Grammar Investigation

Vadim V. Zaytsev
2011
What to expect

✓ C++ is bigger than C.
✓ C# is more complex than Java.
✓ There are 11 bugs in Rascal.
✓ Modula can have 2 sublanguages.
✓ Fortran and Oberon are equally hard to learn.
✓ It was more difficult to develop Rascal than XPath.
✓ C# grammar is hard to extend, can be improved.
✓ JDK grammars underuse the grammar notation.
What to recall

- Formal grammars
- Complexity theory
- Software metrics
- Mathematical statistics
- Program impurity classes
- Psychiatry
- Software science
- Lorenz curve
- Control flow analysis
- Product quality standard
- Pattern recognition
- Graph theory
Grammar investigation

Grammar  ???  Profit!
Grammar investigation

Grammar → ??? → Profit!

Usually a number

Usually an indication or a forecast
Motivation

✓ Compare size and effort.
✓ Estimate the quality of the grammar.
✓ Predict future complications (detect smells).
✓ Improve grammar quality.
✓ Compare language implementations.
✓ Evaluate productivity impacts of new techniques.
What do metrics measure?

- Length
- Size
- Quality, complexity
  - Language complexity
  - Structural complexity
  - Cognitive complexity, learnability
- Functionality, usability
- Defect density, reliability
- Modularity, coupling/cohesion, reusability
- Nobody knows exactly
### Grammar analysis

<table>
<thead>
<tr>
<th>TERM</th>
<th>VAR</th>
<th>LAB</th>
<th>PROD</th>
<th>DEAD</th>
<th>DEADP</th>
<th>UNDEF</th>
<th>ROOT</th>
<th>LOC</th>
<th>AVSN</th>
<th>AVSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMET</td>
<td>UOPS</td>
<td>MET</td>
<td>OPS</td>
<td>VOC</td>
<td>LEN</td>
<td>LEN^</td>
<td>UOPS*</td>
<td>VOL</td>
<td>PVOL</td>
<td>BVOL</td>
</tr>
<tr>
<td>NPAT</td>
<td>NPATC</td>
<td>MPAT</td>
<td>MPATC</td>
<td>WPAT</td>
<td>FImin</td>
<td>FIavg</td>
<td>FImax</td>
<td>ONCE</td>
<td>FOmin</td>
<td>FOavg</td>
</tr>
<tr>
<td>LEV</td>
<td>CLEV</td>
<td>RLEV</td>
<td>NLEV</td>
<td>HEI</td>
<td>DEP</td>
<td>TIMPI</td>
<td>TIMP</td>
<td>MCC</td>
<td>MI</td>
<td>BUG</td>
</tr>
</tbody>
</table>
## Primitive grammar measurements

<table>
<thead>
<tr>
<th>TERM</th>
<th>VAR</th>
<th>LAB</th>
<th>PROD</th>
<th>DEAD</th>
<th>DEADP</th>
<th>UNDEF</th>
<th>ROOT</th>
<th>LOC</th>
<th>AVSN</th>
<th>AVSP</th>
</tr>
</thead>
</table>

## Metrics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FImin</td>
<td>Minimum Feature Impact</td>
</tr>
<tr>
<td>FIavg</td>
<td>Average Feature Impact</td>
</tr>
<tr>
<td>FImax</td>
<td>Maximum Feature Impact</td>
</tr>
<tr>
<td>FOmin</td>
<td>Minimum Feature Observation</td>
</tr>
<tr>
<td>FOavg</td>
<td>Average Feature Observation</td>
</tr>
<tr>
<td>FOmax</td>
<td>Maximum Feature Observation</td>
</tr>
</tbody>
</table>

## Heights

<table>
<thead>
<tr>
<th>Heights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEI</td>
<td>Height of the tree</td>
</tr>
<tr>
<td>DEP</td>
<td>Depth of the tree</td>
</tr>
</tbody>
</table>

## Timing

<table>
<thead>
<tr>
<th>Timing</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMPI</td>
<td>Timing of the parser</td>
</tr>
<tr>
<td>TIMP</td>
<td>Timing of the parser (microseconds)</td>
</tr>
</tbody>
</table>

