Formal Foundations for Semi-parsing

Vadim Zaytsev, Universiteit van Amsterdam
CSMR-WCRE 2014 ERA

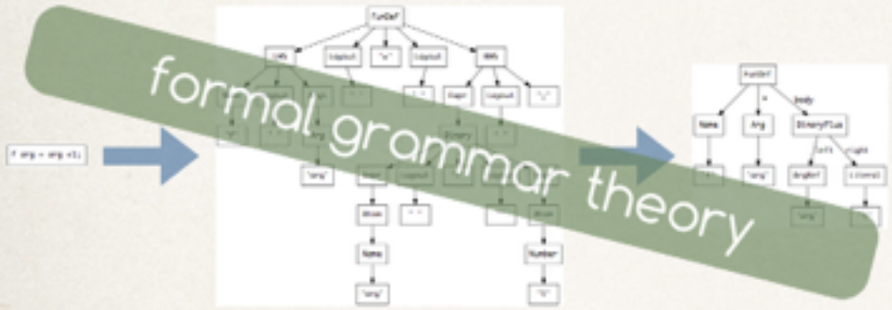
Poster in the next room

Formal Foundations for Semi-parsing

CSMR-WCRE 2014: Early Research Achievements

Vadim Zaytsev, Universiteit van Amsterdam, The Netherlands

Parsing:



(hard to automate for legacy code, unknown dialects, language cocktails, ...)

Lexical analysis:

- grep
- sed
- awk
- perl
- regular expressions
- search & replace
- ...

string manipulation

(easy to set up, very error prone)

Boolean grammars:

$A ::= B$

$A ::= B\ C\ D$

$A ::= B \mid C\ D \mid E\ F$

$A ::= (B \mid (C \,\&\, D) \mid E\ F) \,\&\, \neg X$

(conjunction, negation, disjunction, seq. composition of symbols)

Parsing schemata:

- deduction system
  - item set (parse forest specs)
  - hypotheses set
  - deduction steps
- inference relation $\vdash$
- ...

Comfortaa by Johan Aakerlund (SIL Open Font License)
Linux Libertine by Andrej Zakhar (SIL Open Font License)
Chinese black dragon by ANGELUS (CC-BY-SA, GNU FDL)
Boolean grammars by Alexander Okhotin (CSR 9, 2013)
Parsing schemata by Klaas Sikkel (Springer, 1997)

V. Zaytsev, Formal Foundations for Semi-parsing, CSMR-WCRE 2014, pp. 313–317.

URGH! THRAK CREATE FIRE! FIRE HOT!

FIRE NEAT, THRAK, BUT ME INVENT CLUB! CLUB FOR HIT THINGS WITH.

OOH...

HELLO, FRIENDS! IT SEEMS THAT I HAVE INVENTED GRAMMAR.

THANKS! ME AND THRAK CAN COMMUNICATE FAR MORE EASILY NOW THAT WE HAVE HELPING VERBS AND ARTICLE ADJECTIVES.

"THRAK AND I"

Grammar Hammer

The Grammar Hammer of 2012*

Vadim Zaytsev, vadim@grammarware.net

Software Analysis and Transformation (SWAT) Team
Centrum Wiskunde & Informatica (CWI)
Amsterdam, The Netherlands

December 19, 2012

Contents

*The title relates both to the folklore story of a steel driving man named John Henry dying with a hammer in his hand instead of losing to a steam drill [Nel08] and to a psychologist Abraham Maslow stating that if the only tool you have is a hammer, it is tempting to treat everything as if it were a nail [Mas62].

1 Introduction

The purpose of this report is documenting personal research results of the year 2012 in a form primarily intended for assessment of their scientific merit as a foundation for future work, not for quantitative assessment of the resulting publication record. This can be considered as an aggressive form of self-archiving initiative [Har01] where scientific and engineering contributions are not only logged, but also put in perspective by a separate first class atomic scientific knowledge object. This report is mostly meant for my SWAT colleagues. However, it is open for broad audience and meant to be readable by any researcher with reasonable degree of familiarity with computer science. It can be consumed as a self-contained document, but many details are not pulled in from available referenced sources.

We start right away with a the overview of the field (§2.1) followed by brief descriptions of major (§2.2) and minor (§2.3) contributions, followed by a more elaborate motivation for creation of this document (§2.4). Next, all research topics are laid out in detail one by one (§3). For the sake of complexity, a separate overview of all involved venues (§4) is included. §5 concludes the report.

