Bisimulation Minimization and Symbolic Model Checking

Sylvain Julmy

December 12, 2017

Bisimulation minimization

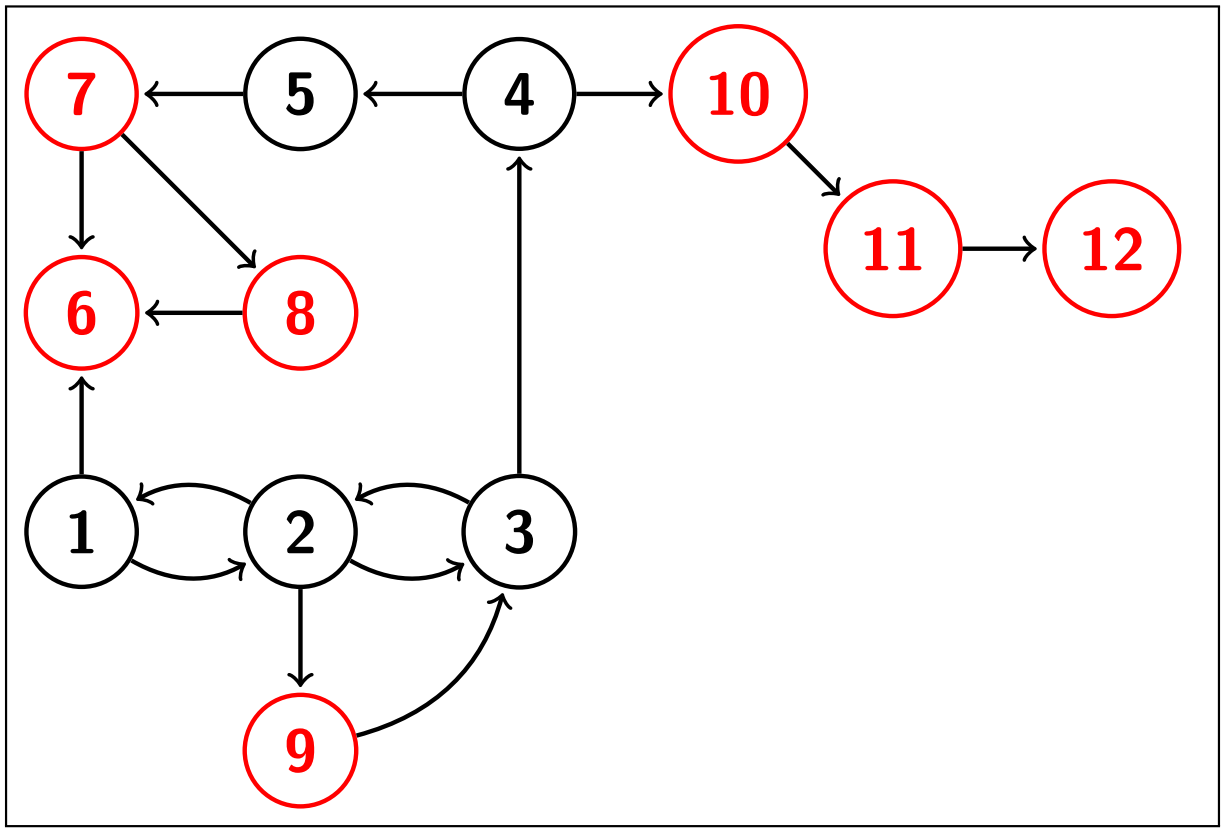

Figure: Initial state

Bisimulation minimization

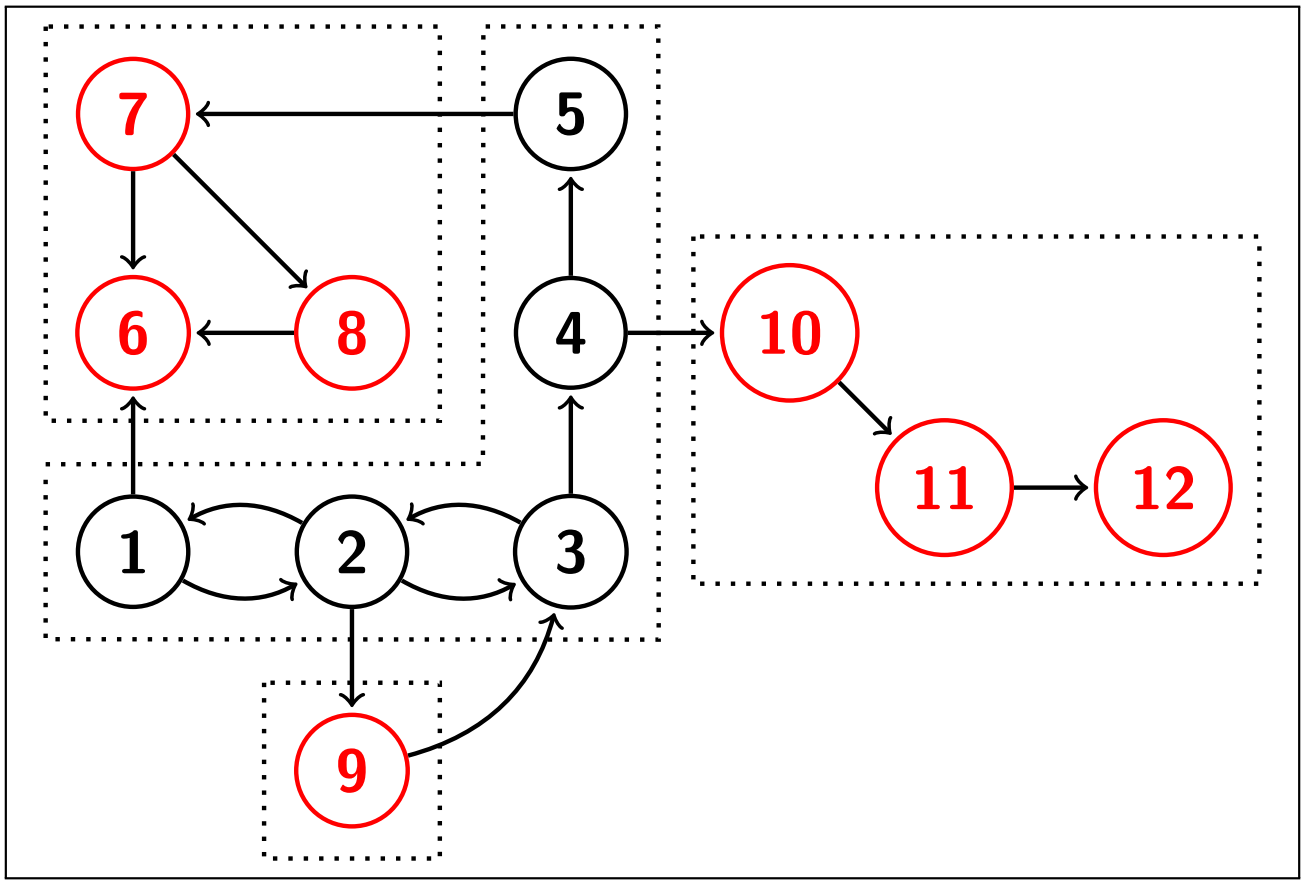

Figure: Initial partition block

Bisimulation minimization

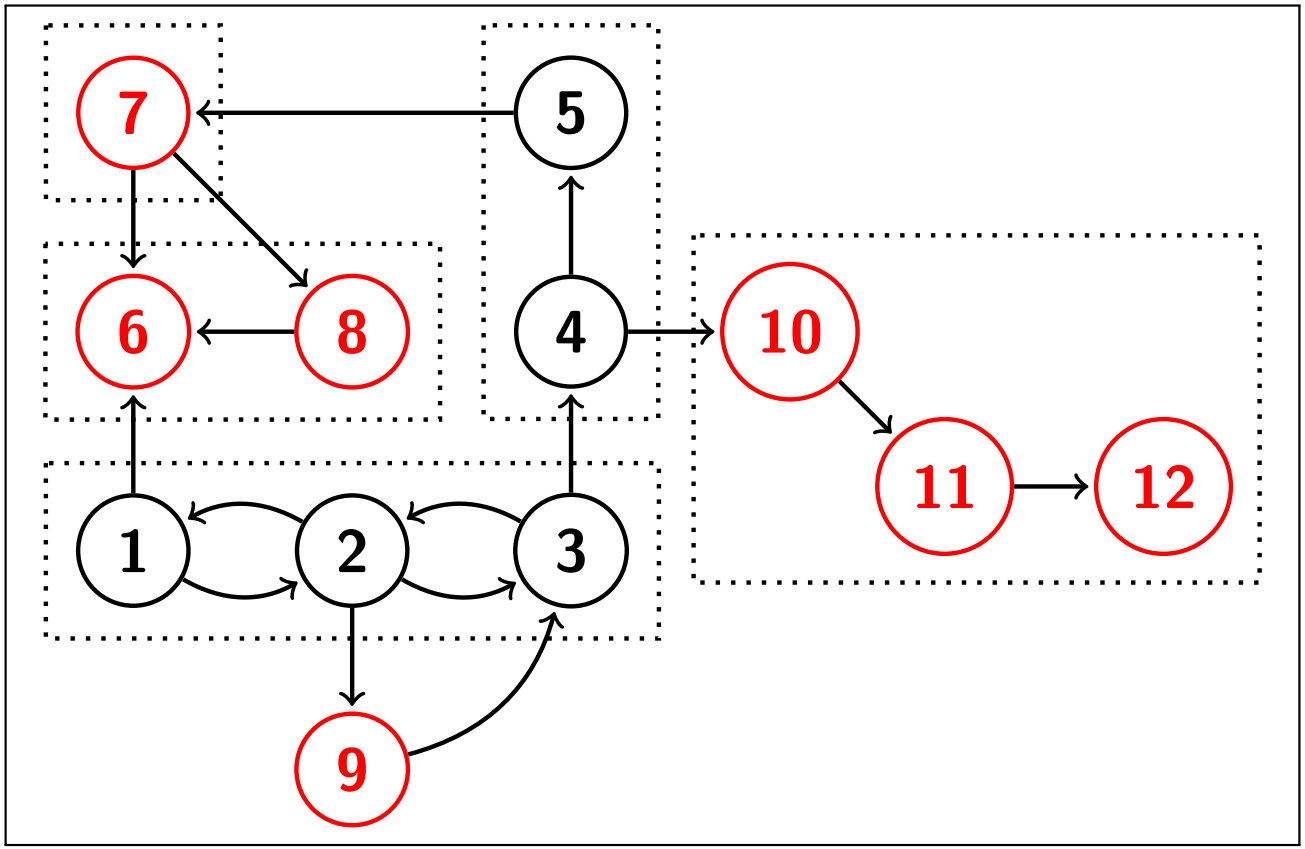

Figure: Computation of equivalence classes