## Volume

<table>
<thead>
<tr>
<th>Volume</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVOL</td>
<td>Primary Volume</td>
</tr>
<tr>
<td>BVOL</td>
<td>Backup Volume</td>
</tr>
</tbody>
</table>

## Other Measurements

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>Message Index</td>
</tr>
<tr>
<td>MCC</td>
<td>Main Code Count</td>
</tr>
<tr>
<td>HLEV</td>
<td>Head Level</td>
</tr>
<tr>
<td>HLEV^</td>
<td>Head Level (raised)</td>
</tr>
<tr>
<td>DIF</td>
<td>Difference</td>
</tr>
<tr>
<td>LLEV</td>
<td>Lower Level</td>
</tr>
<tr>
<td>IC</td>
<td>Information Content</td>
</tr>
<tr>
<td>EFF</td>
<td>Efficiency</td>
</tr>
<tr>
<td>BUGS</td>
<td>Bugs</td>
</tr>
</tbody>
</table>

---

The table above lists various primitive grammar measurements, including terms, variables, labels, productions, dead states, dead productions, undefined states, roots, locations, average symbol name, average symbol position, feature impact minimum, average, maximum, feature observation minimum, average, maximum, tree height, tree depth, timing of the parser, timing of the parser (microseconds), primary volume, backup volume, message index, main code count, head level, head level (raised), difference, lower level, information content, efficiency, and bugs.
TERM: number of terminal symbols

- ✓ Solid size metric
- ✓ Easy to compute (traversal needed)
- ✓ Almost no correlation with any other metrics (except, quite surprisingly, for NPAT)
- ✓ TERM ≡ 0 for some meta-syntaxes (XSD, EMF)
VAR: number of nonterminal symbols

✓ Best to add the number of built-in primitives
✓ Solid size metric
✓ Easy to compute (traversal needed)
✓ Extremely high correlation with most size metrics
✓ Seems like a proper target for normalisations (except it is not, $r = 0.9783$)
✓ Claims that “larger VAR implies greater maintenance overhead”
LAB: number of descriptive labels

- Expression selectors and production labels
- More of a documentation metric
- Does it capture readability?
- Easy to compute (traversal needed)
- Being the only documentation metric, does not correlate with anything
**PROD**: number of production rules

- ✔ Trivial to compute (no traversal)
- ✔ Conceptually different from VAR, but always correlates heavily ($r = 0.9890$)
- ✔ It is known that: $\text{VAR} \leq \text{PROD}$
DEAD: number of dead nonterminals

✓ Nonterminal symbols unreachable from the root
✓ Easy to compute (traversal)
DEADP: number of dead productions

✓ Production rules unreachable from the root
✓ Relatively easy to compute (traversal)
✓
UNDEF: number of bottom nonterminals

✓ Nonterminals that are used but not defined
✓ Relatively easy to compute (traversal)
✓
ROOT: number of start symbols

- In theory, one and only one start
- In practice, multiple or none are possible
- Trivial to compute (no traversal)
LOC: lines of EBNF code

Following LOC counting traditions

Secondary metric computed as:

\[ \text{LOC} = \text{VAR} + \text{PROD} \]

...
AVSN: average right hand side size

✓ Per nonterminal symbol
✓ Relatively easy to compute
AVSP: average right hand side size

- Per production rule
- Relatively easy to compute
UMET: unique meta-symbols

✓ Tells more about grammar notation
✓ Or about the extent to which notation is exercised
✓ For the notation, there exists UMET:

\[ 2 \leq \text{UMET} \leq \text{UMET} \]

✓ …
UOPS: unique operands

✓ Can be computed as:

\[ UOPS = \text{VAR} + \text{TERM} + \text{LAB} \]

✓ There exists UOPS*:

\[ UOPS* \leq UOPS \]

✓ …
**UOPS*: minimum required operands**

✓ Can be computed as:

\[ \text{UOPS*} = \text{TERM} + \text{ROOT} + \text{UNDEF} \]

