

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА

Разработка контроллера светофоров и его верификация

по дисциплине «Верификация распределенных алгоритмов и систем»

Выполнил
студент гр. *****

***** *. *

Руководитель

<подпись>

Шошмина И. В.

Санкт-Петербург
2025

Оглавление

Введение	3
Постановка задачи	4
Решение задачи	6
Заключение	8
Список литературы	9

Введение

Данная курсовая работа выполнена на средний уровень сложности. В ней требовалось разработать контроллер светофоров сложного перекрестка [1], далее провести его верификацию по определенным свойствам с помощью Spin [2].

Моделирование контроллера светофоров заключалось в создании алгоритма контроллера, а также окружения для дальнейшей верификации. В качестве окружения требовалось создать симуляцию траффика на дорогах сложного перекрестка. Траффик на каждом направлении должен появляться в случайные моменты времени. Контроллер, в свою очередь, должен реагировать на появление траффика на направлениях своего перекрестка, разрешая проезд машинам, путем переключения цветов светофоров.

Постановка задачи

Направления	Номера вариантов			
	34	30	17	8
		NS	SW	NE
	DN	DE	ES	

Таблица 1 – Вариант курсовой работы и его интерпретация.

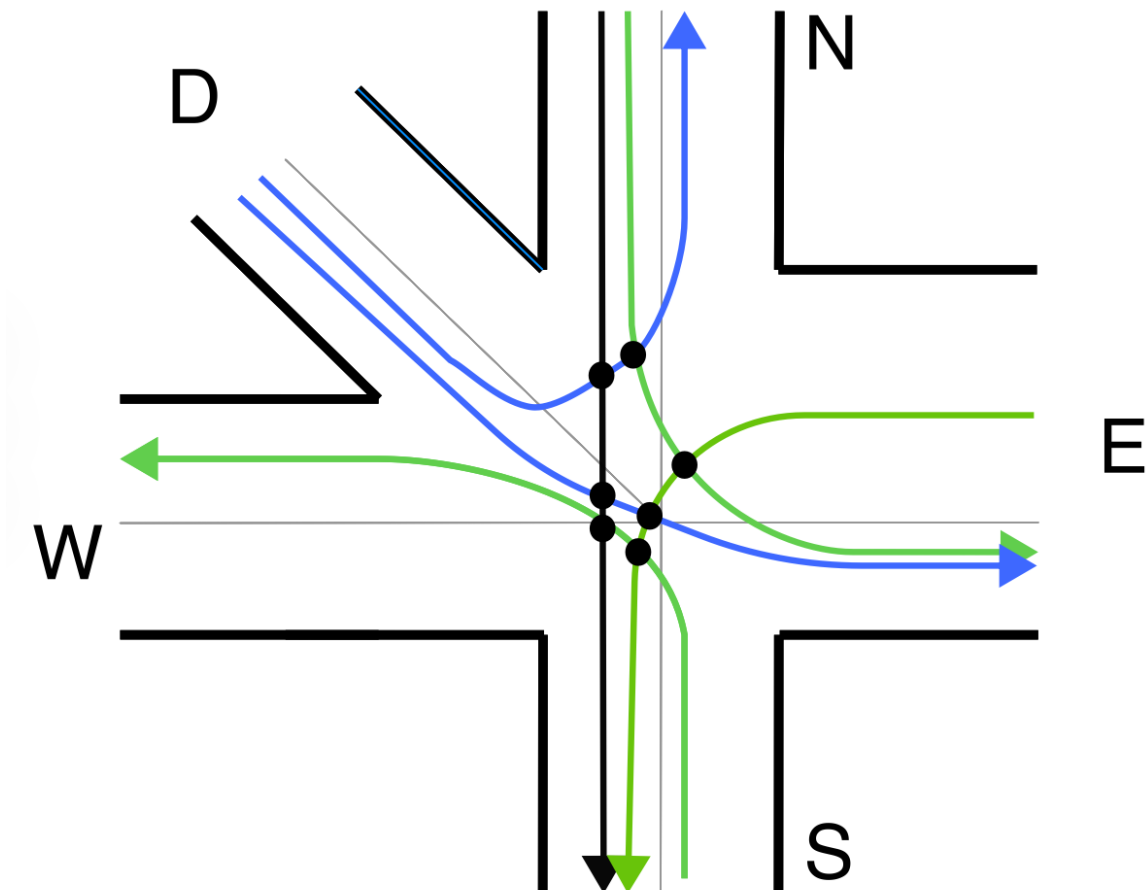


Рисунок 1 – Графическая интерпретация направления согласно варианту.

Согласно данному варианту, на сложном перекрестке присутствуют 6 направлений движения трафика, которые образуют 7 пересечений между собой.

