

Spin2Core - Cohen Eli

d_step vs. atomic (cont.)

Review:

In Spin we have two kind of atomic steps: “atomic” and “d_step”.

The first one is a weak atomic step: You can branch into it, and from it to an outside statement and it can contain non-deterministic steps.

The second one is a stronger one: You can't branch into it, you're not allowed to branch from it to any outside statement, and all statements must be deterministic (if a non-deterministic does appear, the first executable statement will be chosen).

The following examples describe simple programs:

A process that wishes to change the equality of two variables (x,y) while another process that wishes to equal them (first time they are changed only).

The LTL sentence that will be tested are:

“Eventually x is always not equal to y” ($\diamond \blacksquare (x \neq y)$)

“x is always equal to y” ($\blacksquare (x == y)$)

Program:

```
byte x=5;
byte y=5;
bit sem = 0;
```

```
proctype Atomic ()
{
  atomic {
    sem = 1;
    x = y - 1;
    (x == y);
    x = y - 1;
  }
}

proctype Equalizer ()
{
  do
    :: (sem == 1) ->
      if
        :: (x == y) -> break;
        :: (x != y) -> x=y; break;
      fi;
  od
}

never { /* []<>(x==y) */
T0_init:
  if
    :: (x==y) -> goto accept_S5
    :: (1) -> goto T0_init
  fi;
accept_S5:
  if
    :: (1) -> goto T0_init
  fi;
accept_all:
  skip
}

init
{
  atomic {
    run Atomic();
    run Equalizer();
  }
}
```

States:

```
proctype Atomic
  state 5 -(tr 24)-> state 2 [id 0 tp 2] [A---G] line 7 => sem = 1
  state 2 -(tr 25)-> state 3 [id 1 tp 2] [A---G] line 9 => x = (y-1)
  state 3 -(tr 26)-> state 4 [id 2 tp 2] [A---G] line 10 => ((x==y))
  state 4 -(tr 27)-> state 6 [id 3 tp 2] [----G] line 11 => x = (y-1)
  state 6 -(tr 28)-> state 0 [id 5 tp 3500] [--e-L] line 13 => -end-
proctype Equalizer
  state 9 -(tr 15)-> state 7 [id 6 tp 2] [----G] line 17 => ((sem==1))
  state 7 -(tr 16)-> state 12 [id 7 tp 2] [----G] line 18 => ((x==y))
  state 7 -(tr 19)-> state 5 [id 9 tp 2] [----G] line 18 => ((x!=y))
  state 12 -(tr 23)-> state 0 [id 17 tp 3500] [--e-L] line 23 => -end-
  state 5 -(tr 20)-> state 12 [id 10 tp 2] [----G] line 20 => x = y
proctype :never:
  state 5 -(tr 4)-> state 9 [id 18 tp 2] [----G] line 27 => ((x==y))
  state 5 -(tr 7)-> state 5 [id 20 tp 2] [----G] line 27 => (1)
  state 9 -(tr 9)-> state 5 [id 24 tp 2] [-a--L] line 32 => (1)
proctype init
  state 3 -(tr 1)-> state 2 [id 30 tp 2] [A---L] line 41 => (run Atomic())
  state 2 -(tr 2)-> state 4 [id 31 tp 2] [----L] line 43 => (run Equalizer())
  state 4 -(tr 3)-> state 0 [id 33 tp 3500] [--e-L] line 45 => -end-
```

Transition Type: A=atomic; D=d_step; L=local; G=global
Source-State Labels: p=progress; e=end; a=accept;

Results:

(Spin Version 3.2.3 -- 1 August 1998)
+ Partial Order Reduction
Full statespace search for:
never-claim +
assertion violations + (if within scope of claim)
acceptance cycles - (not selected)
invalid endstates - (disabled by never-claim)

State-vector 28 byte, depth reached 22, errors: 0
14 states, stored
5 states, matched
19 transitions (= stored+matched)
11 atomic steps

hash conflicts: 0 (resolved)
(max size 2¹⁸ states)

