Верификация диаграмм по методу Model Checking в QReal

Автор: Копытов Д.С., 471 гр.

Научный руководитель:
ст. преподаватель Кириленко Я.А.

Рецензент:
ст. преподаватель Брыксин Т.А.

TRIK Studio

- QReal
- Программирование роботов
- Графический язык программирования
- Диаграммы
- Более одного исполняемого потока
- Общение между потоками

Верификация

- Model Checking
- Конечная модель системы
- Проверка выполнимости свойств

Постановка задачи

- Исследовать существующие системы верификации, использующие данный метод
- На основе полученных данных выбрать наиболее подходящую
- Внедрить выбранную систему верификации в среду
- Апробировать

Обзор существующих решений

- UPPAAL
  - Отсутствуют средства для описания асинхронного взаимодействия
- NuSMV
  - Синхронизация процессов только посредством разделяемых переменных
- NuXMV
  - Расширение NuSMV
- SPIN

SPIN

- Параллельные и распределённые системы
- Межпроцессное взаимодействие
- Язык Promela для описание модели
- Динамическое создание процессов
- Рандеву и асинхронные каналы
- Темпоральная логика линейного времени (LTL)

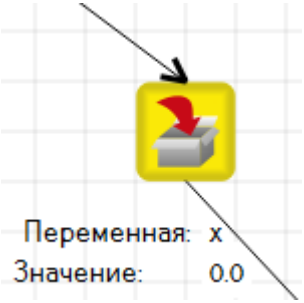
Реализация

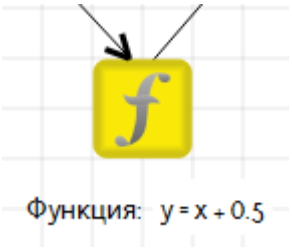
- Связь в одну и другую сторону
  - Трансляция диаграмм во входной язык Promela
  - Отображение контрпримера на диаграмме

Генератор в язык Promela

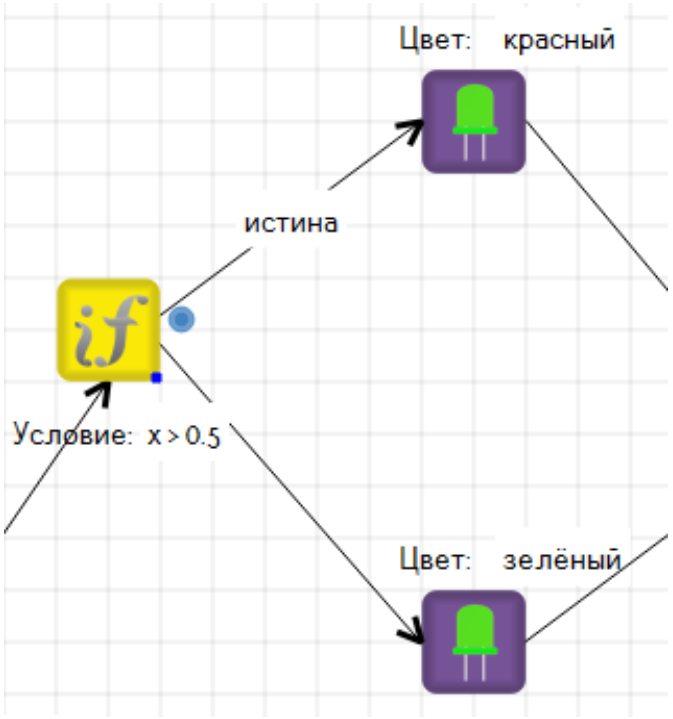
- Типы данных
- Сенсоры и таймер
- Прием/передача сообщений процессами
- Потоки и подпрограммы

Типы данных





```
skip;
```



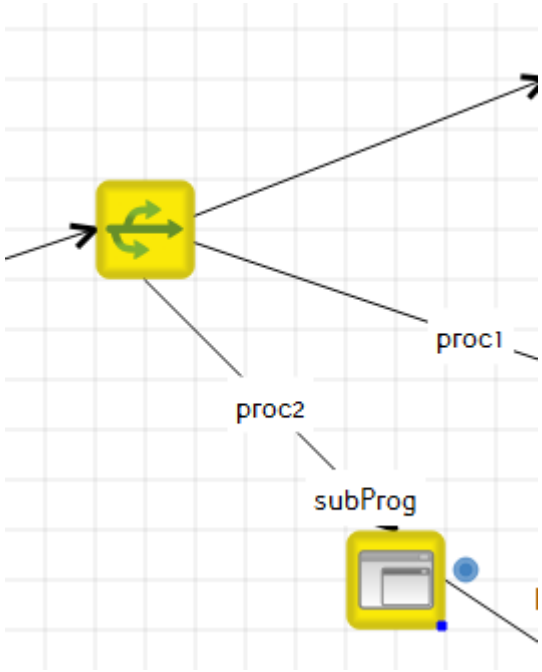
```
if
:: true ->
        ledColor = red;
:: true ->
        ledColor = green;
fi;
```

