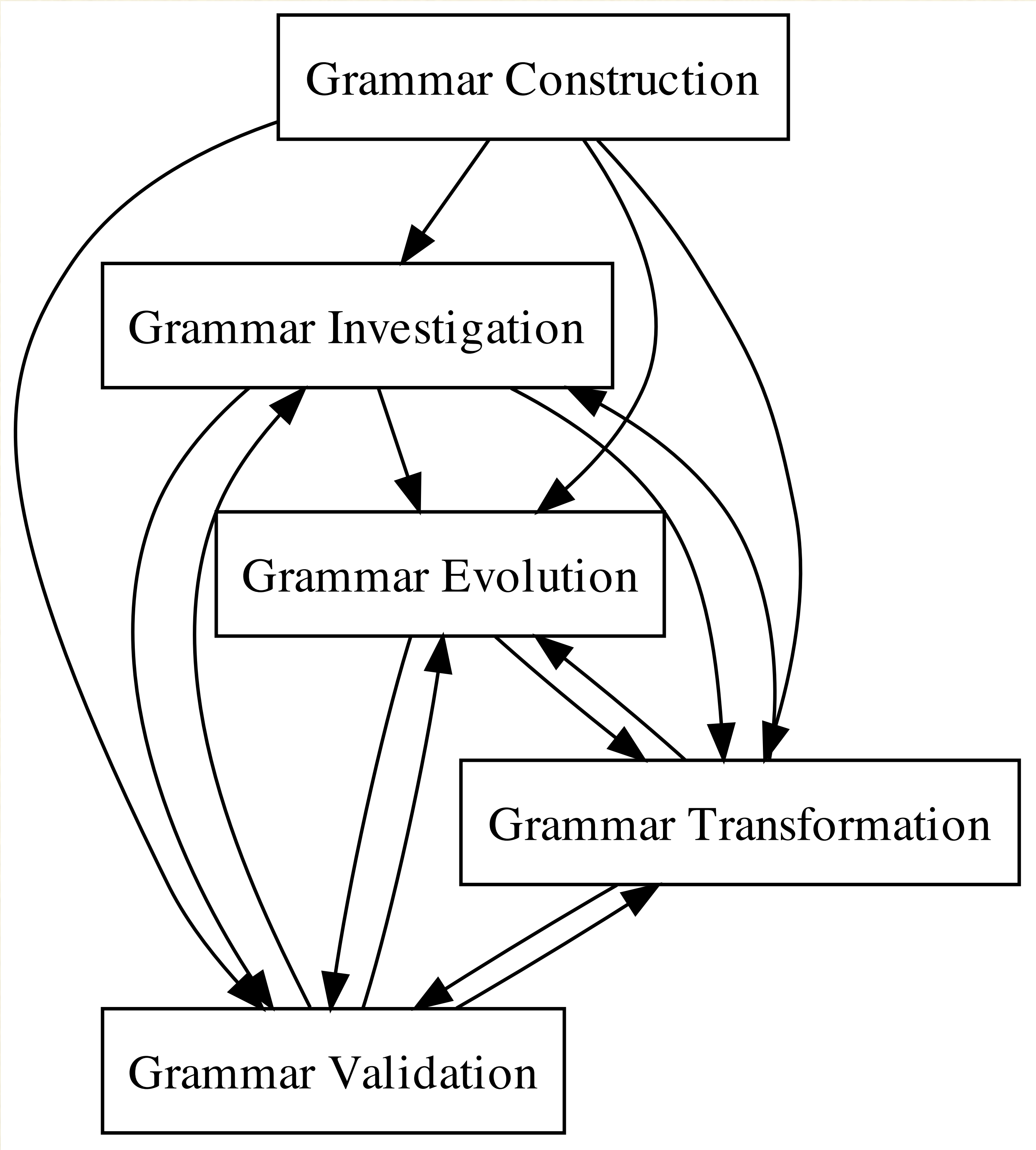
Grammar Investigation

Vadim V. Zaytsev
2011

Grammar life cycle



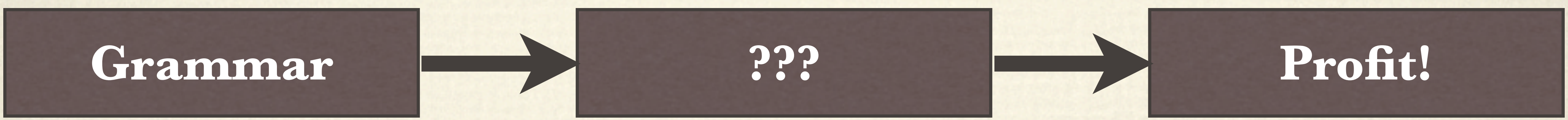
What to expect

- ✓ 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.

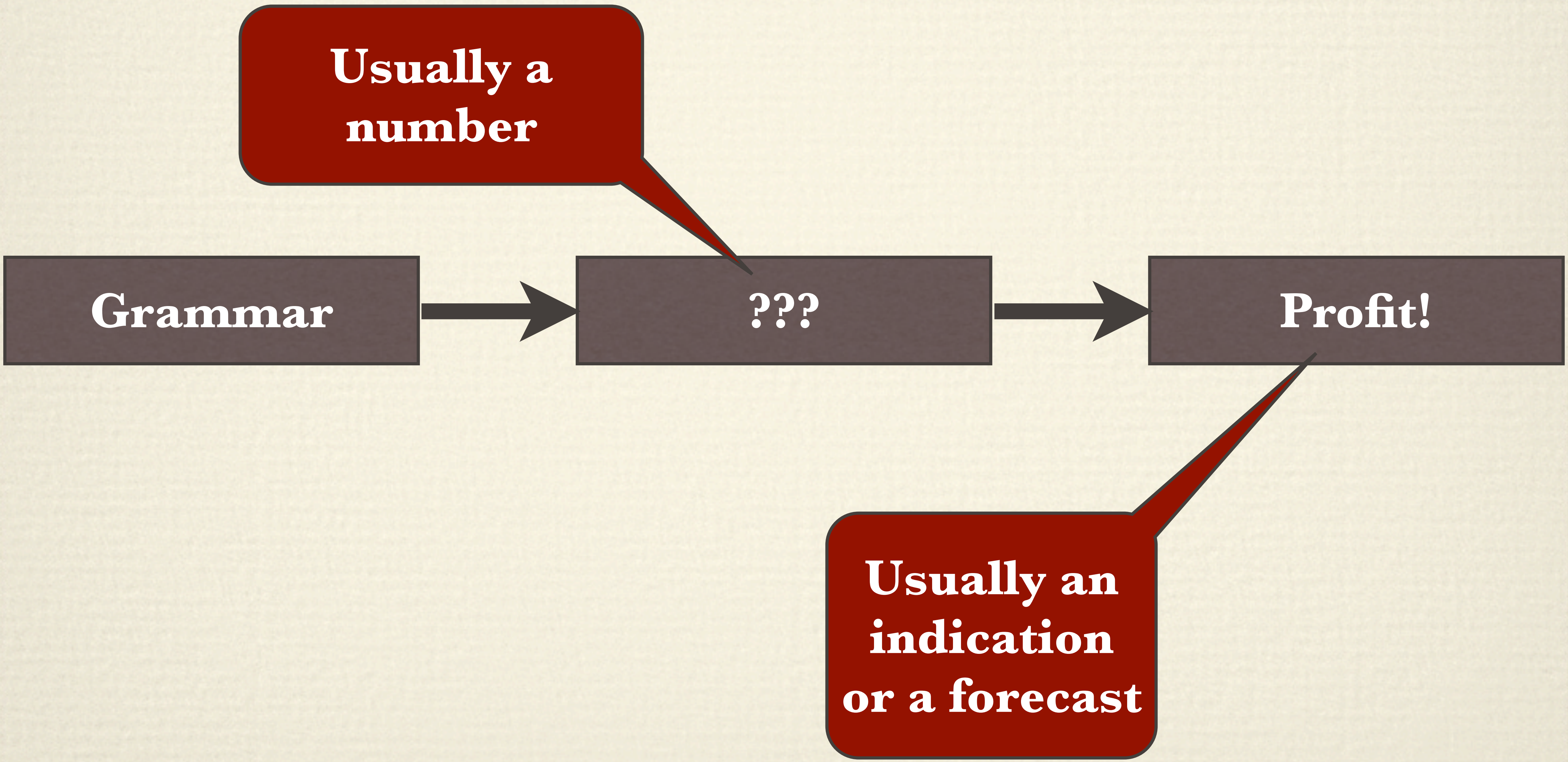
What to recall

- ✓ 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 investigation



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

TERM	UMET	NPAT	FImin	LEV
VAR	UOPS	NPATC	FIavg	CLEV
LAB	MET	MPAT	FImax	RLEV
PROD	OPS	MPATC	ONCE	NLEV
DEAD	VOC	WPAT	FOmin	HEI
DEADP	LEN		FOavg	DEP
UNDEF	LEN^		FOmax	TIMPI
ROOT	UOPS*		LEAF	TIMP
LOC	VOL			
AVSN	PVOL			
AVSP	BVOL			
	HLEV			
	HLEV^			
	DIF, IC	MCC		
	LLEV			
	EFF			
	EFF^	MI		
	BUG			

Primitive grammar measurements

TERM
VAR
LAB
PROD
DEAD
DEADP
UNDEF
ROOT
LOC
AVSN
AVSP

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:

$$LOC = VAR + PROD$$

- ✓ ...

