

## Ruby trunk - Feature #6678

### Precedence of ^ operator

07/01/2012 06:32 AM - trans (Thomas Sawyer)

<b>Status:</b>	Rejected
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	Next Major
<b>Description</b>	
I wrote about this here: <a href="http://trans.github.com/2009/12/13/a-failure-of-precedence.html">http://trans.github.com/2009/12/13/a-failure-of-precedence.html</a> (or <a href="https://github.com/trans/trans.github.com/wiki/2009-12-13-a-failure-of-precedence">https://github.com/trans/trans.github.com/wiki/2009-12-13-a-failure-of-precedence</a> if you prefer)	
In short I would like to see ^ operator move up in precedence to be the same as **. We need another option besides ** at the top of precedence list and ^ fits the bill well since many languages already think of it as "power of".	
Of course, this has downside that effects current use for XOR definition. A new operator would be needed for that but ^ could alias it by default, so only thing that changes is precedence. Thankfully XOR is not among the most commonly used operators and often when used the precedence will be okay either way. So there should be very few cases where new operator or parenthesis will need to be put in.	
Ultimately it would be great if precedence could be controlled on per-class basis. But that is big change, and might not be altogether practical. So I propose this one modest adjustment instead in which it would be an improvement.	

### History

#### #1 - 10/25/2012 07:22 PM - yhara (Yutaka HARA)

- Target version changed from 2.0.0 to Next Major

#### #2 - 10/26/2012 11:54 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

' is not a power operator in Ruby.

#### #3 - 10/27/2012 01:04 AM - trans (Thomas Sawyer)

I take it you read the article? Do you see the issue I am having in need of an operator with a precedence above + - \* and /? So what solution would you recommend?

Note, I am not saying ' should stop being XOR, or that it should become power operator for Numeric. I just was asking if it could get a higher precedence, so it could be used for such in custom classes. As a broader question, why are the bitwise operators lower in precedence than math operators? Is there a reason for that (other than that's how everyone has always done it). I've read that even Dennis Ritchie admits that the precedence of operators in C, which everyone has copied, isn't optimal.

#### #4 - 10/27/2012 06:29 AM - duerst (Martin Dürst)

On 2012/10/27 1:04, trans (Thomas Sawyer) wrote:

Issue [#6678](#) has been updated by trans (Thomas Sawyer).

I take it you read the article? Do you see the issue I am having in need of an operator with a precedence above + - \* and /? So what solution would you recommend?

<http://trans.github.com/2009/12/13/a-failure-of-precedence.html> gives me a 404 (There isn't a GitHub Page hosted here, sorry.).

<https://github.com/trans/trans.github.com/wiki/2009-12-13-a-failure-of-precedence> works, and I have read it. There's just one single example/use case. It seems inappropriate to claim that this is a big problem with just a single example/use case.

Also, what about alternatives such as 2.square.meters, or 2.meters.squared ?

Also, you say: "On the other hand, some simple issues doggedly persist for little good reason (backward compatibility notwithstanding)". This

sounds to me as if you think that backwards compatibility wasn't a problem at all. But who is going to fix all these scripts out there that rely on the current precedence?

Note, I am not saying ' should stop being XOR, or that it should become power operator for Numeric. I just was asking if it could get a higher precedence, so it could be used for such in custom classes. As a broader question, why are the bitwise operators lower in precedence than math operators? Is there a reason for that (other than that's how everyone has always done it). I've read that even Dennis Ritchie admits that the precedence of operators in C, which everyone has copied, isn't optimal.

This is third-hand knowledge, and not in any way specific to "" or the absence of other operators with the same precedence as "" (because C doesn't have that). Also, I don't think Dennis Ritchie would agree changing operator precedences in a widely used programming language. (unfortunately, we can't ask him anymore)

Regards, Martin.

---

Feature [#6678](#): Precedence of ^ operator  
<https://bugs.ruby-lang.org/issues/6678#change-31614>

Author: trans (Thomas Sawyer)  
Status: Rejected  
Priority: Normal  
Assignee:  
Category: core  
Target version: Next Major

I wrote about this here: <http://trans.github.com/2009/12/13/a-failure-of-precedence.html> (or <https://github.com/trans/trans.github.com/wiki/2009-12-13-a-failure-of-precedence> if you prefer)

In short I would like to see ^ operator move up in precedence to be the same as \*\*. We need another option besides \*\* at the top of precedence list and ^ fits the bill well since many languages already think of it as "power of".

Of course, this has downside that effects current use for XOR definition. A new operator would be needed for that but ^ could alias it by default, so only thing that changes is precedence. Thankfully XOR is not among the most commonly used operators and often when used the precedence will be okay either way. So there should be very few cases where new operator or parenthesis will need to be put in.

Ultimately it would be great if precedence could be controlled on per-class basis. But that is big change, and might not be altogether practical. So I propose this one modest adjustment instead in which it would be an improvement.

#### #5 - 10/27/2012 10:33 AM - trans (Thomas Sawyer)

gives me a 404 (There isn't a GitHub Page hosted here, sorry.).

It was just the same write-up with different formatting. So nothing missed.

This sounds to me as if you think that backwards compatibility wasn't a problem at all. But who is going to fix all these scripts out there that rely on the current precedence?

It's a valid reason. I was just saying what I concluded putting that particular reason aside. It is important, of course. How much code would break if ^ had a high precedence? It can be hard to say. I would expect very little given typical usage I've encountered. Of course, there will still be cases, even if few.

This is third-hand knowledge, and not in any way specific to "" ...

Pretty sure Dennis Ritchie did specifically say that the bitwise operators should have not been lower than == and that he would do it differently if he had to do over again. Inferring from his statements about this, it seems operator precedence arose more organically rather than being fully rationalized out. I too wish we could ask him.