✓ If the above expression is zero, 2nd assumption:

\[ \text{UOPS*} = \text{DEAD} \]

✓ …
MET: used metasymbols

✓ Number of applications of sequential composition, repetition, optionality, …

✓ Known property:

\[ \text{UMET} \leq \text{MET} \]

✓ …
OPS: used operands

✓ Number of occurrences of nonterminals, terminals, labels, …

✓ Known property:

\[ \text{UOPS} \leq \text{OPS} \]

✓ …
VOC: grammar vocabulary

Can be computed as:

\[ \text{VOC} = \text{UMET} + \text{UOPS} \]
LEN: grammar length

✓ Can be computed as:

LEN = MET + OPS

✓ …
PUR: purity ratio

Can be computed as:

\[ \text{PUR} = \frac{\overline{\text{LEN}}}{\text{LEN}} \]

...
VOL: grammar volume

✓ Can be computed as:

\[ \text{VOL} = \text{LEN} \log_2 \text{VOC} \]

✓ …
PVOL: potential (minimal) volume

✓ Can be computed as:

\[ PVOL = (2 + UOPS^*) \log_2 (2 + UOPS^*) \]

✓ …
BVOL: boundary volume

Can be computed as:

\[ \text{BVOL} = (2 + \text{UOPS} \times \log_2 \text{UOPS}^*) \log_2 (2 + \text{UOPS}^*) \]

...
HLEV: grammar level

✓ Can be computed as:

HLEV: PVOL/VOL

✓ Known property:

0 \leq HLEV \leq 1

✓ …
HLEV: estimated grammar level

✓ Can be computed as:

\[
\text{HLEV}: \frac{2 \times \text{UOPS}}{\text{UMET} \times \text{OPS}}
\]

✓ …
DIF: difficulty

✓ Can be computed as:

\[
DIF = \frac{1}{HLEV}
\]

✓ …
LLEV: meta-language level

Can be computed as:

\[ \text{LLEV} = \text{HLEV} \times \text{PVOL} \]

- For English: 2.16
- For Algol: 1.21
- For Assembly: 0.88
- For BNF: 0.00002–0.00437
EFF: engineering effort

✓ Can be computed as:

\[
EFF = \frac{VOL}{HLEV}
\]
EFF: estimated engineering effort

✓ The most commonly used metric “by Halstead”

✓ Was not suggested by Maurice Halstead.

✓ Computed as:

\[
\hat{\text{EFF}} = \frac{\text{VOL}}{\text{HLEV}} = \frac{\text{UMET} \times \text{OPS} \times \text{LEN} \times \log_2 \text{VOC}}{2 \times \text{UOPS}}
\]

✓ …
BUG: estimated number of errors

✓ Can be computed as:

\[ \text{BUG} = \frac{\text{EFF}^{2/3}}{3000} \]

✓ Or (more accurate):

\[ \text{BUG} = \frac{\text{VOL}}{3000} \]

✓ …
Pattern (clone type II) analysis

<table>
<thead>
<tr>
<th>NPAT</th>
<th>NPATC</th>
<th>MPAT</th>
<th>MPATC</th>
<th>WPAT</th>
</tr>
</thead>
</table>

**TERM**
- VAR
- LAB
- PROD
- DEAD
- DEADP
- UNDEF
- AVSP
- HLEV
- WPAT

**UMET**
- UOPS
- MET
- OPS
- LEN
- VOC
- LEN^*
- UOPS*
- VOL
- PVOL
- BVOL
- HLEV
- HLEV^*
- DIF
- LLEV
- IC
- EFF
- BUGS

**MI**
- MI
- FImin
- FIavg
- FImax
- FOmin
- FOavg
- FOmax
- LEAF
- LEV
- CLEV
- RLEV
- NLEV
- HEI
- DEP
- TIMPI
- TIMP
- MCC
- HLEV^*