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

Grammarware science

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

UMET: unique meta-symbols

- ✓ Tells more about grammar notation
- ✓ Or about the extent to which notation is exercised
- ✓ For the notation, there exists $\underline{\text{UMET}}$:

$$2 \leq \text{UMET} \leq \underline{\text{UMET}}$$

- ✓ …

UOPS: unique operands

- ✓ Can be computed as:

$$UOPS = VAR + TERM + LAB$$

- ✓ There exists UOPS*:

$$UOPS^* \leq UOPS$$

- ✓ …

UOPS*: minimum required operands

- ✓ Can be computed as:

  UOPS* = TERM + ROOT + UNDEF

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

  UOPS* = 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:

$$\mathrm{VOC} = \mathrm{UMET} + \mathrm{UOPS}$$

- ✓ …

LEN: grammar length

- Can be computed as:

$$\text{LEN} = \text{MET} + \text{OPS}$$

- …

PUR: purity ratio

- ✓ Can be computed as:

$$\text{PUR} = \widehat{\text{LEN}}/\text{LEN}$$

- ✓ …

VOL: grammar volume

- Can be computed as:

$$\mathrm{VOL} = \mathrm{LEN}\ \log_2 \mathrm{VOC}$$

- …

PVOL: potential (minimal) volume

- ✓ Can be computed as:

$$PVOL = (2+UOPS^*)\log_2(2+UOPS^*)$$

- ✓ …

BVOL: boundary volume

- ✓ Can be computed as:

  $$BVOL = (2+UOPS^* \log_2 UOPS^*)\log_2(2+UOPS^*)$$

- ✓ …

HLEV: grammar level

- ✓ Can be computed as:

$$\text{HLEV: PVOL/VOL}$$

- ✓ Known property:

$$0 \le \text{HLEV} \le 1$$

- ✓ …

$\widehat{\text{HLEV}}$: estimated grammar level

- ✓ Can be computed as:

$$\widehat{\text{HLEV}}\text{: } (2\times\text{UOPS})/(\text{UMET}\times\text{OPS})$$

- ✓ …

DIF: difficulty

- ✓ Can be computed as:

$$DIF = 1/HLEV$$

- ✓ ...

LLEV: meta-language level

- ✓ Can be computed as:

$$LLEV = HLEV \times PVOL$$

- ✓ 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:

$$EFF = VOL/HLEV$$

- ✓ …

$\widehat{\mathrm{EFF}}$: estimated engineering effort

- The most commonly used metric "by Halstead"
- Was not suggested by Maurice Halstead.
- Computed as:

$$\widehat{\mathrm{EFF}} = \frac{\mathrm{VOL}}{\widehat{\mathrm{HLEV}}} = \frac{\mathrm{UMET} \times \mathrm{OPS} \times \mathrm{LEN} \times \log_2 \mathrm{VOC}}{2 \times \mathrm{UOPS}}$$

- …

BUG: estimated number of errors

- Can be computed as:

$$BUG = EFF^{2/3}/3000$$

- Or (more accurate):

$$BUG = VOL/3000$$

- …

Pattern (clone type II) analysis

NPAT
NPATC
MPAT
MPATC
WPAT

NPAT: number of patterns

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

$$1 \leq NPAT \leq PROD$$

- ✓ …

NPATC: normalised NPAT

- Computed as:

$$NPATC = NPAT/PROD \times 100\%$$

- It is obvious that:

$$PROD^{-1} \leq NPATC \leq 1$$

- …

MPAT: max number of pattern uses

- ✓ It is obvious that:

$$1 \leq \text{MPAT} \leq \text{PROD}$$

- ✓ …

MPATC: normalised MPAT

- ✓ Computed as:

$$\text{MPATC} = \text{MPAT}/\text{PROD} \times 100\%$$

- ✓ It is obvious that:

$$\text{PROD}^{-1} \leq \text{MPATC} \leq 1$$

- ✓ …

WPAT: length of the longest pattern

$$0 \leq \text{WPAT} < \infty$$

✓ ∃ NPAT*: max number of patterns

$$\text{NPAT} \leq \text{NPAT*}$$

WPAT	**NPAT***
0	1
1	3
2	7
3	21
4	73
5	279
…	???

Control flow (fan-in & fan-out)

FImin
FIavg
FImax
ONCE
FOmin
FOavg
FOmax
LEAF

Nonterminal fan-in

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

$$FImin \geq 2$$

$$0 \leq FIavg \leq FImax \leq VAR$$

- ✓ Coupling metric

Nonterminal fan-out

- ✓ Number of distinct nonterminals referenced
- ✓ Fan-out = 0 ⇒ LEAF

$$\text{FOmin} \geq 1$$

$$0 \leq \text{FOavg} \leq \text{FOmax} \leq \text{VAR}$$

- ✓ Cohesion metric
- ✓ If VAR = PROD,

$$\text{FOmax} \leq \text{WPAT}$$

Grammatical levels & call graph

[redacted]

LEV
CLEV
RLEV
NLEV
HEI
DEP
TIMPI
TIMP

[redacted]

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:

$$\mathrm{CLEV} = \mathrm{LEV}/\mathrm{VAR} \times 100\%$$

- ✓ 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 ⇔ 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:

$$\mathrm{DEP} \leqslant \frac{\mathrm{VAR} - \mathrm{LEV}}{\mathrm{NLEV}} + 1$$

- ✓ 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:

$$\mathrm{HEI} \leq \mathrm{LEV}$$

- All metrics derived from grammatical levels are pairwise strongly independent on the class of context-free languages.

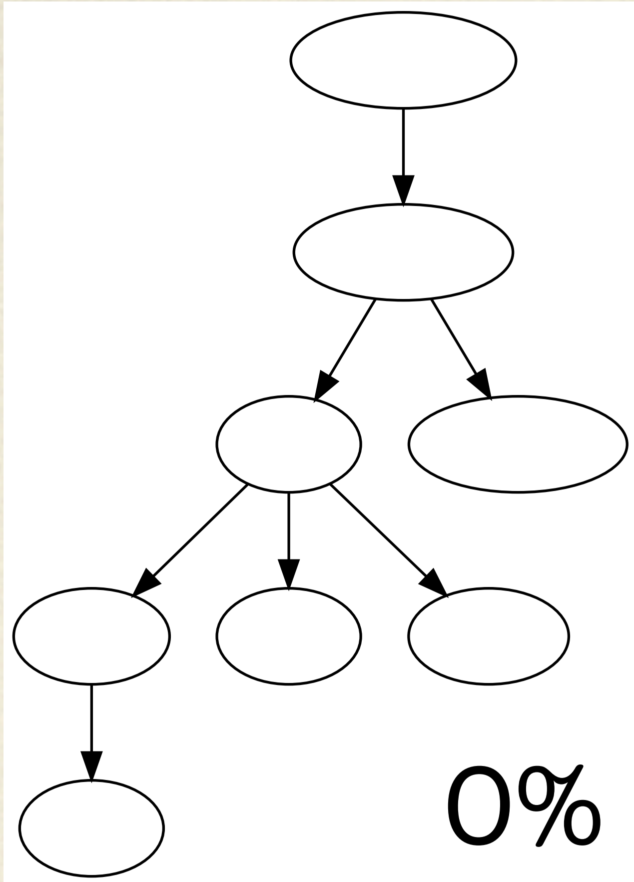
TIMPI: (immediate) tree impurity

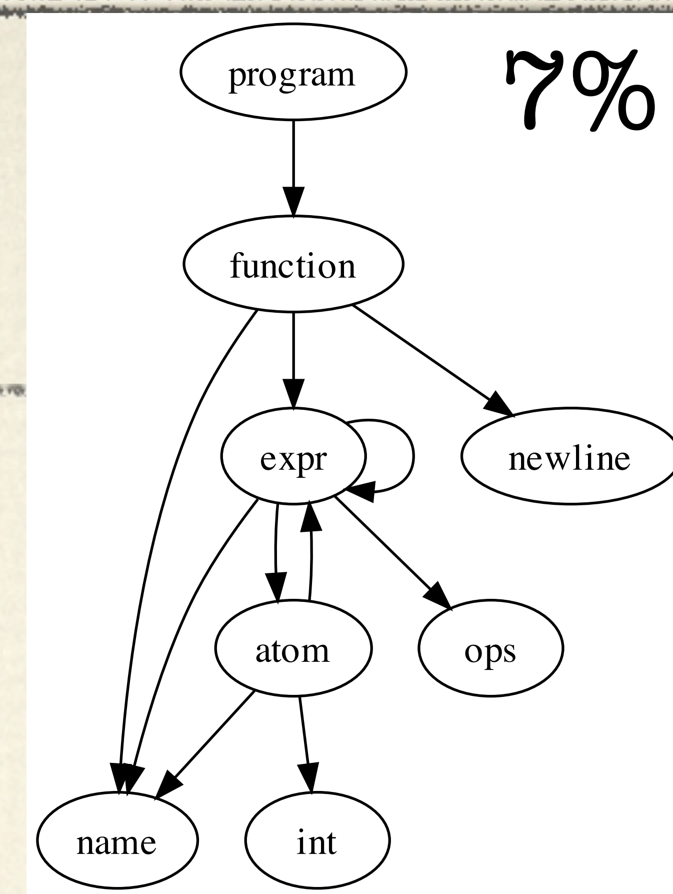
- ✓ A call graph is always between a tree and a complete digraph
- ✓ How far is the immediate call graph from a tree?

