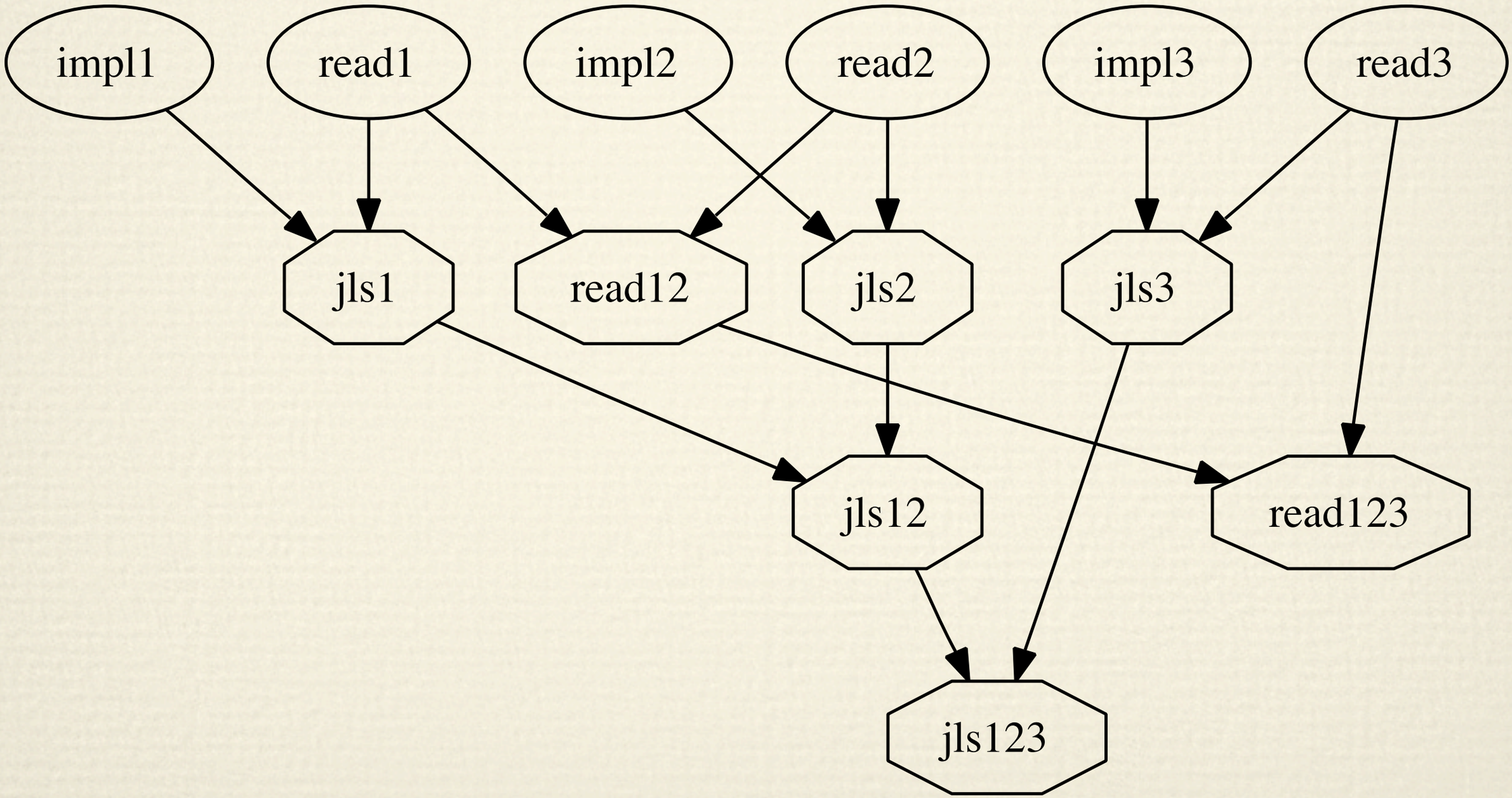
Recovering Grammar Relationships for the Java Language Specification