**AVSN**
- AVSP
- NPATC
- MPATC
- WPAT
- FImax
- FOmin
- FOmax
- TIMPI
- TIMP
- MCC
- HLEV^*

---

**NPAT**
- NPAT
- NPATC
- MPAT
- MPATC
- WPAT

---
NPAT: number of patterns

✓ Conceptual clone detection
✓ Map all productions to \{N,T,(,),|\}^*
✓ It is known that:

\[ 1 \leq \text{NPAT} \leq \text{PROD} \]
✓ …
NPATC: normalised NPAT

✓ Computed as:

\[
\text{NPATC} = \frac{\text{NPAT}}{\text{PROD}} \times 100\%
\]

✓ It is obvious that:

\[
\text{PROD}^{-1} \leq \text{NPATC} \leq 1
\]

✓ …
MPAT: max number of pattern uses

✓ It is obvious that:

\[ 1 \leq \text{MPAT} \leq \text{PROD} \]

✓ ...
MPATC: normalised MPAT

✓ Computed as:

\[ \text{MPATC} = \frac{\text{MPAT}}{\text{PROD}} \times 100\% \]

✓ It is obvious that:

\[ \text{PROD}^{-1} \leq \text{MPATC} \leq 1 \]

✓ ...
WPAT: length of the longest pattern

\[0 \leq WPAT < \infty\]

\(\exists \ NPAT^*\): max number of patterns

\[NPAT \leq NPAT^*\]

<table>
<thead>
<tr>
<th>WPAT</th>
<th>NPAT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>73</td>
</tr>
<tr>
<td>5</td>
<td>279</td>
</tr>
<tr>
<td>\ldots</td>
<td>???</td>
</tr>
</tbody>
</table>
Control flow (fan-in & fan-out)
Nonterminal fan-in

✓ Number of uses of a nonterminal within a grammar
✓ Fan-in = 0 ⇒ DEAD
✓ Fan-in = 1 ⇒ ONCE

$\text{FI}_{\text{min}} \geq 2$

$0 \leq \text{FI}_{\text{avg}} \leq \text{FI}_{\text{max}} \leq \text{VAR}$

✓ Coupling metric
Nonterminal fan-out

✓ Number of distinct nonterminals referenced

✓ Fan-out = 0 $\Rightarrow$ LEAF

$$F\text{O}_{\text{min}} \geq 1$$

$$0 \leq F\text{O}_{\text{avg}} \leq F\text{O}_{\text{max}} \leq \text{VAR}$$

✓ Cohesion metric

✓ If $\text{VAR} = \text{PROD}$,

$$F\text{O}_{\text{max}} \leq \text{WPAT}$$
Grammatical levels & call graph

<table>
<thead>
<tr>
<th>TERM</th>
<th>VAR</th>
<th>LAB</th>
<th>PROD</th>
<th>DEAD</th>
<th>DEADP</th>
<th>UNDEF</th>
<th>AVSP</th>
<th>HLEV</th>
<th>WPAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPAT</td>
<td>NPATC</td>
<td>MPAT</td>
<td>MPATC</td>
<td>WPAT</td>
<td>FImin</td>
<td>FIavg</td>
<td>FImax</td>
<td>FOmin</td>
<td>FOavg</td>
</tr>
</tbody>
</table>
LEV: number of grammatical levels

✓ Grammatical level: a subset of mutually dependent nonterminals

✓ It is known that:

\[ 1 \leq \text{LEV} \leq \text{VAR} \]

✓ …
CLEV: percentage of gram.levels

✓ LEV normalised by nonterminal count
✓ Computed as:

$$CLEV = \frac{LEV}{VAR} \times 100\%$$

✓ Low CLEV ⇒ nonterminals are clustered into few equivalence classes, subjects to modularisation
RLEV: number of recursive levels

✓ Levels that are either nontrivial or self-referring

✓ It is known that:

\[ 0 \leq \text{RLEV} \leq \text{LEV} \]

✓ RLEV reveals the number of syntactic components

✓ RLEV=0 \iff the language is finite
NLEV: number of nontrivial levels

✓ Levels that consist of more than one nonterminal
✓ It is known that:

\[ 0 \leq \text{NLEV} \leq \text{RLEV} \]

