

Ruby trunk - Bug #5227

Float#round fails on corner cases

08/25/2011 06:03 AM - marcandre (Marc-Andre Lafortune)

Status: Closed	
Priority: Normal	
Assignee: marcandre (Marc-Andre Lafortune)	
Target version: 1.9.3	
ruby -v: -	Backport:
Description	
Float#round fails on some corner cases:	
42.0.round(300) # => 42.0 42.0.round(308) # => Infinity, should be 42.0 42.0.round(309) # => 42.0	
1.0e307.round(1) # => 1.0e307 1.0e307.round(2) # => Infinity, should be 1.0e307	
These occur when the exponent of the intermediate value overflows.	
The original code already had criteria for extreme values, but we can find much tighter ones, as explained in the patch below. This fixes the bugs above and optimizes for most trivial cases.	
I'd be grateful if someone could look it over before I commit it, thanks.	
<pre>diff --git a/numeric.c b/numeric.c index 272bbd1..22608c9 100644 --- a/numeric.c +++ b/numeric.c @@ -1491,18 +1491,37 @@ flo_round(int argc, VALUE *argv, VALUE num) VALUE nd; double number, f; int ndigits = 0; • int binexp; long val; if (argc > 0 && rb_scan_args(argc, argv, "01", &nd) == 1) { ndigits = NUM2INT(nd); } number = RFLOAT_VALUE(num); - f = pow(10, abs(ndigits)); • if (isinf(f)) { • if (ndigits < 0) number = 0; • } • else { • frexp (number, &binexp); + +/* Let exp be such that number is written as: "0.{digits}e#{exp}", • i.e. such that 10 ** (exp - 1) <= number < 10 ** exp • Recall that up to 17 digits can be needed to represent a double, • so if ndigits + exp >= 17, the intermediate value (number * 10 ** ndigits) • will be an integer and thus the result is the original number. • If ndigits + exp <= 0, the result is 0 or "1e#{exp}", so • if ndigits + exp < 0, the result is 0. • We have: • 2 ** (binexp-1) <= number < 2 ** binexp • 10 ** ((binexp-1)/log_2(10)) <= number < 10 ** (binexp/log_2(10)) • If binexp >= 0, and since log_2(10) = 3.322259: • 10 ** (binexp/4 - 1) < number < 10 ** (binexp/3)</pre>	

- `binexp/4 <= exp <= binexp/3`
- If `binexp <= 0`, swap the `/4` and the `/3`
- So if `ndigits + binexp/(3 or 4) >= 17`, the result is number
- If `ndigits + binexp/(4 or 3) < 0` the result is `0 +*/`
- if `((long)ndigits * (4 - (binexp < 0)) + binexp < 0) {`
- `number = 0;`
- `}`
- else if `((long)(ndigits - 17) * (3 + (binexp < 0)) + binexp < 0) {`
- `f = pow(10, abs(ndigits)); if (ndigits < 0) { double absnum = fabs(number); if (absnum < f) return INT2FIX(0);`

Associated revisions

Revision 64e0cae2 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33140 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision 33140 - 08/31/2011 04:13 AM - marcandre (Marc-Andre Lafortune)

- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

Revision cc6407f8 - 08/31/2011 04:20 AM - marcandre (Marc-Andre Lafortune)

- backport r33140 from trunk
- `numeric.c (flo_round)`: Avoid overflow by optimizing for trivial cases [Bug #5227]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@33141 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 08/25/2011 10:49 PM - nobu (Nobuyoshi Nakada)

Totally seems fine.

- A magic number 17 is `DBL_DIG+2`?
- `frexp` call doesn't feel matching with the rest of the file.

#2 - 08/26/2011 05:29 AM - Anonymous

Hi,

The 17 is the maximum number of digits needed to represent uniquely the double (see [ruby-core:30145] and [ruby-core:30163]).

On Thu, Aug 25, 2011 at 9:49 AM, Nobuyoshi Nakada nobu@ruby-lang.org wrote:

Issue [#5227](#) has been updated by Nobuyoshi Nakada.

A magic number 17 is DBL_DIG+2?
T

#3 - 08/26/2011 10:46 AM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

r33061 tried to fix this but the result isn't changed.

Additional to say:

- add test for this to test/ruby/test_float.rb
- write Ticket number in ChangeLog and commit message (this is current limitation of Redmine-commit association)

#4 - 08/27/2011 04:53 AM - Anonymous

Hi

On Thu, Aug 25, 2011 at 9:46 PM, Yui NARUSE naruse@airemix.jp wrote:

r33061 tried to fix this but the result isn't changed.

