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

Misformat
```
a:
   b;
   c:
   d;
```

Underuse
```
args: arg+
stmts:
 stmt |
 stmt stmts
```

Priorities
```
xor: and |
 xor"^"and;
and: cmp |
 and"&"cmp;
```

Combo
```
a : b?
b : c?
```

Deprecated
```
a : [b];
```

Almost
```
D: [1-9]+;
L: [b-zA-Z]
```

Uncluster
```
a: "A" b;
c: "C" d;
d: "D" c;
b: "B" a;
```

Misnomer
```
stmt ::= IF
 condition
 _THEN b;
```

Overspec
```
a:
  ! "b"
  & "c";
```

Singleton
```
foo: one of
 bar
```

Chant
```
D : [0-9];
L :
 any letter
```

Exotic
```
syntax A
 = "a"
 > "b";
```

Confusing Entry
```
an: "a"+ bn
bn: "b"+ cn
cn: "c"+ an
```

Unsequence
```
a: b c;
c: d e;
d: f e;
b: e f;
```

SayMyName
```
staetment
seperate
arguement
grammer
```

Factoring
```
if:
 IF c TH s
|IF c TH s
  EL s
```

Shotgun
```
args: ![)]+
```

Echo
```
foo: bar;
bar: b|a|r;
foo: bar;
```

Common
```
a: b c d;
e: b c f;
g: b c h;
```

Dead
```
foo: bar;
z: foo+;
```

MiddleStart
```
an: "a"+;
bn: "b"+;
S:an bn cn;
cn: "c"+;
```

ZigZag
```
a ::= b|c;
a ::= d;
```

Recursion
```
exp:
 exp op exp
```

NoDefault
```
stmt: map
 /case /if;
```

Clone
```
case:"CASE"
cof:c "OF"
c: "CASE"
```

Permuted
```
foo: a | b;
bar: b | a;
```

Bottom
```
Int ::= D+;
```

BadLayout
```
L:[\n\t\ ]+
| "//" .* $
```

Splat
```
a ::= b c;
x ::= y z;
a ::= d e;
```

Superset
```
Id: [a-z]+
```

Action
```
a: b {c();}
```

Foldable
```
case:"CASE"
cof:"CASE"
 "OF"
```

Lookalike
```
if: "IF"
 ifB "END"
do: "DO"
 doB "END"
```

Debt
```
Expr := ε;
 // TODO
```

Preprocessor
```
pragma:
 "#" ppword
 ppcontent
 "#end";
```

Ouroboros
```
a : b c;
b : d e;
d : f g;
f : a h;
```

Soulmates

