

## hifrog - To do #7860

### follow up in EUF summaries (Grisha's idea)

07/09/2018 22:15 - Sepideh Asadi

<b>Status:</b>	New	<b>Start date:</b>	07/09/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Sepideh Asadi	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>Description</b>			
<p>The point is about the loop in function func which calculates a new value for variable b. This loop is not needed for verifying any of assertions. It cannot be removed by slicing because variable b is used later in the program. But it can be identified by our summarization procedure.</p> <p>So idea is to 1) skip the loop at all, and thus treat b nondeterministically, 2) solve the whole BMC formula with EUF and get the UNSAT result, 3) interpolate and analyze the EUF summary for func. If variable b does not appear in the summary, then we can safely remove the loop and trust the summary. If b appears in the summary, then the result is not trustable, and we go back to the encoding of main program with loop, (unroll the loop perhaps, lazily?) and re-check. If the result is SAT, then of course we also need to go back to the unrolling.</p> <p><b>Note:</b> if you replace the assignment to b after the loop by nondet() and the formula is UNSAT, then you can trust the summary. But if you create an under-approximation of the loop (e.g., unroll it 1 / 2 / 5 / 10 times) and the formula is UNSAT then the summary is not trustable any more, and you may try the above trick.</p>			

### History

#1 - 07/09/2018 22:17 - Sepideh Asadi

- Assignee set to Sepideh Asadi