Ruby - Bug #14959

Writing a "link_to" method and a "url_helper" with a request parameter under certain "if else" statement in Rails helper crashes with KERN_INVALID_ADDRESS at 0x0000000000000000

08/02/2018 07:00 PM - y4m4p (Masahiro Yamashita)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-darwin14]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Disclaimer

Sorry about the problem being rails, and not ruby specific. I couldn't narrow down the problem and wording it correctly.
As it turns out that the problem derived from the ruby vm used in my Rails application, I decided to create the issue here.

Backstory

I have created this issue in rails/rails. (https://github.com/rails/rails/issues/33460)
I became aware that this issue derives from Shopify/bootsnap (https://github.com/Shopify/bootsnap/issues/182)
The member there told me that this is an issue with the ruby vm itself, so I have created this issue here.

Please refer for more information in the above link.

Reproduction

Reproduction Project:

https://github.com/y4m4p/rails_issue_33460

Project Dependency

program version
ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-darwin17]
Rails 5.2.0
Bundler 1.16.1
Mysql 5.7.21 Homebrew

Reproduction method

```
def helper_method
  if true
    link_to 'page', some_view_path(foo: 'true')
  else
    link_to 'somewhere', some_view_path(foo: 'false')
  end
end
```

Crash condition

Writing this helper method under "app/helper" directory in Rails which satisfies all of the next particular conditions will crash the Rails server application with
KERN_INVALID_ADDRESS at 0x0000000000000000.

- Writing a condition with

```
if true
```

```
  ...
else
  ...
```

OR

```
if false
  ...
else
  ...
```

- Write a path using the "link_to" helper and a "url_helper (xxx_path)" which contains a "request parameter" (xxx_path(some_parameter: 'x')).
- Writing the process under the statement that will never be used.
  For example, if you write the "if" sentence with "if true" then you must write your process under the "else" statement and vice versa.

Non crashing condition

- Writing the "if" condition with a instance variable or constant.

```
def helper_method
  condition = true
  if condition
    ...
  else
    ...
  end
end
```

- Do not write the specific "link_to" helper process under the statement that will never be used.

```
def helper_method
  if true
    link_to 'somewhere', web_pages_some_view_path(foo: 'true')
  else
    # link_to 'somewhere', web_pages_some_view_path(foo: 'false')  <= comment out
  end
end
```

- Do not write the request parameter for the "url_helper" path.

```
def helper_method
  if true
    link_to 'somewhere', web_pages_some_view_path(foo: 'true')
#<= Writing the request parameter in the used statement is OK
  else
    link_to 'somewhere', web_pages_some_other_view_path
  end
end
```

- Do not write the process in helper. But rather, writing the process in the view itself will not crash.

```
# app/view/web_pages/some_view.html.erb
<%=  if true
       link_to 'somewhere', web_pages_some_view_path(foo: 'true')
      else
        link_to 'somewhere', web_pages_some_view_path(foo: 'false')
      end %>
```

Expected

Should render correct views with link and not crash.
Even though I am writing the "if" condition with "true" which makes the "else" statement useless, but this should work as intended. (Just render the "true" statement.)

Related issues:

Related to Ruby - Bug #14897: Unexpected behavior of `if` in specific code	**Closed**
Related to Ruby - Bug #14960: Segmentation fault	**Closed**
Related to Ruby - Bug #14894: Segfault loading iseqs	**Closed**
Has duplicate Ruby - Bug #15163: [BUG] Segmentation fault at 0x0000000000000000	**Closed**
Is duplicate of Ruby - Bug #14553: Maybe this is too early for this but i enc...	**Closed**

History

#1 - 08/02/2018 07:01 PM - y4m4p (Masahiro Yamashita)

- Description updated

#2 - 08/02/2018 07:46 PM - rafaelfranca (Rafael França)

Minimal reproduction steps.

```
code = <<~CODE
module ApplicationHelper
  def broken_helper_in_application_helper
    if true
    else
      link_to 'somewhere', some_view_path(foo: 'false')
    end
  end
end
CODE
RubyVM::InstructionSequence.compile(code).to_binary
```

#3 - 08/03/2018 01:21 AM - sam.saffron (Sam Saffron)

also see: https://bugs.ruby-lang.org/issues/14894 probably the same issue.

#4 - 08/03/2018 07:36 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Bug #14897: Unexpected behavior of `if` in specific code added

#5 - 08/03/2018 07:38 AM - nobu (Nobuyoshi Nakada)

Thank you, @rafaelfranca.
It a fixed bug in the trunk, but doesn't seem backported to 2.5 yet.

#6 - 08/03/2018 07:39 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

#7 - 09/21/2018 08:17 AM - ujihisa (Tatsuhiro Ujihisa)

I'm not sure if it's going to be backported to current stable version 2.5 but let me clarify couple things just in case for future.

- This is issue does not depend neither rails nor bootsnap
- Yet another minimal reproducible code to segfault

```
RubyVM::InstructionSequence.compile(<<~CODE).to_binary
  f(x: 1) if false
CODE
```

Workaround for users who use old versions: Just replace "false" with "!true". "nil" or "unless true" will stil segfault.

```
# workaround
RubyVM::InstructionSequence.compile(<<~CODE).to_binary
  f(x: 1) if !true
CODE
```

#8 - 09/21/2018 08:36 AM - nobu (Nobuyoshi Nakada)

The same issue is already marked to backport in #14897.
Please wait for 2.5.2.

#9 - 09/27/2018 09:12 PM - naruse (Yui NARUSE)

- Has duplicate Bug #15163: [BUG] Segmentation fault at 0x0000000000000000 added

#10 - 12/05/2018 03:46 PM - matssigge (Mats Sigge)

nobu (Nobuyoshi Nakada) wrote:

> The same issue is already marked to backport in #14897.
> Please wait for 2.5.2.

I'm still seeing the problem in 2.5.3, so I'm guessing this was never backported? Will it be?

#11 - 12/09/2018 02:17 AM - nobu (Nobuyoshi Nakada)

Seems 2.5.4 has been fixed.

#12 - 12/09/2018 02:18 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #15365: backport r62776 (test_iseq.rb: skip iseq with coverage) added

#13 - 12/09/2018 02:18 AM - nobu (Nobuyoshi Nakada)

- Has duplicate deleted (Bug #15365: backport r62776 (test_iseq.rb: skip iseq with coverage))

#14 - 12/09/2018 02:22 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Bug #14553: Maybe this is too early for this but i encountered issues when using jit with rails added

#15 - 12/09/2018 02:23 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of deleted (Bug #14897: Unexpected behavior of `if` in specific code)

#16 - 12/09/2018 02:23 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #14897: Unexpected behavior of `if` in specific code added

#17 - 06/19/2019 07:03 PM - jeremyevans0 (Jeremy Evans)

- Related to Bug #14960: Segmentation fault added

#18 - 06/19/2019 07:31 PM - jeremyevans0 (Jeremy Evans)

- Related to Bug #14894: Segfault loading iseqs added

Files

ruby_2018-08-03-030208_y4m4p.crash	63.6 KB	08/02/2018	y4m4p (Masahiro Yamashita)