Sorry, I do not understand. r33061 fixes Integer#round, which had a completely different issue from Float#round (see redmine [#5228](#))

Additional to say:

- add test for this to test/ruby/test_float.rb

I already committed tests in RubySpec, which I believe is where they are the most useful.

- write Ticket number in ChangeLog and commit message (this is current limitation of Redmine-commit association)

Ah, thanks, I didn't realize this had changed. I updated the contributor guidelines wiki page (<http://redmine.ruby-lang.org/projects/ruby/wiki/CommitterHowto>) but I don't have write access to another that should be updated accordingly: <http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

Can someone either give me access or update it?

Thanks

Bug [#5227](#): Float#round fails on corner cases
<http://redmine.ruby-lang.org/issues/5227>

Author: Marc-Andre Lafortune
Status: Assigned
Priority: Normal
Assignee: Marc-Andre Lafortune
Category: core
Target version: 1.9.3
ruby -v: r32601

Float#round fails on some corner cases:

À 42.0.round(300) #

#5 - 08/27/2011 05:23 PM - nobu (Nobuyoshi Nakada)

- ruby -v changed from r32601 to -

Hi,

At Sat, 27 Aug 2011 04:40:16 +0900,
Marc-Andre Lafortune wrote in [ruby-core:39126]:

Additional to say:

- add test for this to test/ruby/test_float.rb

I already committed tests in RubySpec, which I believe is where they are the most useful.

No. In fact, your patch breaks test/ruby/test_float.rb.

1) Failure:
test_round_with_precision(TestFloat) [test/ruby/test_float.rb:326]:

expected but was

.

--

Nobu Nakada

#6 - 08/27/2011 05:53 PM - naruse (Yui NARUSE)

(2011/08/27 4:40), Marc-Andre Lafortune wrote:

On Thu, Aug 25, 2011 at 9:46 PM, Yui NARUSE naruse@airemix.jp wrote:

r33061 tried to fix this but the result isn't changed.

Sorry, I do not understand. r33061 fixes Integer#round, which had a completely different issue from Float#round (see redmine [#5228](#))

```
./ruby -ve'p 42.0.round(308)'  
ruby 1.9.4dev (2011-08-26 trunk 33073) [x86_64-freebsd8.2]  
Infinity
```

Additional to say:

- add test for this to test/ruby/test_float.rb

I already committed tests in RubySpec, which I believe is where they are the most useful.

Adding tests to RubySpec is not enough.
You should also add tests to CRuby's test-all.

Moreover it fails on my environment:

```
2)  
Float#round works for corner cases FAILED  
Expected Infinity  
to equal 42.0
```

```
/usr/home/chkbuild/build/ruby-trunk/20110827T070101Z/rubyspec/core/float/round_spec.rb:30:in block (4 levels) in <top (required)>'  
/usr/home/chkbuild/build/ruby-trunk/20110827T070101Z/rubyspec/core/float/round_spec.rb:3:in '  
http://59.106.172.211/~chkbuild/ruby-trunk/log/20110827T070101Z.log.html.gz  
http://59.106.172.211/~chkbuild/ruby-trunk/recent.html
```

- write Ticket number in ChangeLog and commit message (this is current limitation of Redmine-commit association)

Ah, thanks, I didn't realize this had changed. I updated the contributor guidelines wiki page (

<http://redmine.ruby-lang.org/projects/ruby/wiki/CommitterHowto>) but I don't have write access to another that should be updated accordingly: <http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

Can someone either give me access or update it?

Added.

--
NARUSE, Yui naruse@airemix.jp

#7 - 08/29/2011 04:53 AM - Anonymous

Hi,

On Sat, Aug 27, 2011 at 4:20 AM, Nobuyoshi Nakada nobu@ruby-lang.org wrote:

In fact, your patch breaks test/ruby/test_float.rb.

Oh, I didn't realize that make test-all does not recompile modified .c files. Is there a good reason for that?

In any case, I write specs in RubySpec, but ruby's tests must still be run too, of course!