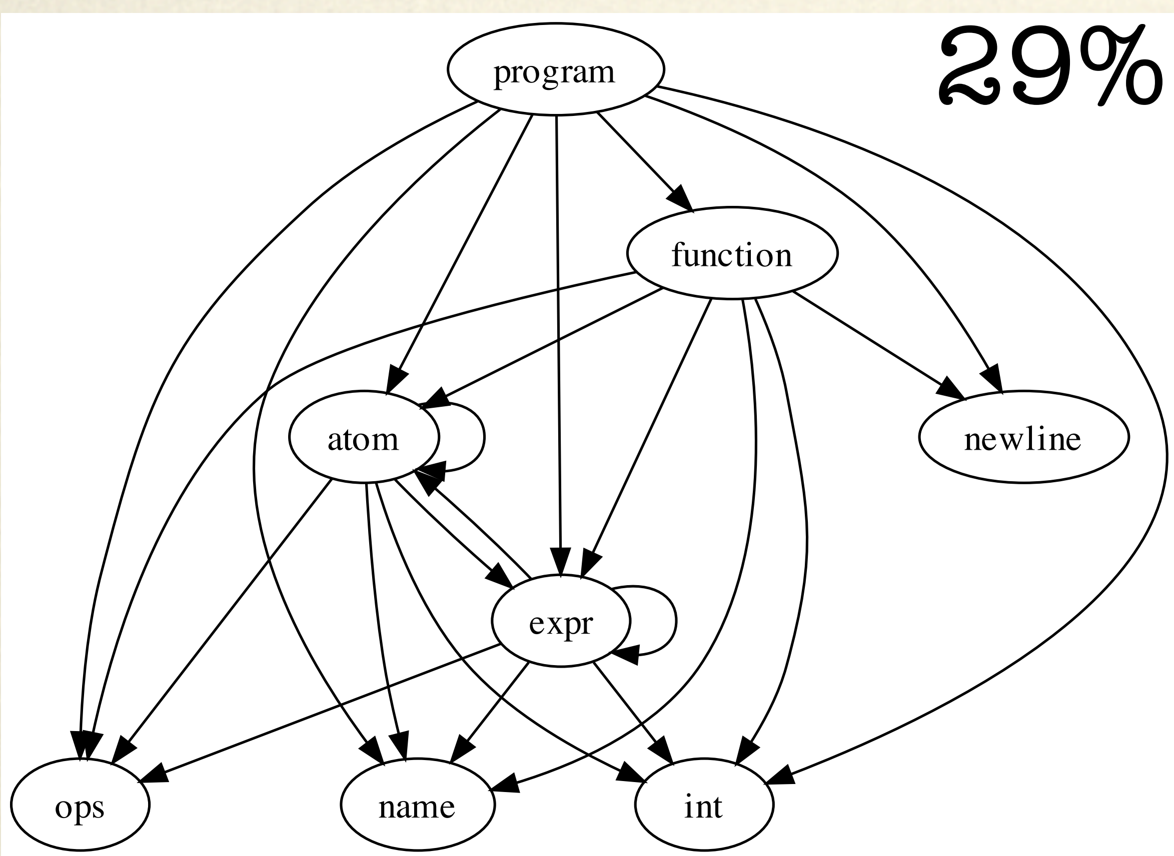
$$\mathrm{TIMP} = \frac{e-n+1}{n(n-1)}100\%$$

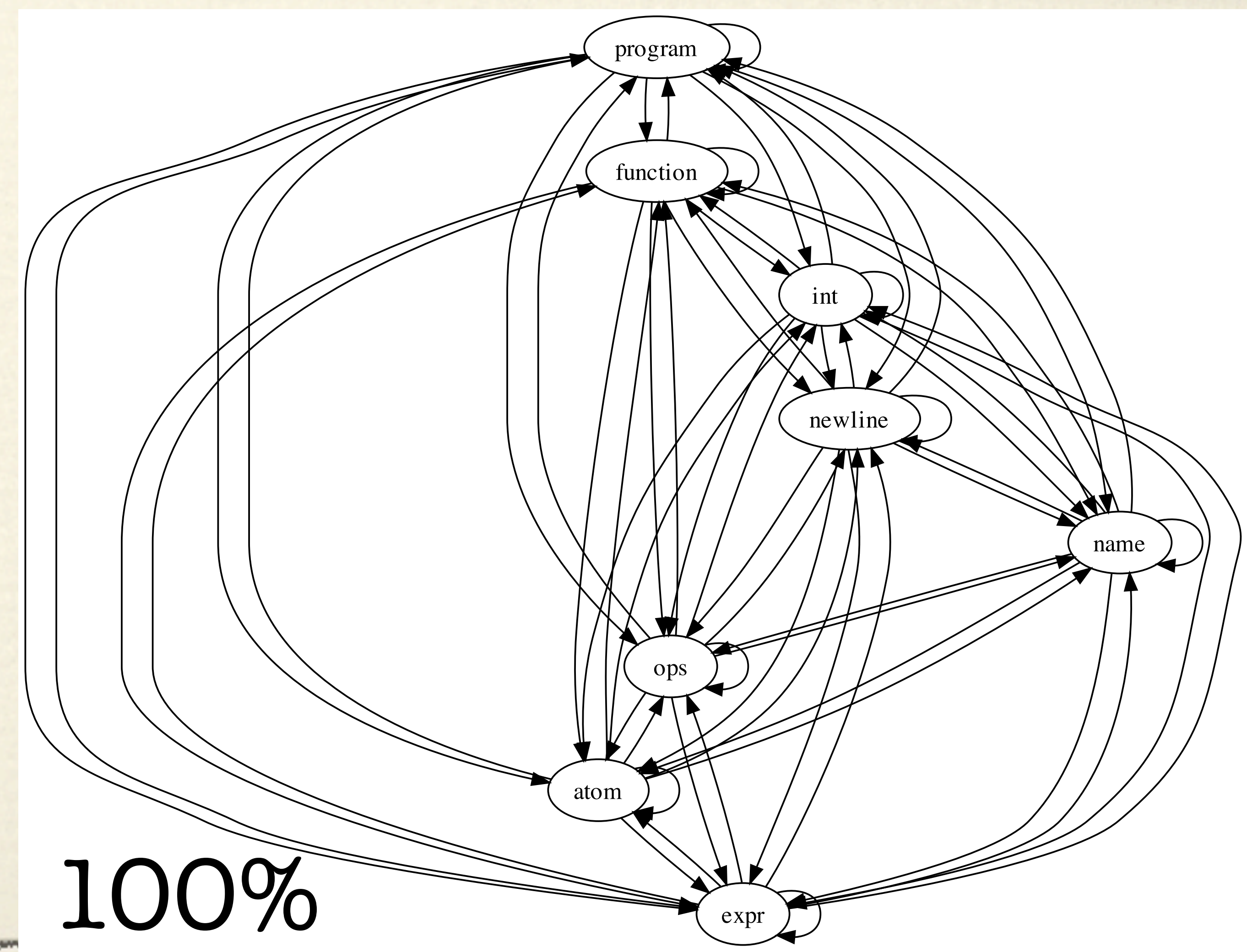
- ✓ where *n* is the number of nodes (nonterminals) and *e* is the number of edges

