Bisimulation Minimization and Symbolic Model Checking

Sylvain Julmy

December 18, 2017

The Lee-Yannakakis algorithm

LY - idea

- Stabilize only reachable blocks.
- Reachable block use a representative that has to bee reachable.
- The first state is the representative for the initial block.
- To find new reachable state, we look for transition from representative of reachable state to state from unreachable block.

LY - idea

Two loops :

- Search new reachable blocks
- Stabilize reachable but unstable blocks

LY - termination

With the exception of the initial block, all new blocks created by the algorithm have paths to the bad block.

LY - termination

Therefore, when a second block becomes reachable, the algorithm should raise a violation and terminate.

LY - new algorithm

Basic idea[1] :

- Search new reachable blocks.
- Stabilize reachable but unstable blocks.
- When a second block becomes reachable $\rightarrow$ raise a violation.

[1] Very similar to BR

LY - search

To search for new reachable block, the algorithm is searching from all the successor of the initial state if one of those is in a different block.

The algorithm also determine if the initial block has to be stabilize or not.

LY - search

$D := post(B)$
for all $\langle C, q \rangle \in post(init)$ **do**
    if $B \neq C$ **then**
        raise violation
    end if
    if $B \cap pre(C) \neq B$ **then** ▷ Not all predecessor of B are in B
        B is not stable
    end if
    $D := D - C$
end for
if $D \neq \emptyset$ **then** ▷ $post(init) = \emptyset$
    B is not stable
end if

LY - search

$queue := \emptyset$

$partition = \{B, Bad\}$

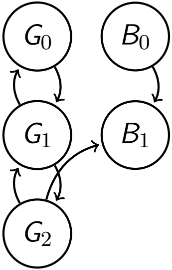
$init = G_0$

$B = \{G_0, \ldots, G_2\}$

$Bad = \{B_0, B_1\}$

$block_{init} = \langle B, init \rangle$

$D = post(B) = \{B, Bad\}$



LY - search

$post(init) = \{B\}$

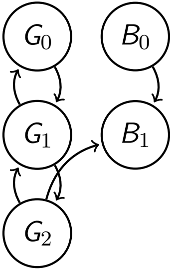
$\langle C, q \rangle = \langle B, init \rangle$

$pre(C) = \{B\}$

$\{B\} \cap pre(C) = \{B\} == \{B\}$

$D = \{B, Bad\} - \{B\} = Bad$

$D \neq \emptyset \rightarrow enqueue(\langle B, init \rangle)$



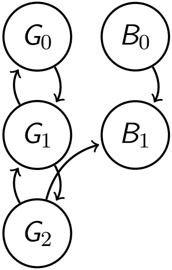
LY - stabilization

while B is not stable **do**
  Mark B as stable
  Compute the frontier of B
  Let B' the state of B that can only reach B
  Let B'' the state of B that can reach a bad block
  if $\emptyset \neq B' \cap \mathit{pre}(B') \neq B'$ or $\emptyset \neq B' \cap \mathit{pre}(B'') \neq B'$ **then**
    Mark B as unstable
  end if
end while

LY - stabilization

Iteration 1

$$init = G_0$$
$$partition = \{B, Bad\}$$
$$B = \{G_0, G_1, G_2\}$$
$$pre(B) = \{B\}$$
$$post(B) = \{B, Bad\}$$



LY - stabilization

Iteration 1

$B_1' = B \cap pre(B) = \{B\}$

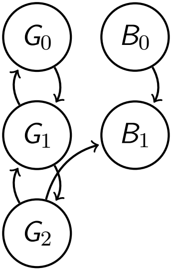
$B_2' = pre(post(B) - B) = pre(\{Bad\}) = \{B\}$

$B' = B_1' - B_2' = \emptyset$

$B'' = B - B' = B$

$partition = \{B, Bad, B\}$

$B := B' = \emptyset$



LY - stabilization

Iteration 1

$B := B' = \emptyset$

$pre(B) = \emptyset$

$B'' = B$

$pre(B'') = B$

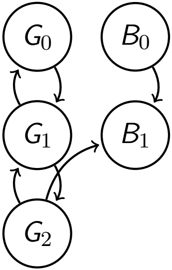
$B \cap pre(B) = \emptyset$

$B \cap pre(B'') = \emptyset$

no enqueue !

$post(init) \cap B'' = \{B\} \cap \{B\} = \{B\}$

$\rightarrow$ raise safety violation !