Ralf Lämmel and Vadim Zaytsev
Software Languages Team
Universität Koblenz-Landau

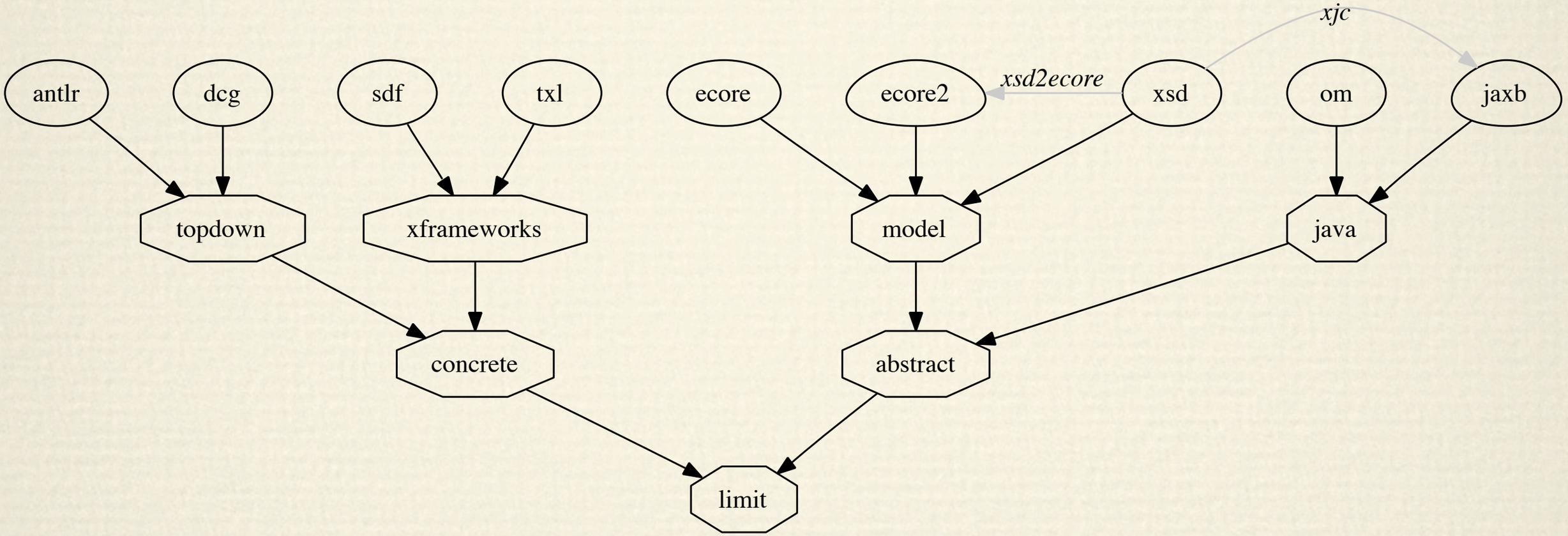
Language convergence motivated

Different versions of a language as documented by specifications



Alternative convergence scenario

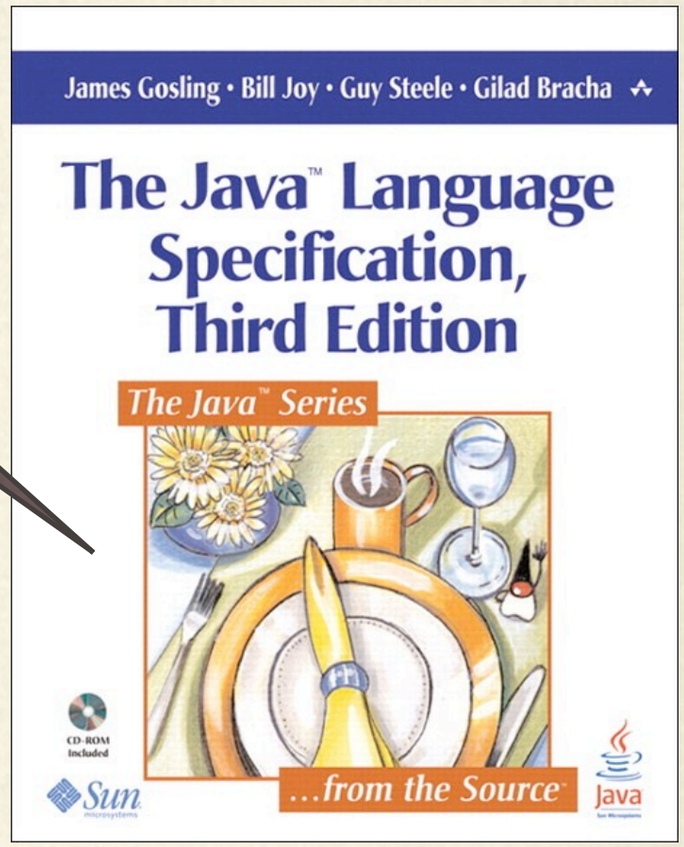
Different implementations of the same language
(parsers, data models, etc.)



Ralf Lämmel and Vadim Zaytsev, *An Introduction to Grammar Convergence*, IFM 2009,
http://www.uni-koblenz.de/~laemmel/convergence/

Java Language Specification

Assumptions?



