From Compilers

to Grammarware

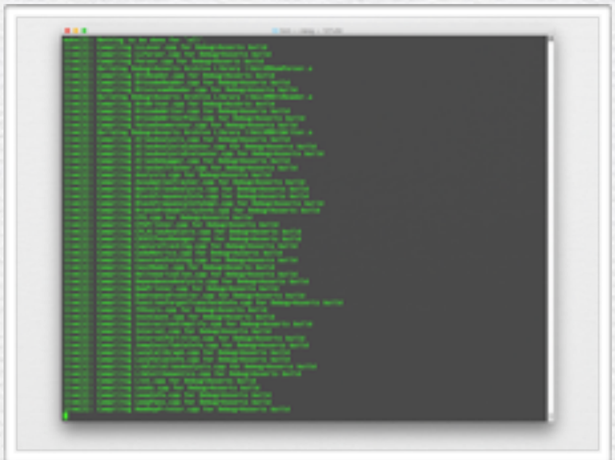
Dr. Vadim Zaytsev

From Compilers

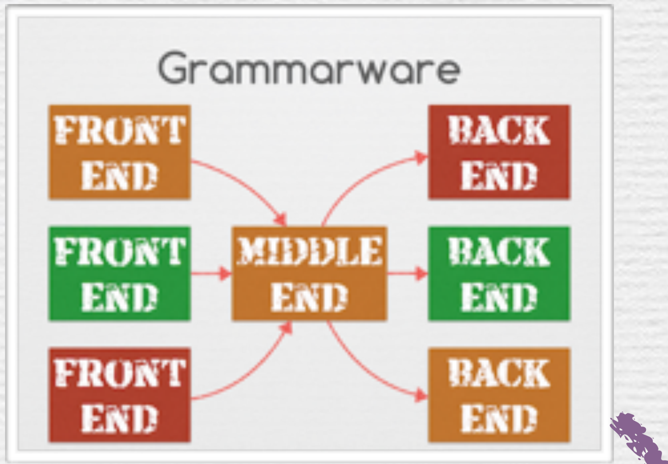
to Grammarware

Dr. Vadim Zaytsev

Introduction



Compilers



Grammarware

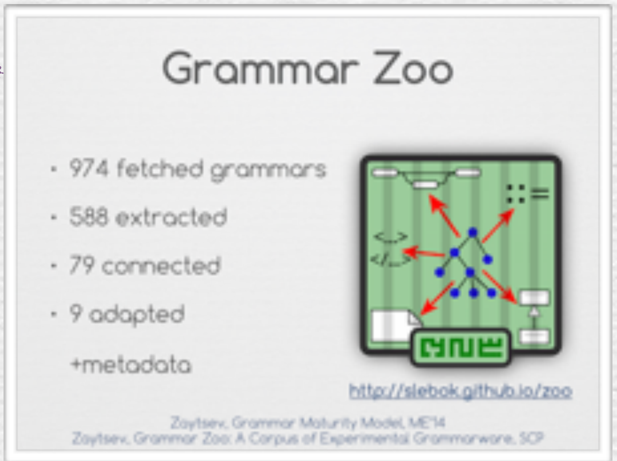
Grammarware

Case study: JLS

The Java Language Specification
The Java Language Specification Second Edition
The Java Language Specification Third Edition
?

Lämmel, Zaytsev, Recovering Grammar Relationships for the JLS, SCAM'14/SQJ

Consistency



Grammar Zoo

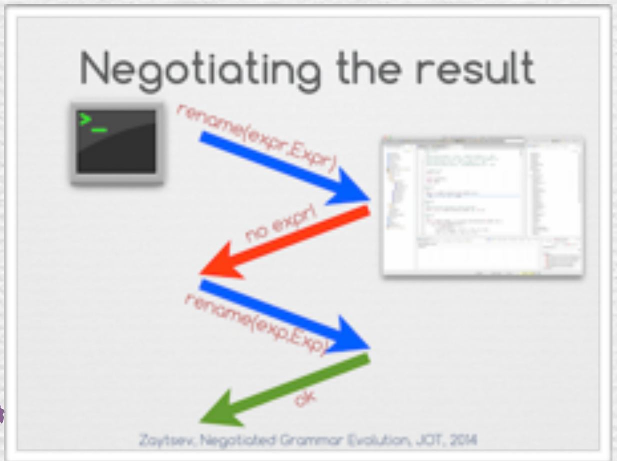
- 974 fetched grammars
- 588 extracted
- 79 connected
- 9 adapted

