

# **Лабораторная работа №12**

**Имитационное моделирование**

Волгин Иван Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>

# Список иллюстраций

4.1	Декларации модели . . . . .	7
4.2	Начальный граф . . . . .	8
4.3	Промежуточные состояния . . . . .	8
4.4	Дополнительные декларации . . . . .	9
4.5	модель простого протокола передачи данных . . . . .	9
4.6	Результат работы модели . . . . .	10
4.7	Отчет о пространстве состояний . . . . .	11
4.8	Граф пространства состояний . . . . .	12

# 1 Цель работы

Изучить пример моделирования простого протокола передачи данных и построить его самостоятельно опираясь на данный пример

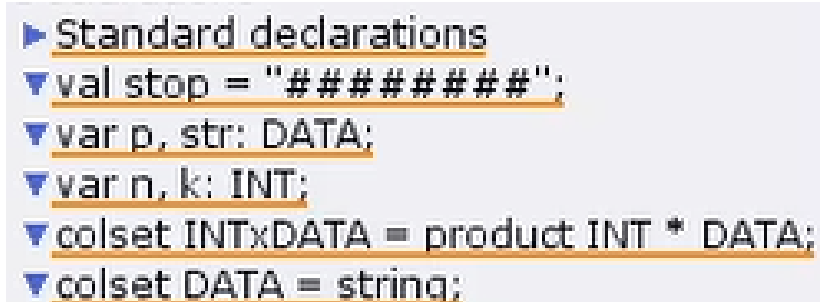
## 2 Задание

1. Изучить пример моделирования простого протокола передачи данных
2. Построить самостоятельно модель
3. Вычислить пространство состояний
4. Построить граф пространства состояний

### **3 Теоретическое введение**

## 4 Выполнение лабораторной работы

Для начала я изучил пример моделирования постого протокола передачи данных. Затем перешел к самостоятельному построению. Для начала нужно было задать декларации модели (рис. 4.1).

A screenshot of a code editor showing several lines of model declarations. The text is as follows:

```
► Standard declarations  
▼ val stop = "#####";  
▼ var p, str: DATA;  
▼ var n, k: INT;  
▼ colset INTxDATA = product INT * DATA;  
▼ colset DATA = string;
```

Рис. 4.1: Декларации модели

Вторым шагом я строю начальный граф и правильно настраиваю все значения и типы состояний (рис. 4.2).

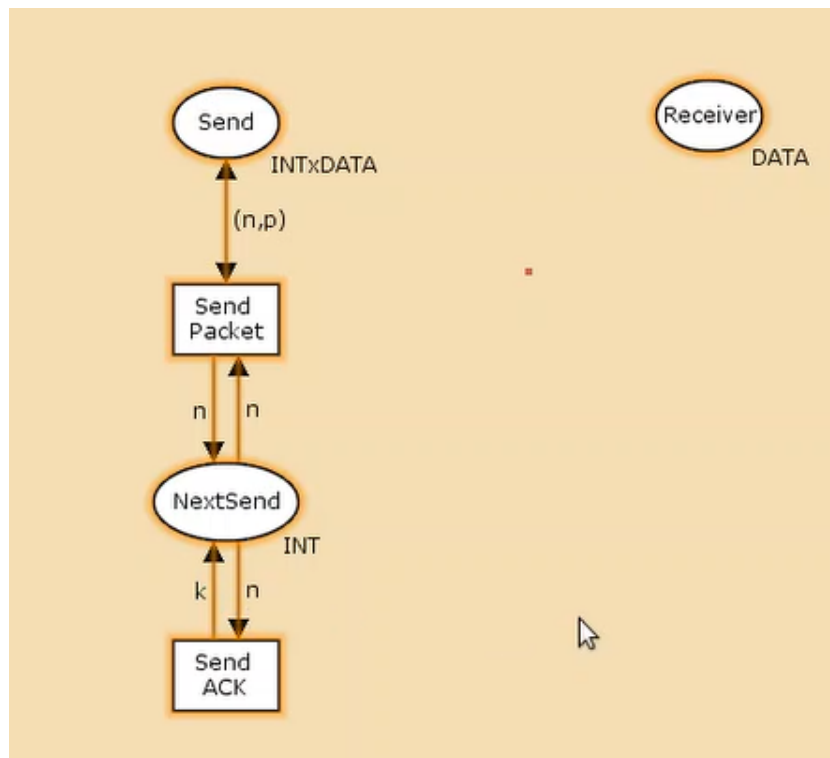


Рис. 4.2: Начальный граф

Далее к начальному графу добавляем промежуточные состояния (рис. 4.3)