2 Preliminaries

2.1 Background notions

Software language is a concept that generalises over programming languages, markup languages, database schemata, data structures, abstract data types, data types, modelling languages, ontologies, etc. Whenever we observe some degree of *commitment to structure*, we can identify it with a language, which elements (symbols) can be separately defined and the allowed combinations of them can be somehow specified. Studying software language engineering is important because of possibly gained insights into relations between the way such languages are defined and used in different technological spaces (e.g., we can study data binding as a way to map a relational database to an object model, or language convergence as a way to compare an XML schema with a syntax definition).

1

V. Zaytsev, The Grammar Hammer of 2012, ACM CoRR 1212.4446, December 2012.

Grammar-based source code analysis

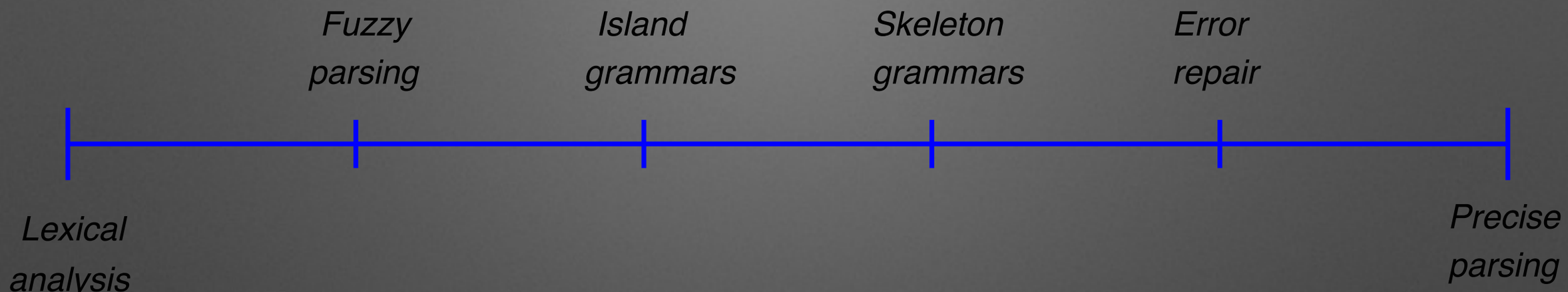


Figure 10. A spectrum of approaches for source code analysis.

- A spectrum of approaches w.r.t. tolerance
- Tolerance increases from right to left
- Figure borrowed (for extension) from:

S. Klusener, R. Lämmel, Deriving Tolerant Grammars from a Base-line Grammar, ICSM 2003


С476РА163

Semi-parsing examples

- Skeleton grammars
  - Productions for interesting constructs are reused
  - Default productions used for the rest
- Robust multilingual parsing
  - island grammars for multiple languages
  - combined and stitched together

S. Klusener, R. Lämmel, Deriving Tolerant Grammars from a Base-line Grammar, ICSM 2003.
N. Synytskyy, J. R. Cordy, T. R. Dean, Robust Multilingual Parsing Using Island Grammars, CASCON'03.

All methods of semi-parsing

- ad hoc lexical analysis
- hierarchical lexical analysis
- lexical conceptual structure
- iterative lexical analysis
- fuzzy parsing
- parsing incomplete sentences
- island grammars
- lake grammars
- robust multilingual parsing
- gap parsing
- noise skipping
- bridge grammars
- skeleton grammars
- breadth-first parsing
- iterative syntactic analysis
- grammar relaxation
- agile parsing
- permissive grammars
- hierarchical error repair
- panic mode
- noncorrecting error recovery
- practical precise parsing

V. Zaytsev, Formal Foundations for Semi-parsing, CSMR-WCRE 2014, pp. 313–317.

Boolean grammars

- Set theory:
  - union, intersection, complement
- Context-free grammars:
  - disjunction
- Conjunctive grammars:
  - disjunction, conjunction
- Boolean grammars:
  - disjunction, conjunction, negation

A. Okhotin, Conjunctive and Boolean Grammars: The True General Case of the Context-Free Grammars, Computer Science Review, vol. 9, pp. 27–59, 2013.

The use for conjunction

- Statement is...
  - keyword, expression, block
- Statement is also a chunk between dots/semicolons/...
- So, we define a statement
  - as a chunk and as a detailed statement

V. Zaytsev. Modelling Robustness with Conjunctive Grammars. SATToSE, July 2013.

The use of negation

- An identifier can be anything... but a keyword
  - filtering
- Embedded SQL query
  - skip until “END-EXEC”
- Existing approaches are hard to compare
  - reject productions, lookahead restrictions, ordered disjunction, production priorities, ...

. T. R. Dean, J. R. Cordy, A. J. Malton, and K. A. Schneider, Agile Parsing in TXL, Journal of Automated Software Engineering, vol. 10, no. 4, pp. 311–336, 2003.

