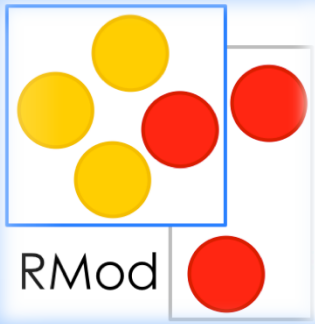
Phorms: Pattern Matching Library for Pharo

Markiyan Rizun

Camille Teruel, Gustavo Santos, Stéphane Ducasse



Pattern matching

Introduced in **functional programming**.

Mainly used for **inductive definition of functions**.

Pattern matching.
Example

```
fib n
    | n == 0 = 1
    | n == 1 = 1
    | n >= 2 = fib (n-1) + fib (n-2)
```

Pattern matching

Integrated in **Newspeak, Scala**.

Mainly used to **decompose object data**.

Pattern matching in Pharo

The Rewrite Engine

- Source code rewriting

Pattern matching in Pharo.
Example

“input”

```
a > b
        ifTrue: [ a ]
        ifFalse: [ b ]
```

“result”

```
a max: b
```

“matching part”

```
``@a > ``@b
        ifTrue: [ ``@a ]
        ifFalse: [ ``@b ]
```

“transforming part”

```
``@a max: ``@b
```

The Rewrite Engine

1. **Efficient** source code **rewriting**.
2. **Smalltalk** style **syntax**.

1. **AST focused**.
2. Sometimes **confusing**.
3. Patterns are **not composable**.
4. **Debug** and **inspect** are **not user-friendly**.
5. **Not extendable**.

Solution

Phorms

- Patterns are objects

Phorms.
Patterns

1 **equals**	“equality”
p1 **&** p2 **&** … **&** pN	“and”
p1 **|** p2 **|** … **|** pN	“or”

Phorms.
Patterns

p1, p2, … , pN	“list”
p1, (p2, p3) inList	“nested list”
p1, p2 **star**, p3	“star list”

Phorms.
Patterns

pattern **==>** [:it :context | …] “rewriting”

1 equals **named:** #aName “named”

#var **any** “any”

Phorms.
Matching

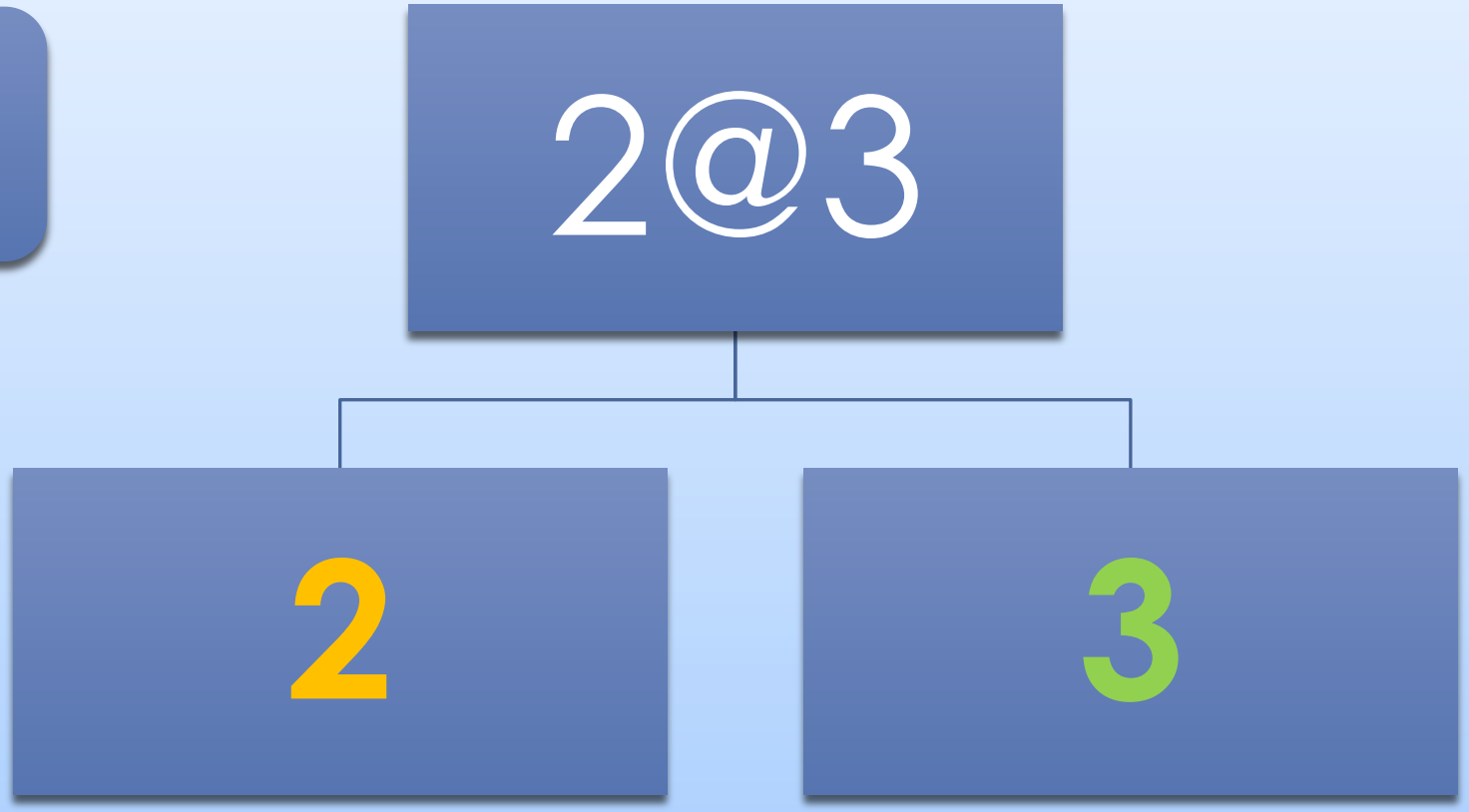
```
pattern match: anObject
```

Phorms.
Matching **Example 1**

1) pattern := **2 equals**, **3 equals**.
2) pattern **match:** **2**@**3**.