Tree impurity examples









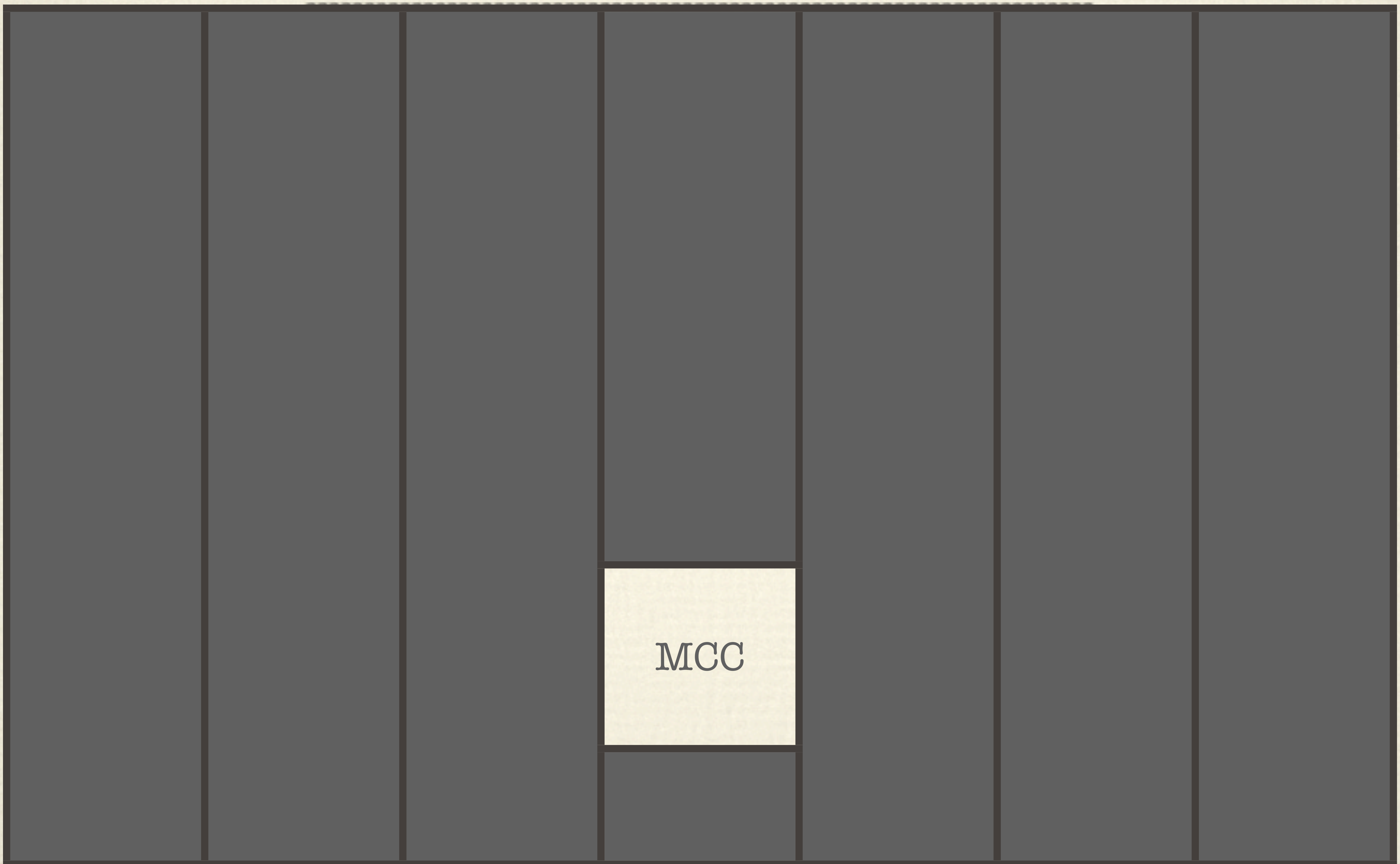
TIMP: tree impurity

- ✓ 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

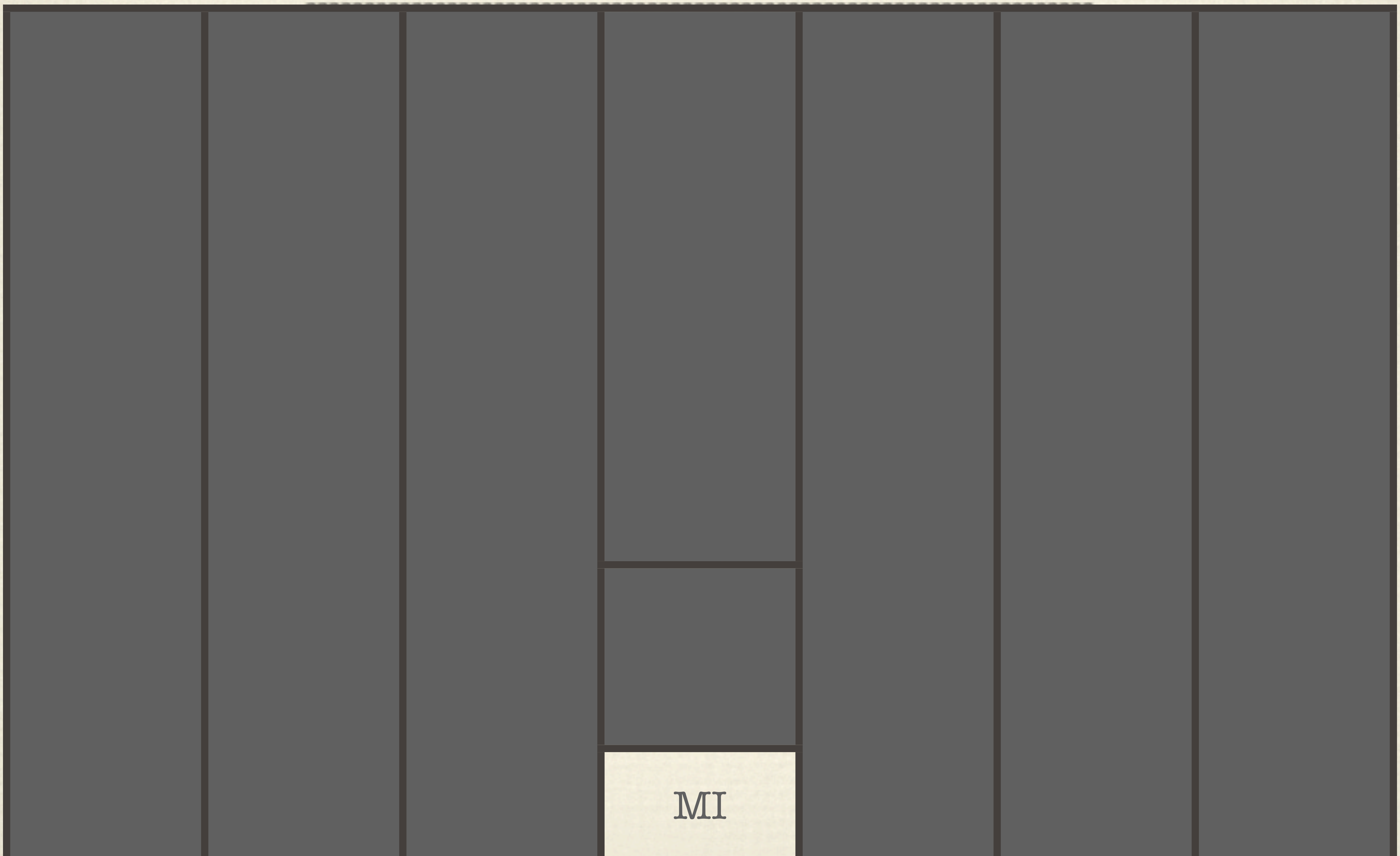
Cyclomatic complexity



MCC: 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:

  $$MI = 171 - 5.2 \ln VOL - 0.23\ MCC - 16.2 \ln LOC$$

- ✓ Observed considerable reverse correlation with the first BUG metric ($r = -0.9080$)

Grammar analysis

TERM	UMET	NPAT	FImin	LEV
VAR	UOPS	NPATC	FIavg	CLEV
LAB	MET	MPAT	FImax	RLEV
PROD	OPS	MPATC	ONCE	NLEV
DEAD	VOC	WPAT	FOmin	HEI
DEADP	LEN		FOavg	DEP
UNDEF	LEN^		FOmax	TIMPI
ROOT	UOPS*		LEAF	TIMP
LOC	VOL			
AVSN	PVOL			
AVSP	BVOL			
	HLEV			
	HLEV^			
	DIF, IC	MCC		
	LLEV			
	EFF			
	EFF^	MI		
	BUG			

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

$$r_{xy} = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}}$$

- ✓ Research how normalisation changes results
- ✓ Research what metrics are heavily interdependent
  - ✓ $\Rightarrow$ measure the same thing

How to compare a metric with itself?

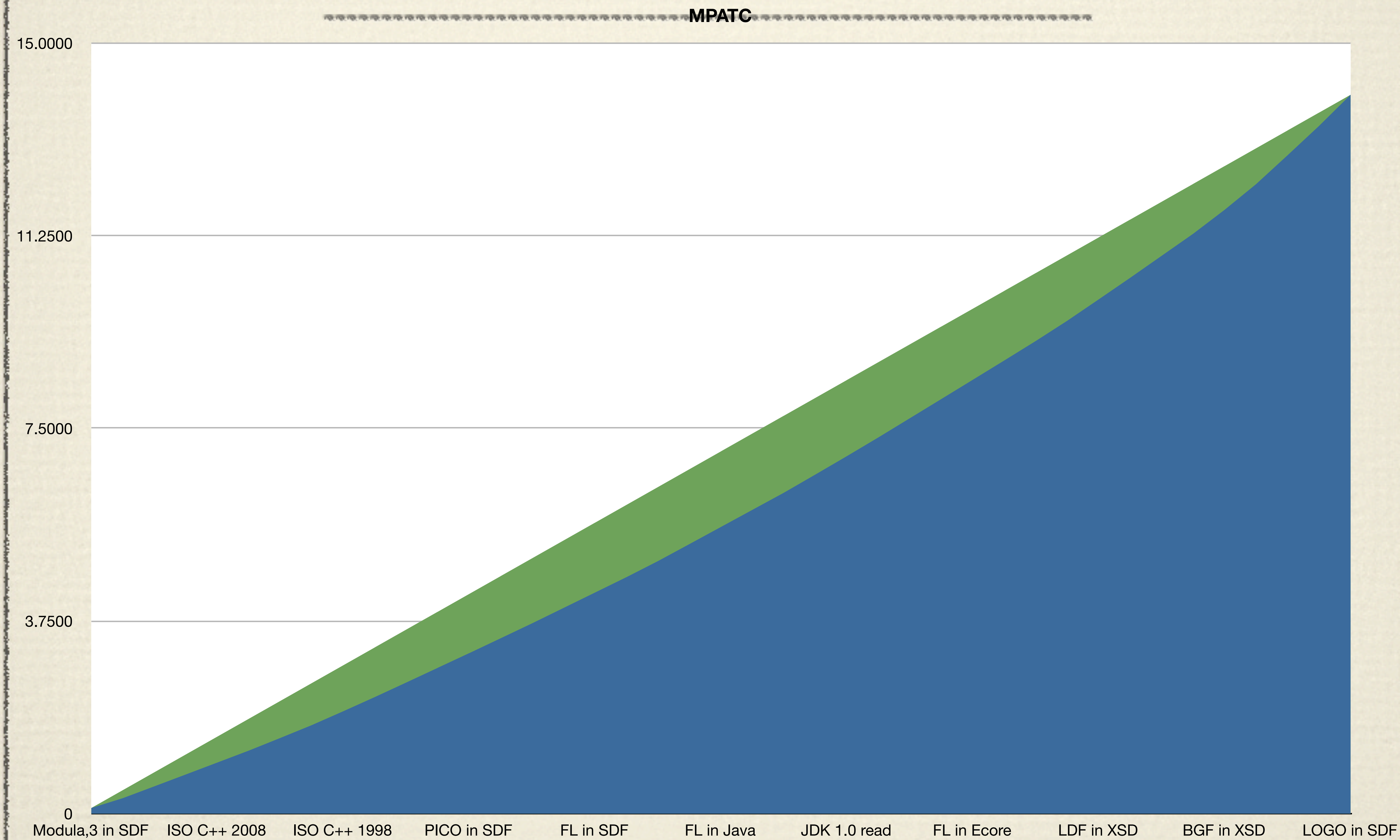
- ✓ Not looking for a correlation with itself ($r \equiv 1.0$)
- ✓ How interesting are the results provided by a metric?
- ✓ Constants are not interesting
- ✓ "Linear" metrics will be detected by their correlation with size (VAR, PROD, …) anyway
- ✓ Versatile results are interesting!
- ✓ Deviation? Variance?