1.493 memory usage (Mbyte)

unreached in proctype Atomic
(0 of 6 states)
unreached in proctype Equalizer
(0 of 12 states)
unreached in proctype :init:
(0 of 4 states)

no atomic.trail

Program:

```
byte x=5;
byte y=5;
bit sem = 0;

proctype Atomic ()
{
    atomic {
        sem = 1;
        x = y - 1;
        (x == y);
        x = y - 1;
    }
}

proctype Equalizer ()
{
    do
        :: (sem == 1) ->
            if
                :: (x == y) -> break;
                :: (x != y) -> x=y; break;
            fi;
    od
}

never { /* <>(x!=y) */
T0_init:
    if
        :: (x!=y) -> goto accept_all
        :: (1) -> goto T0_init
    fi;
accept_all:
    skip
}

init
{
    atomic {
        run Atomic();
        run Equalizer();
    }
}
```

States:

```
proctype Atomic
state 5 -(tr 21)-> state 2 [id 0 tp 2] [A---G] line 7 => sem = 1
state 2 -(tr 22)-> state 3 [id 1 tp 2] [A---G] line 9 => x = (y-1)
state 3 -(tr 23)-> state 4 [id 2 tp 2] [A---G] line 10 => ((x==y))
state 4 -(tr 24)-> state 6 [id 3 tp 2] [----G] line 11 => x = (y-1)
state 6 -(tr 25)-> state 0 [id 5 tp 3500] [--e-L] line 13 => -end-
proctype Equalizer
state 9 -(tr 12)-> state 7 [id 6 tp 2] [----G] line 17 => ((sem==1))
state 7 -(tr 13)-> state 12 [id 7 tp 2] [----G] line 18 => ((x==y))
state 7 -(tr 16)-> state 5 [id 9 tp 2] [----G] line 18 => ((x!=y))
state 12 -(tr 20)-> state 0 [id 17 tp 3500] [--e-L] line 23 => -end-
state 5 -(tr 17)-> state 12 [id 10 tp 2] [----G] line 20 => x = y
proctype :never:
state 5 -(tr 4)-> state 7 [id 18 tp 2] [----G] line 27 => ((x!=y))
state 5 -(tr 7)-> state 5 [id 20 tp 2] [----G] line 27 => (1)
state 7 -(tr 9)-> state 8 [id 24 tp 2] [-a--L] line 32 => (1)
state 8 -(tr 10)-> state 0 [id 25 tp 3500] [--e-L] line 33 => -end-
proctype init
state 3 -(tr 1)-> state 2 [id 26 tp 2] [A---L] line 37 => (run Atomic())
state 2 -(tr 2)-> state 4 [id 27 tp 2] [----L] line 39 => (run Equalizer())
state 4 -(tr 3)-> state 0 [id 29 tp 3500] [--e-L] line 41 => -end-
```

Transition Type: A=atomic; D=d_step; L=local; G=global
Source-State Labels: p=progress; e=end; a=accept;

Results:

pan: claim violated! (at depth 9)

pan: wrote atomic2.trail

(Spin Version 3.2.3 -- 1 August 1998)

Warning: Search not completed
+ Partial Order Reduction

Full statespace search for:

never-claim +
assertion violations + (if within scope of claim)
acceptance cycles - (not selected)
invalid endstates - (disabled by never-claim)

State-vector 28 byte, depth reached 9, errors: 1

4 states, stored
0 states, matched
4 transitions (= stored+matched)
3 atomic steps

hash conflicts: 0 (resolved)
(max size 2¹⁸ states)

1.493 memory usage (Mbyte)

```
1: proc - (:never:) line 29 "atomic2" (state 3) [(1)]
2: proc 1 (:init:) line 38 "atomic2" (state 1) [(run Atomic())]
3: proc 1 (:init:) line 39 "atomic2" (state 2) [(run Equalizer())]
4: proc - (:never:) line 29 "atomic2" (state 3) [(1)]
5: proc 2 (Atomic) line 8 "atomic2" (state 1) [sem = 1]
6: proc 2 (Atomic) line 9 "atomic2" (state 2) [x = (y-1)]
7: proc - (:never:) line 28 "atomic2" (state 1) [(x!=y)]
8: proc 3 (Equalizer) line 18 "atomic2" (state 1) [(sem==1)]
9: proc - (:never:) line 32 "atomic2" (state 7) [(1)]
```

spin: trail ends after 9 steps

#processes: 4
x = 4
y = 5
sem = 1

```
9: proc 3 (Equalizer) line 18 "atomic2" (state 7)
9: proc 2 (Atomic) line 10 "atomic2" (state 3)
9: proc 1 (:init:) line 41 "atomic2" (state 4)
9: proc - (:never:) line 33 "atomic2" (state 8)
```