co-changed

Spillover
```
block:
(stmt ";")+
```

Mythic

never used

Chain
```
condstmt:
  ifstmt;
ifstmt:
  "IF" … ;
```

Throwaway

used once

Weak
```
if: ifstart
 ifmid ifend;
ifstart:
 "IF" "(";
ifmid:
 cond ")" "{";
```

Rivalry
```
name : id
 | stdname;
```

2Modular

too many modules

2Ramose

high cyclomatic complexity

2Recursive

too many recursive nonterminals

2Nested

too many subgroups

2Coupled

high coupling low cohesion

Ghost
```
f:
 Id("."Id)+
 "(" args
 ")";
```

Diamond
```
t : l | r;
l : b;
r : b;
```

2Tall

some nonterminals are too far from root

Lonely

should be split into several

Greedy

some modules do too much

Lazy

some modules do too little

2Wide

rhs too long

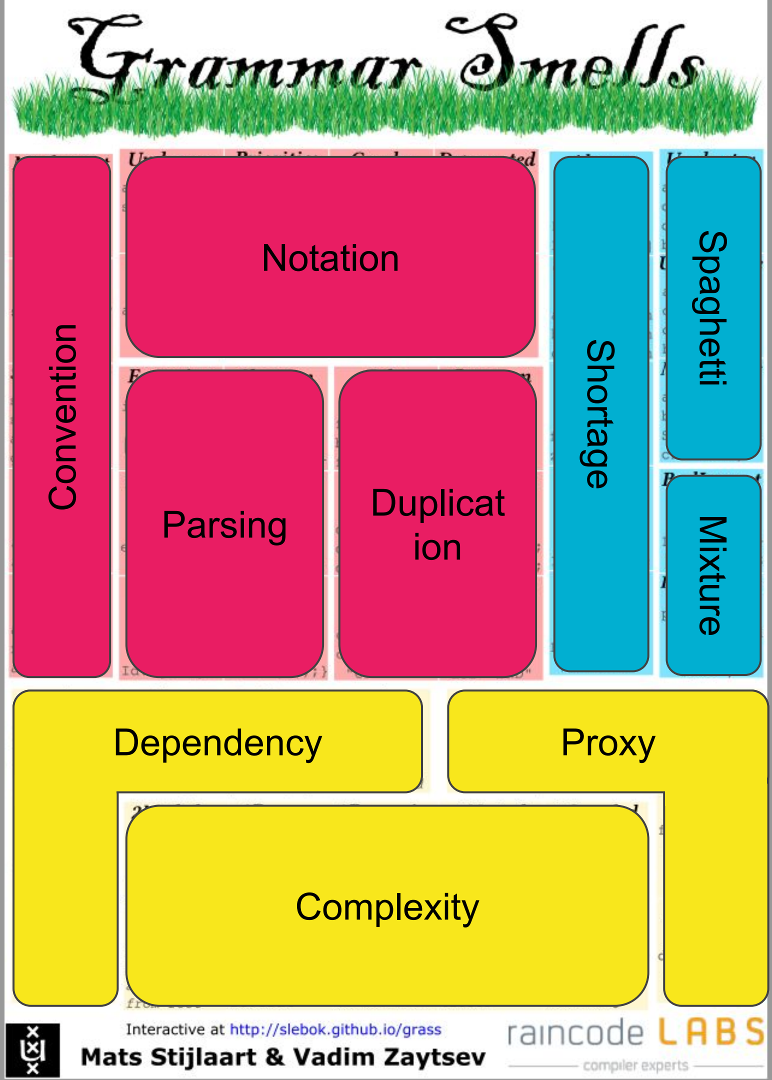
Multitool
```
decl:
 name ":"
 name;
```

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

Mats Stijlaart & Vadim Zaytsev

Organisation

Navigation



Grammar Smells

Convention

Notation

Parsing

Duplication

Shortage

Spaghetti

Mixture

Dependency

Proxy

Complexity

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

Mats Stijlaart & Vadim Zaytsev

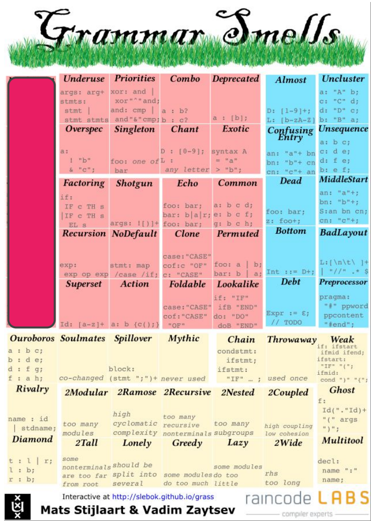
raincode LABS compiler experts

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

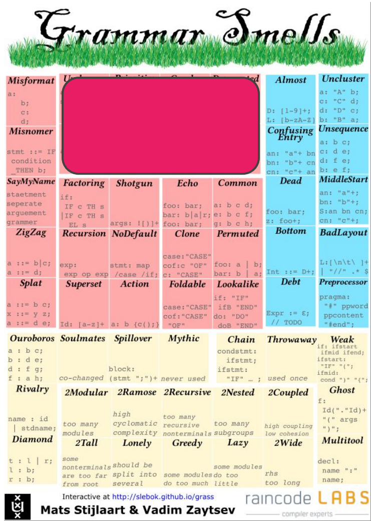


/ Organisation / Convention / Misformat

```
abc :
          def;
          ghi:
          Jkl;
mno :
          pqr;
```

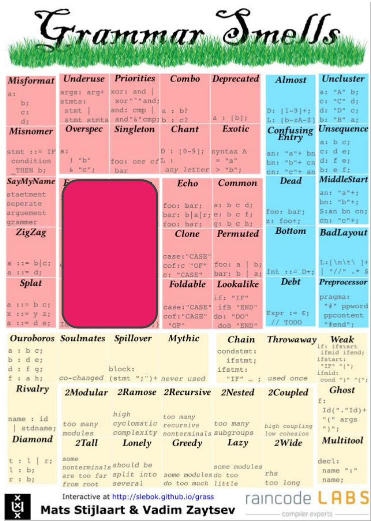
/ Organisation / Notation

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



/ Organisation / Parsing

- Factoring
- 1SidedRecursion
- Superset
- Shotgun
- NoDefault
- Action
  - semantic action is code

