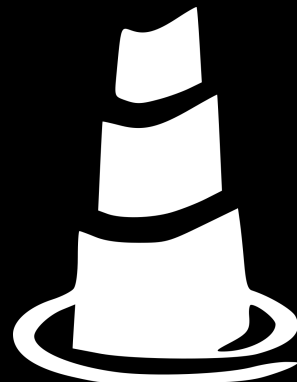
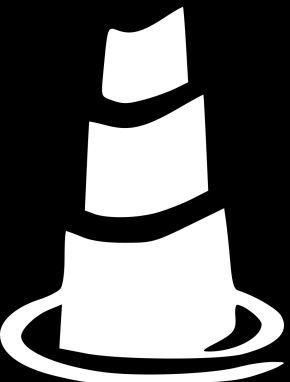
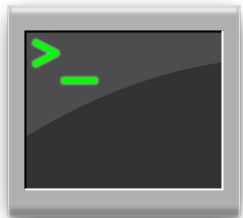


# Towards a Taxonomy of Grammar Smells



V. Zaytsev @ SLE'17 @ SPLASH

joint work with M. Stijlaart



# Mäntylä-Vanhanen-Lassenius Taxonomy of Bad Smells

- Bloaters
- OO abusers
- Change preventers
- Dispensables
- Encapsulators
- Couplers
- Others
  - incomplete + comments

# Our Taxonomy

- Organisation
  - Convention
  - Notation
  - Parsing
  - Duplication

- Navigation
  - Spaghetti
  - Shortage
  - Mixture

- Structure
  - Proxy
  - Dependency
  - Complexity

# Grammar Smells

<b>Misformat</b>	<b>Underuse</b>	<b>Priorities</b>	<b>Combo</b>	<b>Deprecated</b>	<b>Almost</b>	<b>Uncluster</b>
a: b; c: d;	args: arg+ stmts: stmt   stmt stmts	xor: and   xor:"^"and; and: cmp   and:"&"cmp;	a : b? b : c?	a : [b];	D: [1-9]+; L: [b-zA-Z]	a: "A" b; c: "C" d; d: "D" c; b: "B" a;
<b>Misnomer</b>	<b>Overspec</b>	<b>Singleton</b>	<b>Chant</b>	<b>Exotic</b>	<b>Confusing Entry</b>	<b>Unsequence</b>
stmt ::= IF condition _THEN b;	a: ! "b" & "c";	foo: one of bar	D : [0-9]; L : any letter	syntax A = "a" > "b";	an: "a"+ bn bn: "b"+ cn cn: "c"+ an	a: b c; c: d e; d: f e; b: e f;
<b>SayMyName</b>	<b>Factoring</b>	<b>Shotgun</b>	<b>Echo</b>	<b>Common</b>	<b>Dead</b>	<b>MiddleStart</b>
statement seperate argument grammar	if: IF c TH s  IF c TH s EL s	args: ![]+;	foo: bar; bar: b a r; foo: bar;	a: b c d; e: b c f; g: b c h;	foo: bar; z: foo+;	an: "a"+; bn: "b"+; S:an bn cn; cn: "c"+;
<b>ZigZag</b>	<b>Recursion</b>	<b>NoDefault</b>	<b>Clone</b>	<b>Permuted</b>	<b>Bottom</b>	<b>BadLayout</b>
a ::= b c; a ::= d;	exp: exp op exp /case /if;	stmt: map (stmt /if;	case:"CASE" cof:c "OF" c: "CASE"	foo: a   b; bar: b   a;	Int ::= D+;	L:[\n\t\ ]+   "/" . * \$
<b>Splat</b>	<b>Superset</b>	<b>Action</b>	<b>Foldable</b>	<b>Lookalike</b>	<b>Debt</b>	<b>Preprocessor</b>
a ::= b c; x ::= y z; a ::= d e;	Id: [a-z]+	a: b {c()};	case:"CASE" cof:"CASE" "OF"	if: "IF" ifB "END" do: "DO" doB "END"	Expr ::= E; // TODO	pragma: "#" ppword ppcontent "#end";
<b>Ouroboros</b>	<b>Soulmates</b>	<b>Spillover</b>	<b>Mythic</b>	<b>Chain</b>	<b>Throwaway</b>	<b>Weak</b>
a : b c; b : d e; d : f g; f : a h;	co-changed	block: (stmt ";" )+;	never used	condstmt: ifstmt; ifstmt; ifstmt: "IF" ... ;	used once	if: ifstart ifmid ifend; ifstart: "IF" "("; ifmid: cond ")" "{";
<b>Rivalry</b>	<b>2Modular</b>	<b>2Ramose</b>	<b>2Recursive</b>	<b>2Nested</b>	<b>2Coupled</b>	<b>Ghost</b>
name : id   stdname;	too many modules	high cyclomatic complexity	too many recursive nonterminals	too many subgroups	high coupling low cohesion	f: Id(" .Id)+ "(" args )";
<b>Diamond</b>	<b>2Tall</b>	<b>Lonely</b>	<b>Greedy</b>	<b>Lazy</b>	<b>2Wide</b>	<b>Multitool</b>
t : l   r; l : b; r : b;	some nonterminals are too far from root	should be split into several	some modules do too do too much little	some modules rhs too long	decl: name ":" name;	



