КУРСОВАЯ РАБОТА

Разработка контроллера светофоров на перекрестке и его верификация

по дисциплине «Верификация распределенных алгоритмов и систем»

Вариант 2, 8, 10

Выполнила

студентка гр. 3540904/10102 Губенко Н. О.

Руководитель Шошмина И. В.

**Оглавление**

[Введение 3](#_Toc121919179)

[Постановка задачи 4](#_Toc121919180)

[Переменные в модели 5](#_Toc121919181)

[Процессы в модели 5](#_Toc121919182)

[Описание проезда по перекрестку 5](#_Toc121919183)

[Верификация модели 6](#_Toc121919184)

[Заключение 8](#_Toc121919185)

[Список литературы 9](#_Toc121919186)

Введение

Существуют задачи и ситуации, когда критично важно проверить правильность работы задачи или системы. Model Checking, метод проверки формулы темпоральной логики в рамках задачи или системы, является одним из подходов, применяемых в таких задачах. [1]

В курсовой работе ставится задача применить данный подход при разработке системы с помощью инструмента верификации Spin. [2]

Поскольку опыт проверки моделей ранее осуществлялся мной с помощью инструмента верификации NuSMV [5], то именно этот инструмент сравню с инструментом Spin. Spin предоставляет высокоуровневый язык моделирования Promela с C-подобным синтаксисом, который упрощает построение и чтение моделей [6]. NuSMV предоставляет относительно более простой язык для описания систем перехода состояний. С другой стороны, SPIN поддерживает только LTL для описания свойств, в то время как NuSMV также может проверять свойства LTL и свойства CTL [6]. В рамках данной курсовой работы достаточным является применение LTL свойств, а применение языка высокого уровня облегчит процесс моделирования, поэтому в данной работе преимущество за инструментом Spin.

Описание принципов работы Spin и синтаксис Promela можно найти в официальной документации [3] и [4].

Постановка задачи

Есть сложный перекресток с четырьмя двусторонними направлениями движения. На рис 1. выделены направления, указанные для варианта 2, 8, 10. Множество пересечений для варианта:

WN, NE,

SN, NE,

SN, EW.

Уникальных маршрутов четыре –EW, SN, WN, NE

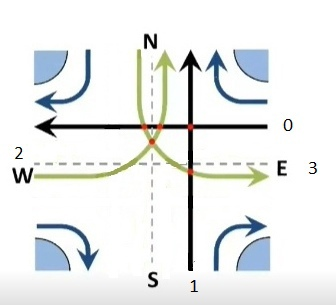


Рисунок Схема перекрестка с учетом варианта

На рисунке цифрами отмечены номера дорог. Каждое направление контролируется своим светофором.

В курсовой работе ставится задача разработки контролирующей системы, реализующей управление светофоров и управление проездов на перекрестке. Каждое направление движения автомобиля на перекрестке регулируется своим светофором. При появлении машины система должна определить, может ли она пропустить машину, а машина безопасно проехать. У каждого светофора есть датчик движения, который посылает сообщение о поступлении машины. В отсутствие машин на направлении светофор горит красным. При наличии машин на разных полосах светофоры периодически переключаются, обеспечивая равномерный проход машин на дорогах.

Переменные в модели

mtype = {CAR}; /\* автомобиль \*/

mtype = {GREEN, RED}; /\* тип сигнала светофора (разрешающий и запрещающий)\*/

chan gen\_cars[NUMBER\_OF\_TRAFFIC\_LIGHTS] = [1] of {mtype}; /\*генерация машин\*/

mtype traffic\_lights[NUMBER\_OF\_TRAFFIC\_LIGHTS] = {RED, RED, RED, RED}; /\*светофоры, с начальным запрещающим значением для каждой дороги\*/

mtype = {GRANTED};

chan cars\_queue = [NUMBER\_OF\_TRAFFIC\_LIGHTS] of {byte}; /\* очереди машин на каждой из дорог\*/

chan cars\_ACK[NUMBER\_OF\_TRAFFIC\_LIGHTS] = [0] of {mtype}; /\*сенсор светофора, предупреждающий о появлении машин\*/

byte semaphore[NUMBER\_OF\_TRAFFIC\_LIGHTS]; /\*блокировка пересечения \*/

Процессы в модели

GenCarsEW() - генератор машин на дороге EW

GenCarsSN() - генератор машин на дороге SN

GenCarsWN() - генератор машин на дороге WN

GenCarsNE() - генератор машин на дороге NE

Intersection() - процесс-семафор для доступа к критической секции

TrafficLightEW() - светофор для направления EW

TrafficLightSN() - светофор для направления SN

TrafficLightWN() - светофор для направления WN

TrafficLightNE() - светофор для направления NE

Описание проезда по перекрестку

В модели реализованы параллельные процессы трёх типов: генераторы потока машин (GenCars), контроллеры светофоров (TrafficLight) и контроллер пересечений (Intersection).

