# OpenSMT 2 - Bug #3491

## Segmentation fault in opensmt when using function summaries

06/09/2016 23:44 - Karine Even Mendoza

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 06/09/2016 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Antti Hyvärinen | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |

**Description**

Code to reproduce the bug: hi-bench/main-bench/funfrog_regression/01_choice/main2_ok.c

When running the code for the first time we get unsat, on the second time we get seg. fault from opensmt.

There are several strange things going on:

1. When using the opensmt terminal the smt code runs ok (returns unsat for both executions). This is also the result when using Z3 website.
2. When using function summaries the most inner function is simply "true", but actually no code really removed (we just add another partition that is "true") - maybe something related to the way the partition is simplified?
3. The smt code when using summaries contains the original code, a partition that is true and a duplicated code (not sure it is related to the seg fault, but it can be a hint). It also says we are not using correctly the function summaries.

The original code (without function summaries): ==========
```
(set-logic QF_LRA)
(declare-fun |funfrog::?callstart_symbol#3| () Bool)
(declare-fun |funfrog::?callend_symbol#3| () Bool)
(declare-fun |c::getchar::1::x!0#2| () Real)
(declare-fun |symex::nondet0| () Real)
(declare-fun |funfrog::c::getchar::?retval_tmp#1| () Real)
(declare-fun |funfrog::c::getchar::?retval#1| () Real)
(declare-fun |funfrog::?callstart_symbol#2| () Bool)
(declare-fun |funfrog::?callend_symbol#2| () Bool)
(declare-fun |funfrog::?error_symbol#1| () Bool)
(declare-fun |goto_symex::guard#1| () Bool)
(declare-fun |goto_symex::guard#2| () Bool)
(declare-fun |goto_symex::guard#3| () Bool)
(declare-fun |goto_symex::guard#4| () Bool)
(declare-fun |c::main::1::x!0#2| () Real)
(declare-fun |c::main::1::t!0#11| () Real)
(declare-fun .oite0 () Real)
(declare-fun |c::main::1::t!0#12| () Real)
(declare-fun .oite1 () Real)
(declare-fun |c::main::1::t!0#13| () Real)
(declare-fun .oite2 () Real)
(declare-fun |c::main::1::t!0#14| () Real)
(declare-fun .oite3 () Real)
(declare-fun |funfrog::?callstart_symbol#1| () Bool)
(declare-fun |funfrog::?callend_symbol#1| () Bool)
(assert
(and
; XXX Partition: 0
(and (= |c::getchar::1::x!0#2| |symex::nondet0|) (= |c::getchar::1::x!0#2| |funfrog::c::getchar::?retval_tmp#1|) (=
|funfrog::c::getchar::?retval_tmp#1| |funfrog::c::getchar::?retval#1|) (or (not |funfrog::?callend_symbol#3|) (and (<= (- 1)
|c::getchar::1::x!0#2|) (and |funfrog::?callstart_symbol#3| (<= (- 255) (* |c::getchar::1::x!0#2| (- 1)))))))
; XXX Partition: 1
(and (= |funfrog::c::getchar::?retval#1| |c::main::1::x!0#2|) (= |goto_symex::guard#1| (= 1 |c::main::1::x!0#2|)) (=
|goto_symex::guard#2| (= |c::main::1::x!0#2| 2)) (= |goto_symex::guard#3| (= |c::main::1::x!0#2| 3)) (= |goto_symex::guard#4| (=
|c::main::1::x!0#2| (- 2))) (= |c::main::1::t!0#11| .oite0) (= |c::main::1::t!0#12| .oite1) (= |c::main::1::t!0#13| .oite2) (= |c::main::1::t!0#14|
