Математические методы верификации схем и программ

mk.cs.msu.ru → Лекционные курсы
→ Математические методы верификации схем и программ

Семинар O1

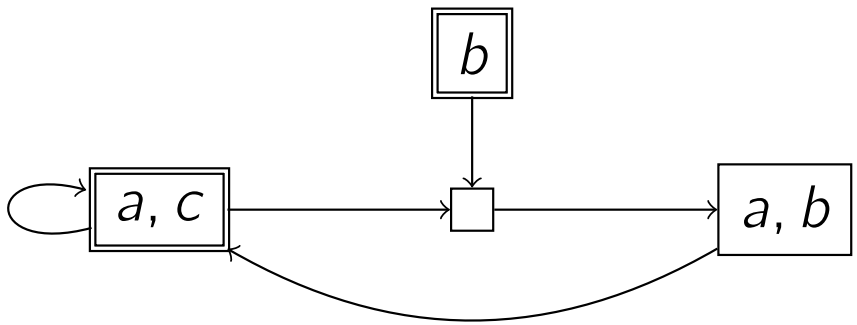
Упражнения для `Spin`

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2023/2024, осенний семестр

Упражнение 1: структура Крипке

Выяснить, выполняется ли заданная формула
на заданной модели Крипке



- $\mathbf{G}(a \to b \vee c)$
- $\mathbf{GF}a$
- $\mathbf{FG}a$
- $\mathbf{GF}\neg c \to \mathbf{F}(a \,\&\, b)$
- $\mathbf{GF}\neg c \to \mathbf{G}(\neg b \mathbf{U} a \,\&\, b)$

Упражнение 2: программы

Три программы выполняются параллельно,
и в каждой из них в бесконечном цикле выполняется команда

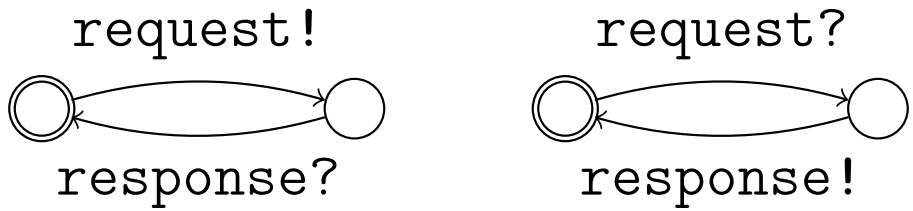
- В первой: `if(x < 10) x = x + 1;`
- Во второй: `if(x > 0) x = x - 1;`
- В третьей: `if(x == 10) x = 0;`

Начальное значение `x` — 0

Выяснить, обязательно ли всевозможные значения `x`, получаемые при выполнении программ, лежат в интервале (а) `[0,10]` или (б) `[-1,11]`, в заданном предположении об атомарности операций:

1. Ветвление состоит из двух атомарных действий:
   проверка условия; присваивание
2. Ветвление атомарно

Упражнение 3: сообщения



Система состоит из клиента (левый автомат),
сервера (правый автомат) и
двух однонаправленных каналов связи между ними, `request` и `response`:

1. Синхронных
2. Асинхронных ёмкости 1

«`m!`»/«`m?`» означает, что сообщение (неважно какое) посылается в канал `m` / принимается из этого канала при выполнении перехода

Проверить, справедливы ли следующие свойства системы:

- После отсылки сообщения в `request` клиент
  рано или поздно получит сообщение из `response`
- Клиент и сервер не могут одновременно ожидать приёма сообщений

Упражнение 4: особенности синхронизации

```
bool b1, b2;
active proctype P() {
  do
  :: b1 = !b1; b2 = !b2;
  od
}
```

Дополнить код, приведённый выше, так,
чтобы можно было проверить справедливость следующего свойства:

«В начале каждого витка цикла значения `b1` и `b2` равны»

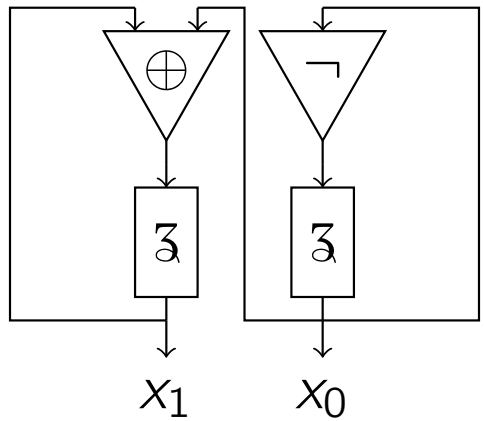
Упражнение 4: особенности синхронизации

```
bool b1, b2;
active proctype P1() {
  do :: b1 = !b1; od
}
active proctype P2() {
  do :: b2 = !b2; od
}
```

Дополнить код, приведённый выше, так, чтобы
для каждого из присваиваний было верно следующее:
если оно (A) выполнилось больше раз, чем другое (B),
то перед следующим выполнением A хотя бы раз выполняется B

Проверить справедливость следующего свойства
для дополненного кода:
«Значения `b1` и `b2` равны $\Leftrightarrow$
присваивания выполнились одинаковое число раз»

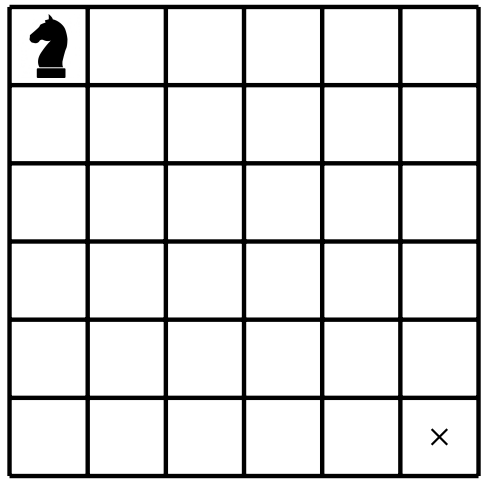
Упражнение 5: схемы



Убедиться, что этой схемой реализуется двухбитовый счётчик:

$$(x_1 x_0)_{t+1} = (x_1 x_0)_t + 1 \ (mod\ 4)$$

Упражнение 6: несложная шахматная загадка



Рассмотрим шахматную доску 6x6 и коня,
ходящего по правилам шахмат

Может ли конь, начав в левом верхнем углу, достичь правого нижнего?

Может ли конь достичь правого нижнего угла из **любой** клетки?