A Tale of Two Grammars

Vadim Zaytsev, SWAT, CWI

Programmable Grammar Transformations

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
);
```

Grammar

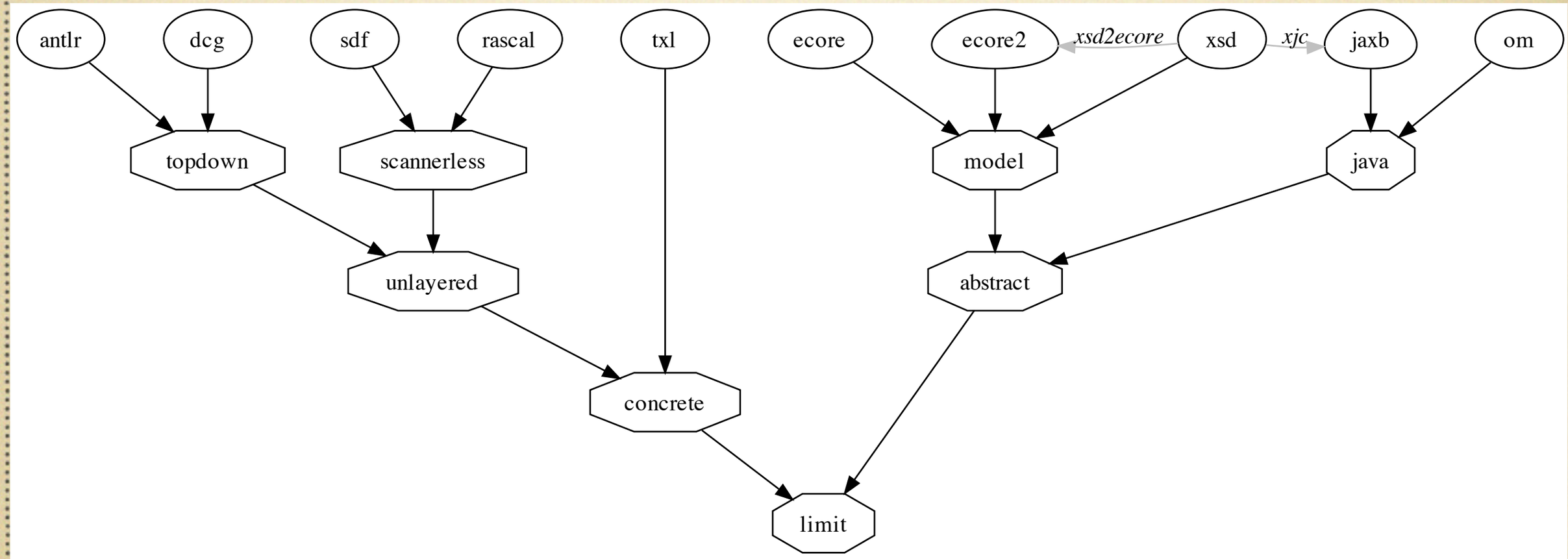
Convergence

Claims we all hear

- “This compiler implements that language”
- “This appendix contains an [insert parsing technique here] optimised grammar of the language”
- “This grammarware produces data suitable to use with that grammarware”
- “These are 100 implementations of one language”
- “This language is a subset/superset of that language”
- “This version of a compiler is backward compatible”

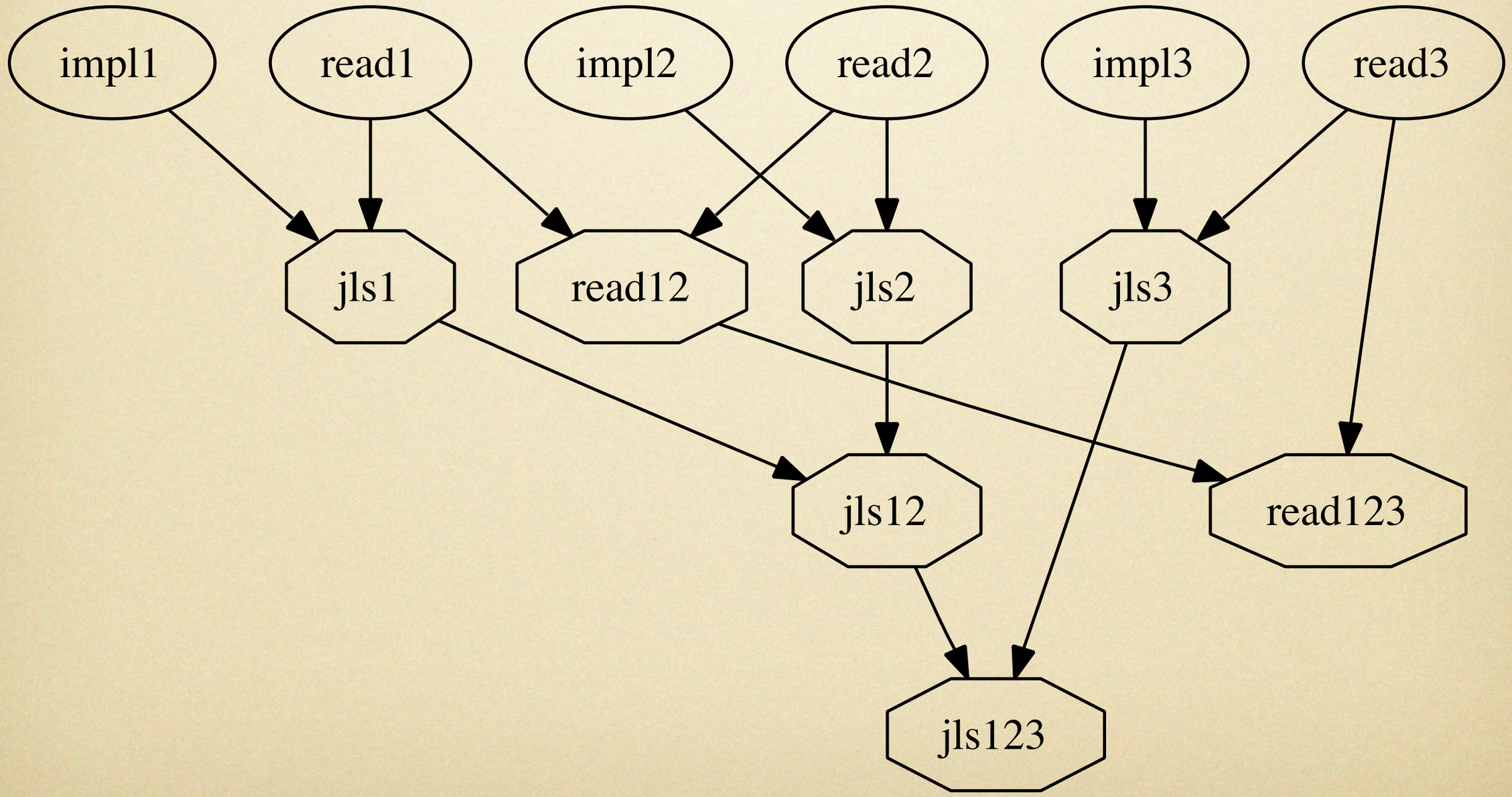
Grammar convergence

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