★ The official language definition

★ Keeps up with language evolution

★ Foundation for compilers, pretty-printers, IDEs,…

★ Freely accessible in three versions

Language convergence method

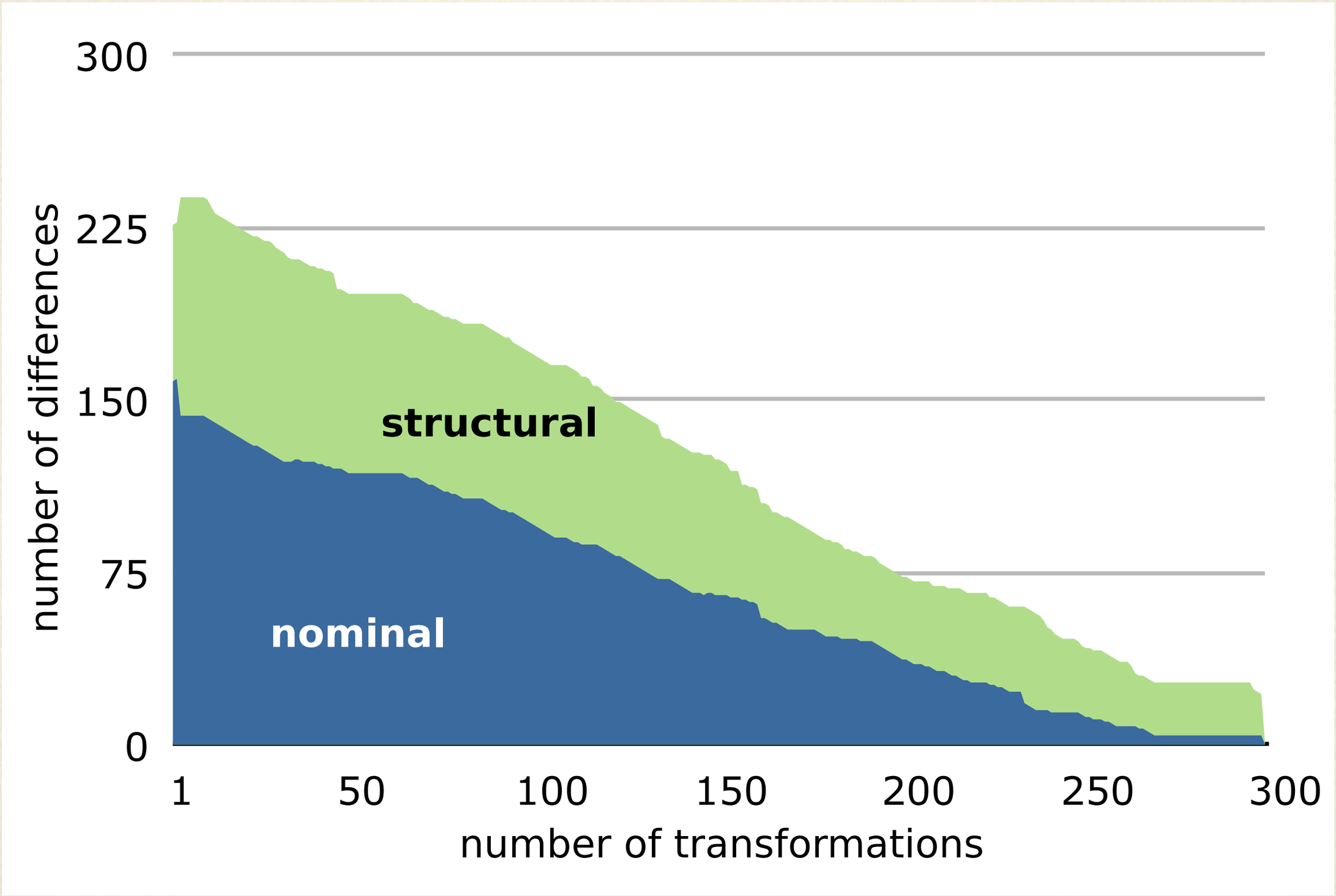
- ★ Grammar *format* free from idiosyncrasies
- ★ Grammar *extraction* for notation mapping
- ★ Grammar *comparison* for spotting grammar differences
- ★ Grammar *transformation*:
  - ✦ Refactoring; extension / restriction; revision
- ★ Grammar *measurement*:
  - ✦ Nominal differences; structural differences

