

Ruby master - Feature #15824

respond_to pattern for pattern match

05/05/2019 12:00 AM - fukajun (Jun Fukaya)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description ruby Interger, Array constant duck typing ruby respond_to pattern class Runner def run end def stop end end runner = Runner.new case runner in .run & .stop :reachable in .start & .stop :unreachable end	
Related issues: Related to Ruby master - Feature #14912: Introduce pattern matching syntax Assigned	

History

#1 - 05/05/2019 03:32 PM - shevegen (Robert A. Heiler)

I don't want to comment so much on the issue as such, but I would like to point out that it may be a better idea to start with a new idea not on highest complexity. I am not sure if everyone fully understood pattern matching yet so in my personal opinion it may be better to see how pattern matching is used before making it more complex (including syntactic changes). But that's just my opinion, anyway; some of the syntax appears to become no longer recognizable ruby (to me).

#2 - 05/22/2019 08:57 AM - matz (Yukihiro Matsumoto)

Interesting idea. We need to investigate the idea further.

Matz.

#3 - 06/14/2019 11:42 PM - ktsj (Kazuki Tsujimoto)

- Related to Feature #14912: Introduce pattern matching syntax added

#4 - 10/06/2019 07:56 AM - wanabe (_ wanabe)

The implementation can be very simple.
(I don't know whether NODE_METHREF is suitable.)

```
diff --git a/compile.c b/compile.c
index 7a88f81daa..f6eafb0ac2 100644
--- a/compile.c
+++ b/compile.c
@@ -5888,6 +5888,11 @@ iseq_compile_pattern_each(rb_iseq_t *iseq, LINK_ANCHOR *const ret, const NODE *c
     ADD_LABEL(ret, fin);
```

```

        break;
    }
+   case NODE_METHREF: {
+       ADD_INSN1(ret, line, putobject, ID2SYM(node->nd_mid));
+       ADD_SEND(ret, line, idRespond_to, INT2FIX(1));
+       break;
+   }
    default:
        UNKNOWN_NODE("NODE_IN", node, COMPILER_ERROR);
}
diff --git a/parse.y b/parse.y
index 287704e14f..dd305479f5 100644
--- a/parse.y
+++ b/parse.y
@@ -3904,6 +3904,10 @@ p_expr_basic      : p_value
    {
        $$ = $2;
    }
+   | '.' operation2
+   {
+       $$ = NEW_METHREF(Qundef, $2, &@$);
+   }
;

p_args      : p_expr

```

By the way, missing "& pattern" seems to be intended. see [ruby-core:88036] <https://bugs.ruby-lang.org/issues/14912#note-7> .