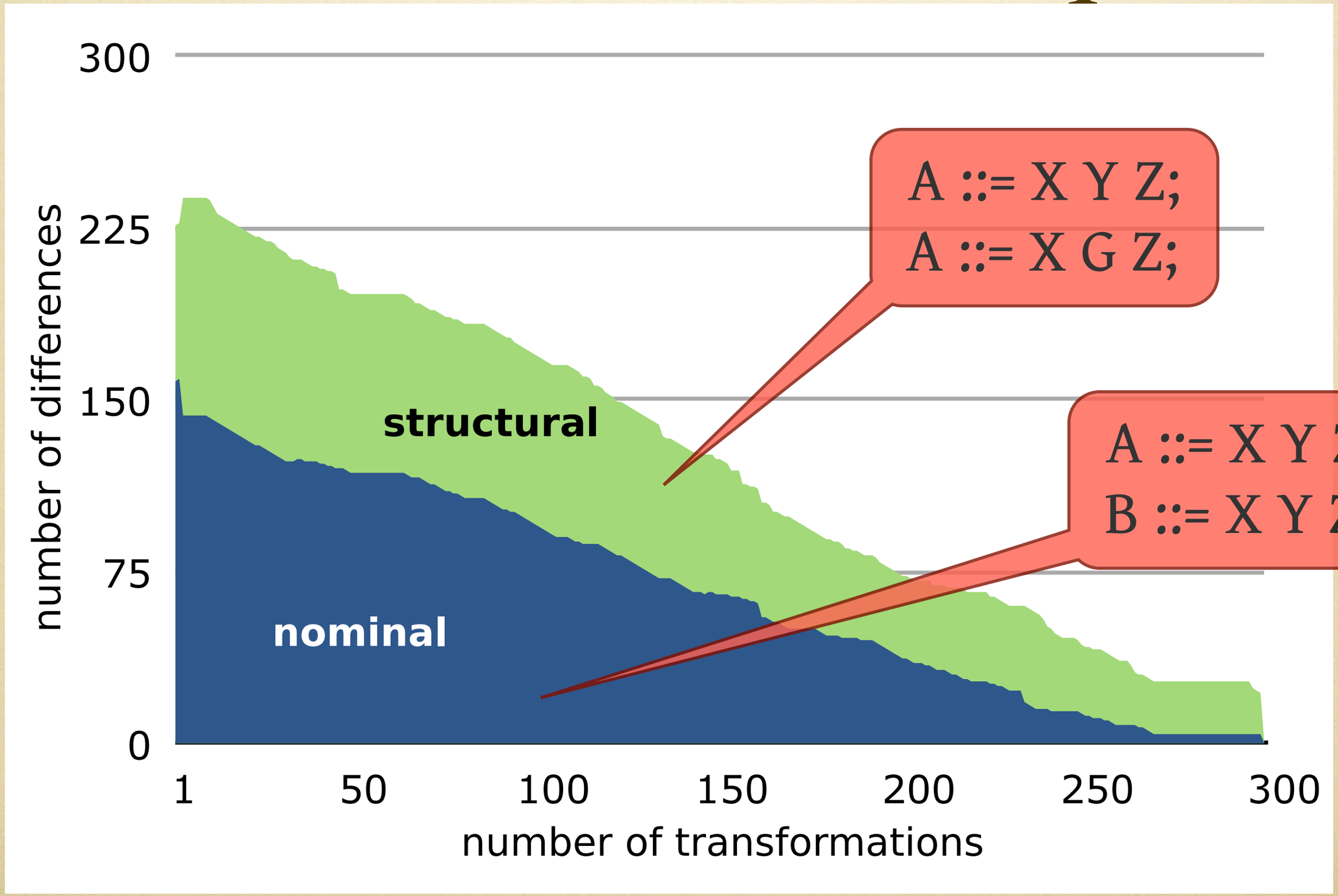


Alternative scenario

Different versions of a language as documented by specifications

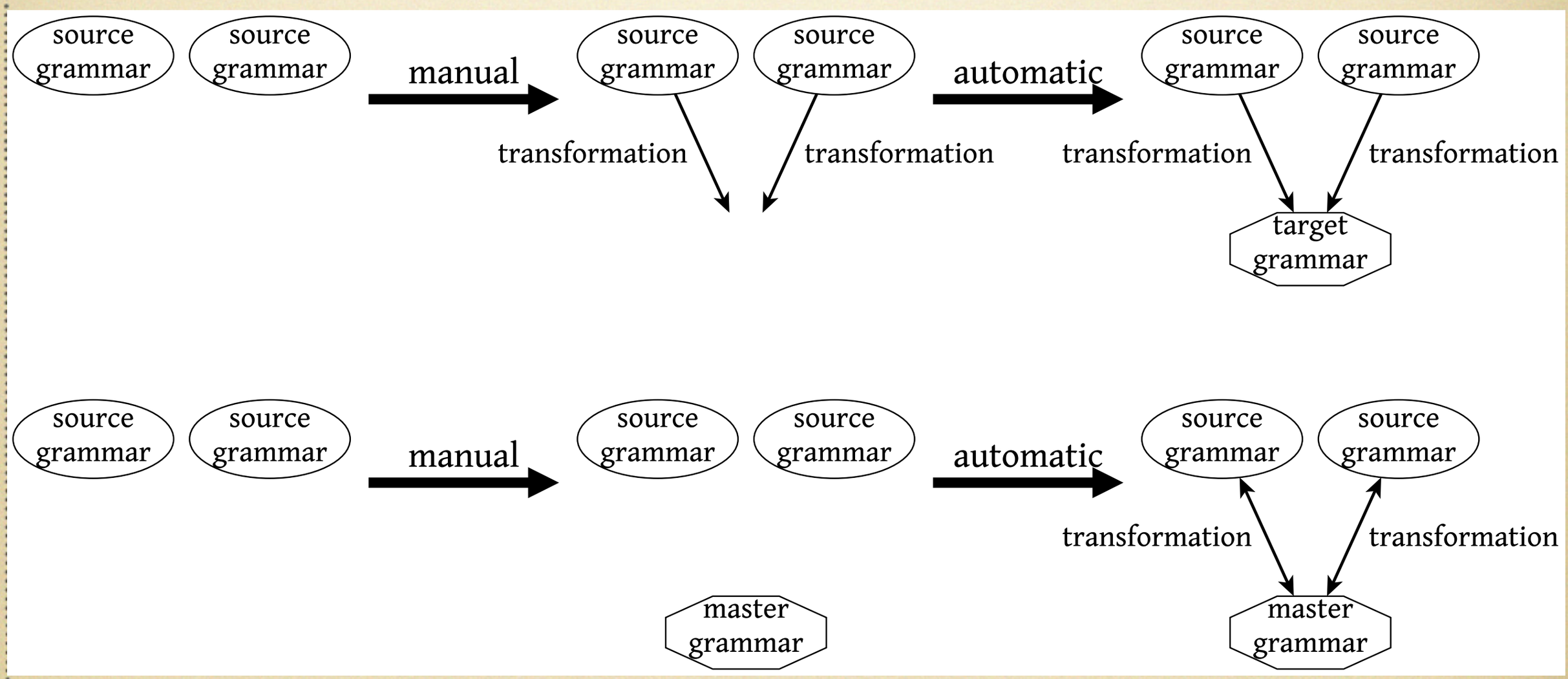


Transform until equal



Guided Grammar Convergence

Guided convergence



Abstract Normal Form

Vadim Zaytsev
@grammarware

Defining a normal form and converting your models/grammars to that normal form does wonders.

9:40 AM - 31 Jan 12 via Twitter for Android · Embed this Tweet

Reply Delete Favorite

Abstract Normal Form

- lack of selectable (named) subexpressions
- lack of labels for production rules
- lack of terminal symbols
- maximal outward factoring of inner choices
- lack of horizontally-defined nonterminals
- the set of starting symbols equals the set of non-leaf tops