Прием/передача сообщений процессами

```
typedef message { int size; int a[n] };

#define  proc1proc  0
#define  proc2proc   1

string  proc1buffer[m];
chan proc1chan = [0] of {int, message};
chan proc2chan = [0] of {int, message};
...
proctype proc1() {
      message temp;
      …
      d_step {
                temp.size = 2;
                temp.a[0] = 0;
                temp.a[1] = 2;
      }
      proc2chan!proc1proc(temp);
      …
}
```

```
proctype proc2() {
      message temp;
      …
      proc2chan?proc1proc(temp);
      d_step {
                int i;
                for (i : 0 .. temp.size - 1) {
                      t.a[i].i = temp.a[i];
                }
                t.size = temp.size;
      }
      …
}
```

Потоки и подпрограммы



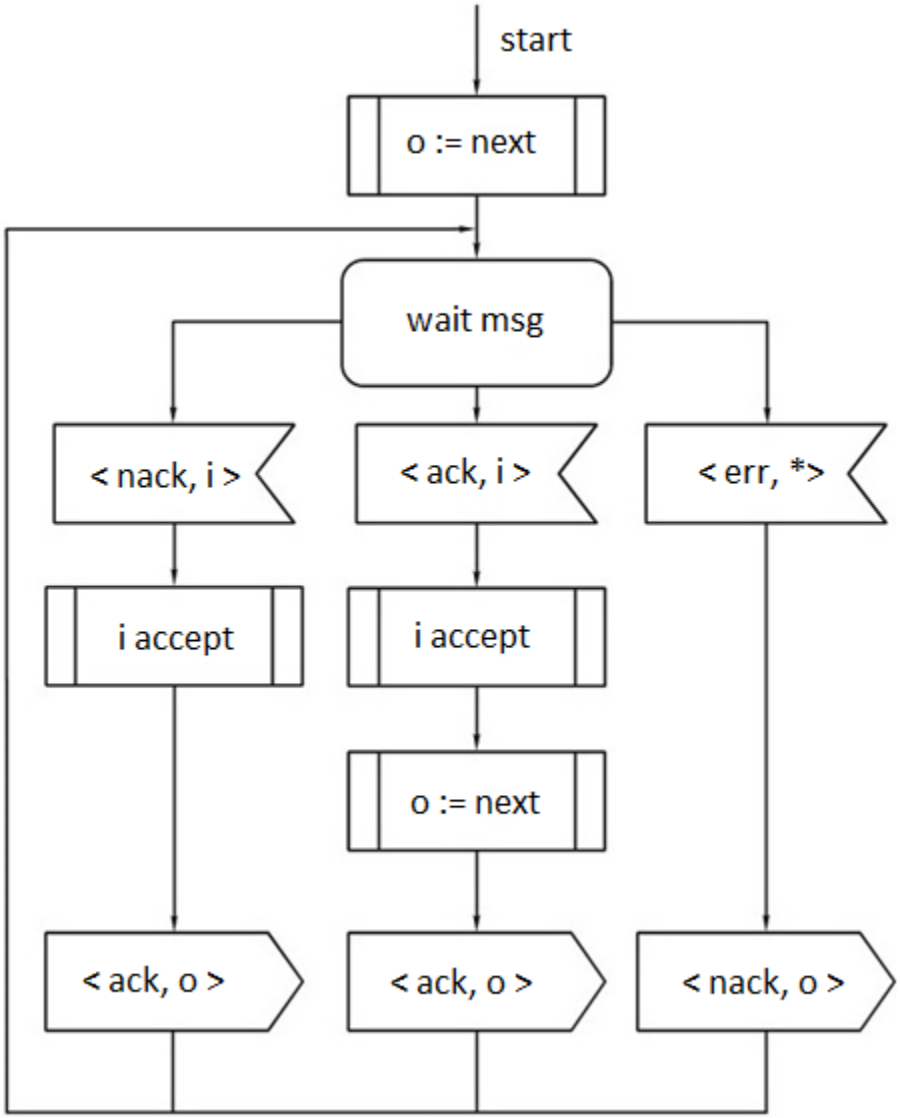

```
inline subProg()
{
    …
}

proctype proc1()
{
    …
}

proctype proc2()
{
    subProg();
    …
}
```

```
proctype main()
{
    run proc1();
    run proc2();
    …
}
```

