24.11.23 ФМ5 Лекция 2

Метод Event-B. Платформа Rodin

Платформа Rodin: https://www3.hhu.de/stups/handbook/rodin/current/pdf/rodin-doc.pdf
http://www.event-b.org/ - сайт **Event-B.org**

Пример. Управление движением автомобилей по мосту
Другое решение задачи управления на мосту
Архитектура платформы **Rodin**
Построение спецификации в **Rodin**:
проект, контекст, машина, теория.
Методы доказательства формул корректности.
Анализ поведения машины в аниматоре.

Автоматное программирование на базисе Event-B

Язык спецификаций Event-B используется в качестве базисного языка для автоматного программирования.

Типы BOOL, INT, NAT.

Оператор присваивания <Переменная> := <Выражение>

Вместо $x \in BOOL$ будем записывать в виде x: BOOL

Начальная инициализация переменных реализуется в секции состояния. Например, x: BOOL := false.

Вместо конструкции partition со сложной семантикой используется описание типа перечисления.

Сегмент **M: if (C) A else B #L** преобразуется к виду

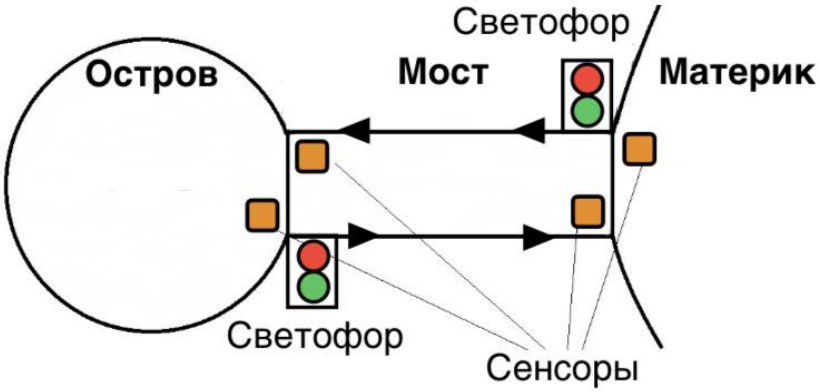
M: C → A #L ранее было: **M: C → A #L**

M: ¬ C → B #L **M: B #L**

Будем использовать комбинированную нотацию двух языков.

Управление движением автомобилей по мосту

Содержательное описание. Узкий односторонний мост соединяет материк и остров. Два светофора (цвета красный и зеленый) при въезде на мост с материка и с острова. Автомобилям запрещено движение на красный светофор при въезде на мост. Имеются четыре сенсора. Каждый сенсор находится на некотором участке автомобильной трассы и способен фиксировать ситуацию, когда автомобиль находится на этом участке трассы. Первый сенсор находится перед въездом на мост с материка. Второй сенсор на мосту перед съездом на остров. Третий сенсор перед въездом на мост с острова. Четвертый сенсор на мосту перед съездом на материк.



Управление движением автомобилей по мосту

Содержательное описание (прод). Число автомобилей, которые могут находиться на острове, ограничено. Необходимо построить контроллер, который переключает светофоры, используя показания сенсоров, и таким образом обеспечивает безопасное движение автомобилей по мосту.

Детализация и анализ требований. Дадим имена светофорам: m_tl – светофор на материке при въезде на мост, isl_tl – светофор на острове при въезде на мост.

Требования окружения:

RE1: Имеются два светофора с именами m_tl и isl_tl.

RE2: Каждый светофор может быть зеленым или красным.

Дадим имена сенсорам: m_Out – при въезде на мост с материка, isl_In – при съезде с моста на остров, isl_Out – при въезде на мост с острова, m_In – при съезде с моста на материк.

