

Лабораторная работа № 13

Задание для самостоятельного выполнения

Шуплецов Александр Андреевич

Содержание

1	Введение	4
1.1	Цели и задачи	4
2	Выполнение лабораторной работы	5
2.1	Реализация задачи в CPN Tools	5
3	Выводы	10
4	Список литературы	11

Список иллюстраций

2.1	Задание деклараций задачи	6
2.2	Модель задачи	7
2.3	Граф достижимости	7
2.4	Граф пространства состояний	8

1 Введение

1.1 Цели и задачи

Реализовать в CPN Tools задание для самостоятельного выполнения.

2 Выполнение лабораторной работы

2.1 Реализация задачи в CPN Tools

В меню заданы новые декларации модели: типы фишек, начальные значения позиций, выражения для дуг(рис. [2.1]). - ram – оперативная память типа RAM; - b1 – первое накопительное устройство типа B1; - b2 – второе накопительное устройство типа B2; - B1xB2 – тип для передачи b1 и b2; - Начальные состояния P1, P2, P3 содержат один элемент типа RAM, B1, B2 соответственно

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

Рис. 2.1: Задание деклараций задачи

Рисуем граф сети. Для этого с помощью контекстного меню создаём новую сеть, добавляем позиции, переходы и дуги, а также зададим типы данных и начальные состояния(рис. [2.2]):

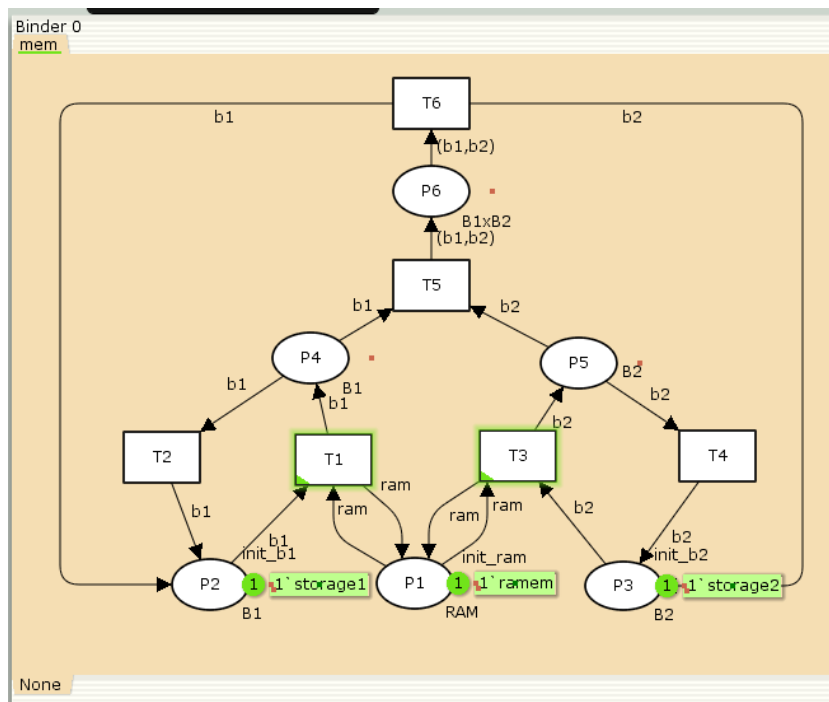


Рис. 2.2: Модель задачи

Построим граф достижимости(рис. [2.3]):