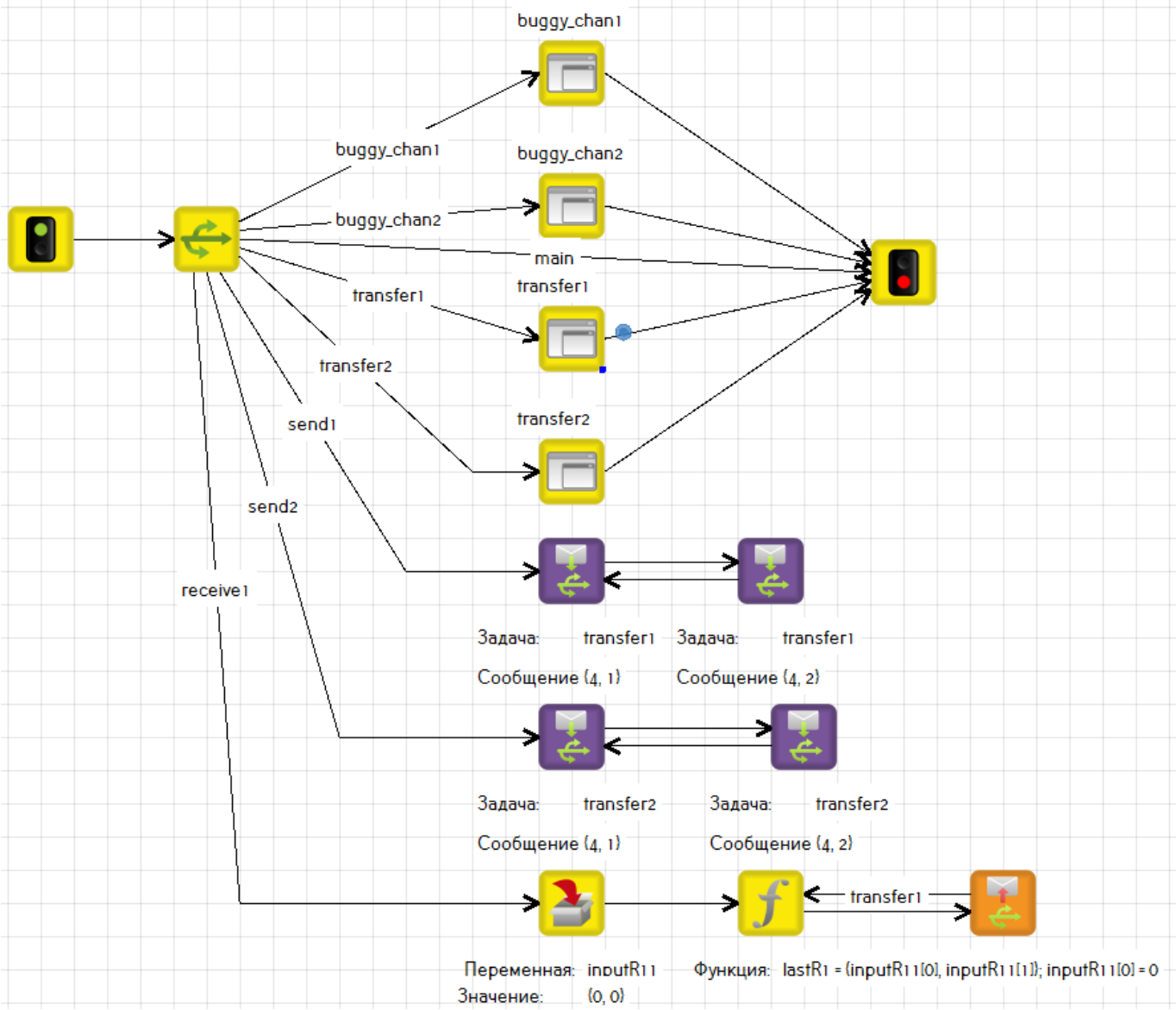
Связь элементов диаграммы с кодом

- Двусторонняя связь
- Подсвечивание контрпримера на диаграмме
- Совместная интерпретация кода и модели