Phorms. Matching **Example 1**

2 equals, **3 equals**

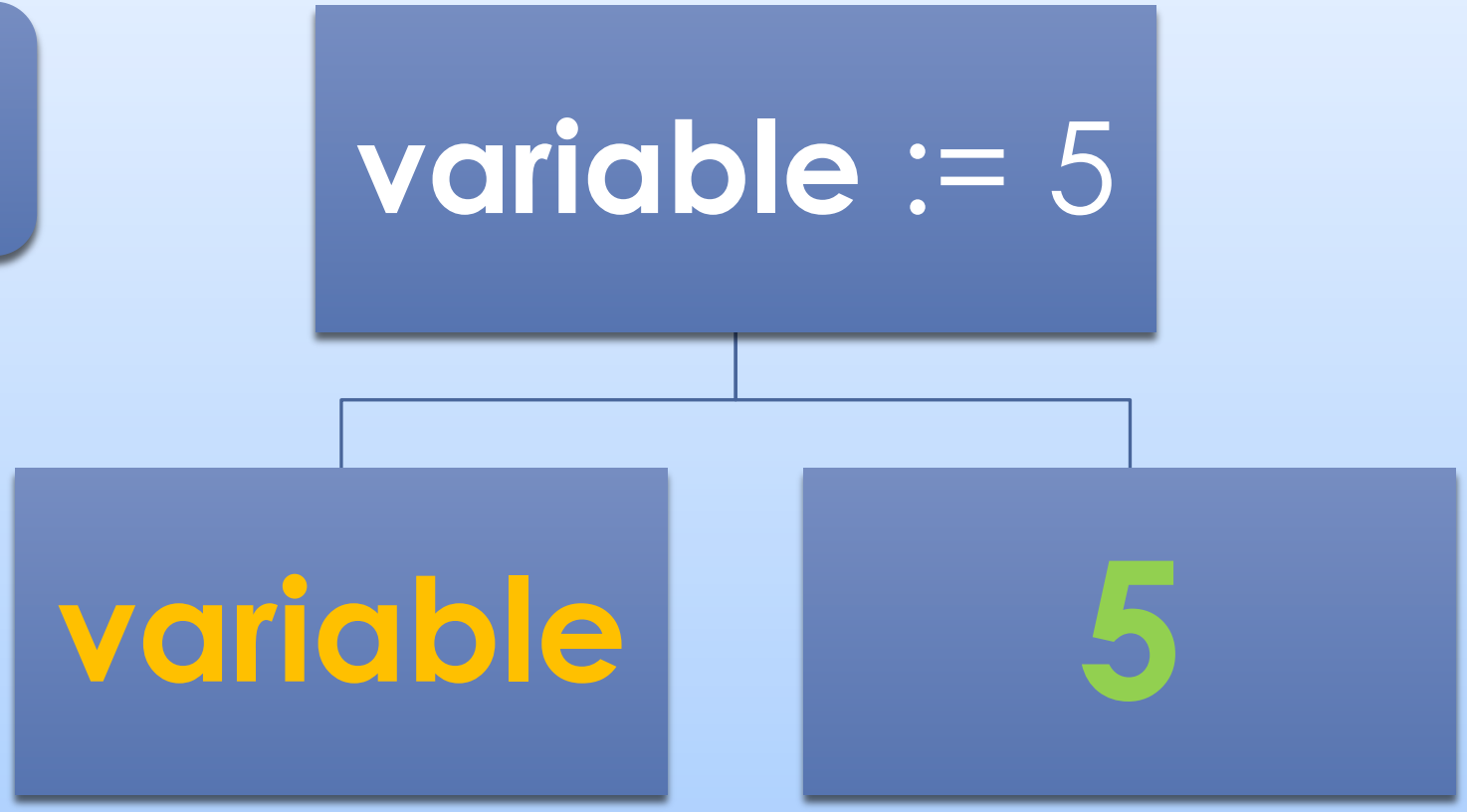


Phorms.
Matching **Example 2**

```
1)  pattern := #var any, 5 equals.
2)  ast := RBParser parseExpression:
3)                          'variable := 5'.
4)  pattern match: ast.
```

Phorms. Matching **Example 2**

#var **any**, **5 equals**



Phorms.
Rewriting

rewritingPattern **rewrite:** anObject

Rewriting pattern = match + transform.

Phorms.
Rewriting **Example**

1) **match** := **#x any**, **#y any**.
2) **transform** := [:it :context | **it y** @ **it x**].
4) pattern := **match ==> transform** .
5) pattern **rewrite: 1@2**.

Result: **(2@1)**

Phorms. **Conclusion**

1. Rewrite **anything**!
2. **Patterns** are objects (**compose** & **convert**).
3. User-friendly **debug** & **inspect**.
4. Easily **extendable**

1. **Not** (yet!) **AST focused**.
2. Sometimes **bulky code**.


Q&A