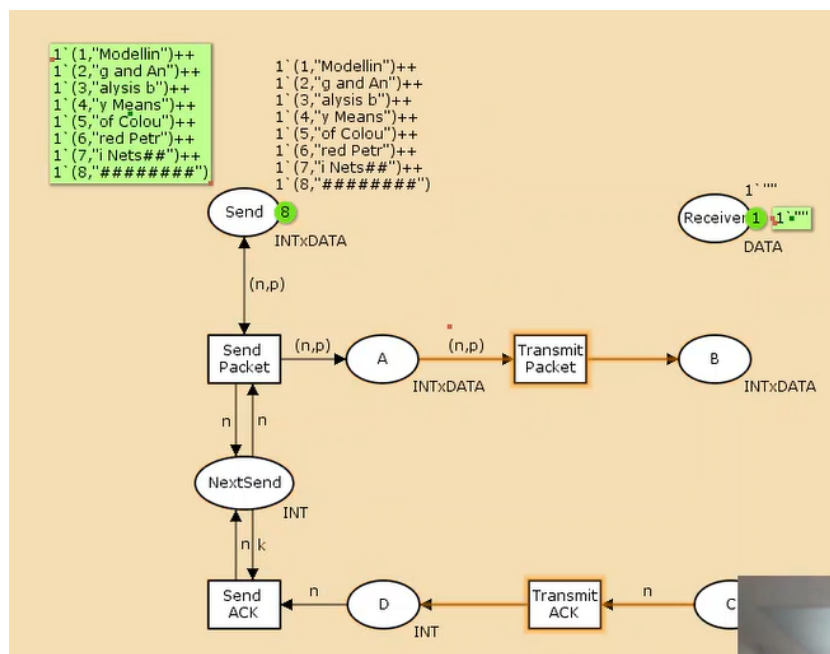


Рис. 4.3: Промежуточные состояния



Затем нужно было добавить дополнительные декларации (рис. 4.4)

```

▶ Standard declarations
▼ val stop = "#####";
▼ var n, k: INT;
▼ var p, str: DATA;
▼ colset INTxDATA = product INT * DATA;
▼ colset Ten0 = int with 0..10;
▼ colset Ten1 = int with 0..10;
▼ var s: Ten0;
▼ var r: Ten1;
▼ fun Ok(s:Ten0, r:Ten1)=(r<=s);

```

Рис. 4.4: Дополнительные декларации

Следующим шагом я заканчиваю строить граф и получаю модель простого протокола передачи данных (рис. 4.5)

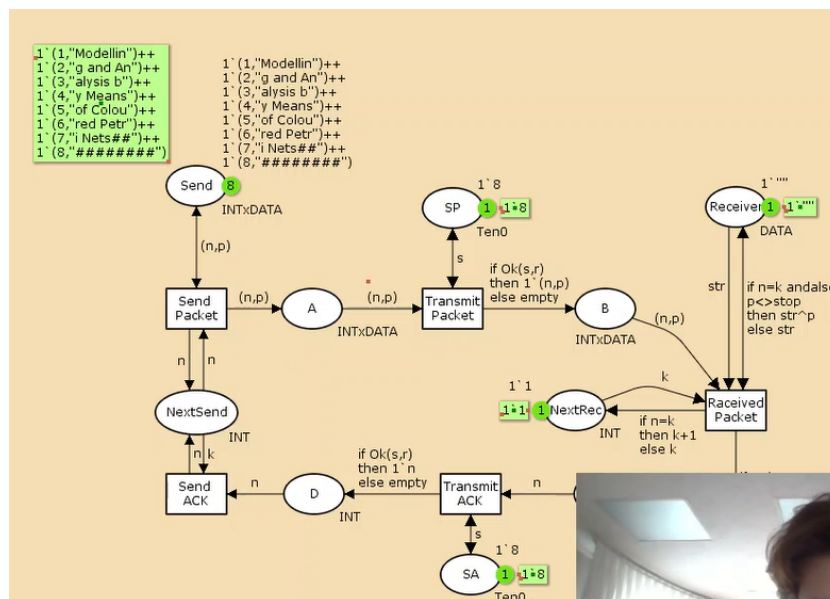


Рис. 4.5: модель простого протокола передачи данных

Чтобы удостовериться в правильности выполнения построения, я запустил модель и в результате получил полное сообщение, что свидетельствует об успешном построении (рис. 4.6)

