Description

When running the following code:

```ruby
require 'bigdecimal'
10000.times { BigDecimal.new('1001.10')**0.75; putc '.' }
```

The call to `**` will sometimes (not always) raise the following trace:

```ruby
.................................TypeError: #<Class:0x0000001cd78f40> can't be coerced into BigDecimal
   from (irb):4:in `**'
   from (irb):4:in `block in irb_binding'
   from (irb):3:in `times'
   from (irb):3
   from /home/tony/.rbenv/versions/1.9.3-p194/bin/irb:12:in `<main>'
```

(The `.'s being the debug print of the putc.) I understand the documentation says only integer powers are supported for BigDecimal, however there is code support for floats and rationals, and it should fail or work reliably either way.

Associated revisions

Revision 1be6a498 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

- ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

- test/bigdecimal/test_bigdecimal.rb: add a reproduction test for the issue [Bug #7044]

Revision 38734 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

- ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

- test/bigdecimal/test_bigdecimal.rb: add a reproduction test for the issue [Bug #7044]
* ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

* test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]

Revision 38734 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

* ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

* test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]

Revision 38734 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

* ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

* test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]

Revision 38734 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

* ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

* test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]

Revision 38734 - 01/07/2013 02:42 PM - mrkn (Kenta Murata)

* ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the immediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]

* test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]

History
#1 - 11/05/2012 08:33 PM - mame (Yusuke Endoh)
- Status changed from Open to Assigned
- Assignee set to mrkn (Kenta Murata)
- Target version set to 2.0.0

Confirmed, good catch!
Looks GC issue. An intermediate object seems to be GC'd too early.
The following patch will fix. Mrkn, could you review and commit it if it looks good?

```
diff --git a/ext/bigdecimal/bigdecimal.c b/ext/bigdecimal/bigdecimal.c
index f58b640..36ca77d 100644
--- a/ext/bigdecimal/bigdecimal.c
+++ b/ext/bigdecimal/bigdecimal.c
@@ -2016,6 +2016,7 @@ static VALUE
      rmpd_power_by_big_decimal(Real const* x, Real const* exp, ssize_t const n)
      { VALUE log_x, multiplied, y;

         volatile VALUE obj = exp->obj;
         if (VpIsZero(exp)) {
            return ToValue(VpCreateRbObject(n, "1"));
@@ -2024,6 +2025,7 @@ rmpd_power_by_big_decimal(Real const* x, Real const* exp, ssize_t const n)
            multiplied = BigDecimal_mult2(exp->obj, log_x, SSIZET2NUM(n+1));
            y = BigMath_exp(multiplied, SSIZET2NUM(n));
         } RB_GC_GUARD(obj);

         return y;
      }
```

--

Yusuke Endoh mame@tsg.ne.jp

#2 - 01/07/2013 11:42 PM - mrkn (Kenta Murata)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r38734.
Tony, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- ext/bigdecimal/bigdecimal.c (rmpd_power_by_big_decimal):
  add RB_GC_GUARD to prevent the intermediate object is GCed too early.
  This patch was made by Yusuke Endoh. [Bug #7044] [ruby-core:47632]
- test/bigdecimal/test_bigdecimal.rb: add a reproduction test for
  the issue [Bug #7044]