Gini coefficients

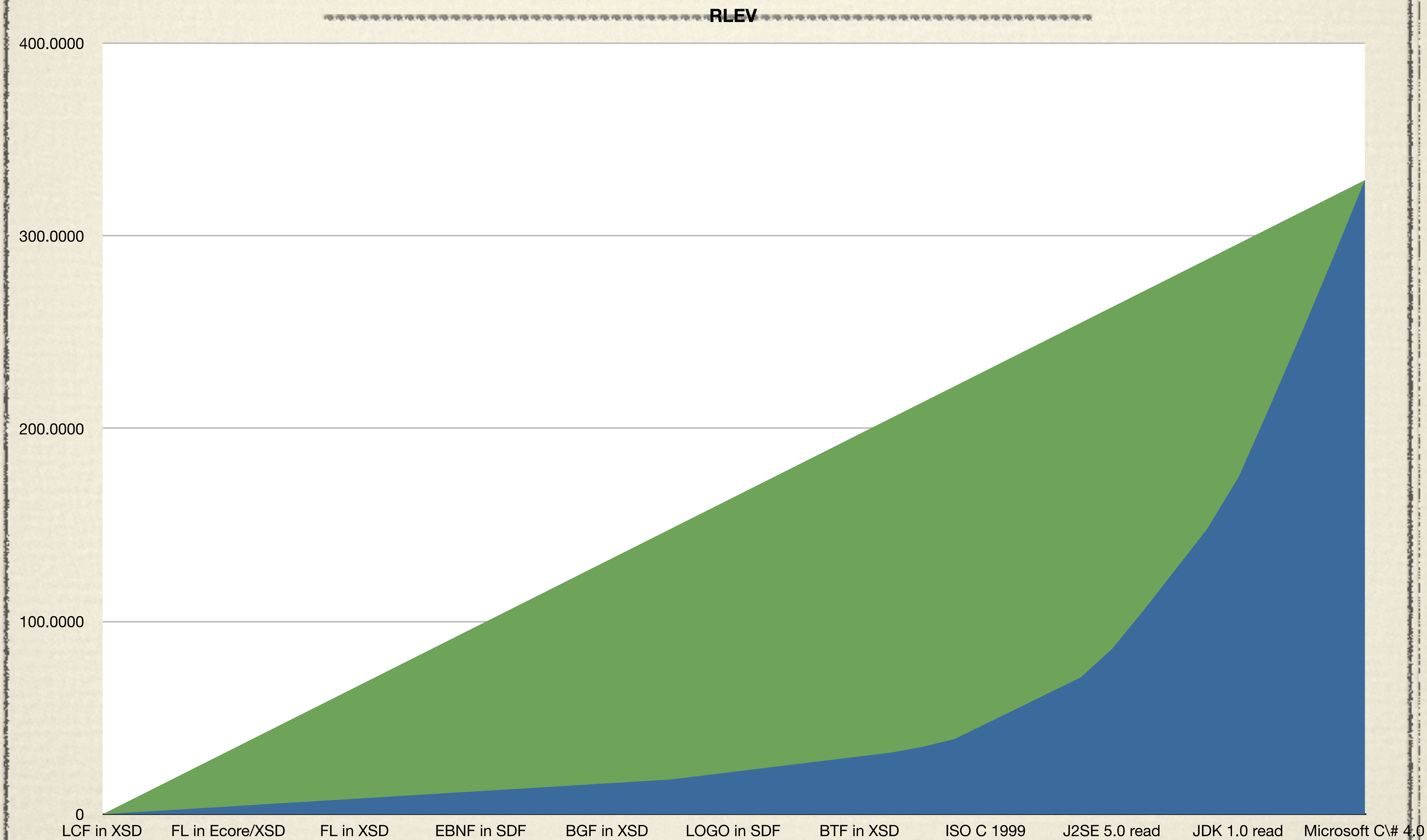
- ✓ Measure the inequality of a distribution
- ✓ G=0 ⇒ total equality
- ✓ G=1 ⇒ total inequality
- ✓ Adjust the formula for our needs:

$$g_x = \frac{2}{n}\left(n - \frac{1 + \sum i x_i}{\sum x_i}\right)$$

Gini coeff: MPATC (g=0.8588)



Gini coeff: RLEV (g=0.3535)



Freshly extracted grammars

TERM'
VAR'
LAB'
PROD'
DEAD'
DEADP'
UNDEF'
AVSP'
HLEV'
WPAT'

[redacted]

Normalised grammars

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	~AVSN ~AVSP ~HLEV^ ~NPATC ~MPATC ~WPAT ~FImin ~FImax ~FOmin ~FOmax ~TIMPI ~TIMP ~MCC

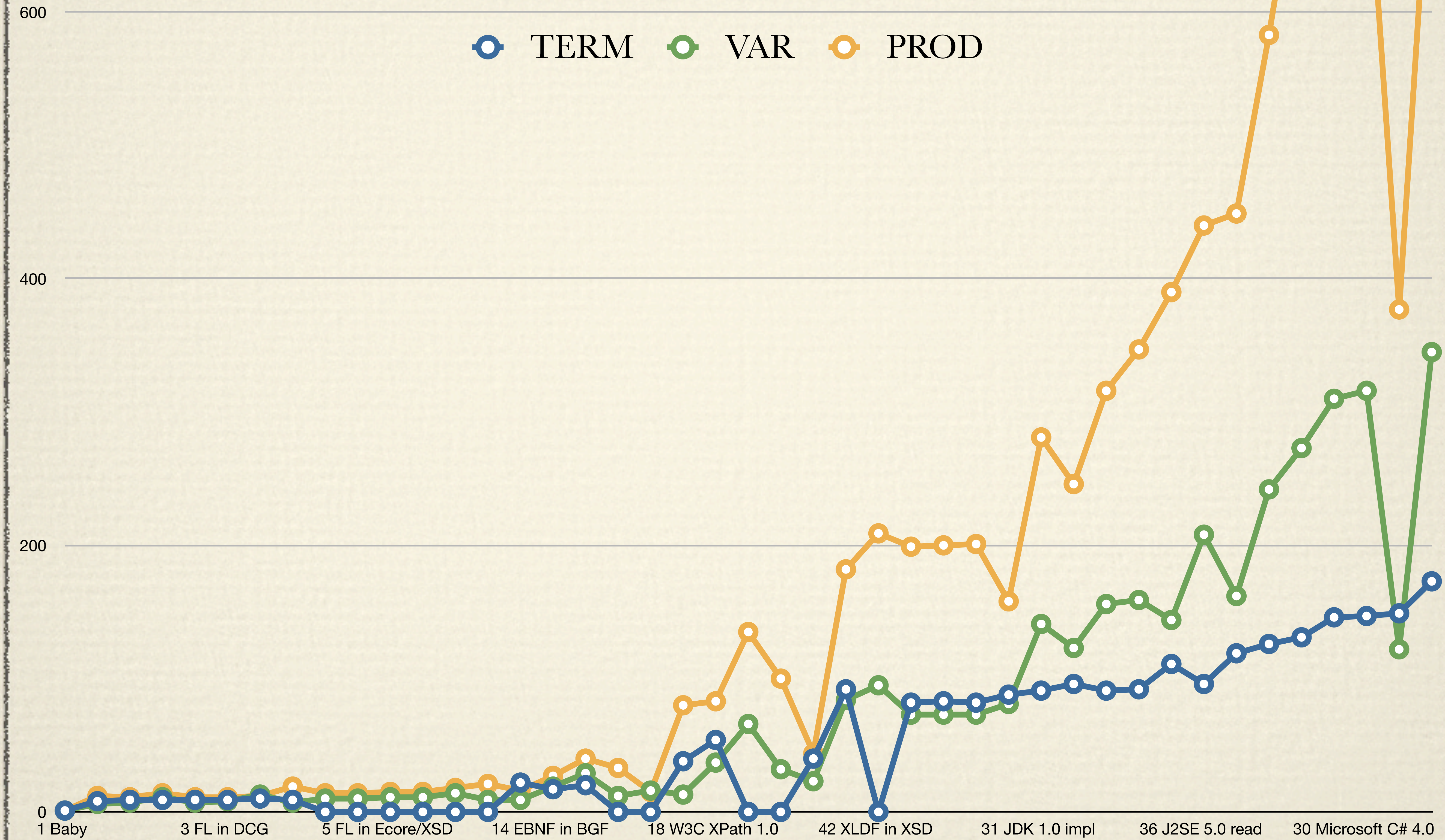
Interesting things found

- ✓ 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: $\widehat{\text{LEN}}$, VOC, UOPS, …
- ✓ VAR correlates with PROD (r =0.9890)
- ✓ AVSN does not correlate with AVSP (???)
- ✓ MI reverse correlates with BUG[1]
- ✓ LEV correlates with maximum fan-in (???)
- ✓ CLEV and TIMP display strong reverse correlation

