  ▼ var b1 : B1;
  ▼ var b2 : B2;
  ▼ var ram : RAM;
  ▼ val init_b1 = 1`storage1;
    val init_b2 = 1`storage2;
    val init_ram = 1`ramem;
  ► Standard declarations

```

Рис. 4.2: Декларации модели

После этого строю схему и задаю начальные состояния, подписываю стрелки (рис. 4.3).

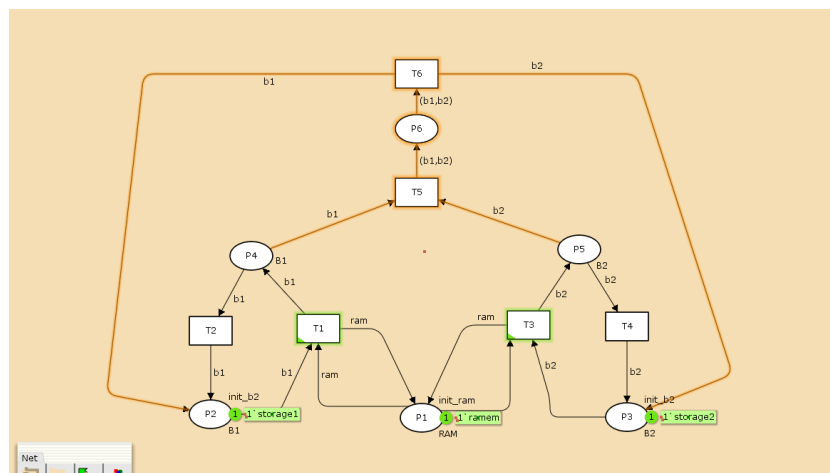


Рис. 4.3: Схема модели

Затем запускаю модель, все работает (рис. 4.4).

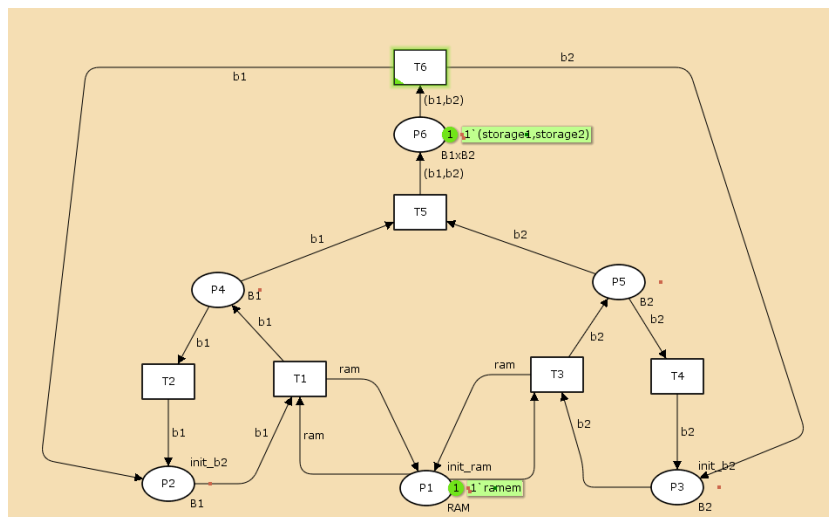


Рис. 4.4: Модель в действии

После того как я вычисляю пространство состояний, я делаю отчет о нем.

- есть 5 состояний и 10 переходов между ними
- 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: Sat May 3 16:42:33 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`ramem
New_Page' P2 1	1`storage1

New_Page'P3	1	1`storage2
New_Page'P4	1	1`storage1
New_Page'P5	1	1`storage2
New_Page'P6	1	1`(storage1,storage2)

Best Lower Multi-set Bounds

New_Page'P1	1	1`ramem
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

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

А также строю граф пространства состояний, он имеет пять вершин (рис. 4.5).

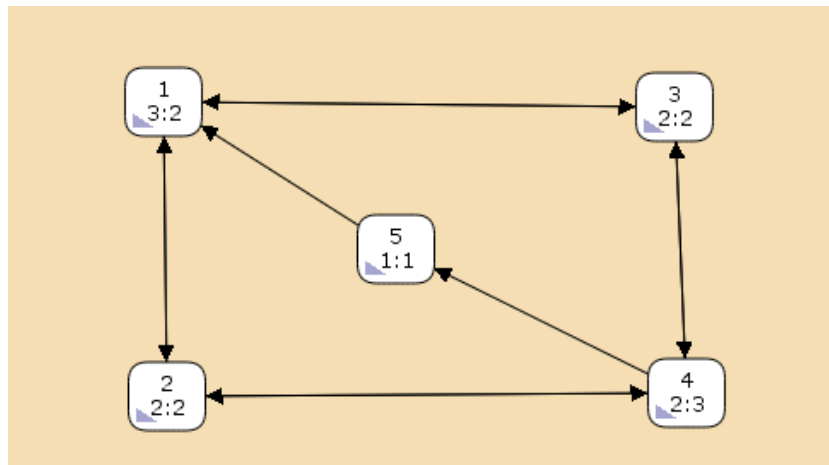


Рис. 4.5: Граф пространства состояний

5 Выводы

Я выполнила задание для самостоятельного выполнения.