Генерация машин по направлениям происходит независимо от контроллеров. Машина считается подъезжающей к светофору, когда по синхронному каналу gen\_cars с соответствующим идентификатором отправляется сообщение. (при симуляции кода если установлен параметр CAR, то значит ничего не сгенерировано, если на этом месте пустое значение, то машина сгенерировалась)

Контроллер пересечений генерирует сразу весь перекресток, и в зависимости от текущей дороги, с помощью семафора (параметр semaphore) блокирует дороги, на которых может произойти авария. Семафор позволяет только одному процессу находиться в критической секции. У каждой дороги свой контроллер. Семафор пишет значение 0 в канал, один из процессов считывает это сообщение и получает разрешение на захват критической секции. Захват дороги происходит в порядке установления параметра 0 для параметра semaphore и блокирует «ненужные» дороги, установив 1 у параметров semaphore с соответствующим идентификатором дорог.

Контроллеры светофоров пытаются захватить ресурсы в случае, если сработал датчик обнаружения автомобиля. Для захвата используются флаги семафоров. Если захват был произведён, то на данном светофоре устанавливается зелёный свет. Захват ресурсов происходит по очереди. Светофор захватывает дороги в порядке установления параметра 0 для параметра semaphore и блокирует «ненужные» дороги, установив 1 у параметров semaphore с соответствующим идентификатором дорог (изначально в процессе пересечения). После захвата необходимых ресурсов светофор переключается в зелёный, освобождает очередь, устанавливает флаг для сенсора светофора (параметр cars\_ACK) как достигнутый, пропускает машины и снова загорается красным, при этом параметры semaphore «ненужных» дорог снова уменьшаются на 1 (обнуляются).

Верификация модели

Для верификации модели проверяется 2 свойства: безопасности и живости.

1. Безопасность – никогда не будет разрешен проезд в пересекающихся направлениях.

ltl safety {

[]!(traffic\_lights[1] == GREEN && traffic\_lights[3] == GREEN) &&

[]!(traffic\_lights[0] == GREEN && traffic\_lights[1] == GREEN) &&

[]!(traffic\_lights[0] == GREEN && traffic\_lights[2] == GREEN) &&

[]!(traffic\_lights[0] == GREEN && traffic\_lights[3] == GREEN) &&

[]!(traffic\_lights[2] == GREEN && traffic\_lights[3] == GREEN)

}

1. Живость – при появлении машины на светофоре он когда-нибудь загорится зелёным.

ltl liveness0 { [] ((gen\_cars[0]) == CAR && traffic\_lights[0] == RED)-> <> (traffic\_lights[0] == GREEN) }

ltl liveness1 { [] ((gen\_cars[1]) == CAR && traffic\_lights[1] == RED)-> <> (traffic\_lights[1] == GREEN) }

ltl liveness2 { [] ((gen\_cars[2]) == CAR && traffic\_lights[2] == RED)-> <> (traffic\_lights[2] == GREEN) }

ltl liveness3 { [] ((gen\_cars[3]) == CAR && traffic\_lights[3] == RED)-> <> (traffic\_lights[3] == GREEN) }

1. Справедливость – ни в каком направлении не движется непрерывный поток машин.

ltl fairness {

[] <> !(traffic\_lights[0] == GREEN && (gen\_cars[0]) == CAR) &&

[] <> !(traffic\_lights[1] == GREEN && (gen\_cars[1]) == CAR) &&

[] <> !(traffic\_lights[2] == GREEN && (gen\_cars[2]) == CAR) &&

[] <> !(traffic\_lights[3] == GREEN && (gen\_cars[3]) == CAR)

}

Заключение

В результате данной работы мной были получены практические знания в верификации распределённых параллельных систем, построена и верифицирована модель системы работы перекрёстка дорог, получены навыки работы с инструментом Spin. Однако, данная модель нуждается в некоторой доработке, которая подразумевает улучшение и увеличения количества рассматриваемых ситуаций логики работы пересечений на дорогах, представленных вариантами 2, 8, 10.

Список литературы

1. Карпов Юрий Глебович Темпоральные логики для спецификации свойств программных и аппаратных систем // КИО. 2009. №2.
2. Документация Spin. Электронный ресурс. URL: <http://spinroot.com/spin/whatispin.html> [03.12.2022]
3. И.В. Шошмина, Ю.Г. Карпов, Введение в язык Promela и систему комплексной верификации Spin
4. Ю.Г. Карпов, И.В. Шошмина. Верификация распределённых систем.
5. <https://github.com/uncttao/NuSMV>
6. Jiang Yong and Qiu Zongyan Model Transformation from SPIN to NuSMV (Tool Paper)