

Ruby trunk - Bug #15312

Ruby gives me wrong result for plus operation

11/16/2018 05:43 PM - kot-begemot (Max Zet)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Run into this weird issue today. Tried that on rubies 2.5.1p57 and ruby 2.5.3p105 (i believe this is latest stable at the moment).

Here is how to reproduce it

```
$ irb
2.5.3 :001 > 2.6 + 0.1
=> 2.7
2.5.3 :002 > 2.7 + 0.1
=> 2.80000000000000003
2.5.3 :003 > 2.8 + 0.1
=> 2.9
```

Here some more cases

```
$ irb
2.5.3 :001 > 2.7 + 0.01
=> 2.71
2.5.3 :002 > 2.79 + 0.01
=> 2.8
2.5.3 :003 > 2.8 - 0.1
=> 2.6999999999999997
2.5.3 :004 > 2.79 + 0.01
=> 2.8
2.5.3 :005 > 2.81 - 0.01
=> 2.80000000000000003
```

Haven't discovered other cases yet, if will catch them, will post them here

History

#1 - 11/16/2018 06:34 PM - oleynikov (Alexander Oleynikov)

This is not a bug, that's how floating point arithmetic works in accordance with the IEEE 754 standard.

You can read more about that on Wikipedia or in this article: https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html

To get mathematically correct results you should be using [BigDecimal](#) or [Rational](#).

```
require 'bigdecimal'
BigDecimal('2.7') + BigDecimal('0.1') #=> 0.28e1
```

#2 - 11/16/2018 06:37 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Rejected

See also <https://0.30000000000000004.com/>

#3 - 11/17/2018 01:28 AM - nobu (Nobuyoshi Nakada)

You can write rational numbers by r suffix.

```
p 0.1r + 0.2r #=> (3/10)
```

As I found these literals (r and i) have not been mentioned in doc/syntax/, added them now ([r65773](#)).

#4 - 11/17/2018 01:28 AM - nobu (Nobuyoshi Nakada)

- Description updated