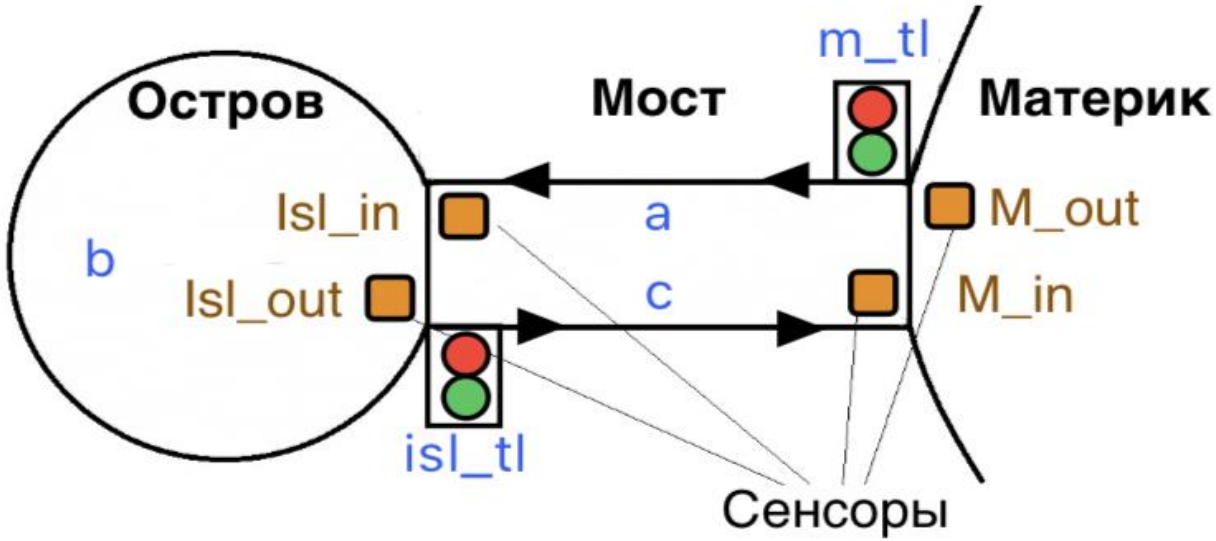
RE3: Имеются четыре сенсора с именами: m_Out, isl_In, isl_Out и m_In.

RE4: Каждый сенсор может быть включен (значение on) или выключен (значение off).

RE5: Сенсор *отключается*, если он изменяет свое значение с on на off.

Сенсоры и светофоры

m – материк, il – остров, tl – traffic light



m_out – при въезде на мост с материка,

isl_in – при съезде с моста на остров,

isl_out – при въезде на мост с острова,

m_in – при съезде с моста на материк

Управление движением по мосту

Функциональные требования:

RF1: Отключение сенсора m_out – очередной автомобиль въехал на мост с материка.

RF2: Отключение сенсора isl_in – очередной автомобиль съехал с моста на остров.

RF3: Отключение сенсора isl_out – очередной автомобиль въехал на мост с острова.

RF4: Отключение сенсора m_in – очередной автомобиль съехал с моста на материк.

RF5: Число автомобилей на острове ограничено.

RF6: Движение автомобилей по мосту возможно либо с материка на остров, либо с острова на материк. Одновременное движение с материка на остров и с острова на материк невозможно.

Требования безопасности:

RS1: При красном светофоре m_tl запрещено движение с материка на мост.

RS2: При красном светофоре isl_tl запрещено движение с острова на мост.

Представленная система управления движением по мосту в целом, и архитектура оборудования в частности, имеют серьезные недостатки и не могут использоваться на практике.

Начальная модель

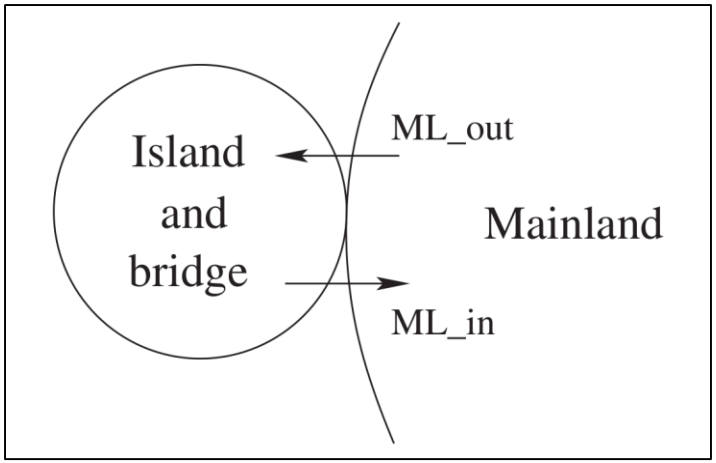
Мост и остров рассматриваются как единое целое.

n – число автомобилей на мосту и на острове.

d – максимальное число автомобилей на острове.

