



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 <cstdint>
#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 А:

- А — начальное состояние,
- S0 — терминальное состояние,
- S2 — счётчик меньше 2,
- S3 — состояние после отправки байта 4,
- S4 — состояние после отправки байта 1,
- S7 — проверка счётчика,
- S10 — счётчик равен двум.

— Процесс В:

- В — начальное состояние,
- S0 — терминальное состояние,
- S2 — счётчик меньше 4,
- S3 — состояние после получения байта,
- S7 — проверка счётчика,
- S10 — счётчик равен 4.

## Граф переходов между состояниями модели

На рисунке 1 представлен граф переходов между состояниями процессов А и В.

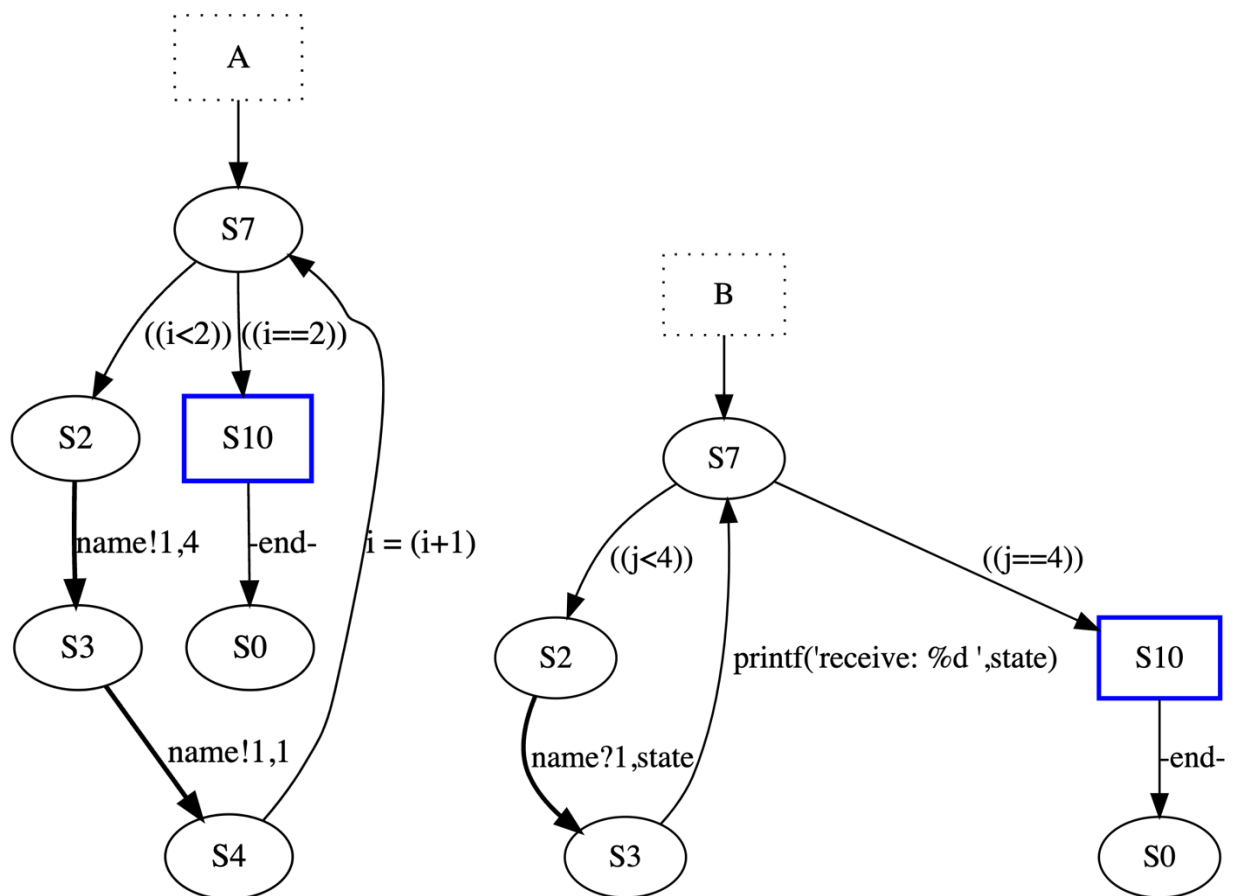


Рисунок 1 — Графы переходов между состояниями процессов А и В

## 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.