+metadata

http://slebok.github.io/zoo

Zaytsev, Grammar Maturity Model, ME'14
Zaytsev, Grammar Zoo: A Corpus of Experimental Grammarware, SCP

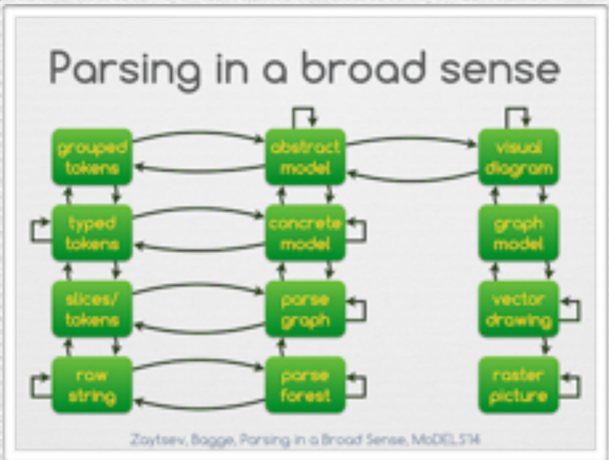
Maturity



Negotiating the result

Zaytsev, Negotiated Grammar Evolution, JOT, 2014

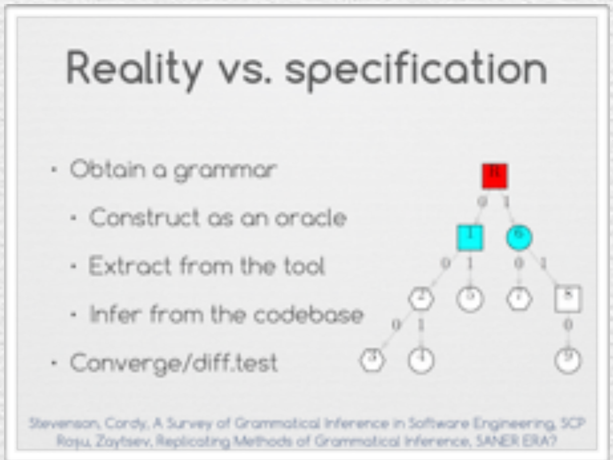
Transformation



Parsing in a broad sense

Zaytsev, Bagge, Parsing in a Broad Sense, MoDELS'14

Understanding



Reality vs. specification

- Obtain a grammar
  - Construct as an oracle
  - Extract from the tool
  - Infer from the codebase
- Converge/diff.test

Stevenson, Cordy, A Survey of Grammatical Inference in Software Engineering, SCP
Raju, Zaytsev, Replicating Methods of Grammatical Inference, SANER ERA?

