Ruby master - Feature #14836
Method to return first/last lineno/column of Proc

06/08/2018 09:48 AM - tagomoris (Satoshi TAGOMORI)

Status: Closed
Priority: Normal
Assignee:
Target version:

Description
As written in [https://bugs.ruby-lang.org/issues/6012](https://bugs.ruby-lang.org/issues/6012), we want to know column no of Proc location. In addition to it, I want to know the last_lineno and last_column.

If we don't have these values, we cannot know whether the "a" in code below is defined in block, or not.

```
b1 = ->() { foo(); a = 1; b2 = ->() { bar() }
```

Associated revisions
Revision ce798d08 - 11/05/2018 02:13 AM - yui-knk (Kaneko Yuichiro)
Implement RubyVM::AST.of [Feature #14836]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@65542 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 65542 - 11/05/2018 02:13 AM - yui-knk (Kaneko Yuichiro)
Implement RubyVM::AST.of [Feature #14836]

Revision 65542 - 11/05/2018 02:13 AM - yui-knk (Kaneko Yuichiro)
Implement RubyVM::AST.of [Feature #14836]

History
#1 - 06/08/2018 10:14 AM - joker1007 (Tomohiro Hashidate)
I want this feature too.

I wrote a rubygem that parses proc and converts to AST::Node (of ast gem).
To inspect proc source code is main purpose.

[https://github.com/joker1007/proc_to_ast](https://github.com/joker1007/proc_to_ast)

Usage example.

```ruby
# this is sample of rspec-parameterized gem

describe "lambda parameter" do
  where(a, :b, :answer) do
    [1, 2, -> {should == 3}],
    [5, 8, -> {should == 13}],
    [0, 0, -> {should == 0}]
  end

  with_them do
    subject {a + b}
    it "should do additions" do
      self.instance_exec(&answer)
    end
  end
end
```

And output example.

```
lambda parameter
  a: 1, b: 2, answer: -> {should == 3}
  should do additions
```
But the gem has very heuristic hacks and it is fragile.
If multi procs exist at same line, it is very difficult to detect particular proc.
Accurate Proc location resolves the problem.

#2 - 11/01/2018 04:31 AM - yui-knk (Kaneko Yuichiro)
If you use location information of proc to get AST nodes, I feel adding class method to AST module which receives proc and return AST nodes directly is useful.
What do you think about it?

For example:

```ruby
def a(block)
  node = RubyVM::AST.of(block)
  p node
  p node.children
  p node.children.last.children
end

a do
  1 + 2
  "abc".upcase
end
```

$ ./miniruby /tmp/of.rb

```
<RubyVM::AST::Node(NODE_SCOPE(0) 8:2, 11:3): >
[[], nil, <!RubyVM::AST::Node(NODE_BLOCK(1) 9:2, 10:14): >]
[<RubyVM::AST::Node(NODE_OPCALL(36) 9:2, 9:7): >, <RubyVM::AST::Node(NODE_CALL(35) 10:2, 10:14): >]
```

Ref: https://github.com/ruby/ruby/compare/trunk...yui-knk:feature/ast_of?expand=1

#3 - 11/01/2018 07:59 AM - joker1007 (Tomohiro Hashidate)
Great work!!
What I want is this!
If ruby-trunk merges this patch, I will try to use.

#4 - 11/05/2018 02:14 AM - yui-knk (Kaneko Yuichiro)
- Status changed from Open to Closed

Applied in changeset trunk/r65542.

Implement RubyVM::AST.of [Feature #14836]