Interactive at <http://slebok.github.io/grass>

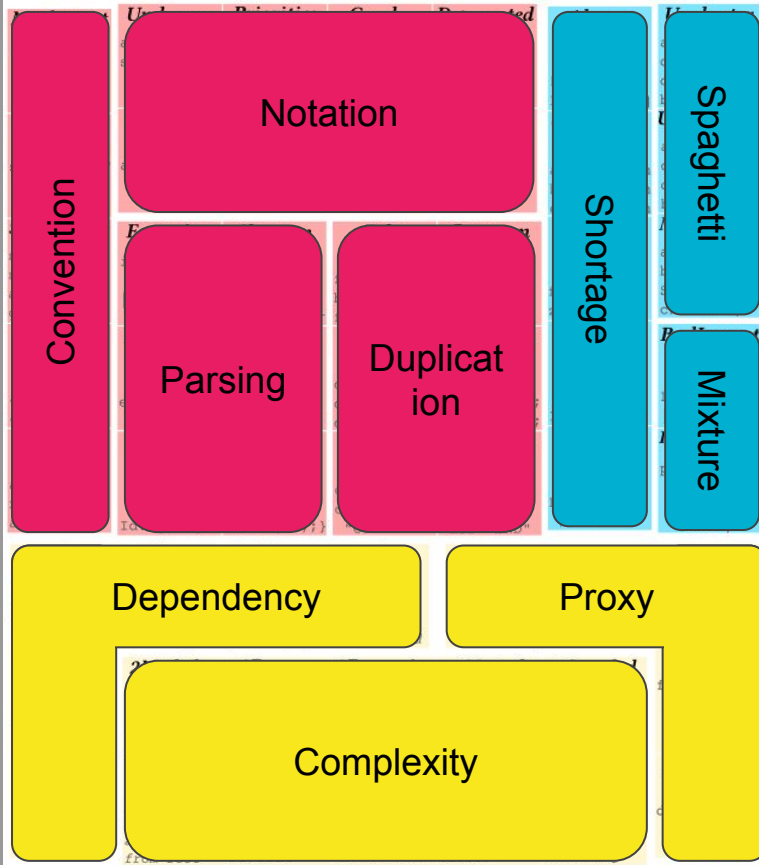
Mats Stijlaart & Vadim Zaytsev

raincode LABS  
compiler experts

Organisation

Navigation

# Grammar Smells



Structure

<http://slebok.github.io/grass>



# / Organisation / Convention

- Misformat
  - misleading formatting
- Misnomer
  - problems with naming policies
- SayMyName
  - a misspelling or improper use of the natural language
- ZigZag
  - mixing horizontal and vertical styles
- Splat
  - nonterminal fragments scattered over the grammar