Testing

Conclusion

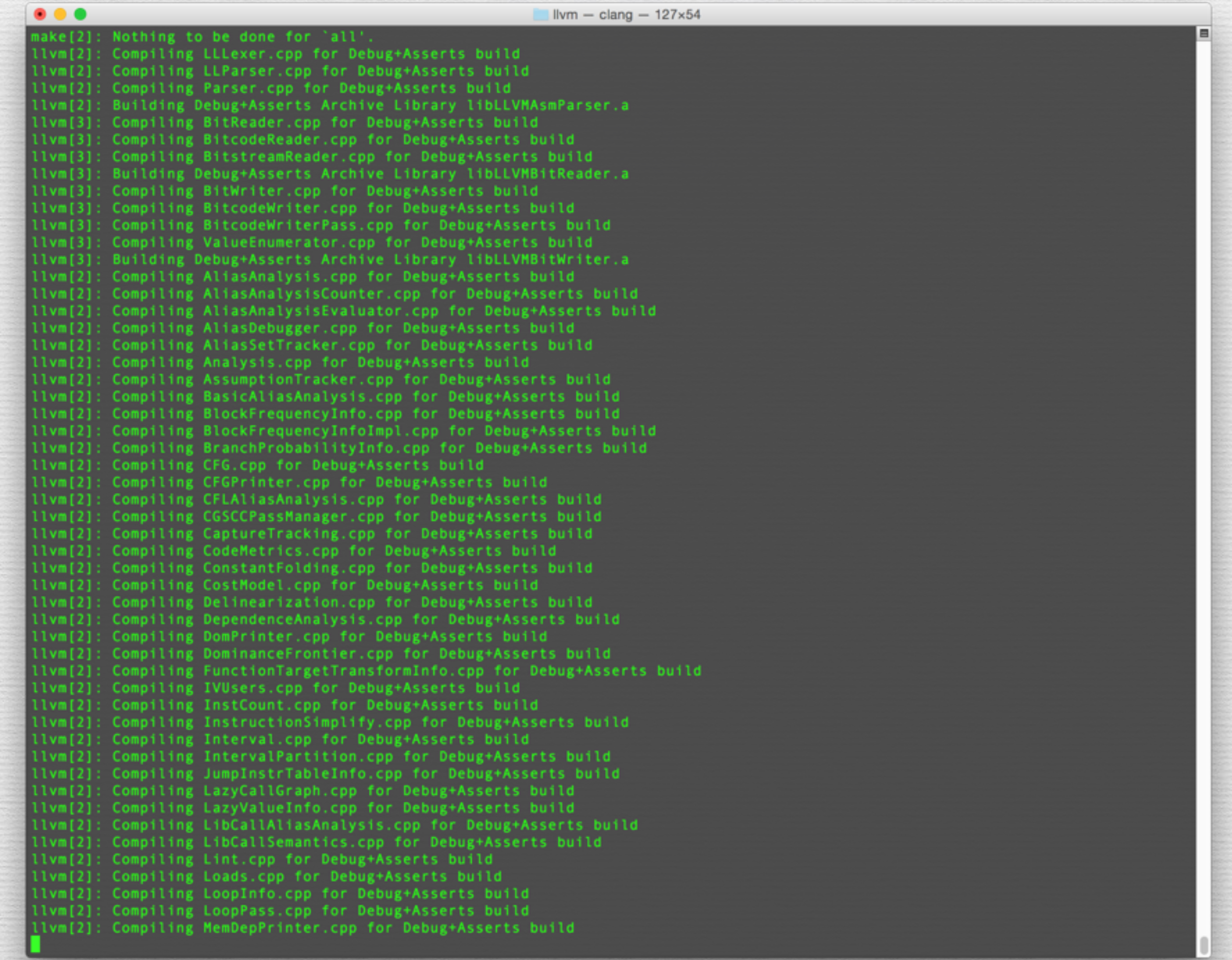
- Grammarware is more than just compilers
- Borrow methods from other domains
- Automate whenever possible
- Compare & combine
- Advance taxonomies & formalisms
- Bet on robust/tolerant methods