Ralf Lämmel and Vadim Zaytsev, *An Introduction to Grammar Convergence*, IFM 2009, http://www.uni-koblenz.de/~laemmel/convergence/

JLS irregularities in extraction

	impl1	**impl2**	**impl3**	**read1**	**read2**	**read3**	**Total**
Arbitrary lexical decisions	2	109	60	1	90	161	423
Well-formedness violations	5	0	7	4	11	4	31
Indentation violations	1	2	7	1	4	8	23
Recovery rules	3	12	18	2	59	47	141
◦ Match parentheses	0	3	6	0	0	0	9
◦ Metasymbol to terminal	0	1	7	0	27	7	42
◦ Merge adjacent symbols	1	0	0	1	1	0	3
◦ Split compound symbol	0	1	1	0	3	8	13
◦ Nonterminal to terminal	0	7	3	0	8	11	29
◦ Terminal to nonterminal	1	0	1	1	17	13	33
◦ Recover optionality	1	0	0	0	3	8	12
Purge duplicate definitions	0	0	0	16	17	18	51
Total	11	123	92	24	181	238	669

Grammar measurement



Grammar refactoring example

BGF (*read2*)

```
ClassBodyDeclarations:
    ClassBodyDeclaration
ClassBodyDeclarations:
    ClassBodyDeclarations ClassBodyDeclaration
ClassBody:
    "{" ClassBodyDeclarations ? "}"
```

```
ClassBody:
    "{" ClassBodyDeclaration * "}"
```

XBGF (*grammar refactoring*)

```
deyaccify(ClassBodyDeclarations);
inline(ClassBodyDeclarations);
massage(
 ClassBodyDeclaration + ? ,
 ClassBodyDeclaration * );
```

Grammar extension example

BGF (*read2*)

ClassModifier:
"public"
"protected"
"private"
"abstract"
"static"
"final"
"strictfp"

FieldModifier:
"public"
"protected"
"private"
"static"
"final"
"transient"
"volatile"

MethodModifier:
"public"
"protected"
"private"
"abstract"
"static"
"final"
"synchronized"
"native"
"strictfp"

XBGF (grammar optimisation)

unite(InterfaceModifier, Modifier);
unite(ConstructorModifier, Modifier);
unite(MethodModifier, Modifier);
unite(FieldModifier, Modifier);
... ...