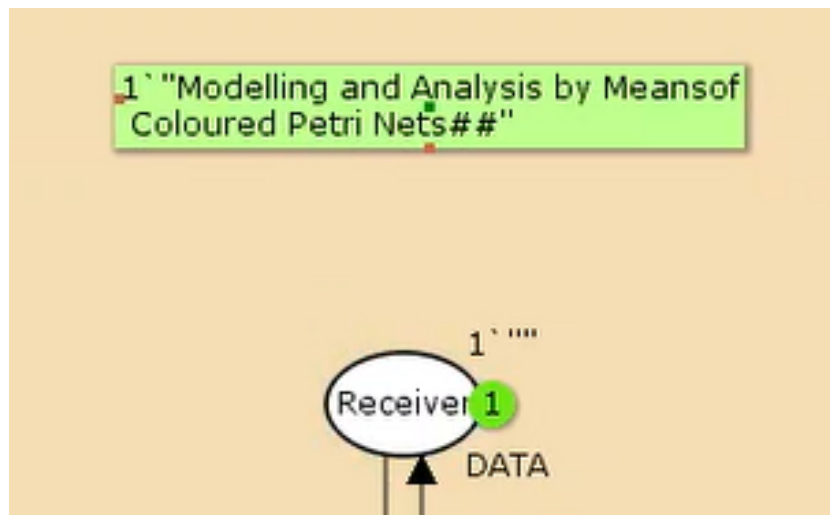


Рис. 4.6: Результат работы модели

Далее я перешел к выполнению упражнения, в котором нужно было вычислить пространство состояний, сформировать о нем отчет и построить граф пространства состояний. Сначала я сформировал отчет и проанализировал его (рис. 4.7). После этого я построил граф (рис. 4.8)

CPN Tools state space report for:  
/home/openmodelica/lab12.cpn  
Report generated: Sat Apr 26 15:58:26 2025

#### Statistics

##### State Space

Nodes: 14974  
Arcs: 285606  
Secs: 300  
Status: Partial

##### Scc Graph

Nodes: 11264  
Arcs: 257879  
Secs: 14

#### Boundedness Properties

##### Best Integer Bounds

	Upper	Lower
New_Page'A 1	14	0
New_Page'B 1	8	0
New_Page'C 1	10	0
New_Page'D 1	8	0
New_Page'NextRec 1	1	1
New_Page'NextSend 1	1	1
New_Page'Receiver 1	1	1
New_Page'SA 1	1	1
New_Page'SP 1	1	1
New_Page'Send 1	8	8

##### Best Upper Multi-set Bounds

New_Page'A 1	9` (7, "i Nets##")++
10` (8, "#####")	
New_Page'B 1	4` (7, "i Nets##")++
6` (8, "#####")	
New_Page'C 1	1` 8++
9` 9	
New_Page'D 1	1` 8++
8` 9	
New_Page'NextRec 1	1` 9
New_Page'NextSend 1	1` 7++

Рис. 4.7: Отчет о пространстве состояний

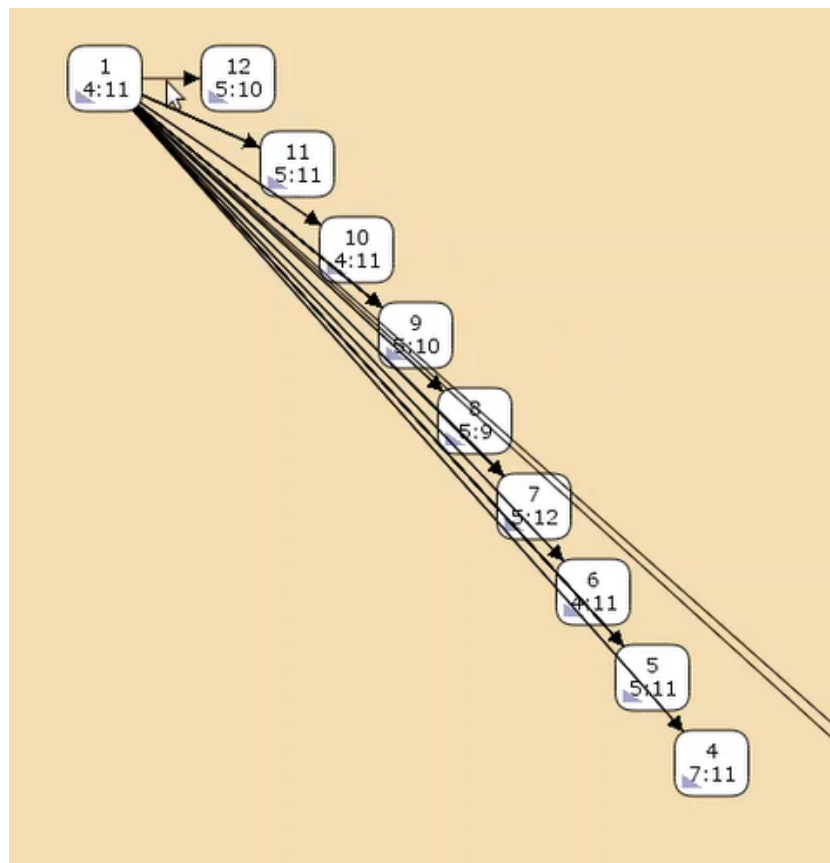


Рис. 4.8: Граф пространства состояний

## **5 Выводы**

В ходе выполнения данной лабораторной работы, я изучил простую модель протокола передачи данных и построил его модель, а также проанализировал отчет о пространстве состояний и построил его граф.