Conclusion

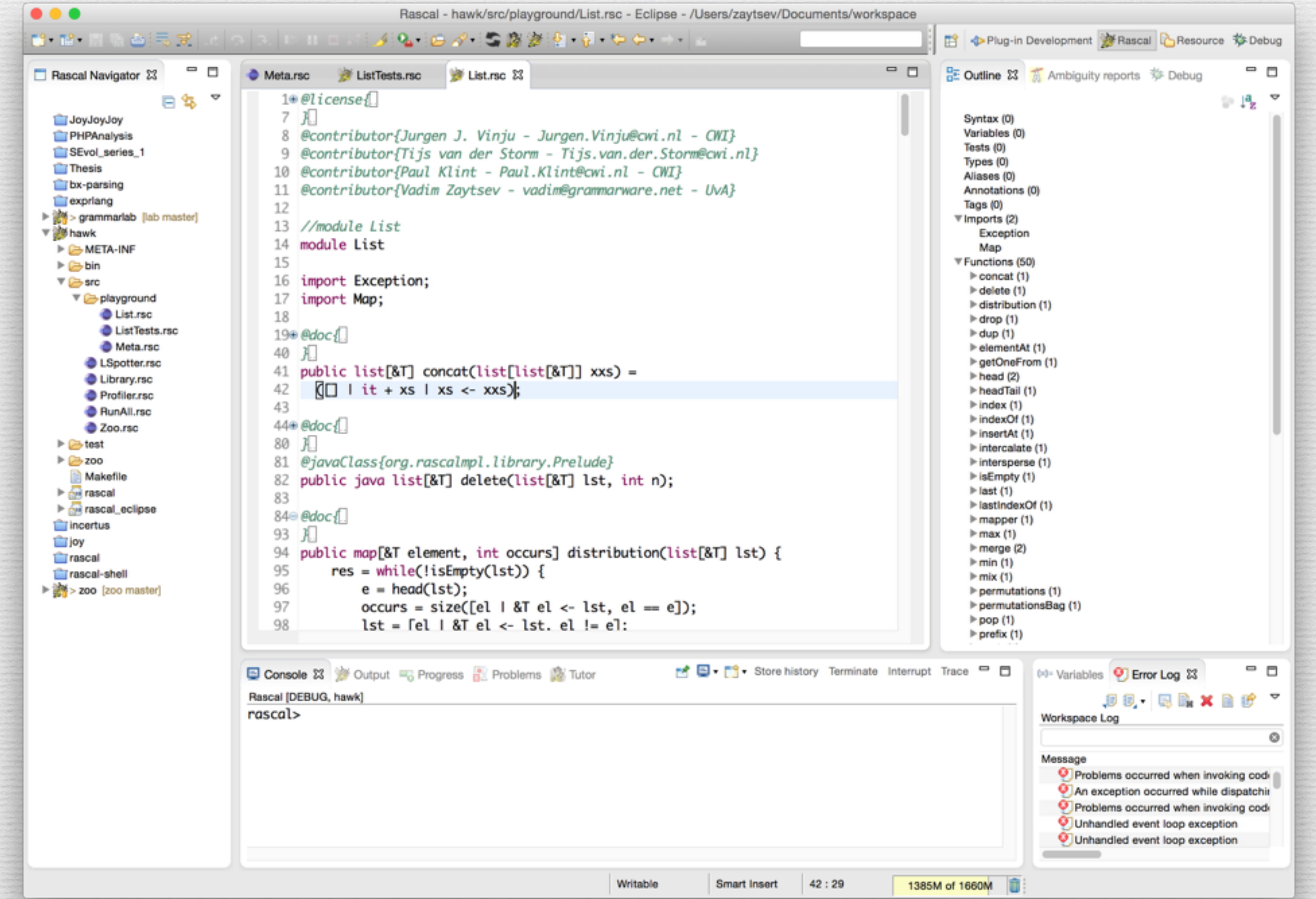
Introduction

- Vadim Zaytsev
- MSc in appl.math (2003) & telematics (2004)
- PhD in softw.lang.eng. (2010)
- Postdoc at CWI (2010–2013)
- Lecturer at UvA (2013–…)