✓ …
DEP: depth

✓ The size of the biggest grammatical level

✓ It can be proven that:

\[
\text{DEP} \leq \frac{\text{VAR} - \text{LEV}}{\text{NLEV}} + 1
\]

✓ High DEP indicates uneven distribution of nonterminals among grammatical levels

✓ The distribution is always uneven!
HEI: Varju height

✓ The longest path from the starting gram.level

✓ It is known that:

HEI \leq LEV

✓ All metrics derived from grammatical levels are pairwise strongly independent on the class of context-free languages.
TIMPI: (immediate) tree impurity

✓ A call graph is always between a tree and a complete digraph

✓ How far is the immediate call graph from a tree?

\[
\text{TIMP} = \frac{e - n + 1}{n(n - 1)} \times 100\%
\]

✓ where \( n \) is the number of nodes (nonterminals) and \( e \) is the number of edges
Tree impurity examples

0%

29%

7%

100%
TIMP: tree impurity

✓ A closure on the call graph is always between a tree and a complete digraph

✓ How far is it from a tree?

✓ Obviously,

\[
\text{TIMPI} \leq \text{TIMP}
\]

✓ Correlates well with CLEV

✓ It is claimed that high TIMP hinders adaptation
Cyclomatic complexity

MCC
MCC: cyclomatic complexity

✓ McCabe, McClure

✓ Number of decision points:
  ✓ choices
  ✓ optionality
  ✓ repetition

✓ Other cyclomatic metrics exist

✓ To be explored
Maintainability index
MI: maintainability index

✓ Coleman-Oman model

✓ Secondary metric computed as:

\[ MI = 171 - 5.2 \ln \text{VOL} - 0.23 \text{MCC} - 16.2 \ln \text{LOC} \]

✓ Observed considerable reverse correlation with the first BUG metric \( (r = -0.9080) \)
## Grammar analysis

<table>
<thead>
<tr>
<th>TERM</th>
<th>VAR</th>
<th>LAB</th>
<th>PROD</th>
<th>DEAD</th>
<th>DEADP</th>
<th>UNDEF</th>
<th>ROOT</th>
<th>LOC</th>
<th>AVSN</th>
<th>AVSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMET</td>
<td>UOPS</td>
<td>MET</td>
<td>OPS</td>
<td>VOC</td>
<td>LEN</td>
<td>LEN</td>
<td>UOPS</td>
<td>VOL</td>
<td>PVOL</td>
<td>BVOL</td>
</tr>
<tr>
<td>NPAT</td>
<td>NPATC</td>
<td>MPAT</td>
<td>MPATC</td>
<td>WPAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FImin</td>
<td>FIavg</td>
<td>FImax</td>
<td>ONCE</td>
<td>FOmin</td>
<td>FOavg</td>
<td>FOmax</td>
<td>LEAF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEV</td>
<td>CLEV</td>
<td>RLEV</td>
<td>NLEV</td>
<td>HEI</td>
<td>DEP</td>
<td>TIMPI</td>
<td>TIMP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC</td>
<td>MI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recall complexity theory

✓ Kolmogorov complexity is about how much resources are needed to specify the entity.

✓ The shortest description in a meta-language.

✓ Hence, related to normal forms.

✓ Also linked to identifiable structured subentities.

✓ Complexity is incomputable.

✓ All proof systems have a complexity threshold.
Metrics tripled

✓ Measure working/baseline/recovered grammars
✓ Measure normalised grammars
✓ Impurity V “Unwarranted Assignment”
✓ Impurity VI “Unfactored Expressions”
✓ Measure freshly extracted grammars
✓ May be incorrect, contain dead production rules
✓ Easier to get than good quality grammars
Grammar normalisations

- Chain productions
- Remove (xbgf:unchain)
- Nonterminals that are used only once
  - Unfold (xbgf:inline)
- Definitions that contain unfactored expressions
  - Factor (xbgf:distribute)
Idea: some metrics tell the same story

- Gather statistical data
- Compute correlations

\[ r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \]

- Research how normalisation changes results
- Research what metrics are heavily interdependent

\[ \checkmark \Rightarrow \text{measure the same thing} \]
How to compare a metric with itself?