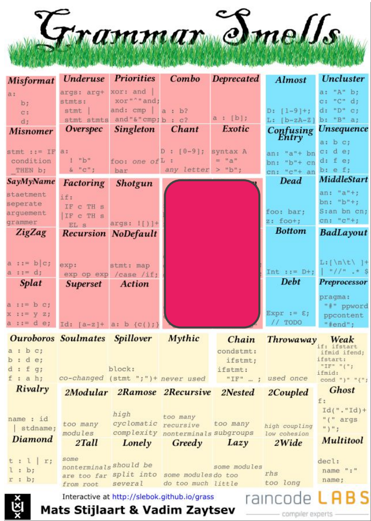


/ Organisation / Parsing / 1SidedRecursion

```
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



/ Navigation

Spaghetti

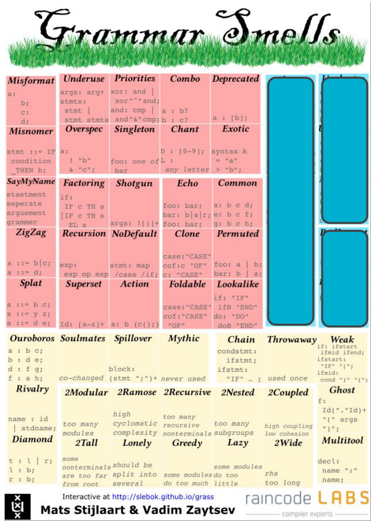
- Uncluster
- Unsequence
- StartInTheMiddle

Mixture

- BadLayout
- Preprocessor

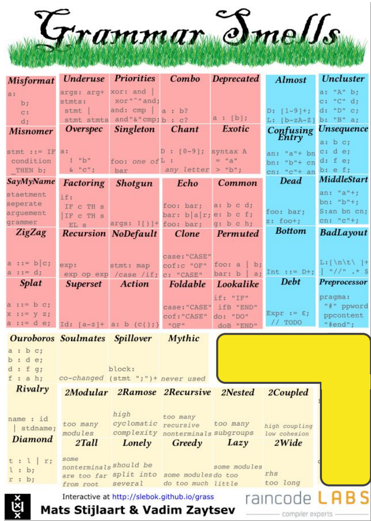
Shortage

- AlmostAlphabet
- ConfusingEntry
- Dead
- Bottom
- Debt


Grammar Smells
Misformat
a: b; c: d;
Underuse
args: arg+ stmts: stmt | stmt stmts
Priorities
xor: and | xor"^"and; and: cmp | and"&"cmp;
Combo
a : b? b : c?
Deprecated
a : [b];
Misnomer
stmt ::= IF condition _THEN b;
Overspec
a: ! "b" & "c";
Singleton
foo: one of bar
Chant
D : [0-9]; L : any letter
Exotic
syntax A = "a" > "b";
SayMyName
staetment seperate arguement grammer
Factoring
if: IF c TH s |IF c TH s EL s
Shotgun
args: ![)]+
Echo
foo: bar; bar: b|a|r; foo: bar;
Common
a: b c d; e: b c f; g: b c h;
ZigZag
a ::= b|c; a ::= d;
Recursion
exp: exp op exp
NoDefault
stmt: map /case /if;
Clone
case:"CASE" cof:c "OF" c: "CASE"
Permuted
foo: a | b; bar: b | a;
Splat
a ::= b c; x ::= y z; a ::= d e;
Superset
Id: [a-z]+
Action
a: b {c();}
Foldable
case:"CASE" cof:"CASE" "OF"
Lookalike
if: "IF" ifB "END" do: "DO" doB "END"
Ouroboros
a : b c; b : d e; d : f g; f : a h;
Soulmates
co-changed
Spillover
block: (stmt ";")+
Mythic
never used
Chain
condstmt: ifstmt; ifstmt: "IF" … ;
Throwaway
used once
Weak
if: ifstart ifmid ifend; ifstart: "IF" "("; ifmid: cond ")" "{";
Rivalry
name : id | stdname;
2Modular
too many modules
2Ramose
high cyclomatic complexity
2Recursive
too many recursive nonterminals
2Nested
too many subgroups
2Coupled
high coupling low cohesion
Ghost
f: Id("."Id)+ "(" args ")";
Diamond
t : l | r; l : b; r : b;
2Tall
some nonterminals are too far from root
Lonely
should be split into several
Greedy
some modules do too much
Lazy
some modules do too little
2Wide
rhs too long
Multitool
decl: name ":" name;
Interactive at http://slebok.github.io/grass
Mats Stijlaart & Vadim Zaytsev
raincode LABS
compiler experts

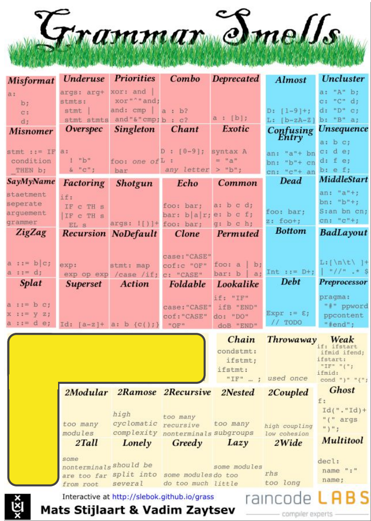
/ Structure / Proxy

- Chain
  - right hand side is a nonterminal
- Throwaway
  - a nonterminal used only once
- Weak
  - a nonterminal does not form a proper abstraction
- Ghost
  - a subexpression that should have been a nonterminal
- Multitool
  - a nonterminal that represents two or more concepts



/ Structure / Dependency

- Diamond
- Rivalry
  - overlapping alternatives
- Ouroboros
  - a circular dependency among modules
- Soulmates
  - nonterminals or modules are always changed together
- Spillover
  - a nonterminal that is too linked to the adjacent symbols
- Mythic
  - an alternative that is never exercised in the codebase



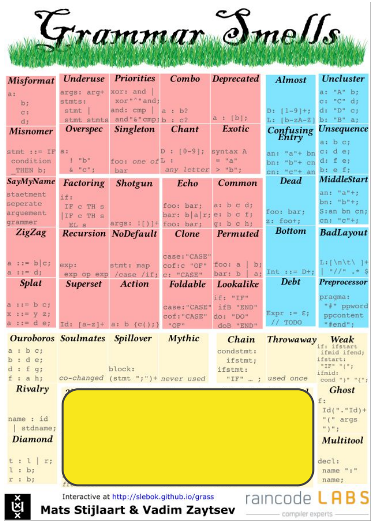
/ Structure / Dependency / Diamond