What is a compiler?



```
make[2]: Nothing to be done for `all'.
llvm[2]: Compiling LLLexer.cpp for Debug+Asserts build
llvm[2]: Compiling LLParser.cpp for Debug+Asserts build
llvm[2]: Compiling Parser.cpp for Debug+Asserts build
llvm[2]: Building Debug+Asserts Archive Library libLLVMAsmParser.a
llvm[3]: Compiling BitReader.cpp for Debug+Asserts build
llvm[3]: Compiling BitcodeReader.cpp for Debug+Asserts build
llvm[3]: Compiling BitstreamReader.cpp for Debug+Asserts build
llvm[3]: Building Debug+Asserts Archive Library libLLVMBitReader.a
llvm[3]: Compiling BitWriter.cpp for Debug+Asserts build
llvm[3]: Compiling BitcodeWriter.cpp for Debug+Asserts build
llvm[3]: Compiling BitcodeWriterPass.cpp for Debug+Asserts build
llvm[3]: Compiling ValueEnumerator.cpp for Debug+Asserts build
llvm[3]: Building Debug+Asserts Archive Library libLLVMBitWriter.a
llvm[2]: Compiling AliasAnalysis.cpp for Debug+Asserts build
llvm[2]: Compiling AliasAnalysisCounter.cpp for Debug+Asserts build
llvm[2]: Compiling AliasAnalysisEvaluator.cpp for Debug+Asserts build
llvm[2]: Compiling AliasDebugger.cpp for Debug+Asserts build
llvm[2]: Compiling AliasSetTracker.cpp for Debug+Asserts build
llvm[2]: Compiling Analysis.cpp for Debug+Asserts build
llvm[2]: Compiling AssumptionTracker.cpp for Debug+Asserts build
llvm[2]: Compiling BasicAliasAnalysis.cpp for Debug+Asserts build
llvm[2]: Compiling BlockFrequencyInfo.cpp for Debug+Asserts build
llvm[2]: Compiling BlockFrequencyInfoImpl.cpp for Debug+Asserts build
llvm[2]: Compiling BranchProbabilityInfo.cpp for Debug+Asserts build
llvm[2]: Compiling CFG.cpp for Debug+Asserts build
llvm[2]: Compiling CFGPrinter.cpp for Debug+Asserts build
llvm[2]: Compiling CFLAliasAnalysis.cpp for Debug+Asserts build
llvm[2]: Compiling CGSCCPassManager.cpp for Debug+Asserts build
llvm[2]: Compiling CaptureTracking.cpp for Debug+Asserts build
llvm[2]: Compiling CodeMetrics.cpp for Debug+Asserts build
llvm[2]: Compiling ConstantFolding.cpp for Debug+Asserts build
llvm[2]: Compiling CostModel.cpp for Debug+Asserts build
llvm[2]: Compiling Delinearization.cpp for Debug+Asserts build
llvm[2]: Compiling DependenceAnalysis.cpp for Debug+Asserts build
llvm[2]: Compiling DomPrinter.cpp for Debug+Asserts build
llvm[2]: Compiling DominanceFrontier.cpp for Debug+Asserts build
llvm[2]: Compiling FunctionTargetTransformInfo.cpp for Debug+Asserts build
llvm[2]: Compiling IVUsers.cpp for Debug+Asserts build
llvm[2]: Compiling InstCount.cpp for Debug+Asserts build
llvm[2]: Compiling InstructionSimplify.cpp for Debug+Asserts build
llvm[2]: Compiling Interval.cpp for Debug+Asserts build
llvm[2]: Compiling IntervalPartition.cpp for Debug+Asserts build
llvm[2]: Compiling JumpInstrTableInfo.cpp for Debug+Asserts build
llvm[2]: Compiling LazyCallGraph.cpp for Debug+Asserts build
llvm[2]: Compiling LazyValueInfo.cpp for Debug+Asserts build
llvm[2]: Compiling LibCallAliasAnalysis.cpp for Debug+Asserts build
llvm[2]: Compiling LibCallSemantics.cpp for Debug+Asserts build
llvm[2]: Compiling Lint.cpp for Debug+Asserts build
llvm[2]: Compiling Loads.cpp for Debug+Asserts build
llvm[2]: Compiling LoopInfo.cpp for Debug+Asserts build
llvm[2]: Compiling LoopPass.cpp for Debug+Asserts build
llvm[2]: Compiling MemDepPrinter.cpp for Debug+Asserts build
```