Bisimulation minimization

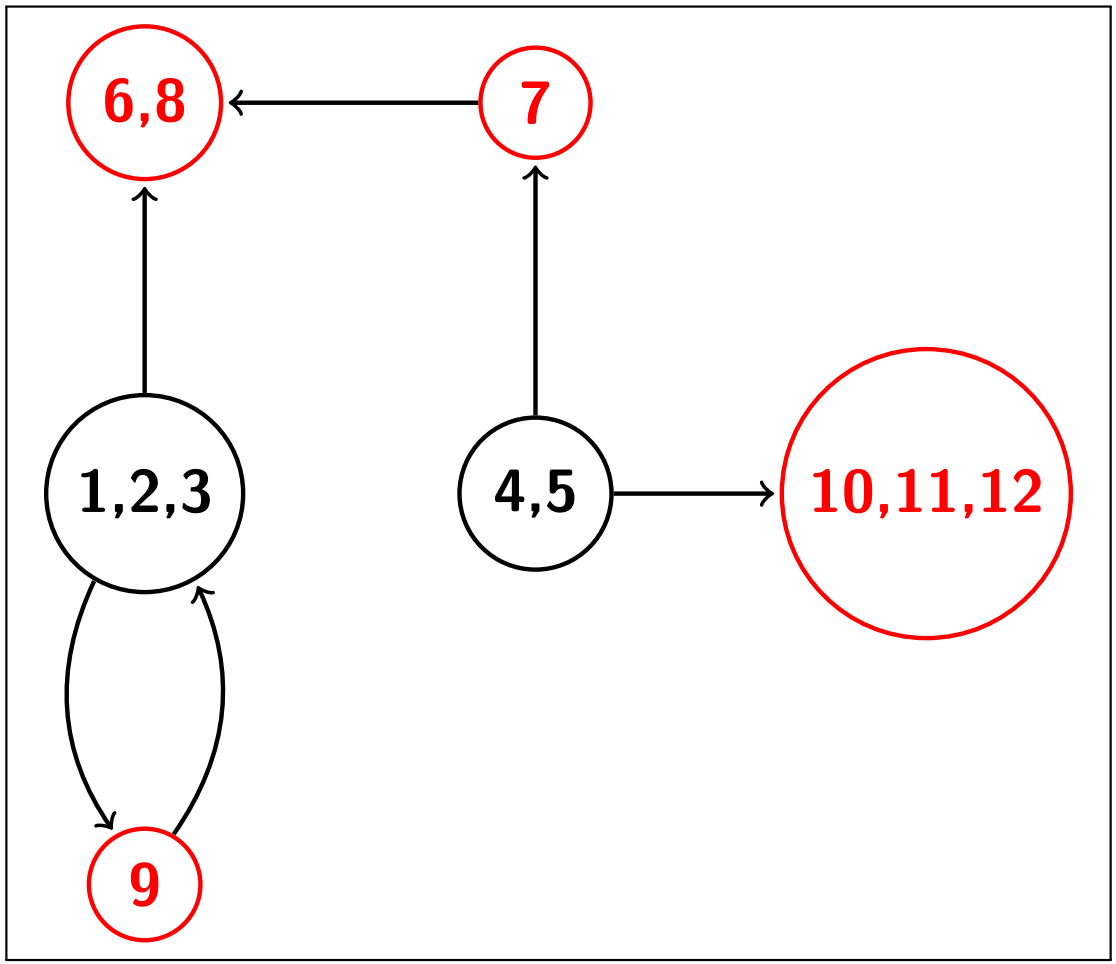

Figure: Final system to model check

BFH

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 - Algorithm

1: $S := \emptyset$ List of stable block
2: $R := \{[init]_p\}$ List of reachable block
3: **while** $R \neq S$ **do**
4: Select a reachable, but unstable block X
5: Stabilize X w.r.t. every block in the partition
6: **if** No new blocks are created **then**
7: Add X to S
8: Block reachable from X are added to R
9: **else**
10: Add new the new blocks to the partition
11: Update the initial block
12: Remove from S the blocks that becomes unstable
13: **end if**
14: **end while**

BFH - New Algorithm

1: $l := [init]_p$
2: Mark the bad block
3: **while** l is not marked **do**
4: Stabilize l
5: **if** No new blocks are created **then**
6: **if** $post_p(l) \setminus \{l\} = \emptyset$ **then**
7: Signal safety violation
8: **else**
9: Break
10: **end if**
11: **else**
12: **end if**
13: **end while**
14: **if** l is marked **then**
15: Signal safety violation
16: **end if**

BFH - Termination

As in LY, BFH could terminate when a second block becomes reachable.

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.