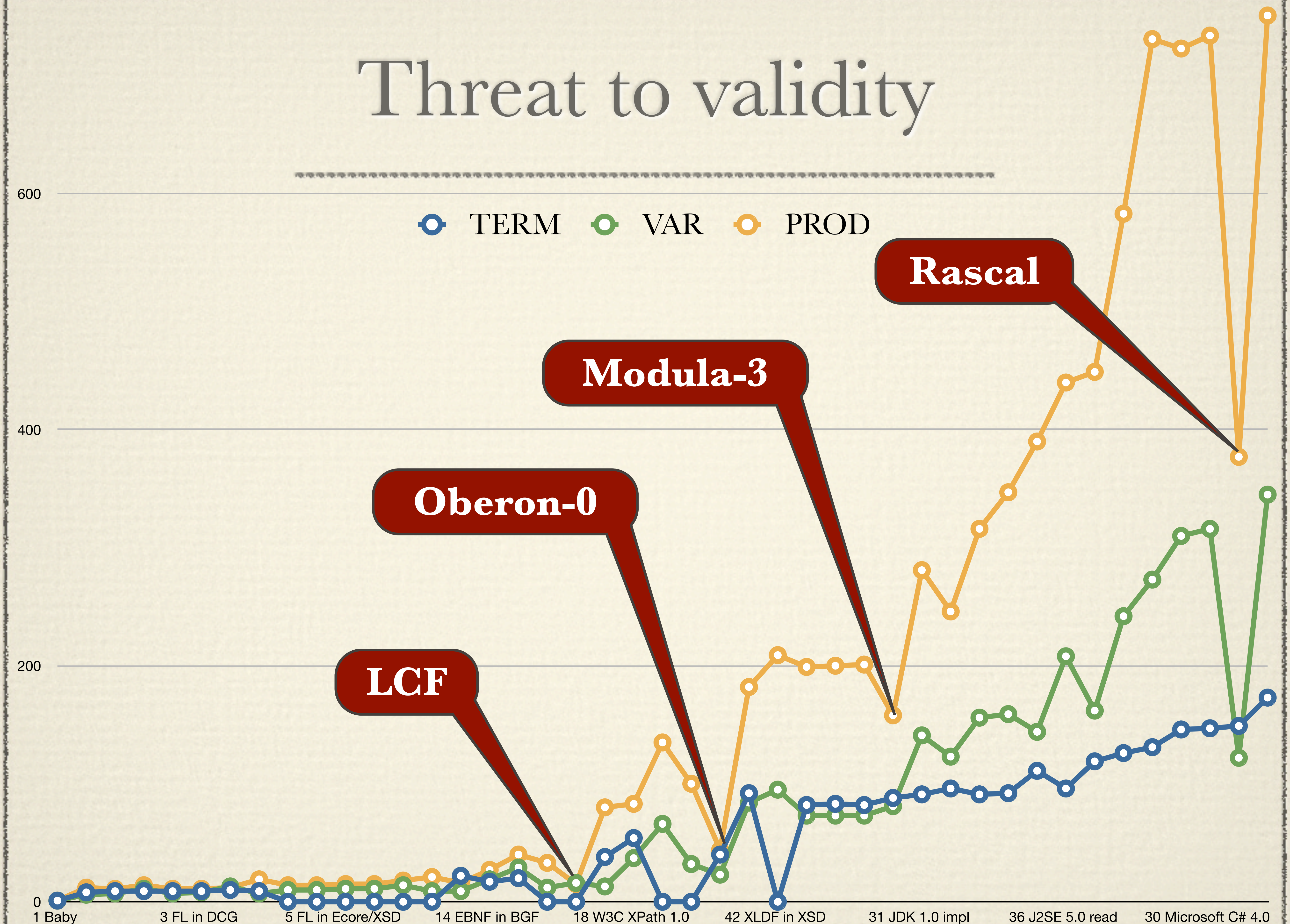
Complete picture (47/159)

TERM'	TERM	UMET	NPAT	FImin	LEV	~AVSN
VAR'	VAR	UOPS	NPATC	FIavg	CLEV	~AVSP
LAB'	LAB	MET	MPAT	FImax	RLEV	~HLEV^
PROD'	PROD	OPS	MPATC	ONCE	NLEV	~NPATC
DEAD'	DEAD	VOC	WPAT	FOmin	HEI	~MPATC
DEADP'	DEADP	LEN		FOavg	DEP	~WPAT
UNDEF'	UNDEF	LEN^		FOmax	TIMPI	~FImin
AVSP'	ROOT	UOPS*		LEAF	TIMP	~FImax
HLEV'	LOC	VOL				~FOmin
WPAT'	AVSN	PVOL				~FOmax
	AVSP	BVOL				~TIMPI
		HLEV				~TIMP
		HLEV^				~MCC
		DIF, IC	MCC			
		LLEV	...			
		EFF				
		EFF^	MI			
		BUG				