.oite3) (= |funfrog::?callstart_symbol#2| |funfrog::?callstart_symbol#3|) (= (not (or (not (and |funfrog::?callend_symbol#3|
|funfrog::?callstart_symbol#2|)) (not (= (- 2) |c::main::1::t!0#14|)))) |funfrog::?error_symbol#1|) (or (not |funfrog::?callend_symbol#2|)
(and |funfrog::?callend_symbol#3| |funfrog::?callstart_symbol#2|)))
```

```
; XXX Partition: 2
(or |funfrog::?callstart_symbol#1| (not |funfrog::?callend_symbol#1|))
; XXX Partition: 3
(and |funfrog::?error_symbol#1| |funfrog::?callstart_symbol#1| (= |funfrog::?callend_symbol#1| |funfrog::?callstart_symbol#2|))
; XXX oite symbol: .oite0
(and (or (not |goto_symex::guard#4|) (= (- 2) .oite0)) (or |goto_symex::guard#4| (= 10 .oite0)))
; XXX oite symbol: .oite1
(and (or (not |goto_symex::guard#3|) (= 5 .oite1)) (or |goto_symex::guard#3| (= |c::main::1::t!0#11| .oite1)))
; XXX oite symbol: .oite2
(and (or (not |goto_symex::guard#2|) (= 3 .oite2)) (or |goto_symex::guard#2| (= |c::main::1::t!0#12| .oite2)))
; XXX oite symbol: .oite3
(and (or (not |goto_symex::guard#1|) (= 1 .oite3)) (or |goto_symex::guard#1| (= |c::main::1::t!0#13| .oite3)))
))
(check-sat)

The code when using function summaries: ================================
(set-logic QF_LRA)
(declare-fun |funfrog::?callstart_symbol#3| () Bool)
(declare-fun |funfrog::?callend_symbol#3| () Bool)
(declare-fun |funfrog::?callstart_symbol#2| () Bool)
(declare-fun |funfrog::?callend_symbol#2| () Bool)
(declare-fun |funfrog::?error_symbol#1| () Bool)
(declare-fun |goto_symex::guard#1| () Bool)
(declare-fun |goto_symex::guard#2| () Bool)
(declare-fun |goto_symex::guard#3| () Bool)
(declare-fun |goto_symex::guard#4| () Bool)
(declare-fun |c::main::1::x!0#2| () Real)
(declare-fun |funfrog::c::getchar::?retval#1| () Real)
(declare-fun |c::main::1::t!0#11| () Real)
(declare-fun .oite0 () Real)
(declare-fun |c::main::1::t!0#12| () Real)
(declare-fun .oite1 () Real)
(declare-fun |c::main::1::t!0#13| () Real)
(declare-fun .oite2 () Real)
(declare-fun |c::main::1::t!0#14| () Real)
(declare-fun .oite3 () Real)
(declare-fun |funfrog::?callend_symbol#1| () Bool)
(declare-fun |c::getchar::1::x!0#2| () Real)
(declare-fun |symex::nondet0| () Real)
(declare-fun |funfrog::c::getchar::?retval_tmp#1| () Real)
(declare-fun .oite4 () Real)
(declare-fun .oite5 () Real)
(declare-fun .oite6 () Real)
(declare-fun .oite7 () Real)
(assert
(and
; XXX Partition: 0
true
; XXX Partition: 1
(and (= |c::main::1::x!0#2| |funfrog::c::getchar::?retval#1|) (= |goto_symex::guard#1| (= 1 |c::main::1::x!0#2|)) (=
|goto_symex::guard#2| (= |c::main::1::x!0#2| 2)) (= |goto_symex::guard#3| (= |c::main::1::x!0#2| 3)) (= |goto_symex::guard#4| (=
|c::main::1::x!0#2| (- 2))) (= |c::main::1::t!0#11| .oite0) (= |c::main::1::t!0#12| .oite1) (= |c::main::1::t!0#13| .oite2) (= |c::main::1::t!0#14|
.oite3) (= |funfrog::?callstart_symbol#2| |funfrog::?callstart_symbol#3|) (= (not (or (not (and |funfrog::?callend_symbol#3|
|funfrog::?callstart_symbol#2|)) (not (= (- 2) |c::main::1::t!0#14|)))) |funfrog::?error_symbol#1|) (or (not |funfrog::?callend_symbol#2|)
(and |funfrog::?callend_symbol#3| |funfrog::?callstart_symbol#2|)))
; XXX Partition: 2