Production signature

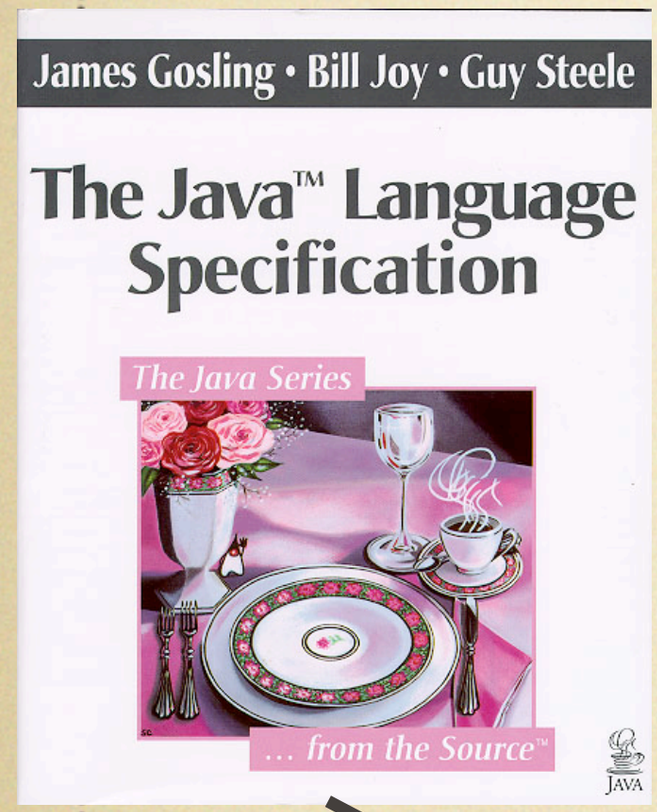
Expression		Production signature
x	nonterminal, $x \neq n$	1
n	recursion	0
$s?$	optional subexpression, $x \neq n$	$?$
$n?$	optional recursion	$\odot$
s^+	iteration, $x \neq n$, one or more	$+$
n^+	iterative recursion, one or more	$\oplus$
s^*	iteration, $x \neq n$, zero or more	$*$
n^*	iterative recursion, zero or more	$\otimes$
α	“any character” metasymbol	α
$(e_1, \ldots)$	sequence	sorted concatenation of prodsigs

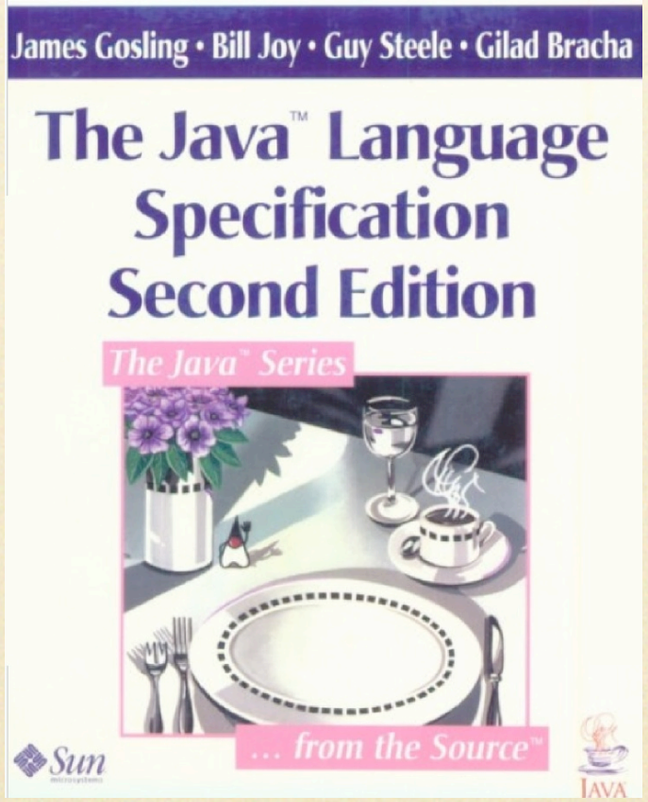
Computation of prodsigs

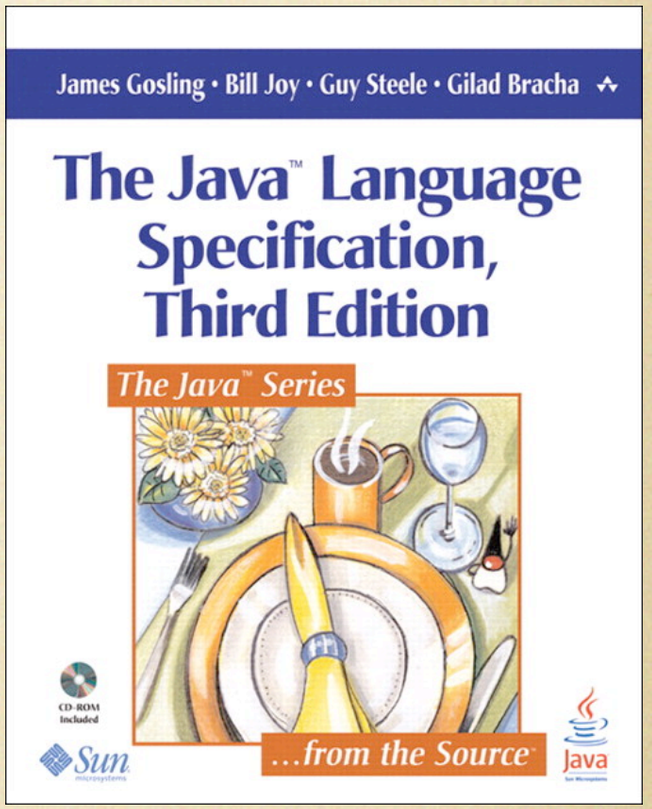
Master grammar		Rascal	
	`expr:`		`Expr:`
1	`STR`	001	`Expr Ops Expr`
	`expr:`		`Expr:`
1	`INT`	1⊕	`Name Expr`$^{+}$
	`expr:`		`Expr:`