```
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
```
a:
   b;
   c:
   d;
```
Underuse
```
args: arg+
stmts:
 stmt |
 stmt stmts
```
Priorities
```
xor: and |
 xor"^"and;
and: cmp |
 and"&"cmp;
```
Combo
```
a : b?
b : c?
```
Deprecated
```
a : [b];
```
Almost
```
D: [1-9]+;
L: [b-zA-Z]
```
Uncluster
```
a: "A" b;
c: "C" d;
d: "D" c;
b: "B" a;
```
Misnomer
```
stmt ::= IF
 condition
 _THEN b;
```
Overspec
```
a:
  ! "b"
  & "c";
```
Singleton
```
foo: one of
 bar
```
Chant
```
D : [0-9];
L :
 any letter
```
Exotic
```
syntax A
 = "a"
 > "b";
```
Confusing Entry
```
an: "a"+ bn
bn: "b"+ cn
cn: "c"+ an
```
Unsequence
```
a: b c;
c: d e;
d: f e;
b: e f;
```
SayMyName
```
staetment
seperate
arguement
grammer
```
Factoring
```
if:
 IF c TH s
|IF c TH s
  EL s
```
Shotgun
```
args: ![)]+
```
Echo
```
foo: bar;
bar: b|a|r;
foo: bar;
```
Common
```
a: b c d;
e: b c f;
g: b c h;
```
Dead
```
foo: bar;
z: foo+;
```
MiddleStart
```
an: "a"+;
bn: "b"+;
S:an bn cn;
cn: "c"+;
```
ZigZag
```
a ::= b|c;
a ::= d;
```
Recursion
```
exp:
 exp op exp
```
NoDefault
```
stmt: map
 /case /if;
```
Clone
```
case:"CASE"
cof:c "OF"
c: "CASE"
```
Permuted
```
foo: a | b;
bar: b | a;
```
Bottom
```
Int ::= D+;
```
BadLayout
```
L:[\n\t\ ]+
| "//" .* $
```
Splat
```
a ::= b c;
x ::= y z;
a ::= d e;
```
Superset
```
Id: [a-z]+
```
Action
```
a: b {c();}
```
Foldable
```
case:"CASE"
cof:"CASE"
 "OF"
```
Lookalike
```
if: "IF"
 ifB "END"
do: "DO"
 doB "END"
```
Debt
```
Expr := ε;
 // TODO
```
Preprocessor
```
pragma:
 "#" ppword
 ppcontent
 "#end";
```
Ouroboros
```
a : b c;
b : d e;
d : f g;
f : a h;
```
Soulmates

co-changed

Spillover
```
block:
(stmt ";")+
```
Mythic

never used

Chain
```
condstmt:
 ifstmt;
ifstmt:
 "IF" … ;
```
Throwaway

used once

Weak
```
if: ifstart
 ifmid ifend;
ifstart:
 "IF" "(";
ifmid:
 cond ")" "{";
```
Rivalry
```
name : id
 | stdname;
```
Ghost
```
f:
 Id("."Id)+
 "(" args
 ")";
```
Diamond
```
t : l | r;
l : b;
r : b;
```
Multitool
```
decl:
 name ":"
 name;
```

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

Mats Stijlaart & Vadim Zaytsev

raincode LABS

compiler experts

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