

Лабораторная работа №13

Имитационное моделирование

Волгин Иван Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	13

Список иллюстраций

4.1	Сеть Петри	9
4.2	Декларации модели	10
4.3	Модель сети Петри	10
4.4	Итог работы модели	11
4.5	Отчет о пространстве состояний	11
4.6	Граф пространства состояний	12

Список таблиц

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 того, чтобы теоретически проанализировать сеть петри, строю дерево доступности (рис. 4.1).

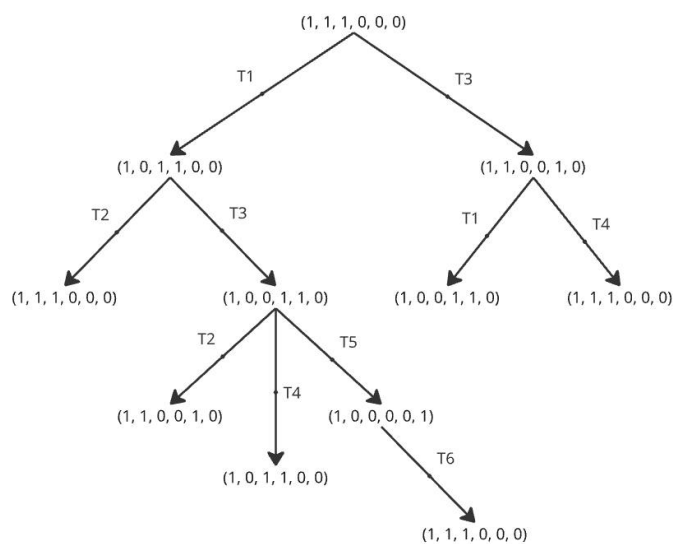


Рис. 4.1: Сеть Петри

Далее перехожу в программу `cpntools`, создаю новую модель и задаю для нее декларации (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;
    ▼ New Block
  ► Standard declarations

```

Рис. 4.2: Декларации модели