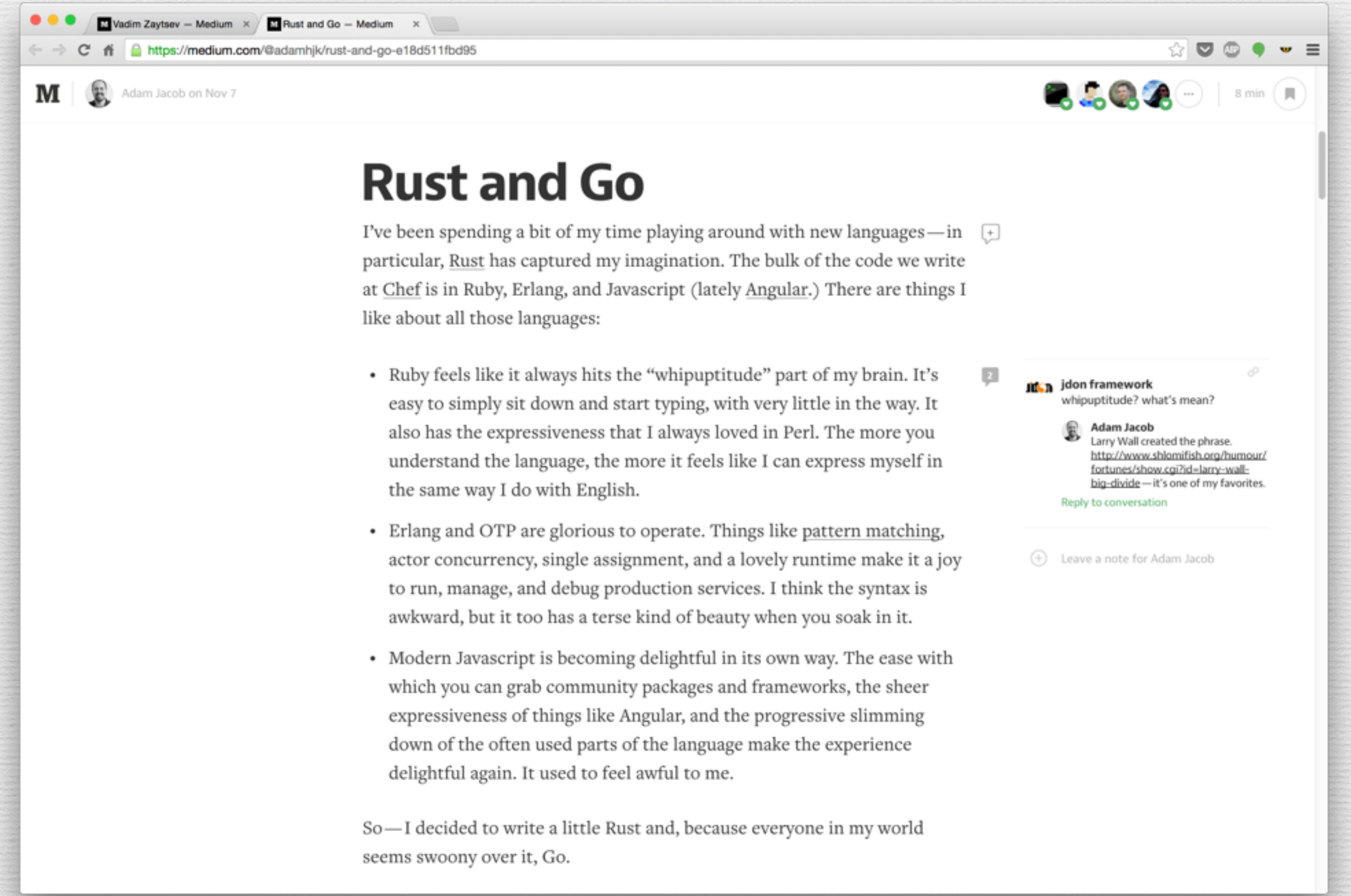


```
@license{□
}□
@contributor{Jurgen J. Vinju - Jurgen.Vinju@cwi.nl - CWI}
@contributor{Tijs van der Storm - Tijs.van.der.Storm@cwi.nl}
@contributor{Paul Klint - Paul.Klint@cwi.nl - CWI}
@contributor{Vadim Zaytsev - vadim@grammarware.net - UvA}

//module List
module List

import Exception;
import Map;

@doc{□
}□
public list[&T] concat(list[list[&T]] xxs) =
  ([] | it + xs | xs <- xxs);

@doc{□
}□
@javaClass{org.rascalmpl.library.Prelude}
public java list[&T] delete(list[&T] lst, int n);

@doc{□
}□
public map[&T element, int occurs] distribution(list[&T] lst) {
    res = while(!isEmpty(lst)) {
        e = head(lst);
        occurs = size([el | &T el <- lst, el == e]);
        lst = [el | &T el <- lst, el != e];
```

Rascal [DEBUG, hawk]

```
rascal>
```

```
Problems occurred when invoking cod
An exception occurred while dispatchi
Problems occurred when invoking cod
Unhandled event loop exception
Unhandled event loop exception
```