Только два события: уход машины с материка и приход машины на материк.

```
section St0 {
    const d: NAT
    n: NAT
    inv  n <= d
}
process carBridge0er {
   cycle: [ML_out:] n := n + 1 |
          [ML_in:]   n := n - 1
          #cycle
}
```



Начальная модель.Ошибки

```
section St0 {
    const d: NAT
    n: NAT
    inv  n <= d
}
process carBridge0er {
   cycle: [ML_out:] n := n + 1 |
          [ML_in:]   n := n - 1
          #cycle
}
```

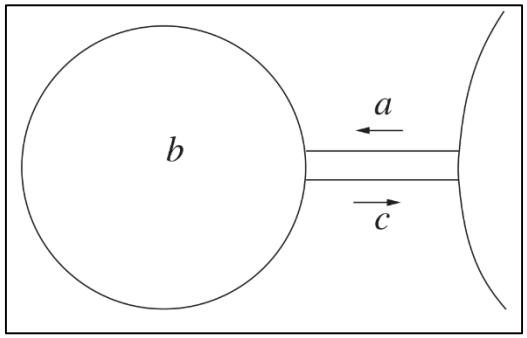
1. После действия n:= n+1 не гарантируется n <= d. n+1<=d
2. Для n := n - 1 генерируется n-1 ∈ NAT
3. В начальный момент не выполняется инвариант n <= d
4. Дедлок. Недоказуема $n \le d \Rightarrow n<d \vee n>0$

Исправленная начальная модель

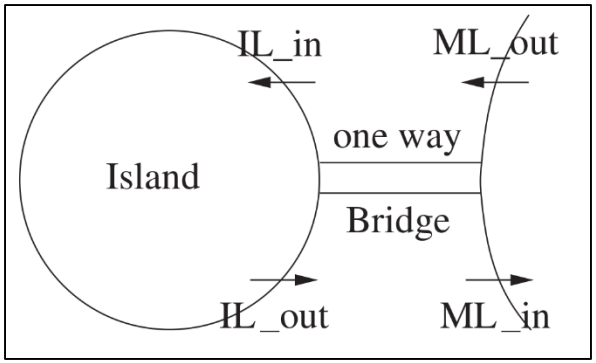
```
section St0 {
    const d: NAT
    axiom d > 0
    n: NAT := 0
    inv  n <= d
}

process carBridge0r {
   cycle: [ML_out:] n < d → n := n + 1 |
          [ML_in:]    n > 0 → n := n - 1
          #cycle
}
```

Уточнение начальной модели




```
section St1 extends St0{
    a: NAT := 0
    b: NAT := 0
    c: NAT := 0
     inv a + b + c = n    //связующий инвариант
     inv a = 0 ∨ c = 0
}
process carBridge1 refines carBridge0 {
    cycle: [ML_out:] a+b<d & c=0 → a := a+1 |
           [ML_in:]    c>0 → c := c-1 |
           [IL_out:]   b>0 & a=0 → b := b-1; c := c+1 |
           [IL_in:]    a>0 →  a := a-1; b := b+1
           #cycle
}
```



Второе уточнение модели

```
section St2 extends St1{
    type Colour = enum {red, green};
    m_tl: Colour := red
     isl_tl: Colour := red
     inv m_tl=red ∨ isl_tl=red
     inv m_tl=green => a+b<d & c=0  //связующий инвариант
     inv isl_tl=green => b>0 & a=0       //связующий инвариант
}
process carBridge2 refines carBridge1{
  cycle: [ML_out:] m_tl=green → a := a + 1 |
          [ML_in:]   c>0 → c:=c-1 |
          [IL_out:]  isl_tl=green →  b:=b-1; c:= c+1  |
          [IL_in:]    a>0 → a := a-1; b:=b+1  |
          [Mtl_green:] m_tl=red & c=0 & b<d  →  m_tl:=green; isl_tl:=red  |
          [Itl_green:]  isl_tl=red & a=0 & b>0 →  isl_tl:=green; m_tl:=red
          #cycle
}
```

Проблемы после второго уточнения

Новые события Mtl_green и Itl_green определяют вроде бы естественные правила переключения светофоров: если мост стал пустым, то оба светофора меняют свет так, чтобы стало возможным движение в противоположную сторону.

Для пустого моста получим быстрое мигание светофоров, что, в частности, может привести к аварии. В книге реализовано следующее решение. Если изменено направление движения, то далее необходимо дождаться появления автомобиля на мосту с противоположной стороны. Данное решение неудовлетворительно. Например. Утром все поехали на остров, где запланирован пикник. Первая партия машин проехала. А остальные окажутся заблокированными до вечера, когда с острова появится первый автомобиль.