Threat to validity



Threat to validity

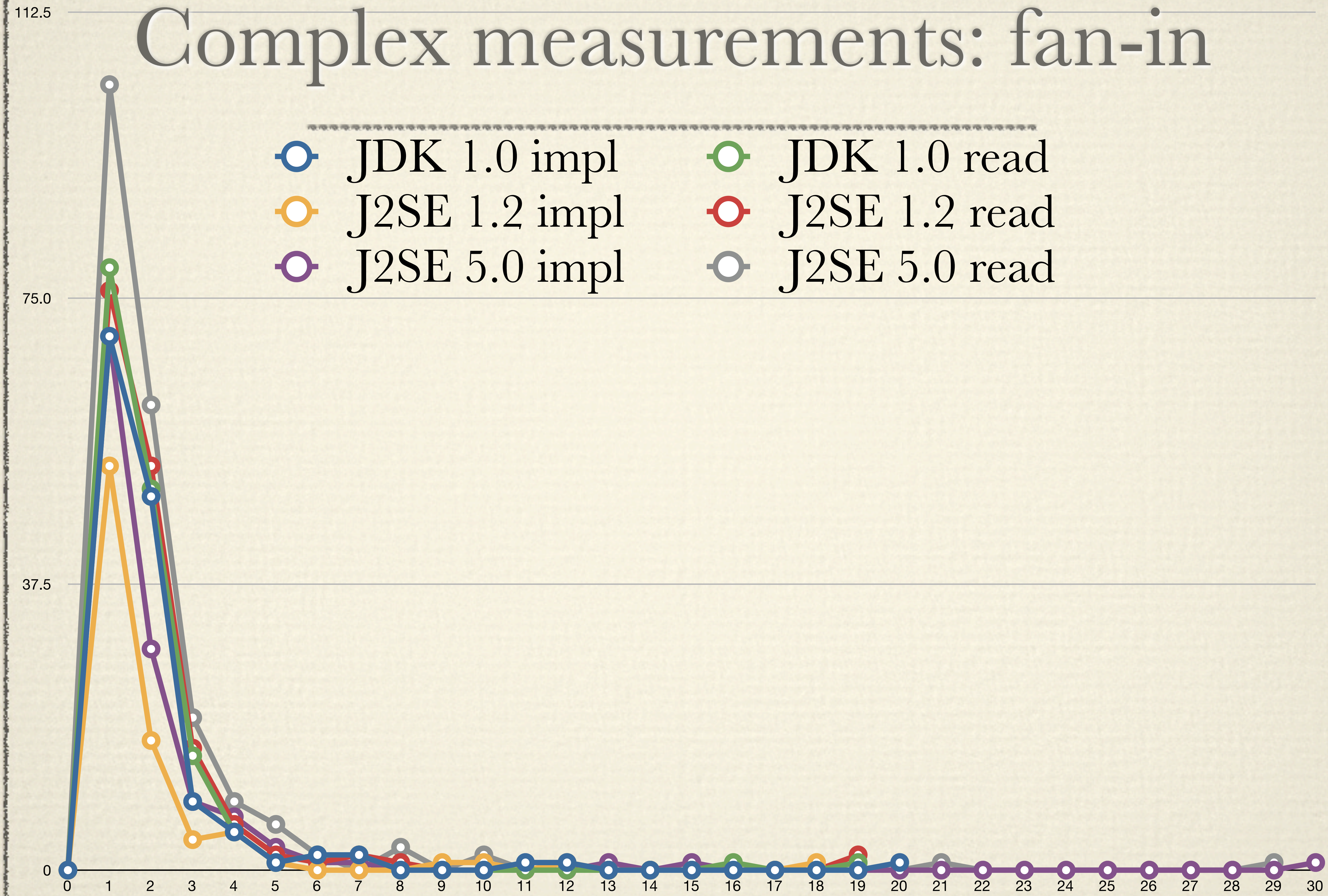


Static vs interactive

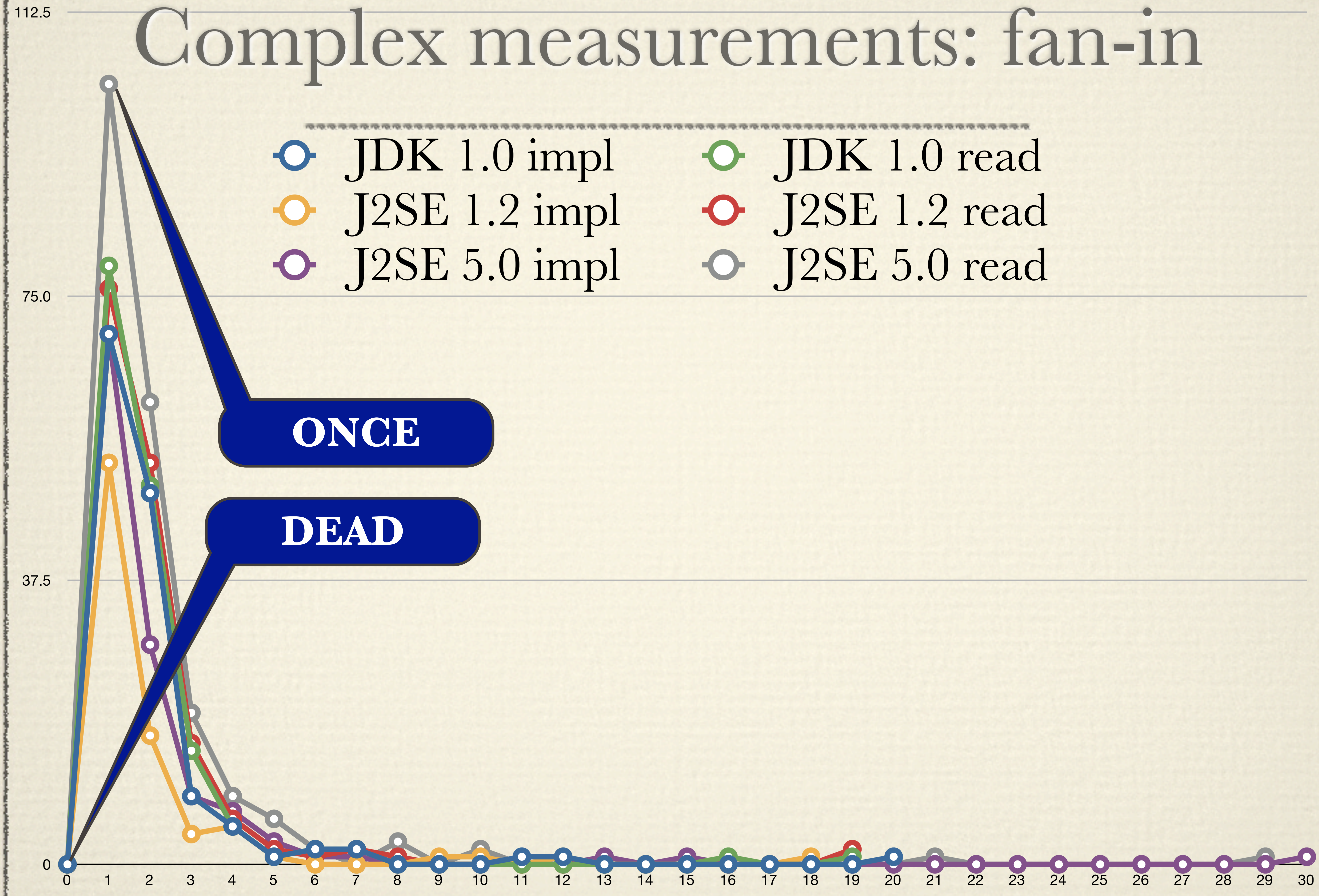
- ✓ Top nonterminals count
- ✓ Average production length
- ✓ Number of subcomponents

- ✓ Top nonterminals list
- ✓ Productions that are too long
- ✓ Indication on how to extract modules from subcomponents

