

hifrog - Bug #4678

hifrog: ../../src/common/FastRational.h:768: FastRational FastRational::inverse() const: Assertion `num != 0' failed.

06/03/2017 16:57 - Karine Even Mendoza

Status:	Closed	Start date:	06/03/2017
Priority:	Normal	Due date:	
Assignee:	Antti Hyvärinen	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
<pre>karinek@karinek-VirtualBox:~/workspace/tools/hi-bench\$ karinek@karinek-VirtualBox:~/workspace/tools/hi-bench\$../../hifrog_theory_ref/hifrog/trunk/cprover/src/funfrog/hifrog --logic qflra main-bench/arithmetic/SimpleArith_Real_3.c --unwind 1 --no-ityp Loading `main-bench/arithmetic/SimpleArith_Real_3.c' ... Parsing main-bench/arithmetic/SimpleArith_Real_3.c file <command-line> line 0: <command-line>:0:0: warning: "__STDC_VERSION__" redefined <built-in>: note: this is the location of the previous definition Converting Type-checking SimpleArith_Real_3 file main-bench/arithmetic/SimpleArith_Real_3.c line 2 function main: function `nondetFloat' is not declared file main-bench/arithmetic/SimpleArith_Real_3.c line 9 function main: function `assert' is not declared Generating GOTO Program Adding CPROVER library Function Pointer Removal Partial Inlining Generic Property Instrumentation LOAD Time: 0.208 sec. Checking claims in program... ; Warning: disabling SATElite preprocessing to track proof Last assertion location: 26 / 34 (3) • INLINING function: __CPROVER_initialize • INLINING function: main Processing a deferred function: __CPROVER_initialize Processing a deferred function: main ◦ WARNING: no body for function nondetFloat SYMEX TIME: 0.004 All SSA steps: 50 Ignored SSA steps after slice: 34 SLICER TIME: 0 hifrog: ../../src/common/FastRational.h:768: FastRational FastRational::inverse() const: Assertion `num != 0' failed. Aborted (core dumped) ===== On Benchmarks: --logic qflra main-bench/arithmetic/SimpleArith_Real_3.c --unwind 1 --no-ityp</pre>			

History

#1 - 07/03/2017 18:19 - Karine Even Mendoza

- Assignee set to Antti Hyvärinen

Any idea why OpenSMT fails?

Antti, if you can state the reason and then pass it to me, as what we did with r < rs bug.

Thanks!

#2 - 08/03/2017 17:05 - Antti Hyvärinen

Is there an explicit division by zero in the code?

#3 - 13/03/2017 18:24 - Antti Hyvärinen

- Status changed from New to Feedback

Ok, I "fixed" this one in a way. It's about the floats being given in the form "3.593100e+0" but the "e-notation" is not supported by OpenSMT, so I just throw an exception in that case. If the exception is not caught, then you can see where it originates.

Another question is whether this e-notation should be supported by opensmt, and at the moment it is not. Let me know what you think.

#4 - 14/03/2017 15:00 - Karine Even Mendoza

Thanks Antti, and yes this is better.

Can you add also the wrong data?

terminate called after throwing an instance of 'opensmt::strConvException' on 3.593100e+0

is it possible?

#5 - 01/04/2017 00:00 - Karine Even Mendoza

- Status changed from Feedback to Closed

See to be fine now.

CONVERSION TIME: 0
SOLVER TIME: 0.003
RESULT: SAT - doesn't hold

WARNING: Use over approximation. Cannot create an error trace.
Use --logic with Different Logic to Try Creating an Error Trace.

VERIFICATION FAILED

A bug found.
WARNING: Possibly due to the Theory conversion.
Initial unwinding bound: 1
Total number of steps: 1
TOTAL TIME FOR CHECKING THIS CLAIM: 0.006
#X: Done.