



Code Transformation by Direct Transformation of ASTs

M. Rizun, J.-C. Bach, S. Ducasse

Example of simple code transformation

Source code

```
foo
  | result a b |
  a := 2.
  b := 3.
  result := a >= b
    ifTrue: [ a ]
    ifFalse: [ b ]
```

transformation

Resulting code

```
foo
  | result a b |
  a := 2.
  b := 3.
  result := a max: b
```

Manual code transformation process (in big system)

Definition of the code to change. (LHS)



Definition of the desired target code. (RHS)



Search for all occurrences of the code to replace in the source code.



Replace all found occurrences by target code.

Find & Replace

Left-Hand Side part

“...code...”

`x >= y`

`ifTrue: [x]`

`ifFalse: [y]`

“...code...”

Right-Hand Side part

“...code...”

`x max: y`

“...code...”

Problems

Time-consuming and annoying task.

Complex and error-prone process.

Rules are not reusable.

Tools to automate process

Tool

Create transformation rules.

Apply rules on desired environment scope.

Save rules for later reuse.

Rewrite Engine



Pros

Engine for code transformation

Unintuitive
syntax & no GUI
to apply rules

Cons

“Scary” transformation rule

Left-Hand Side part

```
| `@temporaries |  
`.InitializationStatement.  
`@condition1  
ifTrue: [  
  | `@otherTemporaries |  
  `@.Statements.  
  `@condition2  
  ifTrue: [ ``@value := ``@calculate ]  
  ifFalse: [ ``@value := `#wrongLiteral ] ]
```

Right-Hand Side part

```
| `@temporaries `@otherTemporaries |  
`.InitializationStatement.  
`@condition1  
  ifFalse: [ ^self ].  
`@.Statements.  
``@value := `@condition2  
ifTrue: [ ``@calculate ]  
ifFalse: [ `#wrongLiteral ]
```


Inattention causes impactful mistakes!



`.InitializationStatement`



`InitializationStatement`

How to apply rules?



NO
YOU
CAN'T

OR

```
| rule result aCollection anEnvironment |  
aCollection := self environment.  
anEnvironment := RBClassEnvironment  
                classes: aCollection.  
rule := (Smalltalk globals at: self rules  
selectedItem) new.  
result := RBSmalllintChecker  
          runRule: rule  
          onEnvironment: anEnvironment.  
(ChangesBrowser changes: result builder  
changes) open
```

Rewrite Tool



- ▶ Intuitive GUI
- ▶ Real time result
- ▶ Apply rules to any scope you want
- ▶ Save rules for later reuse

- A. Input code**
- B. Resulting code**
- C. LHS part of rule**
- D. RHS part of rule**

The screenshot shows the RewriteRuleBuilder application interface. It is divided into several sections:

- Input (A):** A text area containing the code:

```
| result a b |  
result := a < b  
  ifTrue: [ a ]  
  ifFalse: [ b ]
```
- Output (B):** A text area containing the transformed code:

```
| result a b |  
result := a max: b
```
- Transformation Rule (C and D):** A central panel with two text areas and control buttons.
 - C (LHS):**

```
| ``@temporaries1 |  
``@variable1 := ``@variable2 < ``@variable3  
  ifTrue: [ ``@variable2 ]  
  ifFalse: [ ``@variable3 ]
```
 - D (RHS):**

```
| ``@temporaries1 |  
``@variable1 := ``@variable2 max: ``@variable3
```
 - Buttons:** Generate rule, Undo, Redo, Browse rules.

Arrows indicate the flow: a green arrow labeled "Input" points from A to C, and a grey arrow labeled "Output" points from D to B. A grey arrow also points from C to D.



RewriteRuleBrowser

ast

Classes

Methods

- AST-Core
- AST-Interpreter-Core
- AST-Interpreter-Test
- AST-Tests-Core
- FontInfrastructure
- FontInfrastructure-Tests

- ASTCache
- CompiledMethod
- Fraction class
- Integer class
- ManifestASTCore
- Number class
- NumberParser
- RBlockArgumentNode
- RBlockArrayNode
- RBlockAssignmentNode
- RBlockAssignmentToken
- RBlockBinarySelectorToken
- RBlockBlockNode
- RBlockBlockReplaceRule
- RBlockCascadeNode
- RBlockClassReference
- RBlockConfigurableFormatter
- RBlockErrorToken
- RBlockExplicitVariableParser
- RBlockFindBlockArgumentAssignment

hasTemporaryNamed:
isInlinedLoop
right:
specialCommands
children
isBlock
colons:
isImmediateNode
references:
scope
isLast:
precedence
replaceNode:withNode:
uses:
defines:
bar:
postCopy
scope:
match:inContext:
needsParenthesis
methodOrBlockNode
hasArgumentNamed:

Apply rule:

RBEqualNilRule

Only custom rules

Browse rule

Create new rule

Live Demo

Download & Info

➤ <http://smalltalkhub.com/#!/~MarkRizun/RewriteTool>

➤ Configuration Browser (Pharo4)

➤ <http://myfuncoding.com>

➤ Pharo For The Enterprise (RewriteTool chapter)

<https://github.com/SquareBracketAssociates/PharoForTheEnterprise-english/tree/master/RewriteTool>