Правильное решение следующее. Изменение направления движение возможно, если с противоположной стороны имеется автомобиль, стоящий на сенсоре в ожидании зеленого светофора. Реализация данного решения плохо стыкуется с проведенными уточнениями. Разработку контроллера нужно проводить другим способом.

Другое решение задачи управления на мосту

```
section St0 {
    const d: NAT (*максимальное число автомобилей на острове*)
    axiom d > 0
    a, c: NAT := 0        // число автомобилей на мосту
    b: NAT := 0
    type Colour = enum {red, green}; (*тип цвета светофора*)
    m_tl: Colour := green     (*сначала зеленый*)
    isl_tl: Colour := red     (*сначала красный*)
    m_on: BOOL := false   (*при въезде на мост с материка*)
    isl_on: BOOL := false   (*при въезде на мост с острова*)
    inv  a + b <= d (*инвариант: число автомобилей на острове <= d *)
    inv m_tl=red ∨ isl_tl=red (*оба светофора не могут быть зелеными*)
}
```

Другое решение. Полная программа

Полная программа управления движением автомобилей на мосту:

```
process Bridge {   M_in || Isl_in || Isl_out || M_out ||
                          ISL_OUT_on || M_OUT_on || LightControl     }
```

Контроллеры сенсоров, ISL_OUT_on и M_OUT_on – включение сенсоров

```
process M_in {
     1: c>0 → c:=c-1; #1                    }
process Isl_in {
     2: a>0 →  a:=a-1; b:=b+1; #2  }
process Isl_out {
      3: inv isl_on → b>0;
            isl_on & isl_tl=green →
                        isl_on:=false;  b:=b-1; c:=c+1;  #3       }
process M_out {
     4: m_on & m_tl=green →
                  m_on:=false; a:=a+1; if (a+b=d) m_tl:=red;  #4 }
```

Программа LightControl

```
process LightControl {
    7: if (isl_on & isl_tl= red & a=0)
                {isl_tl := green; m_tl:=red}
       else if (m_on & m_tl=red & c=0 & a+b< d & not isl_on)
                                        {m_tl := green; isl_tl:=red}
   #7
}

process M_OUT_on { 5:  m_on: = true;  #5}

process ISL_OUT_on { 6:  b>0 → isl_on := true ;  #6}
```

Обнаруженные ошибки

1.Сначала была обнаружен дедлок в состояниях right0 и left0, когда первый автомобиль на мосту выезжает с него раньше срабатывания сегмента в состояниях right0 или left0. Был введен признак empty для пустого моста.

2. Следствием исправления данной ошибки стал дедлок в состоянии zero. Для исправления второй ошибки введено дополнительное охранное условие: b < d.

3. Третья ошибка была обнаружена на одном из вариантов анимации. После въезда на мост первого автомобиля с острова этот автомобиль неожиданно возвращался на остров. Здесь оказалось возможным срабатывание сегмента off процесса IL_in. Для исправления ошибки введено дополнительное охранное условие itl = red.

Обнаруженные ошибки

4. Обнаружена при написании статьи. В программе ML_out в сегменте on использовалось условие a+b<d для пропуска автомобиля на мост с материка. Отметим, что сенсор реагирует только на появление на нем или съезде с него автомобиля. А автомобиль может съехать только при зеленом светофоре; условие a+b<d не может воздействовать на автомобиль. После пересчета a:=a+1 в сегменте on далее сегмент left программы LightControl проверяет условие a+b=d и если оно истинно устанавливает красный светофор. Потенциально, хотя и крайне маловероятно, следующий автомобиль может успеть съехать на мост раньше (при зеленом светофоре), чем сегмент left выполнит оператор mtl:=red.

```
process ML_out {    inv a+b<=d
    off:  ml_out:=true #on
    on:    if (mtl=green){ a:=a+1; ml_out:=false  #on1 } else #on
    on1:  if (a+b=d){ mtl:=red  #off } else #off
}
```

Rodin

Rodin – платформа для разработки и верификации спецификаций на Event-B

Платформа Rodin содержит:

- Текстовый редактор спецификаций на Event-B
- Набор систем автоматического доказательства
- Поддержку интерактивного доказательства

Полезные ссылки

- Основной сайт
  http://www.event-b.org/
- Туториал по платформе Rodin
  https://www3.hhu.de/stups/handbook/rodin/
- Полное описание языка Event-B (4 страницы)
  http://wiki.event-b.org/images/EventB-Summary.pdf

- Книга J.-R. Abrial. Modeling in Event-B. System and Software Engineering