✓ Not looking for a correlation with itself ($r \equiv 1.0$)

✓ How interesting are the results provided by a metric?

✓ Constants are not interesting

✓ “Linear” metrics will be detected by their correlation with size (VAR, PROD, …) anyway

✓ Versatile results are interesting!

✓ Deviation? Variance?
Gini coefficients

✓ Measure the inequality of a distribution
✓ G=0 ⇒ total equality
✓ G=1 ⇒ total inequality
✓ Adjust the formula for our needs:

\[ g_x = \frac{2}{n} \left( n - \frac{1 + \sum i x_i}{\sum x_i} \right) \]
Gini coeff: MPATC \((g=0.8588)\)
Gini coeff: RLEV \( (g=0.3535) \)
Freshly extracted grammars

<table>
<thead>
<tr>
<th>TERM’</th>
<th>VAR’</th>
<th>LAB’</th>
<th>PROD’</th>
<th>DEAD’</th>
<th>DEADP’</th>
<th>UNDEF’</th>
<th>AVSP’</th>
<th>HLEV’</th>
<th>WPAT’</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table contains various terms and labels extracted from the document, such as TERM’, VAR’, LAB’, PROD’, DEAD’, DEADP’, UNDEF’, AVSP’, HLEV’, and WPAT’. These terms are likely part of a grammar or set of rules extracted from the document.
Normalised grammars

<table>
<thead>
<tr>
<th>TERM</th>
<th>VAR</th>
<th>LAB</th>
<th>PROD</th>
<th>DEAD</th>
<th>DEADP</th>
<th>UNDEF</th>
<th>AVSP</th>
<th>HLEV</th>
<th>WPAT</th>
<th>NPATC</th>
<th>MPATC</th>
<th>WPATC</th>
<th>FImin</th>
<th>FIavg</th>
<th>FImax</th>
<th>FOmin</th>
<th>FOavg</th>
<th>FOmax</th>
<th>TIMPI</th>
<th>TIMP</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

~AVSN ~AVSP ~HLEV^ ~NPATC ~MPATC ~WPAT ~FImin ~FImax ~FOmin ~FOmax ~TIMPI ~TIMP ~MCC
Interesting things found

✓ A cluster of plain size metrics
   (farther from TERM ⇔ closer to VAR)

✓ LEAF complements NPAT and correlates with size metrics that are far from VAR: LEN, VOC, UOPS, ...

✓ VAR correlates with PROD ($r = 0.9890$)

✓ AVSN does not correlate with AVSP (???)

✓ MI reverse correlates with BUG¹

✓ LEV correlates with maximum fan-in (???)

✓ CLEV and TIMP display strong reverse correlation
<table>
<thead>
<tr>
<th>TERM’</th>
<th>VAR’</th>
<th>LAB’</th>
<th>PROD’</th>
<th>DEAD’</th>
<th>DEADP’</th>
<th>UNDEF’</th>
<th>AVSP’</th>
<th>HLEV’</th>
<th>WPAT’</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM</td>
<td>VAR</td>
<td>LAB</td>
<td>PROD</td>
<td>DEAD</td>
<td>DEADP</td>
<td>UNDEF</td>
<td>AVSN</td>
<td>AVSP</td>
<td></td>
</tr>
<tr>
<td>UMET</td>
<td>UOPS</td>
<td>MET</td>
<td>OPS</td>
<td>VOC</td>
<td>LEN</td>
<td>LEN^</td>
<td>UOPS*</td>
<td>VOL</td>
<td>BVOL</td>
</tr>
<tr>
<td>NPAT</td>
<td>NPATC</td>
<td>MPAT</td>
<td>MPATC</td>
<td>WPAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FImin</td>
<td>FIavg</td>
<td>FImax</td>
<td>ONCE</td>
<td>FOmin</td>
<td>FOavg</td>
<td>FOmax</td>
<td>LEAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEV</td>
<td>CLEV</td>
<td>RLEV</td>
<td>NLEV</td>
<td>HEI</td>
<td>DEP</td>
<td>TIMPI</td>
<td>TIMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>~AVSN</td>
<td>~AVSP</td>
<td>~HLEV</td>
<td>~NPATC</td>
<td>~MPATC</td>
<td>~WPAT</td>
<td>~FImin</td>
<td>~FImax</td>
<td>~FOmin</td>
<td>~FOmax</td>
</tr>
<tr>
<td>~TIMPI</td>
<td>~TIMP</td>
<td>~MCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Static vs interactive

