Ruby master - Feature #10378

[PATCH 0/3] It's better (1 + 0i).real? return true

10/13/2014 07:51 PM - gogotanaka (Kazuki Tanaka)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Right now, Complex#real? return false anytime.

I suppose #is_a?(Complex) is enough to check whether a object is Complex's object or not.
(I have to admire #real? is more useful than #is_a?(Complex))

But what we really want to know is whether a object whose class has Numeric as superclass is equal to real number or not.

Actually whichever is ok, modifying #real? or implementing as new method(e.g #real_num? ... :(

Anyway, I wanna method like that for Complex.

```
static VALUE
nucomp_real_p(VALUE self)
{
    get_dat1(self);
    if (rb_equal(dat->imag, INT2FIX(0))) {
        return Qtrue;
    }
    return Qfalse;
}
```

By the way, I can find two coding styles through ruby source code.

Which is prefer? it doesn't matter?

```
if(...)
    return ...
retrun ...
```

or

```
if(...) {
    return ...
}
retrun ...
```

History

#1 - 10/14/2014 05:30 PM - gogotanaka (Kazuki Tanaka)

I can find more useful function f_zero_p.

```
static VALUE
nucomp_real_p(VALUE self)
{
    get_dat1(self);
    return f_zero_p(dat->imag);
}
```

#2 - 10/26/2014 03:22 AM - gogotanaka (Kazuki Tanaka)

- File update_NEWS.patch added

- File add_tests.patch added

- File update_Complex#real_.patch added

There are not any arguments or opinions, I've made patches(implement, test, updating NEWS) anyway.

Please check it.

Thanks.

#3 - 10/26/2014 04:12 AM - sawa (Tsuyoshi Sawada)

gogo tanakaさん、

他スレッドでのコメントありがとうございます。

#4 - 10/26/2014 06:29 AM - gogotanaka (Kazuki Tanaka)

Tsuyoshi Sawada さん
こんにちは．22年間誤用しておりました．．お恥ずかしい．

ご指摘をするのも心苦しいと察します、わざわざお伝え頂けるとは本当に嬉しく思います．
お陰さまで苦手な英語を一歩前進させる事が出来ました．ありがとうございました！！

話は変わりますが、Tsuyoshi Sawada さんが面白いアイディアを沢山投稿されて、ステキだと思っています．
実際に実装されるまでは色々考えなければいけない所があるんでしょうが（もちろん僕も偉そうな事は決して言えません）

単純に考えを深めたりアイディアを生むきっかけとして僕自身大変楽しく拝見させて頂いております．

今後ともよろしくお願い致します．

#5 - 10/28/2014 08:39 AM - t-nissie (Takeshi Nishimatsu)

Objection.

Especially for Float, Complex(1.0,0.0) and Complex(1.0,-0.0) have meanings:
sqrt[-x+i(+0)]=i*sqrt(x)
sqrt[-x+i(-0)]=-i*sqrt(x)
So, they are complex. Not real.
And, please see the man page of cproj(3), though cproj is not implemented in Ruby.

Please see Goldberg's review.
http://docs.oracle.com/cd/E19957-01/806-4847/ncg_goldberg.html (EUC encoded)
http://iss.ndl.go.jp/books/R100000039-I001404293-00
@article{goldberg1991ecs,
title={What every computer scientist should know about floating-point arithmetic},
author={David Goldberg},
journal={ACM Computing Surveys},
volume={23},
number={1},
pages={5--48},
year={1991}}
http://dl.acm.org/citation.cfm?id=103163
This good review is also linked from https://bugs.ruby-lang.org/projects/ruby/wiki/HowToReportJa .

BTW, I do not like
-1.0-0.0i => (-1.0+0.0i) ,
though I know that it is translated as
-1.0-0.0i => Complex(-1.0,0.0)-Complex(0.0,-0.0) => (-1.0+0.0i) .

#6 - 10/28/2014 04:22 PM - gogotanaka (Kazuki Tanaka)

@Takeshi Nishimatsu san
Thank you for your comments and sharing good article.

From this article, your point-out might be right. I admit Complex(x, 0.0) is not truly real.
And I suppose -1.0-0.0i not be -1.0+0.0i too.

Aside from that, as I said #is_a?(Complex) or current #real? can be the way we check whether a number is truly real or not.
Actually whichever is ok, modifying #real? or implementing as new method.

What I want right now is checking whether a act as real.

So lets run through a few scenarios (I'd better find more practical example... * (

```
c = (2.0 + 1.0i)

a1 = (-1.0+0.0i)
p c * a1
```

```
a2 = (-1.0-0.0i)
p c * a2

a3 = (-1.0+0i)
p c * a3

a4 = -1.0
p c * a4
```

a1~4 act as same. When I check a5 is also same, I have to write

```
a5.is_a?(Complex) && a5.imag.zero?
```

It's little bit diffuse.

Thanks.

#7 - 10/29/2014 12:49 AM - t-nissie (Takeshi Nishimatsu)

FYI, on Julia:

```
julia> VERSION
v"0.3.1"

julia> Complex(1.0,-0.0)
1.0 - 0.0im

julia> 1.0-0.0im
1.0 - 0.0im

julia> -1.0-0.0im
-1.0 - 0.0im

julia> bool(Complex(1.0,-0.0))
true

julia> bool(Complex(1.0,0.0))
true

julia> bool(Complex(0.0,0.0))
false

julia> bool(Complex(1.0,1.0))
ERROR: InexactError()
 in bool at bool.jl:10
```

#8 - 10/30/2014 08:49 PM - gogotanaka (Kazuki Tanaka)

@Takeshi Nishimatsu san

Thank for info, how do you think about my points?

#9 - 10/31/2014 01:20 AM - t-nissie (Takeshi Nishimatsu)

> to know whether a object whose class has Numeric as superclass is equal to real number or not.

Simplly, a.real? || a.imag.zero? may be enough for me, so far.

```
irb(main):016:0> a = 2.16
=> 2.16
irb(main):017:0> a.real? || a.imag.zero?
=> true
irb(main):018:0> a = Complex(1.0,-0.0)
=> (1.0-0.0i)
irb(main):019:0> a.real? || a.imag.zero?
=> true
```

Sorry, but I cannot find any reason to change the current
definition of Numeric#real? and to define a new method.

#10 - 11/05/2014 09:40 PM - gogotanaka (Kazuki Tanaka)

@Takeshi Nishimatsu san

OK, it does make sense. thanks.

#11 - 01/05/2018 09:01 PM - naruse (Yui NARUSE)

- Target version deleted (2.2.0)

Files

update_NEWS.patch	716 Bytes	10/26/2014	gogotanaka (Kazuki Tanaka)
add_tests.patch	848 Bytes	10/26/2014	gogotanaka (Kazuki Tanaka)
update_Complex#real_.patch	1.18 KB	10/26/2014	gogotanaka (Kazuki Tanaka)