

Ruby - Bug #19004

Complex can be nested by Complex.polar

09/10/2022 05:46 PM - msnm (Masahiro Nomoto)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	3.1.2	Backport: 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE
Description Complex.polar with one argument can return a "nested" Complex instance, whose real part is also a Complex one. <pre>puts RUBY_DESCRIPTION # ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [x86_64-linux] p Complex.polar(1+0i) # ((1+0i)+0i) p 5.times.inject(1) { num, _ Complex.polar(num) } # (((((1+0i)+0i)+0i)+0i)+0i)+0i</pre> In Ruby < 2.7 , it simply raises an error when the argument is an instance of Complex (i.e. obj.real? == false). <pre>puts RUBY_DESCRIPTION # ruby 2.6.10p210 (2022-04-12 revision 67958) [x86_64-linux] p Complex.polar(1+0i) # TypeError (not a real)</pre>		

Associated revisions

Revision 54cad3123a07583c90e85bcfc55ebd87124c1250 - 10/23/2022 03:59 AM - stevegeek (Stephen Ierodiaconou)

[Bug #19004] Complex.polar handles complex singular abs argument

Complex.polar accepts Complex values as arguments for the polar form as long as the value of the complex has no imaginary part (ie it is 'real'). In `f_complex_polar` this is handled by extracting the real part of the arguments. However in the case polar is called with only a single argument, the absolute value (`abs`), then the Complex is created without applying a check on the type of `abs`, meaning it is possible to create a Complex where the real part is itself an instance of a Complex. This change removes the short circuit for the single argument case meaning the real part extraction is performed correctly (by `f_complex_polar`).

Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that the real part of a complex argument is correctly extracted and used in the resulting Complex real and imaginary parts.

Revision 54cad3123a07583c90e85bcfc55ebd87124c1250 - 10/23/2022 03:59 AM - stevegeek (Stephen Ierodiaconou)

[Bug #19004] Complex.polar handles complex singular abs argument

Complex.polar accepts Complex values as arguments for the polar form as long as the value of the complex has no imaginary part (ie it is 'real'). In `f_complex_polar` this is handled by extracting the real part of the arguments. However in the case polar is called with only a single argument, the absolute value (`abs`), then the Complex is created without applying a check on the type of `abs`, meaning it is possible to create a Complex where the real part is itself an instance of a Complex. This change removes the short circuit for the single argument case meaning the real part extraction is performed correctly (by `f_complex_polar`).

Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that the real part of a complex argument is correctly extracted and used in the

resulting Complex real and imaginary parts.

Revision 54cad312 - 10/23/2022 03:59 AM - stevegeek (Stephen Ierodiaconou)

[Bug #19004] Complex.polar handles complex singular abs argument

Complex.polar accepts Complex values as arguments for the polar form as long as the value of the complex has no imaginary part (ie it is 'real'). In f_complex_polar this is handled by extracting the real part of the arguments. However in the case polar is called with only a single argument, the absolute value (abs), then the Complex is created without applying a check on the type of abs, meaning it is possible to create a Complex where the real part is itself an instance of a Complex. This change removes the short circuit for the single argument case meaning the real part extraction is performed correctly (by f_complex_polar).

Also adds an example to spec/ruby/core/complex/polar_spec.rb to check that the real part of a complex argument is correctly extracted and used in the resulting Complex real and imaginary parts.

Revision 597ce7966c38e4f7fc7368a860ac2d056de794ba - 11/05/2022 05:24 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 54cad3123a07583c90e85bcfc55ebd87124c1250: [Backport #19004]

```
[Bug #19004] `Complex.polar` handles complex singular `abs` argument
```

```
`Complex.polar` accepts Complex values as arguments for the polar form as long
as the value of the complex has no imaginary part (ie it is 'real'). In
`f_complex_polar` this is handled by extracting the real part of the arguments.
However in the case `polar` is called with only a single argument, the absolute
value (abs), then the Complex is created without applying a check on the type
of abs, meaning it is possible to create a Complex where the real part is itself
an instance of a Complex. This change removes the short circuit for the single
argument case meaning the real part extraction is performed correctly
(by f_complex_polar).
```

```
Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that
the real part of a complex argument is correctly extracted and used in the
resulting Complex real and imaginary parts.
```

```
---
complex.c | 13 ++++++-----
spec/ruby/core/complex/polar_spec.rb | 16 ++++++-----
2 files changed, 22 insertions(+), 7 deletions(-)
```