Grammar revision example

BGF (*impl2*, *impl3*)

```
Expression2:
    Expression3 Expression2Rest ?
Expression2Rest:
    ( Infixop Expression3 )*
Expression2Rest:
    Expression3 "instanceof" Type
```

XBGF (*grammar correction*)

```
project(
 Expression2Rest:
    < Expression3 > "instanceof" Type
);
```

Transformation statistics for JLS

	jls1	**jls12**	**jls123**	**jls2**	**jls3**	**read12**	**read123**	**Total**
Number of lines	682	5116	2847	6772	10715	1639	3082	30853
Number of transformations	67	298	111	395	544	77	135	1627
◦ Semantics-preserving	45	239	80	283	381	31	78	1137
◦ Semantics-increasing or -decreasing	22	58	31	102	150	39	53	455
◦ Semantics-revising	—	1	—	10	13	7	4	35
Preparation phase	1	—	—	15	24	11	14	65
◦ Known bugs (Ex. 3.7)	—	—	—	1	11	—	4	16
◦ Post-extraction (Ex. 3.8)	—	—	—	7	8	7	5	27
◦ Initial correction (Ex. 3.9)	1	—	—	7	5	4	5	22
Resolution phase	21	59	31	97	139	35	43	425
◦ Extension (Ex. 3.4)	—	17	26	—	—	31	38	112
◦ Relaxation (Ex. 3.5)	18	39	5	75	112	—	2	251
◦ Correction (Ex. 3.6)	3	3	—	22	27	4	3	62

	jls1	jls12	jls123	jls2	jls3	read12	read123	Total
○ *rename*	9	4	2	9	10	—	2	36
○ *reroot*	2	—	—	2	2	2	1	9
○ *unfold*	1	10	8	11	13	2	3	48
○ *fold*	4	11	4	11	13	2	5	50
○ *inline*	3	67	8	71	100	—	1	250
○ *extract*	—	17	5	18	30	—	5	75
○ *chain*	1	—	2	—	—	1	4	8
○ *massage*	2	13	—	15	32	5	3	70
○ *distribute*	3	4	2	3	6	—	—	18
○ *factor*	1	7	3	5	24	3	1	44
○ *deyaccify*	2	20	—	25	33	4	3	87
○ *yaccify*	—	—	—	—	1	—	1	2
○ *eliminate*	1	8	1	14	22	—	—	46
○ *introduce*	—	1	30	4	13	3	34	85
○ *import*	—	—	2	—	—	—	1	3
○ *vertical*	5	7	7	8	22	5	8	62
○ *horizontal*	4	19	5	17	31	4	4	84
○ *add*	1	14	13	7	20	28	20	103
○ *appear*	—	8	11	8	25	2	17	71
○ *widen*	1	3	—	1	8	1	3	17
○ *upgrade*	—	8	—	14	20	2	2	46
○ *unite*	18	2	—	18	21	5	4	68
○ *remove*	—	10	1	11	18	—	1	41
○ *disappear*	—	7	4	11	11	—	—	33
○ *narrow*	—	—	1	—	4	—	—	5
○ *downgrade*	—	2	—	8	3	—	—	13
○ *define*	—	6	—	4	9	1	6	26
○ *undefine*	—	11	—	13	3	—	—	27
○ *redefine*	—	3	—	8	7	6	2	26
○ *inject*	—	—	—	2	4	—	1	7
○ *project*	—	1	—	1	2	—	—	4
○ *replace*	3	1	2	3	6	1	1	17
○ *unlabel*	—	—	—	—	—	—	2	2

~~Conclusion~~ Discussion

★ Language documentation is often a mess

★ Automated extraction of grammar knowledge

★ Language convergence as a method to represent relationships between grammars

★ Check out **S**oftware **L**anguage **P**rocessing **S**uite: http://slps.sf.net/