Description

Consider `((1 ** expt))` and `(((-1) ** expt))`.

When `((expt))` is a Fixnum, the result is always 1, -1, Rational(1/1), or Rational(-1,1) depending on the signs of the operands and parity of the exponent.

When `((expt))` is a Bignum, Float 1.0 is always returned. Either the behavior for Fixnum exponents should be followed exactly (preferred), or at the very least -1.0 should be returned when the base is -1 and the exponent is odd.

```
$VERBOSE = nil

wordsize = 8 * 1.size
fixnum_max = 2 ** (wordsize - 2) - 1
fixnum_min = -2 ** (wordsize - 2)

[1, -1].each do |a|
  [1, 2, fixnum_max, fixnum_max + 1, fixnum_max + 2, -1, -2, fixnum_min, fixnum_min - 1, fixnum_min - 2].each do |b|
    puts "%5s ** %20s (%s) == %5s" % [a, b, b.class, a ** b]
  end
end
```

Output:

```
1 **                    1 (Fixnum) ==     1
1 **                    2 (Fixnum) ==     1
1 **  4611686018427387903 (Fixnum) ==     1
1 **  4611686018427387904 (Bignum) ==     1
1 **  4611686018427387905 (Bignum) ==     1
1 **                   -1 (Fixnum) ==   1/1
1 **                    -2 (Fixnum) ==   1/1
1 ** -4611686018427387904 (Fixnum) ==   1/1
1 ** -4611686018427387905 (Bignum) ==   1.0 # Bug
1 ** -4611686018427387906 (Bignum) ==   1.0 # Bug
-1 **                    1 (Fixnum) ==  -1
-1 **                    2 (Fixnum) ==  -1
-1 **  4611686018427387903 (Fixnum) ==  -1
-1 **  4611686018427387904 (Bignum) ==  -1
-1 **  4611686018427387905 (Bignum) ==  -1
-1 **                   -1 (Fixnum) ==  -1/1
-1 **                    -2 (Fixnum) ==  -1/1
-1 ** -4611686018427387904 (Fixnum) ==  1/1
-1 ** -4611686018427387905 (Bignum) ==  1.0 # Bug
-1 ** -4611686018427387906 (Bignum) ==  1.0 # Bug
=end
```

Related issues:

Related to Ruby master - Bug #5713: Fixnum#** returns Infinity for 0 ** negat...  Closed  12/06/2011

Associated revisions

Revision 8797ebd6 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision 39063 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]

Revision d22ce4a5 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

Revision 39064 - 02/05/2013 05:39 AM - marcandre (Marc-Andre Lafortune)
  * numeric.c (fix_pow): Handle special cases when base is 0, -1 or +1 [Bug #5713] [Bug #5715]

History
#1 - 12/10/2011 02:00 AM - john_firebaugh (John Firebaugh)
The failing cases are the ones that go through the Rational(+/1) ** Bignum code path, so this is closely related to #5713.

#2 - 12/10/2011 03:44 AM - marcandre (Marc-Andre Lafortune)
- Category set to core
- Assignee set to marcandre (Marc-Andre Lafortune)

Just saw this update. Yes, as I stated, same issue as 5713. Yes the case for (-1) and bignum exponents can be fixed too. The case for 1 and 0 must be addressed for rational and float exponents too (but -1 will go to float for those)

#3 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Assigned

#4 - 01/25/2013 02:46 PM - marcandre (Marc-Andre Lafortune)
- Target version set to 2.0.0

#5 - 02/05/2013 02:39 PM - marcandre (Marc-Andre Lafortune)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r39063.
John, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- rational.c (nurat_expt): Deal with special cases for rationals 0, ±1 [bug #5713] [bug #5715]