1⊕	`STR expr`$^{+}$	000	`Expr Expr Expr`
	`expr:`		`Expr:`
001	`expr operator expr`	0	`Expr`
	`expr:`		`Expr:`
000	`expr expr expr`	1	`Name`
			`Expr:`
		1	`Int`

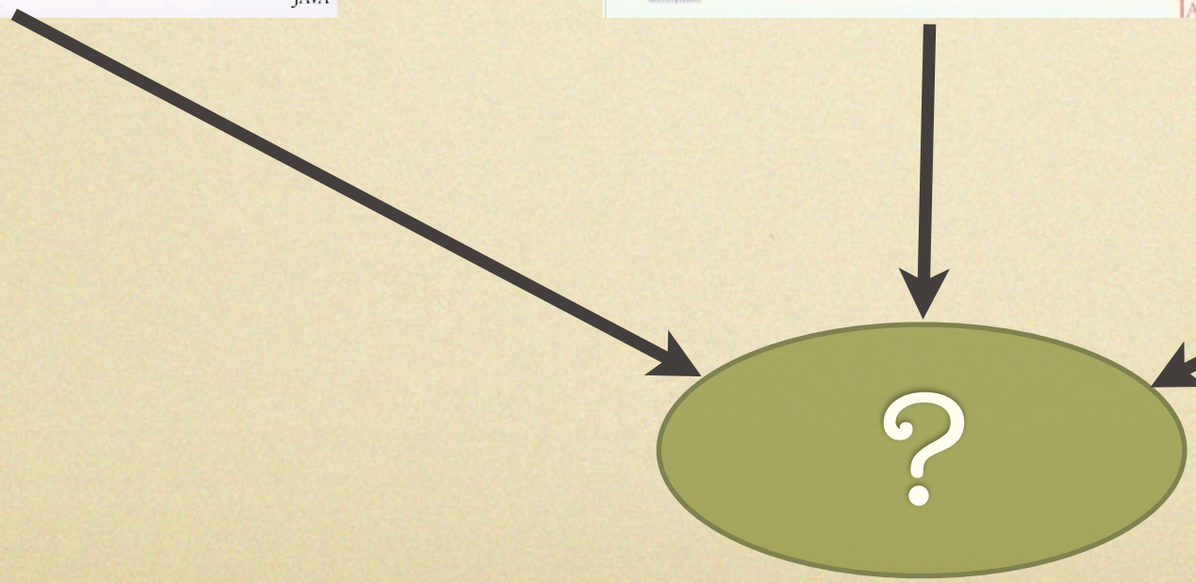
Case Studies

Case study: JLS









JLS convergence results

	jls1	**jls12**	**jls123**	**jls2**	**jls3**	**read12**	**read123**	**Total**
Number of lines	682	5114	2847	6774	10721	1639	3082	30859
Number of transformations	67	290	111	387	544	77	135	1611
○ Semantics-preserving (§4.2.2)	45	231	80	275	381	31	78	1121
○ Semantics-increasing/-decreasing	22	58	31	102	150	39	53	455
○ Semantics-revising	—	1	—	10	13	7	4	35
Preparation phase (§4.2.1)	1	—	—	15	24	11	14	65
○ Known bugs	—	—	—	1	11	—	4	16
○ Post-extraction	—	—	—	7	8	7	5	27