Parsing schemata

- Parsing process as a deduction system
  - Initial items (partial parse trees)
  - Deduction steps (based on production rules)
  - Final items (full parse trees, fact representations)
- Uniform spec of an algorithm
  - implementations can vary

K. Sikkel, Parsing Schemata — a Framework for Specification and Analysis of Parsing Algorithms. Springer, 1997.

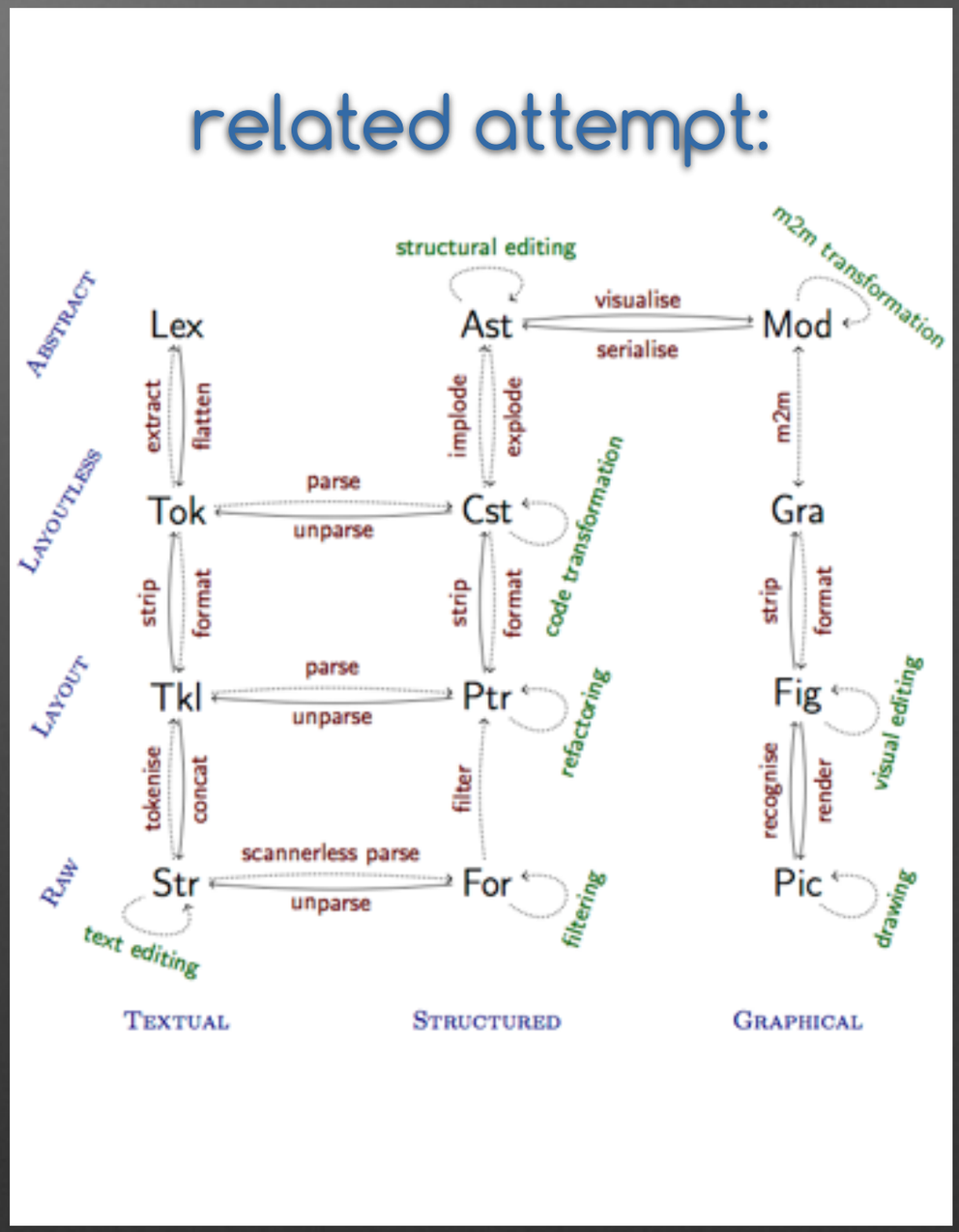
Conclusion

- Lots of methods => good
- Understanding => ???
- Mess needs to be cleaned up
- Can formal methods help?
- I'll try
  - Boolean grammars
  - parsing systems
- Comfortaa by Johan Aakerlund (SIL OFL)

Conclusion

- Lots of methods => good
- Understanding => ???
- Mess needs to be cleaned up
- Can formal methods help?
- I'll try
  - Boolean grammars
  - parsing systems
- Comfortaa by Johan Aakerlund (SIL OFL)

related attempt:



V. Zaytsev, A. H. Bagge, A Bidirectional Megamodel of Parsing. Submitted to ECMFA 2014.