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

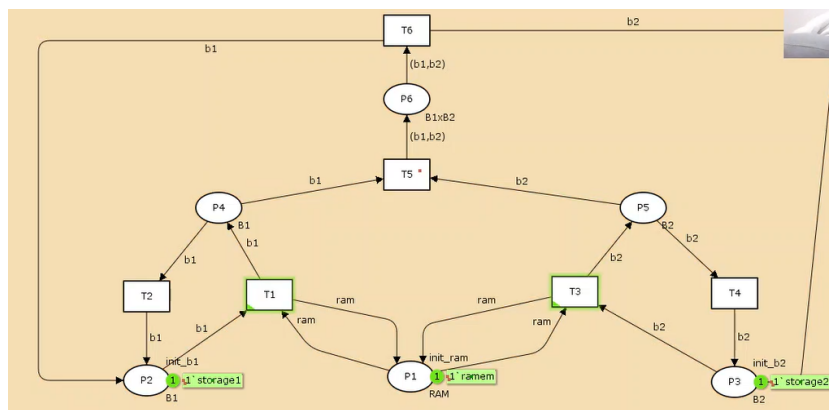


Рис. 4.3: Модель сети Петри

Далее запускаю модель, все корректно работает (рис. 4.4)

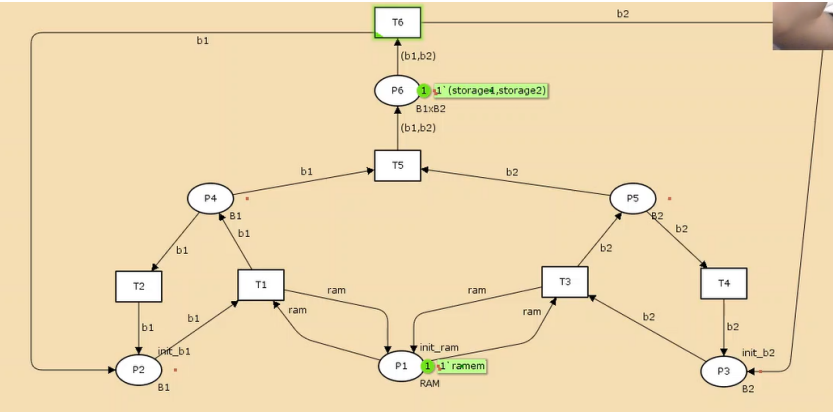


Рис. 4.4: Итог работы модели

После этого делаю отчет о вычисленном пространстве состояний (рис. 4.5)

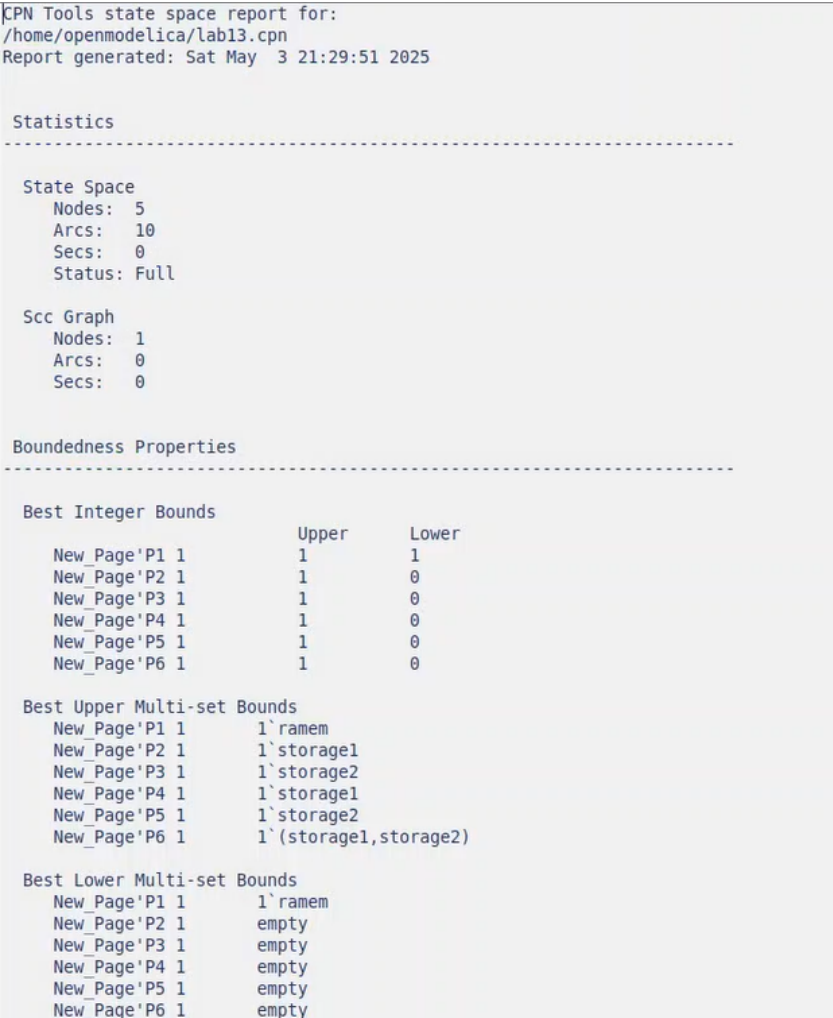


Рис. 4.5: Отчет о пространстве состояний

- 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.

Затем я построил граф пространства состояний, который состоит из 5 вершин и 10 переходов (рис. 4.6)

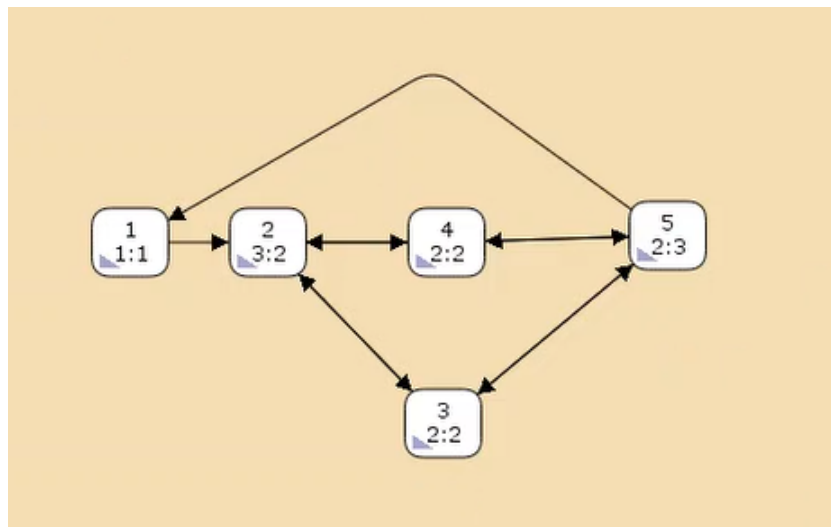


Рис. 4.6: Граф пространства состояний

5 Выводы

В ходе выполнения лабораторной работы я изучил сеть Петри и построил ее модель в cpntools.