○ Initial correction	1	—	—	7	5	4	5	22
Resolution phase	21	59	31	97	139	35	43	425
○ Extension (§4.2.3)	—	17	26	—	—	31	38	112
○ Relaxation (§4.2.4)	18	39	5	75	112	—	2	251
○ Correction (§4.2.5)	3	3	—	22	27	4	3	62

Convergence reveals relationships

	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*	—	3	—	5	3	—	—	11
○ *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

Guided convergence of FL

	antlr	*dcg*	*sdf*	*rascal*	*txl*	*ecore*	$ecore_2$	*xsd*	*jaxb*	*om*
One to many nonterminals	−	−	−	−	−	+	−	+	−	−
Nominal mismatches	+	+	+	+	+	+	+	+	+	+
More liberal definitions	−	−	−	−	−	−	−	−	+	+
Superfluous nonterminals	+	+	+	+	+	−	−	−	−	−
Disconnected nonterminals	−	−	−	−	−	−	−	+	+	+
Maximum number of versions	1	1	1	2	2	4	1	1	1	1
Chain production rules	+	−	−	−	−	+	+	+	+	+
Permutations	−	−	−	−	−	±	+	+	+	+
Reflexive chain rules	+	+	+	+	+	+	−	−	−	−
Undefined matched as...	ε	ε	ε	ε	ε	φ	ε	ε	ε	ε
Aggregation	−	−	−	−	−	+	−	−	−	−
Layered definitions	+	+	−	−	−	−	−	−	−	−
Meaningful chain rules	−	−	−	−	−	+	−	−	−	−

Bibliography

Resources

- Lämmel, Zaytsev, *An Introduction to Grammar Convergence*, iFM 2009, LNCS 5423.
- Lämmel, Zaytsev, *Reverse Engineering Grammar Relationships*, WSR 2010.
- Zaytsev, *Language Convergence Infrastructure*, GTTSE 2009, LNCS 6491.
- Lämmel, Zaytsev, *Recovering Grammar Relationships for the JLS*, SCAM 2009, SQJ 19:2, CoRR abs/1008.4188.
- Zaytsev, *Guided Grammar Convergence*, draft.

Thank you!

grammarware.net

slps.sf.net