LY - search (2)

$queue := \emptyset$

$partition = \{B, Bad\}$

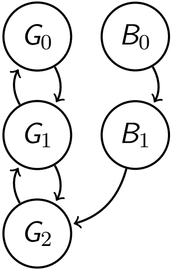
$init = G_0$

$B = \{G_0, \ldots, G_2\}$

$Bad = \{B_0, B_1\}$

$block_{init} = \langle B, init \rangle$

$D = post(B) = \{B\}$



LY - search (2)

$post(init) = \{B\}$

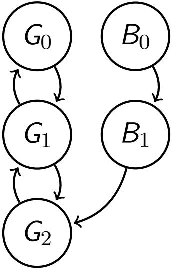
$\langle C, q \rangle = \langle B, init \rangle$

$pre(C) = \{B, Bad\}$

$\{B\} \cap pre(C) = \{B\} == \{B\}$

$D = \{B\} - \{B\} = \emptyset$

no safety violation, terminate



LY - complexity

$$(n-1) * 5M + 4I + 3D + 4E$$

where

- n : number of BR iterations
- M : number of image iterations
- I : number of intersection operations
- D : number of set difference operations
- E : number of equality check
- U : number of union operations

BFH

The Bouajjani-Fernandez-Halbwachs algorithm

BFH - idea

- BFH, like LY, selects reachable blocks to stabilize but differ in how to stabilize a block.
- BFH stabilize a block w.r.t. all the other blocks (either reachable or unreachable).
- The algorithm become simplier but unnecessary work is done.

BFH - termination

As in LY, BFH could terminate when a second block becomes reachable. The algorithm correctly determine violations of invariants but not as soon as they occur.

BFH - termination

The algorthim may traverse a path from the bad block to the initial state before the initial block becomes stable.
Thus, the algorithm take more iteration to terminate.

BFH - new Algorithm

```
Mark the bad block
I = [init]_p
while I is not marked do
    N := split(I, p)
    if N = {I} then
        if post(I) − I ≠ ∅ → violation, else break
    else
        p := (p − {I}) ∪ N
        I := [init]_p
    end if
end while
if I is marked then
    Signal safety violation
end if
```

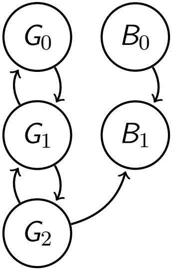
BFH - new algorthim (split)

function SPLIT(X : block, p : partition)
    $N = \{X\}$
    for all Y : block $\in p$ **do**
        $M := \emptyset$
        for all W : state $\in N$ **do**
            $W_1 = W \cap pre(Y)$
            if $W_1 = W$ or $W_1 = \emptyset$ **then**
                $M := M \cup \{W\}$
            else
                $M := M \cup \{W_1, W - W_1\}$
            end if
        end for
    end for
    return N
end function

BFH - example

Init

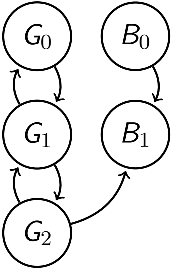
$$I = \{B\}$$
$$p = \{B, Bad\}$$
$$init = G_0$$



BFH - example

Iteration 1

$$N = split(I, p) = ???$$



BFH - example

Iteration 1 - split(1)

$X = B, p = \{B, Bad\}$

$N = \{B\}$

foreach $Y \in p \to$

$Y = B, M = \emptyset$

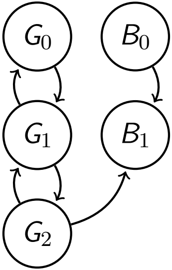
foreach $W \in N \to$

$W = B$

$W_1 = W \cap pre(Y) = \{B\}$

$\to M := M \cup \{W\} = \emptyset \cup B = \{B\}$

$N := M = \{B\}$



BFH - example

Iteration 1 - split(2)

$X = B, p = \{B, Bad\}$

$N = \{B\}$

foreach $Y \in p \rightarrow$

$Y = Bad, M = \emptyset$

foreach $W \in N \rightarrow$

$W = B$

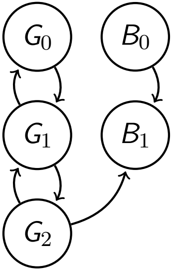
$W_1 = W \cap pre(Y) = B$

Y is marked $\rightarrow$ B is marked

$\rightarrow M := M \cup \{W\} = \emptyset \cup \{B\} = \{B\}$

$N := M = \{B\}$

$return(B)$



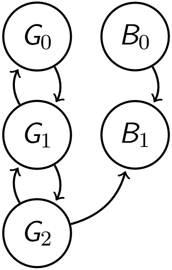
BFH - example

Iteration 1

$N = split(I, p) = \{B\}$

$N = \{I\} \rightarrow post(I) - \{I\} = \{Bad\} \neq \emptyset$

$\rightarrow$ raise safety violation !



BFH - complexity

$$(M + I + 2E) * \frac{n^2 + 3n}{2} + n * D$$

where

- n : number of BR iterations
- M : number of image iterations
- I : number of intersection operations
- D : number of set difference operations
- E : number of equality check
- U : number of union operations

Experimental comparisons

Experimental comparisons

Experimental comparisons

Lower bounds

- BR : $n * (M + U + D + 2E + I)$
- PT : $n * (2M + D + I + E)$
- LY : $(n - 1) * (5M + 4I + 3D + 4E)$
- BFH : $(M + I + 2E) * \frac{n^2+3n}{2} + n * D$