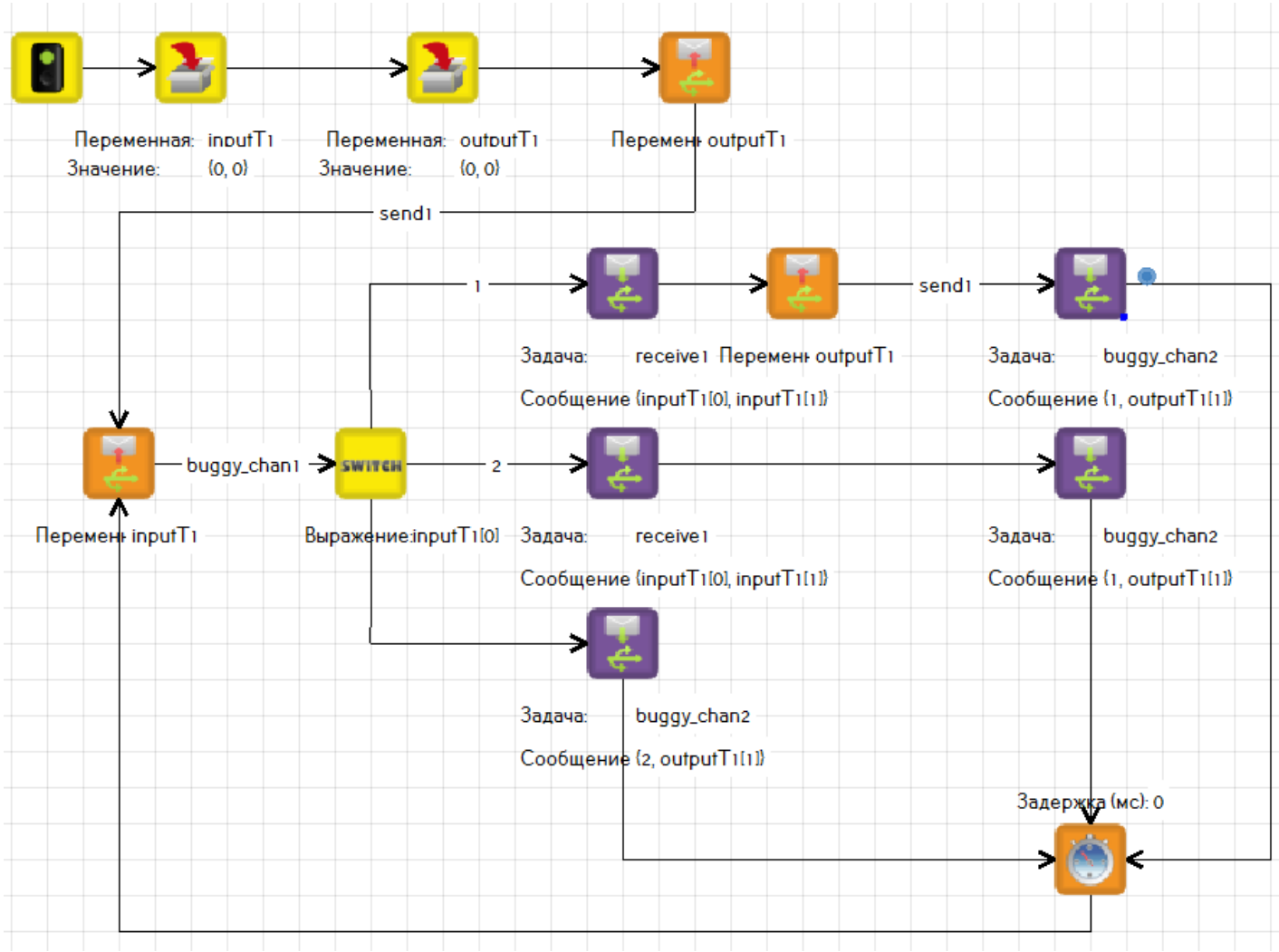
Протокол Линча



Реализация в TRIK Studio



Реализация в TRIK Studio (подпрограмма transfer1)



Заключение

- Было проведено исследование существующих верификаторов
  - SPIN
- Проведено внедрение в TRIK Studio
- Была проведена апробация на конкретном примере (протокол Линча)