4 processes created

atomic2.trail:

```
-2:2:-2
1:0:20
2:1:26
3:1:27
4:0:20
5:2:0
6:2:1
7:0:18
8:3:6
9:0:24
```

Program:

```
byte x=5;
byte y=5;
bit sem = 0;

proctype Dstep ()
{
    d_step {
        sem = 1;
        x = y - 1;
        (x == y);
        x = y - 1;
    }
}

proctype Equalizer ()
{
    do
        :: (sem == 1) ->
            if
                :: (x == y) -> break;
                :: (x != y) -> x=y; break;
            fi;
    od
}

never { /* []<>(x==y) */
T0_init:
    if
        :: (x==y) -> goto accept_S5
        :: (1) -> goto T0_init
    fi;
accept_S5:
    if
        :: (1) -> goto T0_init
    fi;
accept_all:
    skip
}

init
{
    atomic {
        run Dstep();
        run Equalizer();
    }
}
```

States:

```
proctype Dstep
state 5 -(tr 24)-> state 6 [id 4 tp 2] [D---G] line 7 => D_STEP
state 6 -(tr 25)-> state 0 [id 5 tp 3500] [--e-L] line 13 => -end-
proctype Equalizer
state 9 -(tr 15)-> state 7 [id 6 tp 2] [----G] line 17 => ((sem==1))
state 7 -(tr 16)-> state 12 [id 7 tp 2] [----G] line 18 => ((x==y))
state 7 -(tr 19)-> state 5 [id 9 tp 2] [----G] line 18 => ((x!=y))
state 12 -(tr 23)-> state 0 [id 17 tp 3500] [--e-L] line 23 => -end-
state 5 -(tr 20)-> state 12 [id 10 tp 2] [----G] line 20 => x = y
proctype :never:
state 5 -(tr 4)-> state 9 [id 18 tp 2] [----G] line 27 => ((x==y))
state 5 -(tr 7)-> state 5 [id 20 tp 2] [----G] line 27 => (1)
state 9 -(tr 9)-> state 5 [id 24 tp 2] [-a--L] line 32 => (1)
proctype init
state 3 -(tr 1)-> state 2 [id 30 tp 2] [A---L] line 41 => (run Dstep())
state 2 -(tr 2)-> state 4 [id 31 tp 2] [----L] line 43 => (run Equalizer())
state 4 -(tr 3)-> state 0 [id 33 tp 3500] [--e-L] line 45 => -end-
```

Transition Type: A=atomic; D=d_step; L=local; G=global
Source-State Labels: p=progress; e=end; a=accept;

Results:

pan: block in step seq (at depth 4)

pan: wrote d_step.trail

(Spin Version 3.2.3 -- 1 August 1998)

Warning: Search not completed
+ Partial Order Reduction

Full statespace search for:

never-claim +
assertion violations + (if within scope of claim)
acceptance cycles - (not selected)
invalid endstates - (disabled by never-claim)

State-vector 28 byte, depth reached 4, errors: 1

2 states, stored
0 states, matched
2 transitions (= stored+matched)
1 atomic steps

hash conflicts: 0 (resolved)
(max size 2¹⁸ states)

1.493 memory usage (Mbyte)

```
1: proc - (:never:) line 28 "d_step" (state 1) [(x==y)]
2: proc 1 (:init:) line 42 "d_step" (state 1) [(run Dstep())]
3: proc 1 (:init:) line 43 "d_step" (state 2) [(run Equalizer())]
4: proc - (:never:) line 33 "d_step" (state 7) [(1)]
```

spin: trail ends after 4 steps

```
#processes: 4
x = 5
y = 5
sem = 0
```

```
4: proc 3 (Equalizer) line 17 "d_step" (state 9)
4: proc 2 (Dstep) line 7 "d_step" (state 5)
4: proc 1 (:init:) line 45 "d_step" (state 4)
4: proc - (:never:) line 27 "d_step" (state 5)
```