Indeed, I made a mistake in the inequalities for the 3 or 4 approximation (should test for $\text{binexp} > 0$, not < 0). Here's the differential patch:

```
diff --git a/numeric.c b/numeric.c
index 4c48355..ac38bb1 100644
--- a/numeric.c
+++ b/numeric.c
@@ -1517,10 +1517,10 @@ flo_round(int argc, VALUE *argv, VALUE num)
So if  $\text{ndigits} + \text{binexp}/(3 \text{ or } 4) \geq 17$ , the result is number
If  $\text{ndigits} + \text{binexp}/(4 \text{ or } 3) < 0$  the result is 0
*/
```


- if ((long)ndigits * (4 - (binexp < 0)) + binexp < 0) {
- if ((long)ndigits * (4 - (binexp > 0)) + binexp < 0) { number = 0; }
- else if ((long)(ndigits - 17) * (3 + (binexp < 0)) + binexp < 0) {
- else if ((long)(ndigits - 17) * (3 + (binexp > 0)) + binexp < 0) { f = pow(10, abs(ndigits)); if (ndigits < 0) { double absnum = fabs(number);

The complete patch becomes:

```
diff --git a/numeric.c b/numeric.c
index ab890c6..2277296 100644
--- a/numeric.c
+++ b/numeric.c
@@ -1491,18 +1491,37 @@ flo_round(int argc, VALUE *argv, VALUE num)
VALUE nd;
double number, f;
int ndigits = 0;
```

- int binexp;
- long val;
- if (argc > 0 && rb_scan_args(argc, argv, "01", &nd) == 1) {
- ndigits = NUM2INT(nd);
- }
- number = RFLOAT_VALUE(num);

• f = pow(10, abs(ndigits));

- if (isinf(f)) {
- 
- }
- else {

- frexp (number , &binexp);
+
+/* Let exp be such that number is written as:"0.#{digits}e#{exp}",
- i.e. such that $10^{exp-1} \leq |number| < 10^{exp}$
- Recall that up to 17 digits can be needed to represent a double,
- so if ndigits + exp \geq 17, the intermediate value (number * 10^{ndigits})
- will be an integer and thus the result is the original number.
- If ndigits + exp \leq 0, the result is 0 or "1e#{exp}", so
- if ndigits + exp < 0, the result is 0.

• We have:

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]

+*/

- if ((long)ndigits * (4 - (binexp > 0)) + binexp < 0) {
- [REDACTED]
- }
- else if ((long)(ndigits - 17) * (3 + (binexp > 0)) + binexp < 0) {
- [REDACTED]

```
if (ndigits < 0) {
    double absnum = fabs(number);
    if (absnum < f) return INT2FIX(0);
}
```

#8 - 08/29/2011 04:53 AM - Anonymous

Hi,

On Sat, Aug 27, 2011 at 4:32 AM, NARUSE, Yui naruse@airemix.jp wrote:

(2011/08/27 4:40), Marc-Andre Lafortune wrote:

On Thu, Aug 25, 2011 at 9:46 PM, Yui NARUSE naruse@airemix.jp wrote:

r33061 tried to fix this but the result isn't changed.

Sorry, I do not understand. r33061 fixes Integer#round, which had a completely different issue from Float#round (see redmine [#5228](#))

```
./ruby -ve'p 42.0.round(308)'
ruby 1.9.4dev (2011-08-26 trunk 33073) [x86_64-freebsd8.2]
Infinity
```

How is this in any way related to r33061?

Adding tests to RubySpec is not enough.
You should also add tests to CRuby's test-all.

I respectfully disagree.

Moreover it fails on my environment:

Could you please double check this, or else give me information so I can try to reproduce this, as I don't see how this could be the case. Maybe you got bitten by the fact that make test-rubyspec does not recompile modified sources, like I was with make test-all (see my previous message to Nobu)?

Can someone either give me access or update it?

Added.

Thanks, but somehow I still can't edit the page
<http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

#9 - 08/29/2011 09:59 AM - naruse (Yui NARUSE)

2011/8/29 Marc-Andre Lafortune ruby-core-mailing-list@marc-andre.ca:

On Sat, Aug 27, 2011 at 4:32 AM, NARUSE, Yui naruse@airemix.jp wrote:

(2011/08/27 4:40), Marc-Andre Lafortune wrote:

On Thu, Aug 25, 2011 at 9:46 PM, Yui NARUSE naruse@airemix.jp wrote:

r33061 tried to fix this but the result isn't changed.

Sorry, I do not understand. r33061 fixes Integer#round, which had a completely different issue from Float#round (see redmine [#5228](#))

```
./ruby -ve'p 42.0.round(308)'  
ruby 1.9.4dev (2011-08-26 trunk 33073) [x86_64-freebsd8.2]  
Infinity
```

How is this in any way related to r33061?

