Санкт-Петербургский Государственный Политехнический Университет

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Самостоятельная работа №1**

1. по дисциплине «Распределенные алгоритмы»

|  |  |
| --- | --- |
| Выполнила  студент гр. 3530202/90202 | Изображение выглядит как текст  Автоматически созданное описание  Потапова А.М. |
|  |  |
| Преподаватель | Шошмина И. В. |
|  |  |

2022 г

**Содержание**

[Постановка задачи 3](#_Toc97734116)

[Введение 4](#_Toc97734117)

[Описание алгоритма 5](#_Toc97734118)

[Модель алгоритма на языке Promela 6](#_Toc97734119)

[Симуляция 8](#_Toc97734120)

[Верификация и контрпример 12](#_Toc97734121)

[Заключение 16](#_Toc97734122)

[Список литературы 17](#_Toc97734123)

# **Постановка задачи**

1. Установить SPIN
2. Смоделировать один из алгоритмов (в качестве рассматриваемого алгоритма я выбрала алгоритм 2.14 из Ben-Ari.)
3. Провести его симуляцию

# **Введение**

**SPIN — утилита для верификации корректности распределенных программных моделей. Служит для автоматизированной проверки моделей.**

Promela **—** абстрактный язык спецификации алгоритмов. Абстрагирование направлено на то, чтобы с помощью минимальных выразительных средств строить такие абстрактные модели реальных параллельных и распределенных систем, которые легко представляются формальной моделью с конечным числом состояний.

В данной самостоятельной работе основной задачей стоит реализация алгоритма на языке Promela и знакомство с режимом симуляции и верификации утилиты SPIN. Установка SPIN производилась при помощи homebrew (менеджер пакетов с открытым исходным кодом): brew install spin.

# **Описание алгоритма**

Алгоритм 2.14: "Головоломка с лягушками"

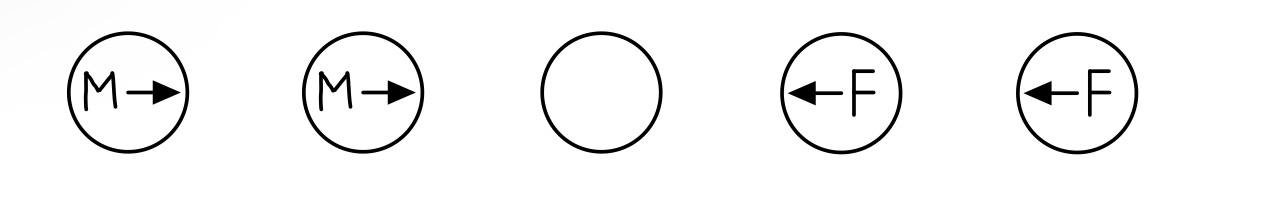
Дано:

Пять камней

2 лягушки-самца справа смотрят влево

2 лягушки-самки слева смотрят направо

*Для наглядности можем изобразить следующим образом:*

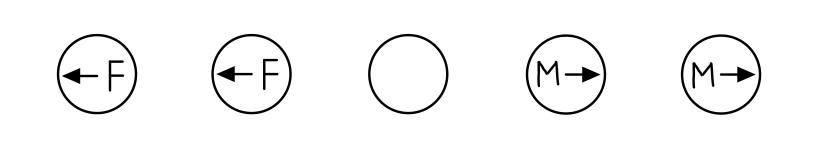
****

Лягушка может двигаться только в направлении своего взгляда, при условии, что перед ней пустой камень, если же нет, то она перепрыгивает препятствие и становится на следующий камень, если он существует и он пуст.

Вопрос:

Существует ли последовательность ходов, которая поменяет местами

самцов и самок лягушек?



# **Модель алгоритма на языке Promela**

#define STONES 5

//для верификации

#define success (\(stones[0]==female) && \(stones[1]==female) && \(stones[3]==male) && \(stones[4]==male)\)

//для верификации

ltl { []!success }

mtype = { none, male, female }

mtype stones[STONES];

proctype mF(byte at) {

end:do

:: atomic {

(at < STONES-1) &&

(stones[at+1] == none) ->

stones[at] = none;

stones[at+1] = male;

at = at + 1;

}

:: atomic {

(at < STONES-2) &&

(stones[at+1] != none) &&

(stones[at+2] == none) ->

stones[at] = none;

stones[at+2] = male;

at = at + 2;

}

od

}

proctype fF(byte at) {

end:do

:: atomic {

(at > 0) &&

(stones[at-1] == none) ->

stones[at] = none;

stones[at-1] = female;

at = at - 1;

}

:: atomic {

(at > 1) &&

(stones[at-1] != none) &&

(stones[at-2] == none) ->

stones[at] = none;

stones[at-2] = female;

at = at - 2;

}

od

}

init {

atomic {

stones[STONES/2] = none;

byte I = 0;

do

:: I == STONES/2 -> break;

:: else ->

stones[I] = male;

run mF(I);

stones[STONES-I-1] = female;

run fF(STONES-I-1);

I++

od

}

}

# **Симуляция**

Произведем простой запуск модели:

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

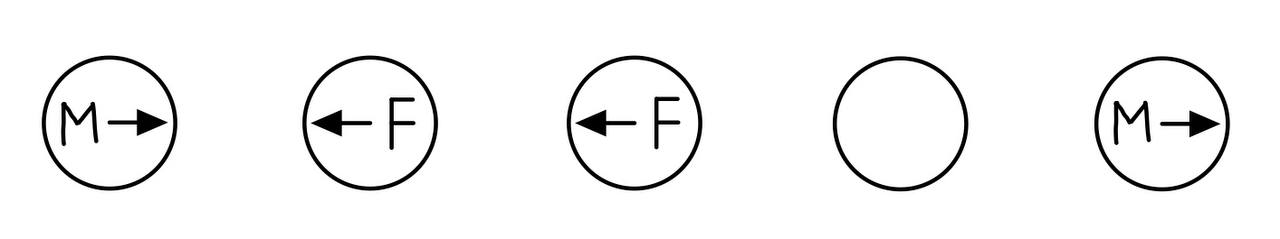
**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

В результате получили следующее расположение (не совпало с желаемым, для ответа на вопрос):



Теперь попробуем ответить на поставленный вопрос. Для этого воспользуемся режимом верификации, добавив в программу желаемое расположение лягушек в переменную success и LTL-формулу, которая будет отрицать это утверждение.

#define success (\(stones[0]==female) && \(stones[1]==female) && \(stones[3]==male) && \(stones[4]==male)\)

ltl { []!success }

# **Верификация и контрпример**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

SPIN нашел контрпример

# **Заключение**

В ходе выполнения первой самостоятельной работы произошло знакомство со средой SPIN. А именно, был смоделирован алгоритм на языке Promela, была произведена его симуляция, а также верификация.

# **Список литературы**

1. Ben-Ari.M. Principles of concurrent and distributed programming. – 2006
2. Ю. Г. Карпов. И. В. Шошмина. Верификация распределенных систем. – 2011
3. Ю. Г. Карпов. И. В. Шошмина. Введение в язык Promela и систему комплексной верификации SPIN. – 2009
4. SPIN Run-Time Options [Электронный ресурс], URL:

http://spinroot.com/spin/Man/Spin.html