

Отчёта по лабораторной работе №13

Задание для самостоятельного выполнения

Надежда Александровна Рогожина

Содержание

1	Задание	5
2	Выполнение лабораторной работы	6
3	Выводы	11
	Список литературы	12

Список иллюстраций

1.1	Схема модели к реализации	5
2.1	Схема сети	6
2.2	Дерево достижимости и анализ сети	7
2.3	Декларации	7
2.4	Сеть	8
2.5	Отчет	8
2.6	Отчет	9

Список таблиц

1 Задание

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.

Схема модели представлена на рис. 1.1.

Схема модели представлена на рис. 13.1.

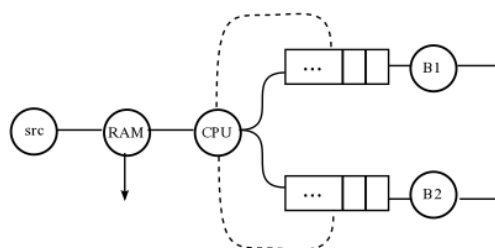


Рис. 1.1: Схема модели к реализации

2 Выполнение лабораторной работы

Нам была дана схема сети для выполнения домашнего задания (рис. 2.1).

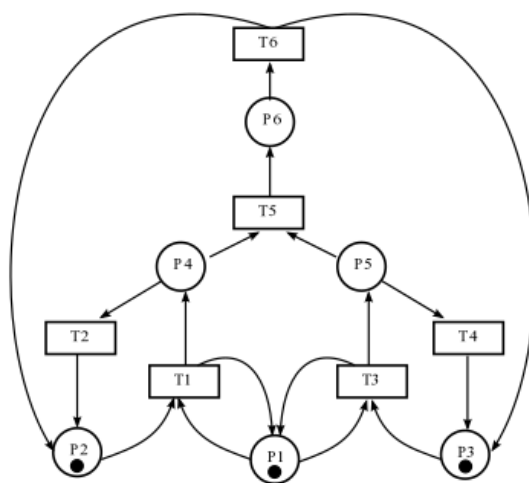


Рис. 13.2. Сеть для выполнения домашнего задания

Рис. 2.1: Схема сети

Первое что необходимо было сделать - проанализировать сеть с помощью дерева достижимости (рис. 2.2).

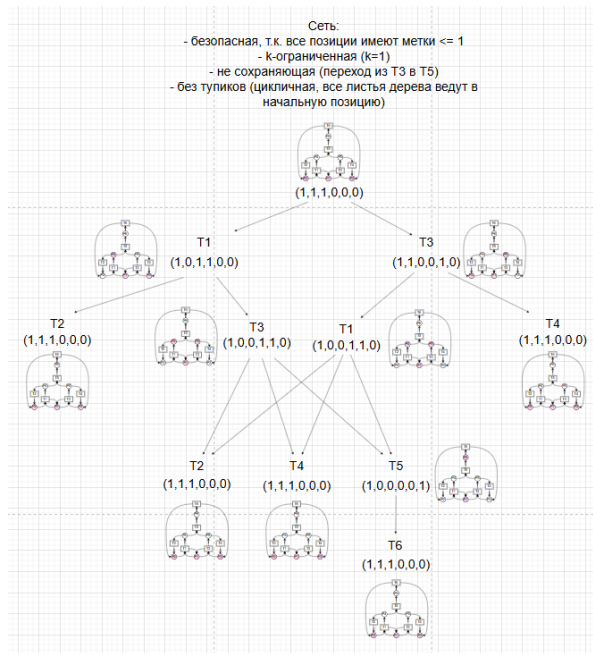


Рис. 2.2: Дерево достижимости и анализ сети

Следующим этапом необходимо было реализовать эту сеть в `SpnTools`. Для начала, объявим декларации (рис. 2.3).

```

▶ History
▼ Declarations
  ▼ Memory
    ▼ colset RAM = unit with memory;
    ▼ colset B1 = unit with storage1;
    ▼ colset B2 = unit with storage2;
    ▼ colset B1xB2 = product B1*B2;
    ▼ var ram:RAM;
    ▼ var b1:B1;
    ▼ var b2:B2;
  ▶ Standard declarations
▶ Monitors
  lab13

```

Рис. 2.3: Декларации

Также, построим саму сеть (рис. 2.4).

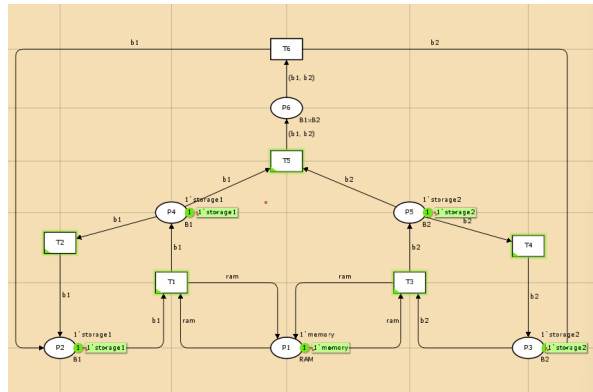


Рис. 2.4: Сеть

Войдя в пространство состояний и построив граф состояний, был сформирован отчет (рис. 2.5, рис. 2.6).

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
lab13'P1 1	1	1
lab13'P2 1	1	0
lab13'P3 1	1	0
lab13'P4 1	1	0
lab13'P5 1	1	0
lab13'P6 1	1	0
Best Upper Multi-set Bounds		
lab13'P1 1	1* memory	
lab13'P2 1	1* storage1	
lab13'P3 1	1* storage2	
lab13'P4 1	1* storage1	
lab13'P5 1	1* storage2	
lab13'P6 1	1* (storage1, storage2)	
Best Lower Multi-set Bounds		
lab13'P1 1	1* memory	
lab13'P2 1	empty	
lab13'P3 1	empty	
lab13'P4 1	empty	
lab13'P5 1	empty	
lab13'P6 1	empty	

Рис. 2.5: Отчет


```

Home Properties
-----
Home Markings
All

Liveness Properties
-----
Dead Markings
None

Dead Transition Instances
None

Live Transition Instances
All

Fairness Properties
-----
Impartial Transition Instances
None

Fair Transition Instances
lab13'T6 1

Just Transition Instances
lab13'T5 1

Transition Instances with No Fairness
lab13'T1 1
lab13'T2 1
lab13'T3 1
lab13'T4 1

```

Рис. 2.6: Отчет

В ходе анализа отчета было определено, что:

1. Система имеет 5 уникальных состояний (узлов).
2. Между состояниями существует 10 переходов (дуг).
3. P1 содержит 1 фишку типа memory.
4. P2 и P4 могут содержать storage1.
5. P3 и P5 могут содержать storage2.
6. Только P1 гарантированно содержит memory.
7. Остальные позиции могут быть пустыми.
8. Из любого состояния можно вернуться в любое другое (все они *домашние* -> всегда можно вернуться в начальное состояние).
9. Нет ни мертвых состояний, ни мертвых переходов, т.е. в любом состоянии рано или поздно переходы могут сработать -> модель “живая”.
10. Нет переходов, требующих беспристрастного выполнения.
11. Переход T6 должен срабатывать бесконечно часто, если он постоянно доступен.

12. Переход T_5 не может быть вечно заблокирован, если он бесконечно часто становится доступным.

3 Выводы

В ходе работы мы смоделировали поведение данной нам модели с помощью CpnTools.

Список литературы