Ruby master - Feature #16137

Add === to UnboundMethod

09/03/2019 12:28 PM - okuramasafumi (Masafumi OKURA)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Abstract

UnboundMethod class should have === so that it can be used in case statement.

Background

Method class has === method so that we can do something like:

```
require 'prime'
case 11
when Prime.method(:prime?) then :prime
end
```

However, we cannot do something like this:

```
positive = Integer.instance_method(:positive?)
case 11
when positive then :positive
end
```

Proposal

Add === method to UnboundMethod class.

Implementation

Minimal implementation in Ruby could be:

```
class UnboundMethod
  def ===(other)
    bind(other).call
  end
end
```

Summary

=== for UnboundMethod can improve readability in case statement.

History

#1 - 09/03/2019 12:58 PM - osyo (manga osyo)

hi.

How about making # bind_call an alias of# ===?

```
class UnboundMethod
  alias_method :===, :bind_call
end
```

see: https://bugs.ruby-lang.org/issues/15955#note-10

Also, you should consider the difference between when Integer.instance_method(:positive?) or when: positive?.to_proc.

```
case 11
when :positive?.to_proc then :positive
end
```

or

```
case 11
when Integer.instance_method(:positive?) then :positive
end
```

Thank you :)

#2 - 09/03/2019 02:50 PM - shevegen (Robert A. Heiler)

Since I love case/when statements in ruby, and I also like the idea to decouple methods
at "runtime", I agree with okuramasafumi.

I can not add much to potential use cases or what osyo asked, but personally I just like
the idea. :)

#3 - 09/04/2019 02:09 PM - nobu (Nobuyoshi Nakada)

As UnboundMethod#bind_call raises a TypeError unless the argument class matches, it isn't for when.

And the example doesn't look more readable to me.

#4 - 09/07/2019 11:05 AM - Eregon (Benoit Daloze)

With the new pattern matching:

```
case 11
in n if n.positive?
  p :positive
end
```

Isn't that more readable and general?