## Grammar Smells

<b>Underuse</b> case: "CASE" of: "OF" end: "END"	<b>Priorities</b> case: "CASE" of: "OF" end: "END"	<b>Combo</b> case: "CASE" of: "OF" end: "END"	<b>Deprecated</b> case: "CASE" of: "OF" end: "END"	<b>Almost</b> case: "CASE" of: "OF" end: "END"	<b>Uncluster</b> case: "CASE" of: "OF" end: "END"
<b>Overpec</b> case: "CASE" of: "OF" end: "END"	<b>Singleton</b> case: "CASE" of: "OF" end: "END"	<b>Chant</b> case: "CASE" of: "OF" end: "END"	<b>Exotic</b> case: "CASE" of: "OF" end: "END"	<b>Confusing Entry</b> case: "CASE" of: "OF" end: "END"	<b>Unsequence</b> case: "CASE" of: "OF" end: "END"
<b>Factoring</b> case: "CASE" of: "OF" end: "END"	<b>Shotgun</b> case: "CASE" of: "OF" end: "END"	<b>Echo</b> case: "CASE" of: "OF" end: "END"	<b>Common</b> case: "CASE" of: "OF" end: "END"	<b>Dead</b> case: "CASE" of: "OF" end: "END"	<b>MiddleStart</b> case: "CASE" of: "OF" end: "END"
<b>Recursion</b> case: "CASE" of: "OF" end: "END"	<b>NoDefault</b> case: "CASE" of: "OF" end: "END"	<b>Clone</b> case: "CASE" of: "OF" end: "END"	<b>Permuted</b> case: "CASE" of: "OF" end: "END"	<b>Bottom</b> case: "CASE" of: "OF" end: "END"	<b>BadLayout</b> case: "CASE" of: "OF" end: "END"
<b>Superset</b> case: "CASE" of: "OF" end: "END"	<b>Action</b> case: "CASE" of: "OF" end: "END"	<b>Foldable</b> case: "CASE" of: "OF" end: "END"	<b>Lookalike</b> case: "CASE" of: "OF" end: "END"	<b>Debt</b> case: "CASE" of: "OF" end: "END"	<b>Propagator</b> case: "CASE" of: "OF" end: "END"
<b>Rivalry</b> case: "CASE" of: "OF" end: "END"	<b>2Modular</b> case: "CASE" of: "OF" end: "END"	<b>2Ramose</b> case: "CASE" of: "OF" end: "END"	<b>2Recursive</b> case: "CASE" of: "OF" end: "END"	<b>2Nested</b> case: "CASE" of: "OF" end: "END"	<b>2Coupled</b> case: "CASE" of: "OF" end: "END"
<b>Ouroboros</b> case: "CASE" of: "OF" end: "END"	<b>Soulmates</b> case: "CASE" of: "OF" end: "END"	<b>Spillover</b> case: "CASE" of: "OF" end: "END"	<b>Mythic</b> case: "CASE" of: "OF" end: "END"	<b>Chain</b> case: "CASE" of: "OF" end: "END"	<b>Throwaway</b> case: "CASE" of: "OF" end: "END"
<b>Diamond</b> case: "CASE" of: "OF" end: "END"	<b>2Tall</b> case: "CASE" of: "OF" end: "END"	<b>Lonely</b> case: "CASE" of: "OF" end: "END"	<b>Greedy</b> case: "CASE" of: "OF" end: "END"	<b>Lazy</b> case: "CASE" of: "OF" end: "END"	<b>2Wide</b> case: "CASE" of: "OF" end: "END"
<b>Multitool</b> case: "CASE" of: "OF" end: "END"					

# / Organisation / Convention / Misformat

abc :

def;

ghi:

Jkl;

mno :

pqr;



# / Organisation / Notation

- Underuse
- Overspec
- Priorities
- Singleton
- Combo
- Chant
- Deprecated
- Exotic

## Grammar Smells



Misformat					Almost	Uncluster
at					at [1-9]+	at [1-9]+
br					at [1-9]+	at [1-9]+
or					at [1-9]+	at [1-9]+
di					at [1-9]+	at [1-9]+
Misnamer					at [1-9]+	at [1-9]+
stat := IF					at [1-9]+	at [1-9]+
condition					at [1-9]+	at [1-9]+
code b:					at [1-9]+	at [1-9]+
SayMyName	Factoring	Shotgun	Echo	Common	Dead	MiddleStart
statement	if				foo bar;	foo bar;
expression	if c TH e				foo bar;	foo bar;
argument	if c TH e				foo bar;	foo bar;
expression	if c TH e				foo bar;	foo bar;
ZigZag	Recursion	NoDefault	Clone	Permuted	Bottom	BadLayout
# := b e						
# := d						
exp := exp						
exp := exp						
Splat	Superset	Action	Foldable	Lookalike	Debt	Propagator
# := b e						
# := d						
# := e a						
# := d e						
Ouroboros	Soulmates	Spillover	Mythic	Chain	Throwaway	Weak
a   b   c						
b   d   e						
d   f   g						
f   a   b						
Rivalry	2Modular	2Ramose	2Recursive	2Nested	2Coupled	Ghost
name   id						
too many						
modules						
Diamond	2Tall	Lonely	Greedy	Lazy	2Wide	Multitool
t   l   f						
l   b						
f   b						



# / Organisation / Parsing / 1SidedRecursion

```
define expression
    [expression] [op] [expression]
|   [id] [expression+]
|   if [expression] then [expression] else [expression]
|   ( [expression] )
|   [id]
|   [number]
end define
```

# / Organisation / Duplication

- Echo
  - same rule occurs twice or more
- Clone
  - same right hand side occurs twice or more
- Foldable
  - nonterminal's right hand side is not unique
- Common
  - common subexpressions
- Permuted
  - permutations in commutative combinators
- Lookalike
  - near-clones