It happens after r33061.
You can know after r33066, RubySpec F is increased.
20110825T020101Z ruby 1.9.4dev (2011-08-25 trunk 33066)
[x86_64-freebsd8.2](#) 1851W rubyspec:3F0E
(diff:src,btest,make,install-nodoc,test-all,rubyspec)
<http://59.106.172.211/~chkbuild/ruby-trunk/summary.html>

It suspects a commit between r33047 and r33066 breaks it.
...but in this case, it is because you add a test to RubySpec, sorry.

Adding tests to RubySpec is not enough.
You should also add tests to CRuby's test-all.

I respectfully disagree.

RubySpec is not CRuby's implementation test, it is a different thing.
You must add a test to test-all when you change CRuby's behavior.

Can someone either give me access or update it?

Added.

Thanks, but somehow I still can't edit the page
<http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

I added you a edit bit; did you unlocked the page?
(I don't know if you unlock it then can edit but...)

--
NARUSE, Yui naruse@airemix.jp

#10 - 08/31/2011 09:53 AM - Anonymous

Hi,

On Sun, Aug 28, 2011 at 8:53 PM, NARUSE, Yui naruse@airemix.jp wrote:

Adding tests to RubySpec is not enough.
You should also add tests to CRuby's test-all.

I respectfully disagree.

RubySpec is not CRuby's implementation test, it is a different thing.
You must add a test to test-all when you change CRuby's behavior.

RubySpec is part of CRuby's implementation test.
Do you have arguments against having tests in RubySpec instead of test/ruby?

Thanks, but somehow I still can't edit the page
<http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

I added you a edit bit; did you unlocked the page?
(I don't know if you unlock it then can edit but...)

Sorry, I don't know what "unlocking" a wiki page is. The only
functionality that I see for that particular page is 'history'. Maybe
you can modify it yourself?

#11 - 08/31/2011 10:23 AM - naruse (Yui NARUSE)

2011/8/31 Marc-Andre Lafortune ruby-core-mailing-list@marc-andre.ca:

Hi,

On Sun, Aug 28, 2011 at 8:53 PM, NARUSE, Yui naruse@airemix.jp wrote:

Adding tests to RubySpec is not enough.
You should also add tests to CRuby's test-all.

I respectfully disagree.

RubySpec is not CRuby's implementation test, it is a different thing.
You must add a test to test-all when you change CRuby's behavior.

RubySpec is part of CRuby's implementation test.
Do you have arguments against having tests in RubySpec instead of test/ruby?

Because I don't think RubySpec is part of CRuby's implementation test.
I believe other CRuby people also think so.
So on current situation you must add a test to test-all.
(If you can persuade other committers, I'll follow you)

Thanks, but somehow I still can't edit the page
<http://redmine.ruby-lang.org/projects/redmine/wiki/VersionControlSystem>

I added you a edit bit; did you unlocked the page?

(I don't know if you unlock it then can edit but...)

Sorry, I don't know what "unlocking" a wiki page is. The only functionality that I see for that particular page is 'history'. Maybe you can modify it yourself?

Commented out ML number cooperation.

--

NARUSE, Yui [Â naruse@airemix.jp](mailto:naruse@airemix.jp)

#12 - 08/31/2011 01:13 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r33140.
Marc-Andre, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- numeric.c (flo_round): Avoid overflow by optimizing for trivial cases [Bug [#5227](#)]

#13 - 08/31/2011 02:29 PM - shyouhei (Shyouhei Urabe)

"So what's the test going to achieve? What's that test for?" said my teacher another day. A test has a perspective. No test can be suitable for everything.

It seems to me RubySpec is meant to be a written behavioral description. Which is not a bad thing of course, but I think Yui needs a regression test instead. It does test CRuby's implementation but doesn't help him.

#14 - 09/05/2011 03:53 AM - Anonymous

Hi,

On Tue, Aug 30, 2011 at 9:19 PM, NARUSE, Yui naruse@airemix.jp wrote:

RubySpec is part of CRuby's implementation test.
Do you have arguments against having tests in RubySpec instead of test/ruby?

Because I don't think RubySpec is part of CRuby's implementation test.
I believe other CRuby people also think so.
So on current situation you must add a test to test-all.
(If you can persuade other committers, I'll follow you)

Although I'm a newbie and not great at persuading people of anything, I'll start a new thread about this.

Commented out ML number cooperation.

Thanks!