Верификация ПО: РК 2

31 марта 2019 г.

Содержание

1 Интерливинг. Асинхронная композиция процессов

Определение. Интерливинг — чередующееся выполнение параллельных процессов. В один момент времени может выполняться действие только одного из процессов, но какое именно не определено.

Определение. Асинхронная композиция процессов

$$P = P_1 \| P_2$$

$$P_1 = (S_1, S_{01}, \rightarrow_1, op_1)$$

$$P_2 = (S_2, S_{02}, \rightarrow_2, op_2)$$

$$P = (S, S_0, \rightarrow, op)$$

$$S = S_1 \times S_2$$

$$S_0 = S_{01} \times S_{02}$$

$$(s_1, s_2) \rightarrow (s_1', s_2') \ \forall (s_2, s_2')$$

2 Архитектура SPIN

1. На входе модель на языке promela.
2. На основе неё строим структуру Крипке.
3. Строим автомат Бюхи на основе заданной LTL-формулы или берём заданный.
4. Проверка пустоты языка или невыполненных assert-ов.

Promela

1. G = [], F = <>, U = U
2. асинхронный процесс: assert(x > b)
3. never-автомат — явное описание автомата Бюхи

Синтаксис:

- `proctype` — процесс
- `!` — записать в канал (синхронное)
- `?` — прочитать из канала (синхронное)
- целые числа `short`, `int`
- массивы фиксированной длины
- структуры `typedef`

- булевский тип `bool`, `bit`
- `do`, `if`

```
do
  :: G1 -> 01
  :: G2 -> 02
...
  :: else -> 0
od
```

Если условия совпали условия — недетерминизм, система может пойти по любой из веток. Аналогично в if-ы.

3 Задачи ???

3.1 Mutex

http://lwn.net/Articles/243851/

```
#define spin_lock(mutex)     \
    do                       \
    :: 1 ->                  \
        atomic {             \
            if               \
            :: mutex == 0 -> \
                mutex = 1;   \
                break        \
            :: else ->       \
                skip         \
            fi               \
        }                    \
    od

#define spin_unlock(mutex) \
    mutex = 0
```