Default argument lookup fails in Ruby 2.2 for circular shadowed variable names

The following code prints nil in Ruby 2.2.0-preview1 but worked in all previous version of Ruby back to 1.8.7:

```ruby
class Foo
  def foo; "abc" end

  # this default param should resolve at runtime to the #foo method call
  def run(foo = foo)
    p foo # print shadowed local var defaulting to attr value
  end
end

puts "Testing #{RUBY_VERSION}:"
Foo.new.run

# Ruby 2.2.0-preview1
# => nil

# Ruby 1.x 2.x etc
# => "abc"
```

My guess is this is happening because "foo" in "foo = foo" is resolving to the argument variable "foo", which currently has the value of nil. It would be equivalent to setting "qux = qux" in a method body, which has been the expected behavior for a long time.

I understand that shadowing variables is something you should probably never do, but unfortunately this code was already written and working for quite a while, so I figured it would be wise to file a bug report for the following reasons:

1. This seems like a breaking change in Ruby 2.2.0-preview1 that was not announced in the changelog. My guess is this change may have been unintentional, but if it was we need a changelog entry at the very least.
2. If this is newly expected behavior, I wanted to chime in that I don't think it makes much sense. I can't think of any time when a user would expect the default value of a "foo = foo" argument to be the same foo argument itself. That would be tautologically nil. Arguably, it doesn't make much sense inside of a method body either when there is a shadowed method that could be called instead.

**Related issues:**

- Related to Ruby master - Bug #9593: Keyword arguments default argument assign...  
  Closed 03/05/2014
- Related to Ruby master - Bug #10280: Regression while evaluating default argm...  
  Rejected 09/22/2014

**Associated revisions**

- Revision 98ea6275 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
  parse.y: warn circular argument reference
  git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48188 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

- Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
  parse.y: warn circular argument reference
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 48188 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
parse.y: warn circular argument reference

- parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]

Revision 6fe9b2b7 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48189 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 48189 - 10/29/2014 02:24 PM - nobu (Nobuyoshi Nakada)
parse.y: warn inside a block

- parse.y (gettable_gen): also warn circular argument reference even inside a block. [ruby-core:65990] [Bug #10314]

Revision 049bbd72 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var
• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 48190 - 10/29/2014 03:01 PM - nobu (Nobuyoshi Nakada)
parse.y: reset current_arg after block var

• parse.y (block_param_def): reset current_arg after block parameter definition, not to warn references in that block body. [ruby-core:65990] [Bug #10314]

Revision 0162e7e6 - 10/17/2019 04:32 PM - jeremyevans (Jeremy Evans)
Make circular argument reference a SyntaxError instead of a warning
Fixes [Bug #10314]

History
#1 - 10/02/2014 05:19 AM - ko1 (Koichi Sasada)
Loren Segal wrote:

1. This seems like a breaking change in Ruby 2.2.0-preview1 that was not announced in the changelog. My guess is this change may have been unintentional, but if it was we need a changelog entry at the very least.

I agree. How about to show a warning?

1. If this is newly expected behavior, I wanted to chime in that I don't think it makes much sense. I can't think of any time when a user would expect the default value of a "foo = foo" argument to be the same foo argument itself. That would be tautologically nil. Arguably, it doesn't make much sense inside of a method body either when there is a shadowed method that could be called instead.

I think this change is to make same as method body ("foo = foo" in method body).

I'm not sure your proposal is natural or not (matz issue?).
I like current behavior because it is simple (define a local variable foo after the sentencefoo="").
#2 - 10/02/2014 06:15 AM - hsbt (Hiroshi SHIBATA)
- Related to Bug #9593: Keyword arguments default argument assignment behaviour not consistent with optional argument added

#3 - 10/02/2014 06:15 AM - hsbt (Hiroshi SHIBATA)
- Related to Bug #10280: Regression while evaluating default arguments of a method added

#4 - 10/29/2014 10:41 AM - matz (Yukihiro Matsumoto)
- Status changed from Open to Rejected

In other part of Ruby, foo = foo makes foo to nil.
To get old behavior, try foo = foo().

Matz.

#5 - 10/29/2014 10:46 AM - ko1 (Koichi Sasada)
- Status changed from Rejected to Assigned
- Assignee set to nobu (Nobuyoshi Nakada)

Maybe we need warning() strongly (without verbose).

#6 - 10/29/2014 12:13 PM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

Applied in changeset r48188.

```
parse.y: warn circular argument reference
  • parse.y (gettable_gen): warn circular argument reference, for transition from 2.1 and earlier. [ruby-core:65990] [Bug #10314]
```

#7 - 12/20/2014 02:37 AM - tmm1 (Aman Karmani)
This change in behavior breaks rails 3.2, for example here:
https://github.com/github/rails/blob/3-2-github/activerecord/lib/active_record/associations/has_many_association.rb#L53-L55

In this case, I think the intended behavior is def has_cached_counter?(reflection = self.reflection) to access the reflection attributed defined in the superclass: https://github.com/github/rails/blob/3-2-github/activerecord/lib/active_record/associations/association.rb#L21

Is adding an explicit self, prefix necessary to preserve compatibility in 2.2 and forward?

#8 - 12/20/2014 02:45 AM - tmm1 (Aman Karmani)
I see, matz (Yukihiro Matsumoto) recommended foo = foo(). I guess this is a breaking change in ruby 2.2.

#9 - 12/20/2014 10:50 AM - hsbt (Hiroshi SHIBATA)

```
tmm1
```

Please request to backport has_many_association patch into Rails 3.2.

#10 - 12/23/2014 05:20 AM - tmm1 (Aman Karmani)
I opened a pull request to add ruby 2.2 compatibility to rails 3-2-stable: https://github.com/rails/rails/pull/18160.

In our large rails app, I only had to fix 5 function definitions, by adding parenthesis to 4 of them. The warnings made it very easy to find and fix the code.

#11 - 01/20/2015 02:54 AM - usa (Usaku NAKAMURA)
- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: DONTNEED, 2.1: DONTNEED

#12 - 01/26/2015 06:30 AM - funny_falcon (Yura Sokolov)
It looks like it is better to make it an error than a warning.
Nobu, what do you think about funny falcon's idea?

I'm in favor of that error.

Do you still want to implement this error?

Also, wasn't def run(foo = self.foo) a viable alternative to def run(foo = foo()) to avoid circular reference warnings?

Empty parens tend to trigger errors from tools like Rubocop.

Also, wasn't def run(foo = self.foo) a viable alternative to def run(foo = foo()) to avoid circular reference warnings?

Unless #foo is private.

Empty parens tend to trigger errors from tools like Rubocop.

Then you should report it to Rubocop, not here.

Attached is a patch to turn circular argument reference from a warning to a SyntaxError. Do we still want to do that?

I think it's OK now to make this error.

Matz.

Make circular argument reference a SyntaxError instead of a warning

Fixes [Bug #10314]

Files
circular-arg-ref-syntax-error-10314.patch 3.82 KB 08/11/2019 jeremyevans0 (Jeremy Evans)