Ruby master - Bug #14897

Unexpected behavior of `if` in specific code

07/06/2018 03:22 AM - peitetsu (teppei takeo)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-darwin17]	**Backport:**	2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONE

Description

I found a strange behavior of if in the following code.

```
def seems_bug(obj)
  if obj || obj
    obj = obj
  else
    raise obj.inspect
  end
  obj
end

seems_bug('foo')
#=> RuntimeError: "foo"
```

This code is expected to return "foo", but the error on the else clause occurs.

The same error occurs in the following code.

```
def seems_bug(obj)
  if obj || any1
    any2 = any2
  else
    raise obj.inspect
  end
  obj
end

seems_bug('foo')
#=> RuntimeError: "foo"
```

Related issues:

Related to Ruby master - Bug #14974: "if" statement executes wrong branch	**Closed**
Related to Ruby master - Bug #14959: Writing a "link_to" method and a "url_he...	**Closed**
Related to Ruby master - Bug #15385: Ruby process hang in ensure	**Closed**
Has duplicate Ruby master - Bug #15021: Segfault when compiling certain code ...	**Closed**

Associated revisions

Revision 727ceb2a - 07/06/2018 04:52 AM - mame (Yusuke Endoh)

Fix a bug of peephole optimization

```
  if   L1
L0:
  jump L2
L1:
  ...
L2:
```

was wrongly optimized to:

```
  unless L2
L0:
```

```
L1:
  ...
L2:
```

To make it conservative, this optimization is now disabled when there is
any label between if and jump instructions.
Fixes [Bug #14897].

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63868 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63868 - 07/06/2018 04:52 AM - mame (Yusuke Endoh)

Fix a bug of peephole optimization

```
  if   L1
L0:
  jump L2
L1:
  ...
L2:
```

was wrongly optimized to:

```
  unless L2
L0:
L1:
  ...
L2:
```

To make it conservative, this optimization is now disabled when there is
any label between if and jump instructions.
Fixes [Bug #14897].

Revision 63868 - 07/06/2018 04:52 AM - mame (Yusuke Endoh)

Fix a bug of peephole optimization

```
  if   L1
L0:
  jump L2
L1:
  ...
L2:
```

was wrongly optimized to:

```
  unless L2
L0:
L1:
  ...
L2:
```

To make it conservative, this optimization is now disabled when there is
any label between if and jump instructions.
Fixes [Bug #14897].

Revision 5e7167f8 - 07/06/2018 08:01 AM - nobu (Nobuyoshi Nakada)

compile.c: remove unreachable jump only

- compile.c (iseq_peephole_optimize): remove unreachable jump instruction only. if it is labeled and referred from other instructions, it is reachable and must not be removed. [ruby-core:87830] [Bug #14897]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63870 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63870 - 07/06/2018 08:01 AM - nobu (Nobuyoshi Nakada)

compile.c: remove unreachable jump only

- compile.c (iseq_peephole_optimize): remove unreachable jump instruction only. if it is labeled and referred from other instructions, it is reachable and must not be removed. [ruby-core:87830] [Bug #14897]

Revision 63870 - 07/06/2018 08:01 AM - nobu (Nobuyoshi Nakada)

compile.c: remove unreachable jump only

- compile.c (iseq_peephole_optimize): remove unreachable jump instruction only. if it is labeled and referred from other instructions, it is reachable and must not be removed. [ruby-core:87830] [Bug #14897]

Revision 06289734 - 10/01/2018 12:02 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 63868,63870: [Backport #14897]

```
    Fix a bug of peephole optimization

    ```
      if   L1
    L0:
      jump L2
    L1:
      ...
    L2:
    ```

    was wrongly optimized to:

    ```
      unless L2
    L0:
    L1:
      ...
    L2:
    ```

    To make it conservative, this optimization is now disabled when there is
    any label between `if` and `jump` instructions.
    Fixes [Bug #14897].

    compile.c: remove unreachable jump only

    * compile.c (iseq_peephole_optimize): remove unreachable jump
      instruction only. if it is labeled and referred from other
      instructions, it is reachable and must not be removed.
      [ruby-core:87830] [Bug #14897]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@64893 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64893 - 10/01/2018 12:02 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 63868,63870: [Backport #14897]

