|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатика, искусственный интеллект и системы управления» |
|  |  |
| КАФЕДРА | «Программное обеспечение ЭВМ и информационные технологии» |

ОТЧЁТ

по лабораторной работе № 1  
по курсу «Математические основы верификации ПО»  
«Знакомство с языком Promela»

**Студент:** Керимов А. Ш.

**Группа:** ИУ7-42М

**Преподаватель:** Кузнецова О. В.

Москва.

2024 г.

# Задание

Для небольшого фрагмента программы необходимо описать модель этой программы на Promela и изучить её (SPIN).

# Фрагмент кода

#include <condition\_variable>

#include <cstddef>

#include <iostream>

#include <mutex>

#include <thread>

std::mutex m;

std::condition\_variable cv;

int name;

bool send = false;

void worker\_thread() {

for (int j = 0; j < 4; ++j) {

// wait until main() sends data

std::unique\_lock lk(m);

cv.wait(lk, []{ return send; });

// after the wait, we own the lock

std::cout << "receive: " << name << '\n';

// processed

send = false;

lk.unlock();

cv.notify\_one();

}

}

int main() {

std::thread worker(worker\_thread);

for (int i = 0; i < 4; ++i) {

// send data to the worker thread

name = i % 2 == 0 ? 4 : 1;

{

std::lock\_guard lk(m);

send = true;

}

cv.notify\_one();

// wait for the worker

{

std::unique\_lock lk(m);

cv.wait(lk, []{ return !send; });

}

}

worker.join();

}

# Описание модели

#define msgtype **1**

**chan** name = [**0**] **of** { **byte**, **byte** };

**active proctype** A() {

**int** i = **0**;

**do**

:: i < **2** ->

name ! msgtype(**4**);

name ! msgtype(**1**);

i++;

:: i == **2** ->

**break**;

**od**

}

**active proctype** B() {

**byte** state;

**int** j = **0**;

**do**

:: j < **4** ->

name ? msgtype(state);

**printf("receive: %d\n"**, state);

j++;

:: j == **4** ->

**break**;

**od**

}

# Перечисление множества состояний

* Процесс A:
  + A — начальное состояние,
  + S0 — терминальное состояние,
  + S2 — счётчик меньше 2,
  + S3 — состояние после посылки байта 4,
  + S4 — состояние после посылки байта 1,
  + S7 — проверка счётчика,
  + S10 — счётчик равен двум.
* Процесс B:
  + B — начальное состояние,
  + S0 — терминальное состояние,
  + S2 — счётчик меньше 4,
  + S3 — состояние после получение байта,
  + S7 — проверка счётчика,
  + S10 — счётчик равен 4.

# Граф переходов между состояниями модели

На рисунке 1 представлен граф переходов между состояниями процессов A и B.

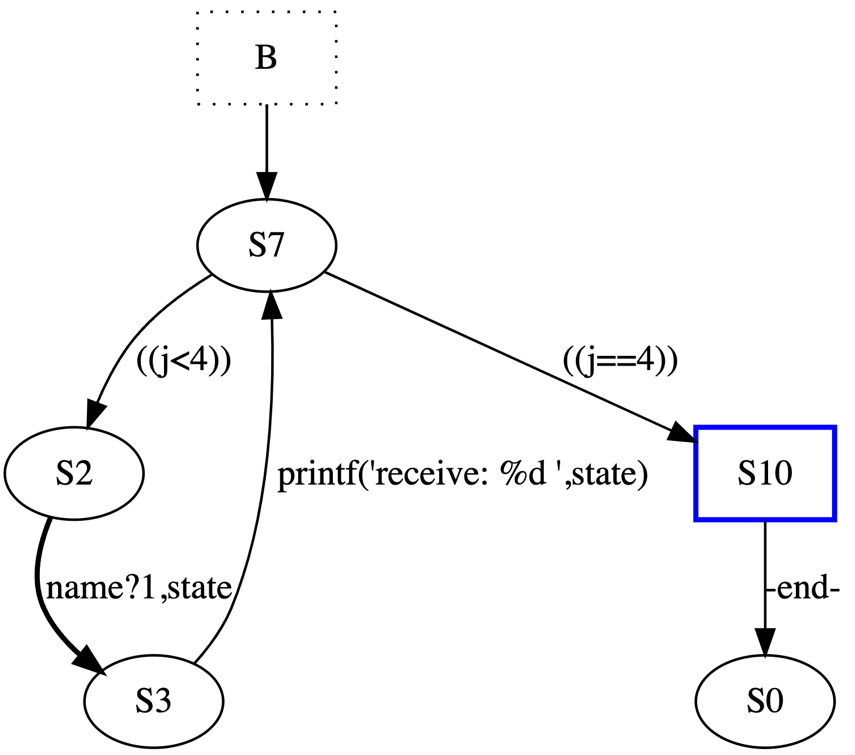
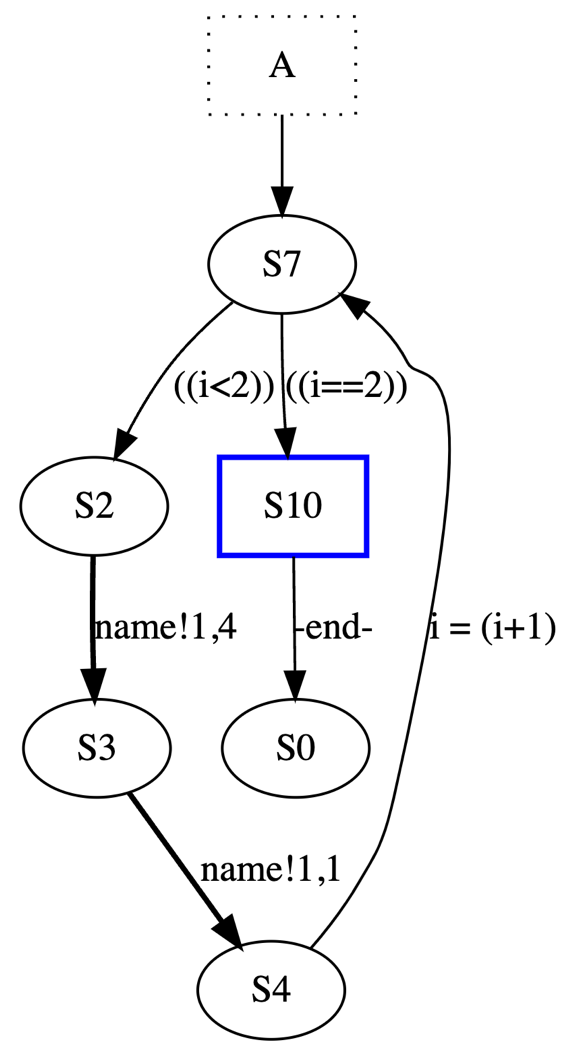


Рисунок 1 — Графы переходов между состояниями процессов A и B

# Выводы

В результате выполнения лабораторной работы № 1 были освоены:

* базовые возможности языка Promela, на котором была описана модель небольшого фрагмента программы;
* базовые возможности верификатора SPIN, с помощью которого были получены графы переходов между состояниями процессов программы, которые представляют собой детерминированные конечные автоматы с выделенными начальными и терминальными состояниями.