Revision 597ce7966c38e4f7fc7368a860ac2d056de794ba - 11/05/2022 05:24 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 54cad3123a07583c90e85bcfc55ebd87124c1250: [Backport #19004]

```
[Bug #19004] `Complex.polar` handles complex singular `abs` argument
```

```
`Complex.polar` accepts Complex values as arguments for the polar form as long
as the value of the complex has no imaginary part (ie it is 'real'). In
`f_complex_polar` this is handled by extracting the real part of the arguments.
However in the case `polar` is called with only a single argument, the absolute
value (abs), then the Complex is created without applying a check on the type
of abs, meaning it is possible to create a Complex where the real part is itself
an instance of a Complex. This change removes the short circuit for the single
argument case meaning the real part extraction is performed correctly
(by f_complex_polar).
```

```
Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that
the real part of a complex argument is correctly extracted and used in the
resulting Complex real and imaginary parts.
```

```
---
complex.c | 13 ++++++-----
spec/ruby/core/complex/polar_spec.rb | 16 ++++++-----
2 files changed, 22 insertions(+), 7 deletions(-)
```

Revision 597ce796 - 11/05/2022 05:24 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 54cad3123a07583c90e85bcfc55ebd87124c1250: [Backport #19004]

```
[Bug #19004] `Complex.polar` handles complex singular `abs` argument
```

```
`Complex.polar` accepts Complex values as arguments for the polar form as long
```

as the value of the complex has no imaginary part (ie it is 'real'). In `f_complex_polar` this is handled by extracting the real part of the arguments. However in the case `polar` is called with only a single argument, the absolute value (`abs`), then the `Complex` is created without applying a check on the type of `abs`, meaning it is possible to create a `Complex` where the real part is itself an instance of a `Complex`. This change removes the short circuit for the single argument case meaning the real part extraction is performed correctly (by `f_complex_polar`).

Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that the real part of a complex argument is correctly extracted and used in the resulting `Complex` real and imaginary parts.

```
---
complex.c | 13 +++++-----
spec/ruby/core/complex/polar_spec.rb | 16 ++++++
2 files changed, 22 insertions(+), 7 deletions(-)
```

History

#1 - 10/16/2022 07:55 PM - stevegeek (Stephen Ierodiaconou)

I thought to try fixing this issue as a first contribution to Ruby. I have a proposed fix here <https://github.com/ruby/ruby/pull/6568>

Details are on the PR, but in summary, I believe that the ability to pass a `Complex` as the `abs` argument is the expected behaviour (given other methods on `Complex` also accept `Complex` value as long as they are real, in the sense they have a zero imaginary part). However `Complex.polar` with a single argument creates the new `Complex` without first extracting the real part of the argument, so `Complex.polar(1+0.0i).real => (1+0i)`

The propose fix simply ensures that the single argument case is handled in the same way as the 2 argument case, so `Complex.polar(1+0.0i).real => 1`

#2 - 10/18/2022 07:45 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED

#3 - 10/23/2022 07:17 AM - stevegeek (Stephen Ierodiaconou)

- Status changed from Open to Closed

Applied in changeset [git|54cad3123a07583c90e85bcfc55ebd87124c1250](https://github.com/ruby/ruby/commit/54cad3123a07583c90e85bcfc55ebd87124c1250).

[Bug #19004] `Complex.polar` handles complex singular `abs` argument

`Complex.polar` accepts `Complex` values as arguments for the polar form as long as the value of the complex has no imaginary part (ie it is 'real'). In `f_complex_polar` this is handled by extracting the real part of the arguments. However in the case `polar` is called with only a single argument, the absolute value (`abs`), then the `Complex` is created without applying a check on the type of `abs`, meaning it is possible to create a `Complex` where the real part is itself an instance of a `Complex`. This change removes the short circuit for the single argument case meaning the real part extraction is performed correctly (by `f_complex_polar`).

Also adds an example to `spec/ruby/core/complex/polar_spec.rb` to check that the real part of a complex argument is correctly extracted and used in the resulting `Complex` real and imaginary parts.

#4 - 11/05/2022 05:25 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED to 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

ruby_3_1 597ce7966c38e4f7fc7368a860ac2d056de794ba merged revision(s) 54cad3123a07583c90e85bcfc55ebd87124c1250.