```
Fix a bug of peephole optimization

```
  if   L1
L0:
  jump L2
L1:
  ...
L2:
```

was wrongly optimized to:

```
  unless L2
L0:
L1:
  ...
L2:
```

To make it conservative, this optimization is now disabled when there is
any label between `if` and `jump` instructions.
Fixes [Bug #14897].

compile.c: remove unreachable jump only

* compile.c (iseq_peephole_optimize): remove unreachable jump
  instruction only. if it is labeled and referred from other
  instructions, it is reachable and must not be removed.
```

```
  [ruby-core:87830] [Bug #14897]
```

History

#1 - 07/06/2018 04:03 AM - shyouhei (Shyouhei Urabe)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: DONTNEED, 2.4: DONTNEED, 2.5: REQUIRED

Seems obj = obj is (mistakenly) optimized out.

```
% ./miniruby --dump=i -v tmp.rb
ruby 2.6.0dev (2018-07-06 trunk 63854) [x86_64-darwin15]
== disasm: #<ISeq:<main>@tmp.rb:1 (1,0)-(10,16)> (catch: FALSE)
0000 putspecialobject             1                                   (   1)[Li]
0002 putobject                    :seems_bug
0004 putiseq                      seems_bug
0006 opt_send_without_block       <callinfo!mid:core#define_method, argc:2, ARGS_SIMPLE>, <callcache>
0009 pop
0010 putself                                                          (  10)[Li]
0011 putstring                    "foo"
0013 opt_send_without_block       <callinfo!mid:seems_bug, argc:1, FCALL|ARGS_SIMPLE>, <callcache>
0016 leave

== disasm: #<ISeq:seems_bug@tmp.rb:1 (1,0)-(8,3)> (catch: FALSE)
local table (size: 1, argc: 1 [opts: 0, rest: -1, post: 0, block: -1, kw: -1@-1, kwrest: -1])
[ 1] obj@0<Arg>
0000 getlocal_WC_0                obj@0                               (   2)[LiCa]
0002 branchif                     8
0004 getlocal_WC_0                obj@0
0006 branchif                     18
0008 putself                                                          (   5)[Li]
0009 getlocal_WC_0                obj@0
0011 opt_send_without_block       <callinfo!mid:inspect, argc:0, ARGS_SIMPLE>, <callcache>
0014 opt_send_without_block       <callinfo!mid:raise, argc:1, FCALL|ARGS_SIMPLE>, <callcache>
0017 pop
0018 getlocal_WC_0                obj@0                               (   7)[Li]
0020 leave                                                            (   8)[Re]
```

#2 - 07/06/2018 04:52 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Applied in changeset trunk|r63868.

Fix a bug of peephole optimization

```
  if   L1
L0:
  jump L2
L1:
  ...
L2:
```

was wrongly optimized to:

```
  unless L2
L0:
L1:
  ...
L2:
```

To make it conservative, this optimization is now disabled when there is
any label between if and jump instructions.
Fixes [Bug #14897].

#3 - 07/06/2018 04:53 AM - mame (Yusuke Endoh)

Good catch. It was a bug of peephole optimization. Fixed. Thank you.

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

- Has duplicate 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 added

#5 - 08/09/2018 06:02 AM - znz (Kazuhiro NISHIYAMA)

- Related to Bug #14974: "if" statement executes wrong branch added

#6 - 10/01/2018 12:02 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.3: DONTNEED, 2.4: DONTNEED, 2.5: REQUIRED to 2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONE

ruby_2_5 r64893 merged revision(s) 63868,63870.

#7 - 10/04/2018 07:27 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #15200: RubyVM::InstructionSequence can not compile to binary from method call with hash in 'if false' expression added

#8 - 10/04/2018 07:43 AM - nobu (Nobuyoshi Nakada)

- Has duplicate deleted (Bug #15200: RubyVM::InstructionSequence can not compile to binary from method call with hash in 'if false' expression)

#9 - 12/09/2018 02:23 AM - nobu (Nobuyoshi Nakada)

- Has duplicate deleted (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)

#10 - 12/09/2018 02:23 AM - nobu (Nobuyoshi Nakada)

- Related to 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 added

#11 - 12/10/2018 12:25 AM - shyouhei (Shyouhei Urabe)

- Related to Bug #15385: Ruby process hang in ensure added

#12 - 06/19/2019 06:53 PM - jeremyevans0 (Jeremy Evans)

- Has duplicate Bug #15021: Segfault when compiling certain code on Ruby 2.5.1 added