# Exotic method parameters bug

## Description

Consider the following code:

```ruby
def test(_, x, *_)
  x
end
test(1,2,3) # => nil
```

By some reason second parameter is nil when the rest of parameters declared as not used.

## Associated revisions

**Revision 62832 - 03/19/2018 03:16 AM - nobu (Nobuyoshi Nakada)**

- parse.y: fix duplicate kwrest
  - parse.y (_kwrest): add the variable name as an argument, as well as an internal variable.
  - parse.y (new_args_tail): now keyword rest argument variable is always placed between keyword arguments and block argument, so just reorder required and optional keyword arguments. no longer kwrest duplicates.

**Revision 9720136d - 03/19/2018 04:08 AM - nobu (Nobuyoshi Nakada)**

- compile.c: arg var index
  - compile.c (iseq_set_arguments): determine argument variable indexes by the order, not by just IDs. arguments begin with _ can be duplicate, so by-ID index may result in a wrong value. [ruby-core:86159] [Bug #14611]

**Revision 62833 - 03/19/2018 04:08 AM - nobu (Nobuyoshi Nakada)**

- compile.c: arg var index
  - compile.c (iseq_set_arguments): determine argument variable indexes by the order, not by just IDs. arguments begin with _ can be duplicate, so by-ID index may result in a wrong value. [ruby-core:86159] [Bug #14611]

**Revision 62833 - 03/19/2018 04:08 AM - nobu (Nobuyoshi Nakada)**

- compile.c: arg var index
  - compile.c (iseq_set_arguments): determine argument variable indexes by the order, not by just IDs. arguments begin with _ can be duplicate, so by-ID index may result in a wrong value. [ruby-core:86159] [Bug #14611]

## History

**#1 - 03/16/2018 03:11 PM - phluid61 (Matthew Kerwin)**

bogdan (Bogdan Gusiev) wrote:
By some reason second parameter is nil when the rest of parameters declared as not used.

Underscore as "not used" here is a convention of Rubocop; your code is no different from:

```ruby
def test(a, x, *a)
  x
end
```

I would say the bug is that it didn’t reject the def in the first place, because of the repeated parameter name.

#2 - 03/16/2018 03:34 PM - Hanmac (Hans Mackowiak)

phluid61 (Matthew Kerwin) not so fast, ruby allows duplicate "_" as parameters so you can say that you don't want them and ruby should just ignore them so you don't get a SyntaxError about duplicated argument name like your change did

That has nothing to do with rubocop

#3 - 03/17/2018 12:22 AM - phluid61 (Matthew Kerwin)

Hanmac (Hans Mackowiak) wrote:

phluid61 (Matthew Kerwin) not so fast, ruby allows duplicate "_" as parameters so you can say that you don't want them and ruby should just ignore them so you don't get a SyntaxError about duplicated argument name like your change did

That has nothing to do with rubocop

... really? When did that happen? I had a look in syntax.rdoc and other places but didn't see any mention of it.

(edit: turns out this is something I completely missed in the 1.9.3 era)

#4 - 03/17/2018 04:08 AM - shevegen (Robert A. Heiler)

I did not know that either.

I guess the problem was that bogdan was using variables with a leading "_".

With this:

```ruby
def x(a,a,b)
  puts 'HI'
end
```

I get in IRB:

```
SyntaxError ((irb):1: duplicated argument name
```

With that, however had:

```ruby
def x(_a,_a,_b)
  puts 'hi'
end
```

x(1,2,3)

```
# => hi
```

I get no error and the method call works.

I have no idea if this is a bug or a feature but to me, just from looking at the above, I would have assumed that _a is equal to _a just as a was (considered) equal to a above, in "def x(a,a,b)".

Perhaps it is because leading _ underscores are always assumed to be special. But ideally this should be documented somewhere nonetheless.

#5 - 03/19/2018 12:12 AM - nobu (Nobuyoshi Nakada)

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

#6 - 03/19/2018 04:08 AM - nobu (Nobuyoshi Nakada)
compile.c: arg var index

- compile.c (iseq_set_arguments): determine argument variable indexes by the order, not by just IDs. Arguments begin with _ can be duplicate, so by-ID index may result in a wrong value. [ruby-core:86159] [Bug #14611]

#7 - 12/24/2018 09:04 AM - mame (Yusuke Endoh)
- Related to Bug #15457: Weird behaviour when having multiple arguments of same name and splat operator added