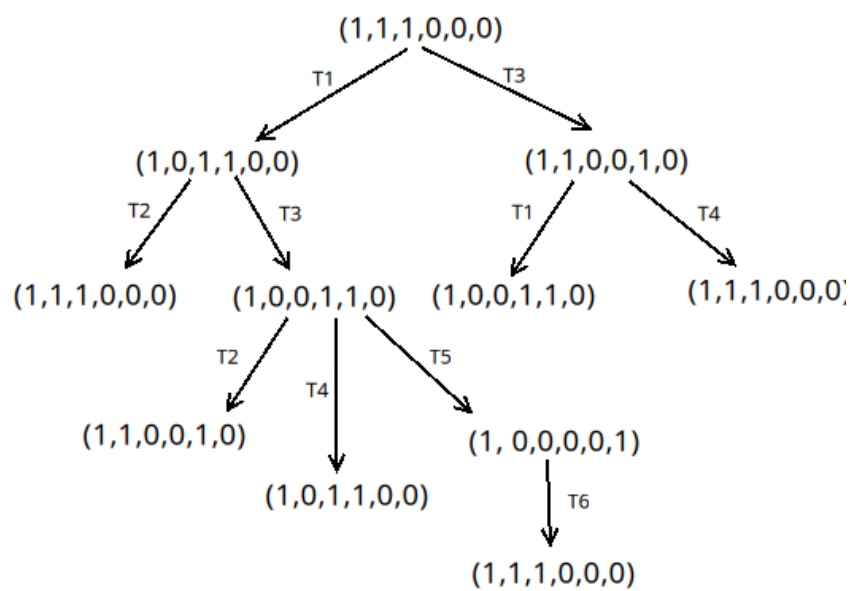


Рис. 2.3: Граф достижимости

Можно увидеть, что рассматриваемая сеть Петри: - безопасна, так как число фишек в каждой позиции не может превысить 1; - ограничена, так как существует такое целое k , что число фишек в каждой позиции не может превысить k (в нашем случае $k = 1$); - не имеет тупиков; - не является сохраняющей, так как при переходе T5 теряется 1 фишка, а при T6 – порождается;

Сформируем граф пространства состояний, всего их 5 (рис. [2.4]):

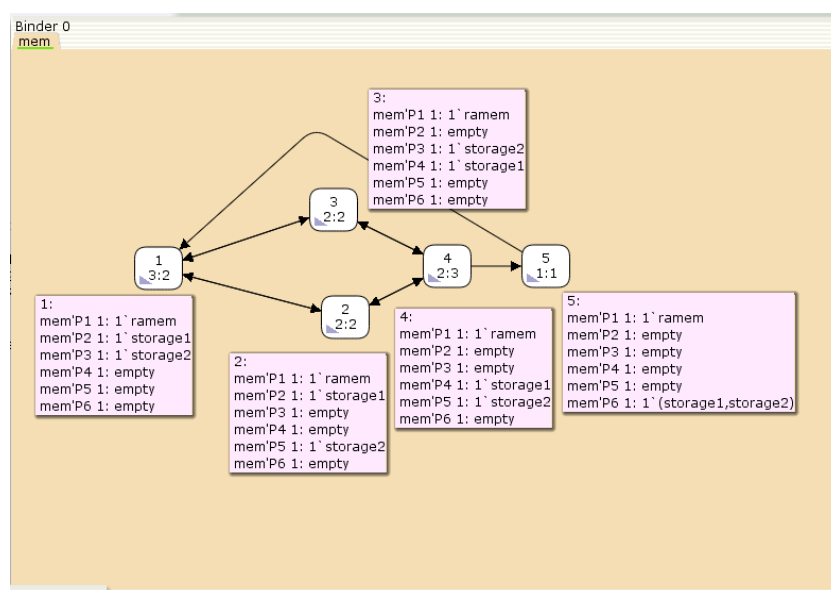


Рис. 2.4: Граф пространства состояний

Затем сформируем отчет пространства состояний. Из него может увидеть:

- есть 5 состояний и 10 переходов между ними, strongly connected components (SCC) graph содержит 1 вершину и 0 переходов, так как нет состояний, из которых можно попасть во все остальные.
- 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, то есть были бесконечные циклы, а состояние T6 происходит всегда, если доступно.

3 Выводы

В результате выполнения работы я выполнил самостоятельное задание: провел анализ сети Петри, построив ее с помощью CPNTools, и также я построил граф состояний и провел его анализ.

4 Список литературы

Королькова А.В., Кулябов Д.С. “Материалы к лабораторным работам”