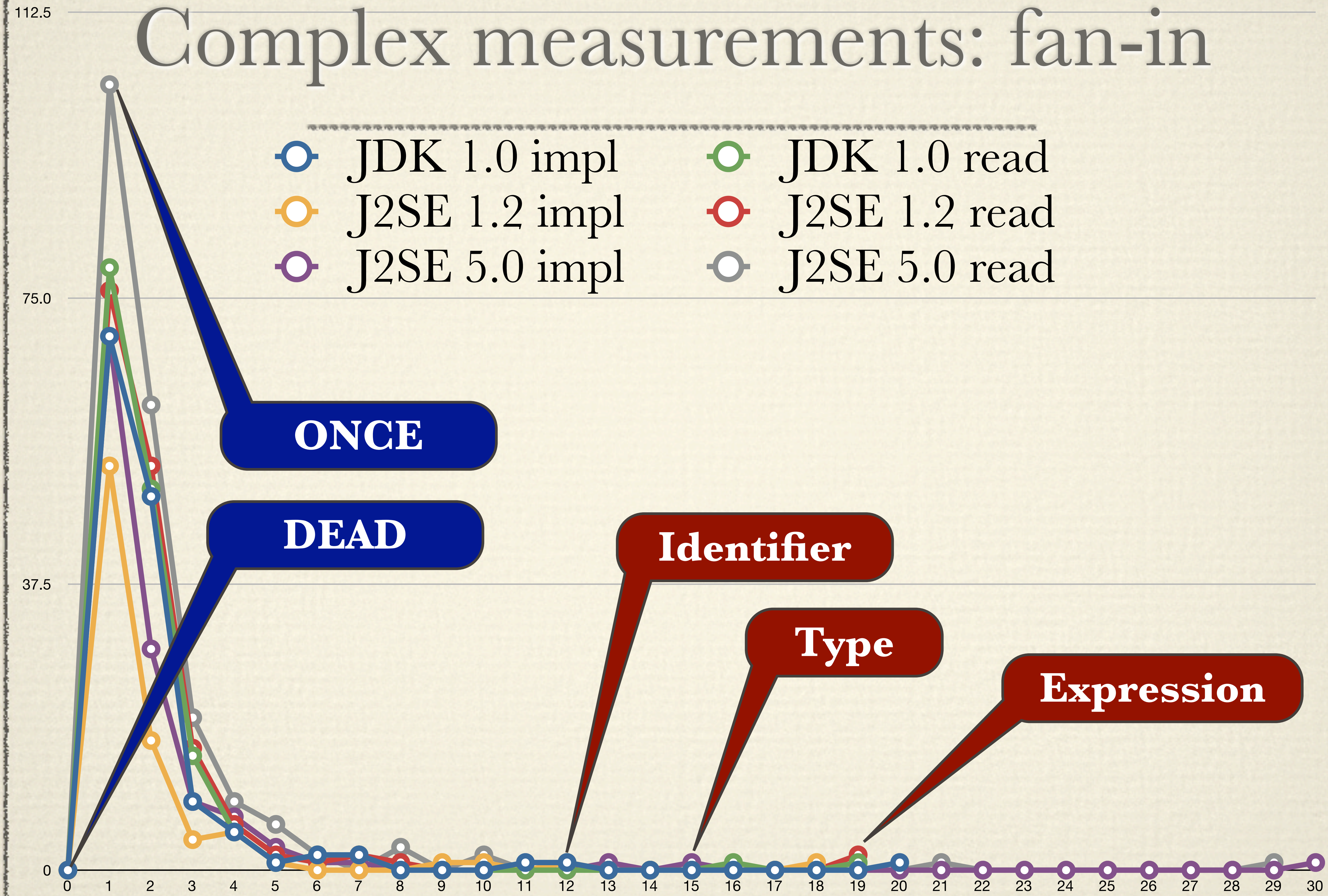
Complex measurements: fan-in



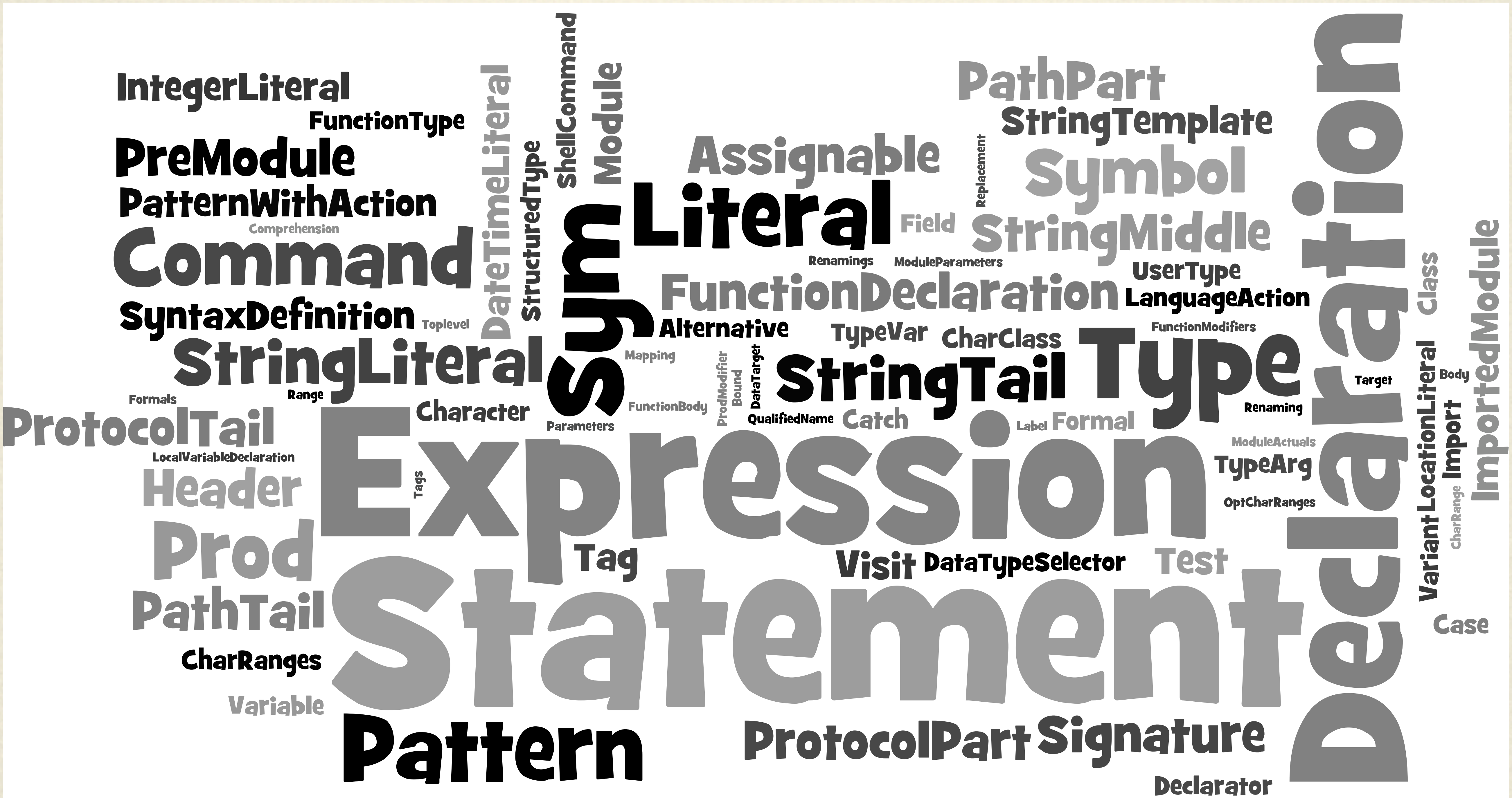
Complex measurements: fan-in



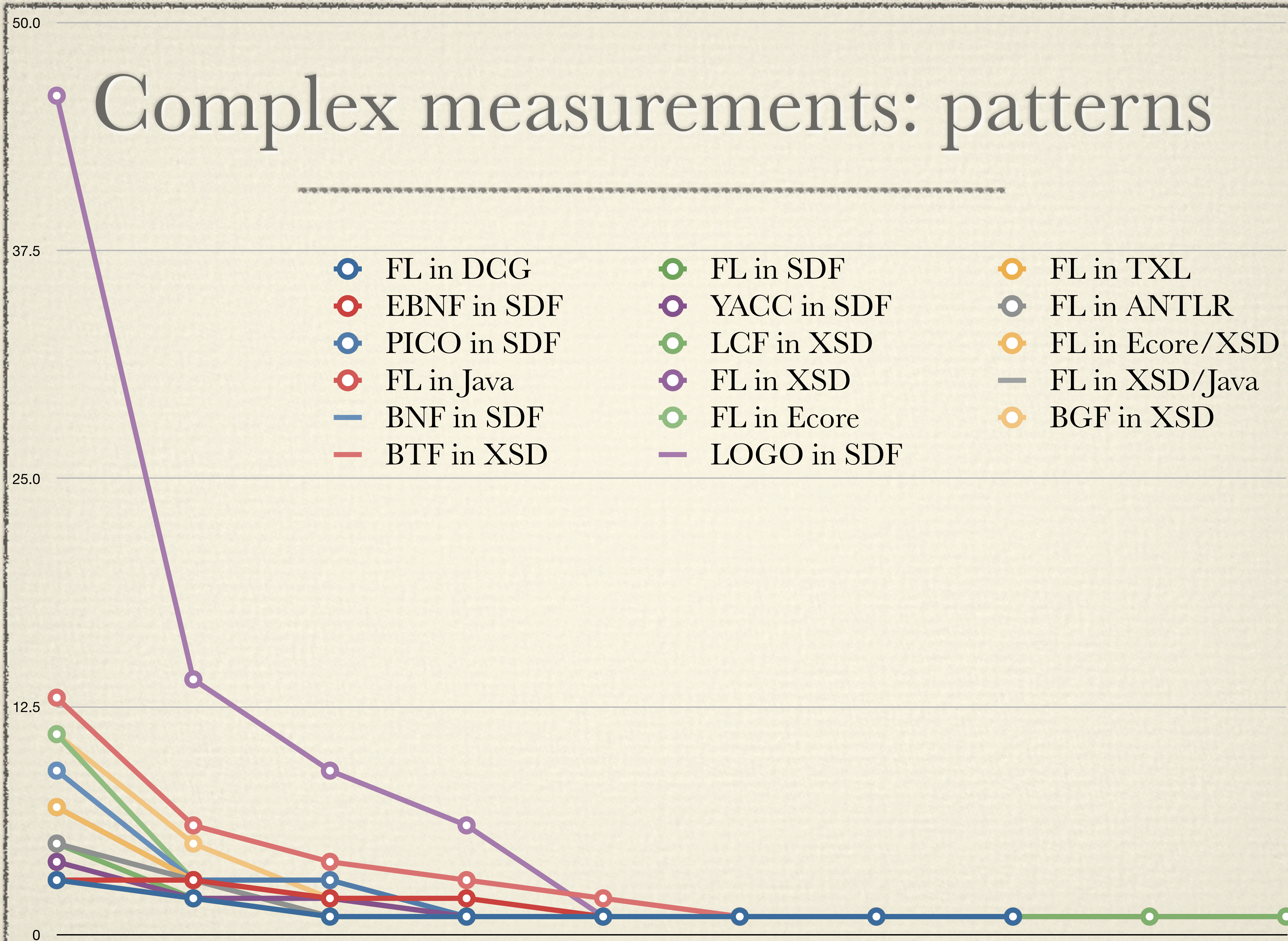
Complex measurements: fan-in



Complex measurements: fan-out



Complex measurements: patterns

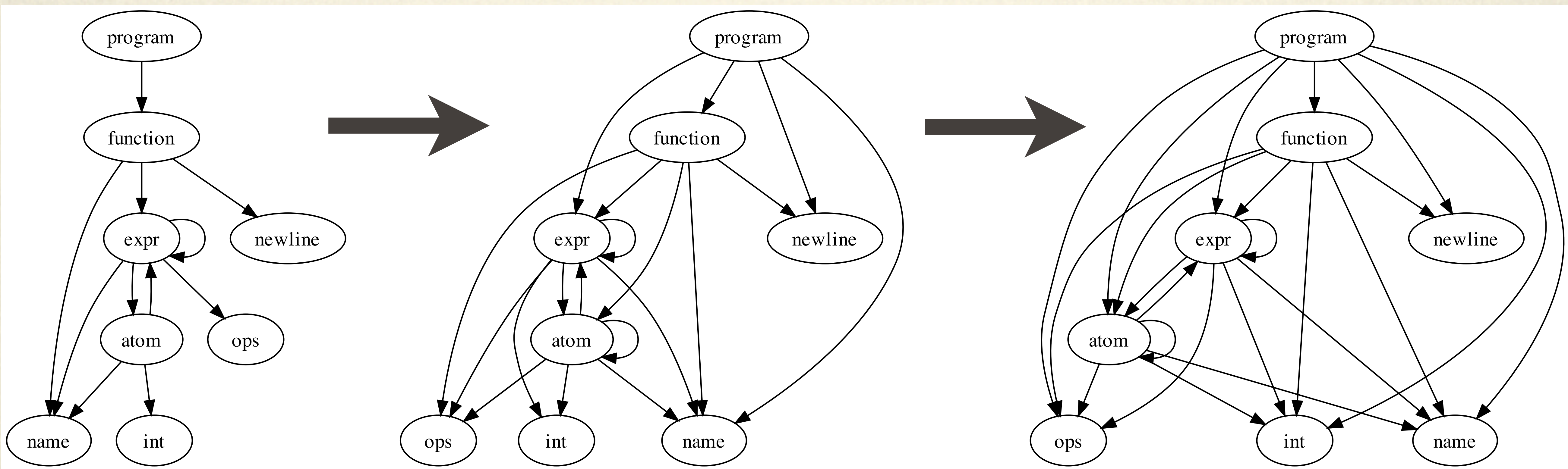


Complex measurements: patterns

✓ The most popular patterns found in all grammars:

Pattern	Uses everywhere	Uses everywhere
N	2682	1635
T	1724	1198
NTN	664	671
NN	346	277
TN	252	212
TNT	150	136
	134	
T{N}T	107	
N{N}	100	68
NT	100	85
TTN		75

Dynamic measurements: call graph



Unsolved questions

✓ Performance

Extract/recover grammarbase	5:23
Normalise grammars	10:19
Calculate correlations between rec & ext, rec & num	3:06
Calculate all possible correlations	4:35
Calculate Gini coefficients	2:04
Compute metrics	1:24

Unsolved questions

$r(CLEV, TIMP) = 0.9518$

- ✓ 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?

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?