

Klever - Feature #9099

Update CPALockator

07/12/2018 04:43 PM - Evgeny Novikov

Status: Closed	Start date: 08/07/2018
Priority: Urgent	Due date: 08/07/2018
Assignee: Pavel Andrianov	% Done: 0%
Category: Addons	Estimated time: 0.00 hour
Target version: 1.1	
Published in build:	
Description Pavel reported that there is a new version of CPALockator that is better in everything than the old one. So, let's update it within Klever after it will pass tests for rule specification <i>sync:race</i> (perhaps CPALockator or tests should be fixed).	
Related issues: Precedes Klever - Feature #9193: Continue sorting out sync:race tests New 08/08/2018 08/08/2018	

History

#1 - 07/26/2018 10:30 AM - Pavel Andrianov

- Status changed from New to Closed

All tests successfully pass. Couple of them required small fixes.

Note, that FunctionPointerTest pass only if function pointers are enabled, and PathIteratorTest shows wrong error trace (this is a problem of the analysis now, not easy to resolve). The new revision is 28815.

#2 - 07/26/2018 02:23 PM - Evgeny Novikov

- Status changed from Closed to Resolved

The issue can't be closed until CI will pass tests.

Pavel Andrianov wrote:

All tests successfully pass. Couple of them required small fixes.

Is there a branch with test fixes?

Note, that FunctionPointerTest pass only if function pointers are enabled, and PathIteratorTest shows wrong error trace (this is a problem of the analysis now, not easy to resolve). The new revision is 28815.

Before all *sync:race* tests passed while function pointers weren't enabled like now, so, I guess that [FunctionPointerTest](#) was wrong. Did you fix it?

What is the branch in the CPAchecker repository?

#3 - 07/26/2018 03:49 PM - Pavel Andrianov

Is there a branch with test fixes?

No, the fixes were in CPAchecker itself.

Before all *sync:race* tests passed while function pointers weren't enabled like now, so, I guess that FunctionPointerTest was wrong.

By default function pointers are enabled. My statement about them was only like a note, because I often disable them manually for debugging, and then I will be surprised, why the test does not pass.

Anyway, your question helps to found a bug in CPAchecker. I found, that the error trace, which was shown, should be found even when the function pointers are disabled. Finally, the bug is fixed and the test pass always.

Return to your question: seems yes, the test is now incorrect.

I am going to fix it, or even better to add one more, because this test is also useful. Is it the same issue?

What is the branch in the CPAChecker repository?

CPALockator-combat-mode

#4 - 07/26/2018 04:37 PM - Evgeny Novikov

Pavel Andrianov wrote:

Before all `sync:race` tests passed while function pointers weren't enabled like now, so, I guess that `FunctionPointerTest` was wrong.

By default function pointers are enabled. My statement about them was only like a note, because I often disable them manually for debugging, and then I will be surprised, why the test does not pass.

Anyway, your question helps to found a bug in CPAChecker. I found, that the error trace, which was shown, should be found even when the function pointers are disabled. Finally, the bug is fixed and the test pass always.

Return to your question: seems yes, the test is now incorrect.

I am going to fix it, or even better to add one more, because this test is also useful. Is it the same issue?

Test fixes and improvements are always welcome! It is great when tests are changed together with stuffs for which they are intended.

#5 - 07/26/2018 04:37 PM - Evgeny Novikov

- Status changed from *Resolved* to *Open*

Wait for a new test case.

#6 - 07/31/2018 01:35 PM - Pavel Andrianov

- Status changed from *Open* to *Resolved*

New tests and CPAChecker update are available in the branch 'new_tests_for_races'. All tests must pass. Not all tests are moved from `TODOs` folder, but one more step is performed anyway.

#7 - 08/01/2018 04:12 PM - Evgeny Novikov

- Status changed from *Resolved* to *Open*

I fixed the branch a bit. Then I noticed that almost all existing preset unsafe marks were not associated with unsafes for `sync:race` at all. The reason is that CPALockator outputs so-called error trace identifier differently, e.g. before it gave "int__ldv_global_var2" (one underscore belongs to the variable name) but now it looks like "int__ldv_global_var2". I assume that this is a bug and you need to get rid of one redundant underscore.

BTW, you can easily test this by populating fresh preset unsafe marks and ensuring that at least some of them associate well with unsafes.

#8 - 08/02/2018 10:53 AM - Pavel Andrianov

This is not a bug, vice versa, the change fixes an existed bug. The identifier is based on variable definition: `int var -> int_var`. However, in the first option there may be a such situation: `struct one_struct_nested var1 -> struct_one_struct_nested_var1` and `struct one_struct nested_var1 -> struct_one_struct_nested_var1`. So, two different variables are mapped to the same identifier. This bug was fixed by changing the identifier construction (adding one more underscore between type and name of variable).

#9 - 08/02/2018 09:08 PM - Evgeny Novikov

Pavel Andrianov wrote:

This is not a bug, vice versa, the change fixes an existed bug. The identifier is based on variable definition: `int var -> int_var`. However, in the first option there may be a such situation: `struct one_struct_nested var1 -> struct_one_struct_nested_var1` and `struct one_struct nested_var1 -> struct_one_struct_nested_var1`. So, two different variables are mapped to the same identifier. This bug was fixed by changing the identifier construction (adding one more underscore between type and name of variable).

Okay. But "struct a__b c" and "struct a b__c" will be mapped to the same name again. Indeed you don't need to map space to any identifier symbol. Are you going to fix that one more time? Perhaps you will also need to fix something within Core.

#10 - 08/03/2018 12:23 PM - Pavel Andrianov

Update CPALockator to 28911 (in branch `new_tests_for_races`). Attributes, including identifier name are printed into witness.

#11 - 08/03/2018 02:42 PM - Evgeny Novikov

As a proper decision of issues with witness attributes we at last switch to translation of witness attributes to error trace ones. So, there are not attributes encoded within witness file names anymore and everybody is welcome to specify specific attributes within witnesses!

I implemented this in branch *new_tests_for_races* that I renamed to *new-CPALockator*. Now wait from Pavel more pretty names and values for new attributes.

#12 - 08/03/2018 03:35 PM - Evgeny Novikov

- Status changed from *Open* to *Resolved*

Pavel updated the branch and it is tested now.

#13 - 08/06/2018 12:54 PM - Evgeny Novikov

Tests revealed that there are numerous problems with preset marks for *sync:race*. I fixed them all and started another testing round.

#14 - 08/06/2018 01:08 PM - Evgeny Novikov

- Follows Feature #9193: Continue sorting out *sync:race* tests added
- Start date changed from 07/12/2018 to 08/07/2018
- Due date set to 08/07/2018

#15 - 08/06/2018 01:08 PM - Evgeny Novikov

- Follows deleted (Feature #9193: Continue sorting out *sync:race* tests)

#16 - 08/06/2018 01:09 PM - Evgeny Novikov

- Follows Feature #9193: Continue sorting out *sync:race* tests added

#17 - 08/06/2018 01:09 PM - Evgeny Novikov

- Follows deleted (Feature #9193: Continue sorting out *sync:race* tests)

#18 - 08/06/2018 01:09 PM - Evgeny Novikov

- Precedes Feature #9193: Continue sorting out *sync:race* tests added

#19 - 08/06/2018 02:27 PM - Evgeny Novikov

- Status changed from *Resolved* to *Closed*

Tests passed, so, I merged the branch to master in [46be6ba39](#).