- ✓ Top nonterminals count
- ✓ Average production length
- ✓ Number of subcomponents

- ✓ Top nonterminals list
- ✓ Productions that are too long
- ✓ Indication on how to extract modules from subcomponents
Complex measurements: fan-in

- JDK 1.0 impl
- JDK 1.0 read
- J2SE 1.2 impl
- J2SE 1.2 read
- J2SE 5.0 impl
- J2SE 5.0 read
Complex measurements: fan-in

- JDK 1.0 impl
- J2SE 1.2 impl
- J2SE 5.0 impl
- JDK 1.0 read
- J2SE 1.2 read
- J2SE 5.0 read

ONCE
DEAD
Complex measurements: fan-in

- JDK 1.0 impl
- J2SE 1.2 impl
- J2SE 5.0 impl
- JDK 1.0 read
- J2SE 1.2 read
- J2SE 5.0 read

ONCE
DEAD
Identifier
Type
Expression
The most popular patterns found in all grammars:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Uses everywhere</th>
<th>Uses everywhere</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>2682</td>
<td>1635</td>
</tr>
<tr>
<td>T</td>
<td>1724</td>
<td>1198</td>
</tr>
<tr>
<td>NTN</td>
<td>664</td>
<td>671</td>
</tr>
<tr>
<td>NN</td>
<td>346</td>
<td>277</td>
</tr>
<tr>
<td>TN</td>
<td>252</td>
<td>212</td>
</tr>
<tr>
<td>TNT</td>
<td>150</td>
<td>136</td>
</tr>
<tr>
<td>T{N}T</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>T{N}T</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>N{N}</td>
<td>100</td>
<td>68</td>
</tr>
<tr>
<td>NT</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>TTN</td>
<td></td>
<td>75</td>
</tr>
</tbody>
</table>
Dynamic measurements: call graph
# Unsolved questions

## Performance

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract/recover grammarbase</td>
<td>5:23</td>
</tr>
<tr>
<td>Normalise grammars</td>
<td>10:19</td>
</tr>
<tr>
<td>Calculate correlations between rec &amp; ext, rec &amp; num</td>
<td>3:06</td>
</tr>
<tr>
<td>Calculate all possible correlations</td>
<td>4:35</td>
</tr>
<tr>
<td>Calculate Gini coefficients</td>
<td>2:04</td>
</tr>
<tr>
<td>Compute metrics</td>
<td>1:24</td>
</tr>
</tbody>
</table>
The relation between the number of cliques in a directed graph normalised per number of nodes, and the distance of that graph from being a tree?

\[ r(CLEV, TIMP) = 0.9518 \]
Unsolved questions

✓ The Coleman-Oman maintainability model is wrong.

✓ Normalisation as explained (unchain/inline/factor)

✓ reduces analysability

✓ reduces changeability

✓ reduces testability

✓ increases the maintenance index

✓ Contradiction with ISO 9126
Unsolved questions

✓ Completeness claims (the lack thereof).

✓ When can we tell that we have measured everything?

✓ When should we just stop measuring everything?
Awesome things ahead

✓ Preserving properties of trafo/normalisations
✓ Dynamic grammar analysis
✓ Grammar smells
✓ Metrics for pairs of grammars
✓ Coverage metrics for grammar testing
✓ Metrics for grammar transformations
To do

- Better classification: measure, metric, counter, …
- Formulae related or values related?
- Information flow metrics
- Parsing influences by metrics
- More research on normal form theory
- More indicators
- Feedback?