Контроллер, управляющий светофорами, должен соответствовать следующим свойствам:

- Безопасность движения: прежде чем разрешить проезд по какому-либо направлению, контроллер должен убедиться в том, что пересекающие рассматриваемый маршрут направления не используются;
- Живость движения: несмотря на необходимость соблюдения безопасности, контроллер должен обязательно выдавать разрешение на проезд по направлениям, трафик на которых присутствует, а цвет светофора красный;

- Честность движения: если трафик по направлению отсутствует, то светофор должен быть красного цвета, чтобы впустую не занимать направление.

Согласно вышеперечисленным свойствам, требуется провести верификацию разработанного контроллера перекрестка.

Решение задачи

Контроллер, определяющий очередность решения, по принципу своей работы повторяет модель Round Robin [3], то есть он по кругу опрашивает направления на наличие траффика, и если он присутствует, то проверяет, будет ли движение по ним безопасным.

Пересечения направлений заданы с помощью матрицы. Данный подход позволяет проводить удобно перебор пересечений в ходе опроса контроллера. Матрица для моего варианта выглядит следующим образом:

	DN	NS	DE	SW	ES	NE
DN	0	1	0	0	0	1
NS	1	0	1	1	0	0
DE	0	1	0	0	1	0
SW	0	1	0	0	1	0
ES	0	0	1	1	0	1
NE	1	0	0	0	1	0

Таблица 2 – Матрица пересечений.

Для каждого направления определены два процесса – контроллер и сенсор [4]. Сенсор отвечает за случайность появления траффика по заданному направлению. Он получает от контроллера запрос на проверку наличия траффика и далее отвечает либо траффик есть, либо он отсутствует. Контроллер, в свою очередь, опрашивает сенсор только в случае, если цвет светофора красный. Если траффик отсутствует, то он не предпринимает никаких действий, если же траффик есть, то он посылает по каналу сообщение на контроллер всего движения, далее именуется Round Robin, который должен выдать разрешение. Контроллер ожидает этого разрешения, после получения которого меняет цвет на зеленый. После этого цвет меняется снова на красный, определяя, что траффик прошел.

Round Robin работает отдельным процессом, работающим бесконечным циклом, перебирающим различные направления. В каждой итерации данного цикла он сначала опрашивает все каналы на наличие сообщения. Далее те направления, которые запросили проезд, заносятся в список на ожидание разрешения. После этого Round Robin проходит по данному списку, проверяя

по матрице пересечений, есть ли среди пересекающих трасс те, которые заняты траффиком. Если таковых не имеется, то выдается разрешение на проезд. Round Robin, опрашивая контроллеры, инициализирует проверку траффика, потому контроллеры и сенсоры находятся в ожидании опроса от RR.

Также каждое направление является объектом структуры, которая в свою очередь состоит из 4-х полей типа `chan` на языке программирования Promela. Данные поля являются каналами [5], с помощью которых происходит синхронный обмен сообщениями между контроллером, сенсором и светофорами.

При старте программы инициализируется матрица пересечений, создаются объекты направлений с каналами, а также запускаются процессы Round Robin'a, сенсоры и контроллеры.

Заключение

В ходе верификации корректно отработали все свойства.

Также в ходе работы изучил материалы по синтаксису Promela, изучил различные подходы решения к данной задаче, а также изучил принцип работы Round Robin'a.

В ходе работы столкнулся со сложностями реализации самого цикла Round Robin'a, так как были проблемы с разработкой подхода к перебору пересечений. В ходе решения данной проблемы пришел к решению с помощью матрицы пересечений, которая позволяет сопоставлять сами пересечения направлений с их занятостью другими путями.

Список литературы

1. Карпов Ю. Г. «Верификация распределенный систем». СПб: Издательство Политехнического университета, 2011 г. 212 стр.
2. SpinRoot: Round Robin Scheduling in Operating System. Электронный ресурс <https://spinroot.com/spin/Man/Quick.html>. Дата обращения 06.12.25
3. GeeksForGeeks: Round Robin Scheduling in Operating System. Электронный ресурс <https://www.geeksforgeeks.org/operating-systems/round-robin-scheduling-in-operating-system>. Дата обращения 05.12.25
4. Karlsruhe Institute of Technology: Formal Specification and Verification. Introduction to Promela. Электронный ресурс <https://formal.kastel.kit.edu/~beckert/teaching/Formale-Verifikation-SS09/11Promela.pdf>. Дата обращения 09.12.25
5. ВМК МГУ: Математические методы верификации схем и программ. Электронный ресурс https://mk.cs.msu.ru/images/archive/c/cf/20171102195318%21Seminar_Verification_7.pdf. Дата обращения 08.12.25