(and |funfrog::?error_symbol#1| (= |funfrog::?callend_symbol#1| |funfrog::?callstart_symbol#2|))
; XXX Partition: 3
(and (= |c::getchar::1::x!0#2| |symex::nondet0|) (= |c::getchar::1::x!0#2| |funfrog::c::getchar::?retval_tmp#1|) (=
|funfrog::c::getchar::?retval#1| |funfrog::c::getchar::?retval_tmp#1|) (or (not |funfrog::?callend_symbol#3|) (and (<= (- 1)
|c::getchar::1::x!0#2|) (and |funfrog::?callstart_symbol#3| (<= (- 255) (* (- 1) |c::getchar::1::x!0#2|))))))
; XXX Partition: 4
(and (= |c::main::1::x!0#2| |funfrog::c::getchar::?retval#1|) (= |goto_symex::guard#1| (= 1 |c::main::1::x!0#2|)) (=
|goto_symex::guard#2| (= |c::main::1::x!0#2| 2)) (= |goto_symex::guard#3| (= |c::main::1::x!0#2| 3)) (= |goto_symex::guard#4| (=
|c::main::1::x!0#2| (- 2))) (= |funfrog::?callstart_symbol#2| |funfrog::?callstart_symbol#3|) (= (not (or (not (and
|funfrog::?callend_symbol#3| |funfrog::?callstart_symbol#2|)) (not (= (- 2) |c::main::1::t!0#14|)))) |funfrog::?error_symbol#1|) (or (not
|funfrog::?callend_symbol#2|) (and |funfrog::?callend_symbol#3| |funfrog::?callstart_symbol#2|)) (= |c::main::1::t!0#11| .oite4) (=
```

```
|c::main::1::t!0#12| .oite5) (= |c::main::1::t!0#13| .oite6) (= |c::main::1::t!0#14| .oite7))
; XXX Partition: 5
(and |funfrog::?error_symbol#1| (= |funfrog::?callend_symbol#1| |funfrog::?callstart_symbol#2|))
; XXX oite symbol: .oite0
(and (or (not |goto_symex::guard#4|) (= (- 2) .oite0)) (or |goto_symex::guard#4| (= 10 .oite0)))
; XXX oite symbol: .oite1
(and (or (not |goto_symex::guard#3|) (= 5 .oite1)) (or |goto_symex::guard#3| (= |c::main::1::t!0#11| .oite1)))
; XXX oite symbol: .oite2
(and (or (not |goto_symex::guard#2|) (= 3 .oite2)) (or |goto_symex::guard#2| (= |c::main::1::t!0#12| .oite2)))
; XXX oite symbol: .oite3
(and (or (not |goto_symex::guard#1|) (= 1 .oite3)) (or |goto_symex::guard#1| (= |c::main::1::t!0#13| .oite3)))
; XXX oite symbol: .oite4
(and (or (not |goto_symex::guard#4|) (= (- 2) .oite4)) (or |goto_symex::guard#4| (= 10 .oite4)))
; XXX oite symbol: .oite5
(and (or (not |goto_symex::guard#3|) (= 5 .oite5)) (or |goto_symex::guard#3| (= |c::main::1::t!0#11| .oite5)))
; XXX oite symbol: .oite6
(and (or (not |goto_symex::guard#2|) (= 3 .oite6)) (or |goto_symex::guard#2| (= |c::main::1::t!0#12| .oite6)))
; XXX oite symbol: .oite7
(and (or (not |goto_symex::guard#1|) (= 1 .oite7)) (or |goto_symex::guard#1| (= |c::main::1::t!0#13| .oite7)))
))
(check-sat)
```

I also trace a similar problem in: (maybe use it, it the code is easier to debug/smaller?)
hi-bench/main-bench/funfrog_regression/03_simple/main4.c
hi-bench/main-bench/funfrog_regression/06_globals/main2.c
hi-bench/main-bench/funfrog_regression/06_globals/main3.c
hi-bench/main-bench/funfrog_regression/06_globals/main4.c

If one of these is better for testing, I'll add the smt code of it (please let me know if needed)

### History

**#1 - 06/09/2016 23:45 - Karine Even Mendoza**

*- Assignee set to Antti Hyvärinen*

**#2 - 12/09/2016 14:14 - Antti Hyvärinen**

*- Status changed from New to Resolved*

*- % Done changed from 0 to 100*

All mentioned files seem to be running without problems currently on Hifrog / master and opensmt / master.