4 processes created

d_step.trail:

```
-2:2:-2
1:0:18
2:1:30
3:1:31
4:0:24
```

Program:

```
byte x=5;
byte y=5;
bit sem = 0;

proctype Dstep ()
{
    d_step {
        sem = 1;
        x = y - 1;
        (x == y);
        x = y - 1;
    }
}

proctype Equalizer ()
{
    do
        :: (sem == 1) ->
            if
                :: (x == y) -> break;
                :: (x != y) -> x=y; break;
            fi;
    od
}

never { /* <>(x!=y) */
T0_init:
    if
        :: (x!=y) -> goto accept_all
        :: (1) -> goto T0_init
    fi;
accept_all:
    skip
}

init
{
    atomic {
        run Dstep();
        run Equalizer();
    }
}
```

States:

```
proctype Dstep
state 5 -(tr 21)-> state 6 [id 4 tp 2] [D---G] line 7 => D_STEP
state 6 -(tr 22)-> state 0 [id 5 tp 3500] [--e-L] line 13 => -end-
proctype Equalizer
state 9 -(tr 12)-> state 7 [id 6 tp 2] [----G] line 17 => ((sem==1))
state 7 -(tr 13)-> state 12 [id 7 tp 2] [----G] line 18 => ((x==y))
state 7 -(tr 16)-> state 5 [id 9 tp 2] [----G] line 18 => ((x!=y))
state 12 -(tr 20)-> state 0 [id 17 tp 3500] [--e-L] line 23 => -end-
state 5 -(tr 17)-> state 12 [id 10 tp 2] [----G] line 20 => x = y
proctype :never:
state 5 -(tr 4)-> state 7 [id 18 tp 2] [----G] line 27 => ((x!=y))
state 5 -(tr 7)-> state 5 [id 20 tp 2] [----G] line 27 => (1)
state 7 -(tr 9)-> state 8 [id 24 tp 2] [-a--L] line 32 => (1)
state 8 -(tr 10)-> state 0 [id 25 tp 3500] [--e-L] line 33 => -end-
proctype init
state 3 -(tr 1)-> state 2 [id 26 tp 2] [A---L] line 37 => (run Dstep())
state 2 -(tr 2)-> state 4 [id 27 tp 2] [----L] line 39 => (run Equalizer())
state 4 -(tr 3)-> state 0 [id 29 tp 3500] [--e-L] line 41 => -end-
```

Transition Type: A=atomic; D=d_step; L=local; G=global
Source-State Labels: p=progress; e=end; a=accept;

Results:

pan: block in step seq (at depth 4)

pan: wrote d_step2.trail

(Spin Version 3.2.3 -- 1 August 1998)

Warning: Search not completed
+ Partial Order Reduction

Full statespace search for:

never-claim +
assertion violations + (if within scope of claim)
acceptance cycles - (not selected)
invalid endstates - (disabled by never-claim)

State-vector 28 byte, depth reached 4, errors: 1

2 states, stored
0 states, matched
2 transitions (= stored+matched)
1 atomic steps

hash conflicts: 0 (resolved)
(max size 2¹⁸ states)

1.493 memory usage (Mbyte)

```
1: proc - (:never:) line 29 "d_step2" (state 3) [(1)]
2: proc 1 (:init:) line 38 "d_step2" (state 1) [(run Dstep())]
3: proc 1 (:init:) line 39 "d_step2" (state 2) [(run Equalizer())]
4: proc - (:never:) line 29 "d_step2" (state 3) [(1)]
```

spin: trail ends after 4 steps

#processes: 4
x = 5
y = 5
sem = 0

```
4: proc 3 (Equalizer) line 17 "d_step2" (state 9)
4: proc 2 (Dstep) line 7 "d_step2" (state 5)
4: proc 1 (:init:) line 41 "d_step2" (state 4)
4: proc - (:never:) line 27 "d_step2" (state 5)
```

4 processes created

d_step.trail:

-2:2:-2
1:0:20
2:1:26
3:1:27
4:0:20