What Have We Done About the Unnecessary Diversity of Notation for Syntactic Definitions
Diversity of notation
Diversity of Syntactic notation

- Software languages and grammarware are everywhere.
- Grammars are extracted from documentation.
- Every language document uses its own notation.

```
<<class methods>> ::= <method definition>*
<<class initializer>> ::= <initializer definition>
```
Diversity of syntactic notation

**ISO/ECMA C#**

\[
\text{member-access:} \\
\quad \text{primary-expression} \ . \ \text{identifier} \ \text{type-argument-list}_{\text{opt}} \\
\quad \text{predefined-type} \ . \ \text{identifier} \ \text{type-argument-list}_{\text{opt}} \\
\quad \text{qualified-alias-member} \ . \ \text{identifier} \ \text{type-argument-list}_{\text{opt}}
\]

**IAL/Algol-58**

\[
\text{de} \ := \ \text{id} \ \text{or} \ \text{integer} \ \text{or} \ \text{id} \ [\text{exp}] \\
\text{del} \ := \ \text{de} \ \text{or} \ \text{del} \ , \ \text{de} \\
\text{switch declar} \ := \ \text{switch} \ \text{id} \ := \ (\text{del})
\]
Diversity of Syntactic notation

Google Dart

```plaintext
normalFormalParameters:
  normalFormalParameter (’, ’ normalFormalParameter)*
;

namedFormalParameters:
  ‘[’ defaultFormalParameter (’, ’ defaultFormalParameter)* ‘]’
;
```

IEEE Scheme

```plaintext
(cond 〈clause1)〉 〈clause2〉 ...) syntax

Syntax: Each 〈clause〉 shall be of the form
  (〈test〉 〈expression〉 ...)```
Diversity of syntactic notation

**Ada 83**

```
subtype_declaration ::= subtype identifier is subtype_indication;
subtype_indication ::= type_mark [constraint]
type_mark ::= type_name | subtype_name
```

**ISO/IEC Eiffel**

```
Note_name ≜ Identifier "::
Note_values ≜ {Note_item ",",...}+
Note_item ≜ Identifier | Manifest_constant
```
Diversity of Syntactic notation

Block ::= \{BlockStat semi\} [ResultExpr]
BlockStat ::= Import
   | ['implicit' | 'lazy'] Def
   | {LocalModifier} TmplDef
   | Expr1
   |

[50] SequenceType ::= ("empty-sequence" "(" ")")
   | (ItemType OccurrenceIndicator?)
[52] ItemType ::= KindTest | ("item" "(" ")") | AtomicType
Notation Specification
Confix metasymbols

- start grammar / end grammar
- start nonterminal / end nonterminal
- start group / end group
- start star repetition / end star repetition
- start star separator list / end star separator list
- ...

Infix metasymbols

- terminator metasymbol
- defining metasymbol
- definition separator metasymbol
- concatenation metasymbol
- exception metasymbol
- ...

Postfix & prefix metasymbols

- postfix optionality metasymbol
- postfix star repetition metasymbol
- postfix star repetition metasymbol
- start one line comment metasymbol
- ...

Conventions

- whitespace reliability
- indentation
- glue consecutive terminals
- uppercase / lowercase / camelcase / mixed case
- ...

Predefined sets

- masked terminals ("" as "x" with x being ")
- nonterminals may contain (space? underscore? slash?)
- built-in nonterminals (string? identifier? number?)
- ...

Conclusion
Results

- Analysed 69 sources with 42 different notations
- Formulated a notation specification
- Prototyped as EDD = EBNF Dialect Definition
- Created Grammar Hunter, an EDD-based recovery tool
- Extracted 64 grammars
Opportunities

Grammar extraction from language documentation
  Zaytsev, Notation-Parametric Grammar Recovery, LDTA 2012

Generating correct documentation
  Zaytsev, Lämmel, A Unified Format for Language Documents, SLE 2010

Syntactic notation evolution
  Zaytsev, Language Evolution, Metasyntactically, BX 2012

Linking documentation to compilers

Grammar readability

Documentation completion

Grammar comparison
Credits

URBAN JUNGLE — font by KC Fonts (free for non-commercial use)

A drippin’ marker — font by Wickhop (freeware)

Wickhop Handwrinting — font by Wickhop (freeware)

“What Can We Do About the Unnecessary Diversity of Notation for Syntactic Definitions?” — a paper by Niklaus Wirth

Photographs — all made by Vadim Zaytsev