## Grammar Smells

<b>Misformat</b>	<b>Underuse</b>	<b>Priorities</b>	<b>Combo</b>	<b>Deprecated</b>	<b>Almost</b>	<b>Uncluster</b>
<b>Manamer</b>	<b>Oversepec</b>	<b>Singleton</b>	<b>Chant</b>	<b>Exotic</b>	<b>Confusing Entry</b>	<b>Unsequence</b>
<b>SkMyName</b>	<b>Factoring</b>	<b>Shotgun</b>			<b>Dead</b>	<b>MiddleStart</b>
<b>ZipZag</b>	<b>Recursion</b>	<b>NoDefault</b>			<b>Bottom</b>	<b>BadLayout</b>
<b>Splat</b>	<b>Superset</b>	<b>Action</b>			<b>Debt</b>	<b>Propocessor</b>
<b>Ouroboros</b>	<b>Soulmates</b>	<b>Spillover</b>	<b>Mythic</b>	<b>Chain</b>	<b>Throwaway</b>	<b>Weak</b>
<b>Rivalry</b>	<b>2Modular</b>	<b>2Ramose</b>	<b>2Recursive</b>	<b>2Nested</b>	<b>2Coupled</b>	<b>Ghost</b>
<b>Diamond</b>	<b>2Tall</b>	<b>Lonely</b>	<b>Greedy</b>	<b>Lazy</b>	<b>2Wide</b>	<b>Multitool</b>

Interactive at <http://sleebk.github.io/grass>

Mats Stijlaart & Vadim Zaitsev

raincode LABS

# / Navigation

## Spaghetti

- Uncluster
- Unsequence
- StartInTheMiddle

## Mixture

- BadLayout
- Preprocessor

## Shortage

- AlmostAlphabet
- ConfusingEntry
- Dead
- Bottom
- Debt

## Grammar Smells

The image shows a grid of code snippets illustrating various 'Grammar Smells'. The grid is organized into several rows and columns, each with a title and a corresponding code snippet. The titles include: Misformat, Underuse, Priorities, Combo, Deprecated, Misanomer, Overspec, Singleton, Chant, Exotic, SxMyName, Factoring, Shotgun, Echo, Common, ZigZag, Recursion, NoDefault, Clone, Permuted, Split, Superset, Action, Foldable, Lookalike, Ouroboros, Soulmates, Spillover, Mythic, Chain, Throwway, Weak, Rivalry, 2Modular, 2Ramose, 2Recursive, 2Nested, 2Coupled, Ghost, Diamond, 2Tall, Lonely, Greedy, Lazy, 2Wide, Multitool. The code snippets are small and difficult to read, but they illustrate various patterns and smells. At the bottom of the grid, there is a footer with the text: 'Interactive at <http://sleebok.github.io/grass> raincode LABS Mats Stijlaert & Vadim Zaytsev'.





# / Structure / Dependency / Diamond

```
reference-type ::= class-type | interface-type  
                | delegate-type | ... ;
```

```
class-type ::= type-name | "object" | "string";
```

```
interface-type ::= type-name ;
```

```
delegate-type ::= type-name ;
```



# / Structure / Complexity

- TooWide
  - right hand side length too big
- TooRamose
  - too much branching
- TooRecursive
  - too much recursion
- TooNested
  - groups are too deeply nested
- TooTall
  - distance from the root too big
- Lonely
  - large non-modular grammar

- TooModular
  - too many modules
- Greedy
  - one module that does too much
- Lazy
  - vice versa
- TooCoupled
  - low cohesion, high coupling

*Grammar Smells*

Misformat	Underuse	Priorities	Combo	Deprecated	Almost	Uncluster
Misnamer	OverSpec	Singleton	Chant	Exotic	Confusing Entry	Unsequence
SayMyName	Factoring	Shotgun	Echo	Common	Dead	MiddleStart
ZipZag	Recursion	NoDefault	Clone	Permuted	Bottom	BadLayout
Splat	Superset	Action	Foldable	Lookalike	Debt	Propagator
Ouroboros	Soulmates	Spillover	Mythic	Chain	Throwaway	Weak
Rivalry						Ghost
Diamond						Multitool

Interactive at <http://sleebok.github.io/grass>

Mats Stijlaert & Vadim Zaytsev

raincode LABS

# What do we know about smells?

- Smells are poor solutions to recurring problems [1]
- Smells are in between design and implementation [1]
- System experts detect smells worse than outsiders [2]
- Detection should be based on metrics [3]
- Systems with smells are worse [\*]
- Effect of smells is negligible compared to size [4]
- Trial-and-error maintenance leads to smells [5]
- Devs rely on smell detectors in unfamiliar contexts [6]

[1] Khomh, Di Penta, Guéhéneuc, WCRE 2006

[2] Mäntylä, Vanhanen, Lassenius, ICSM 2003, ICSM 2004

[3] Marinescu, TOOLS 2001, ICSM 2004

[4] Sjøberg, Yamashita, Anda, Mockus, Dybå, TSE 2012

[5] Al-Kofahi, Kothari, Kästner, GPCE 2017

[6] Sousa, Oliveira, Garcia, Lee, Conte, Oizumi, de Mello, Lopes, Valentim, Oliveira, Lucena, SBES'17

**It is our responsibility to learn more**



<http://slebok.github.io/grass>