Rust and Go

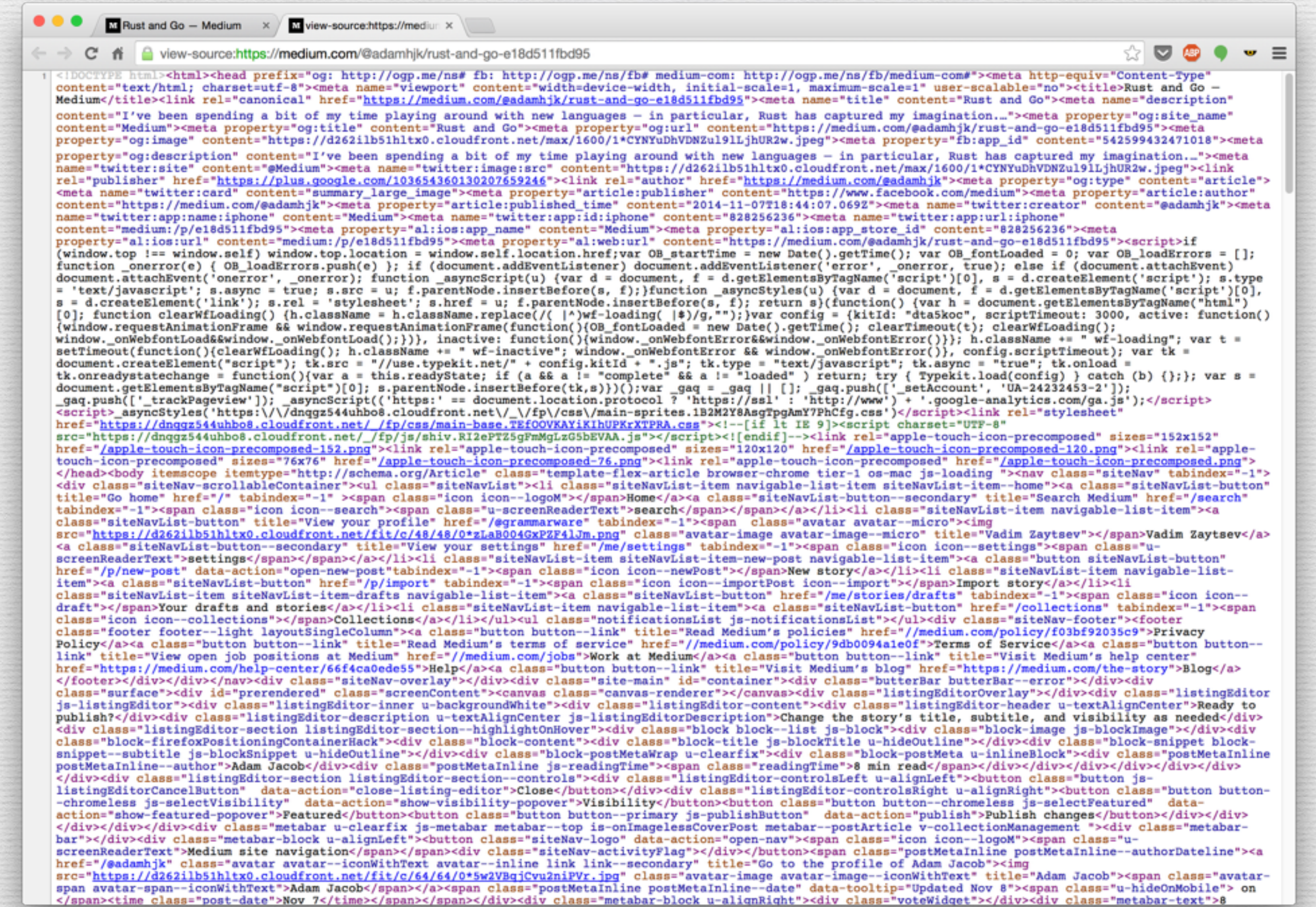
I've been spending a bit of my time playing around with new languages — in particular, Rust has captured my imagination. The bulk of the code we write at Chef is in Ruby, Erlang, and Javascript (lately Angular.) There are things I like about all those languages:

- Ruby feels like it always hits the "whipuptitude" part of my brain. It's easy to simply sit down and start typing, with very little in the way. It also has the expressiveness that I always loved in Perl. The more you understand the language, the more it feels like I can express myself in the same way I do with English.
- Erlang and OTP are glorious to operate. Things like pattern matching, actor concurrency, single assignment, and a lovely runtime make it a joy to run, manage, and debug production services. I think the syntax is awkward, but it too has a terse kind of beauty when you soak in it.
- Modern Javascript is becoming delightful in its own way. The ease with which you can grab community packages and frameworks, the sheer expressiveness of things like Angular, and the progressive slimming down of the often used parts of the language make the experience delightful again. It used to feel awful to me.

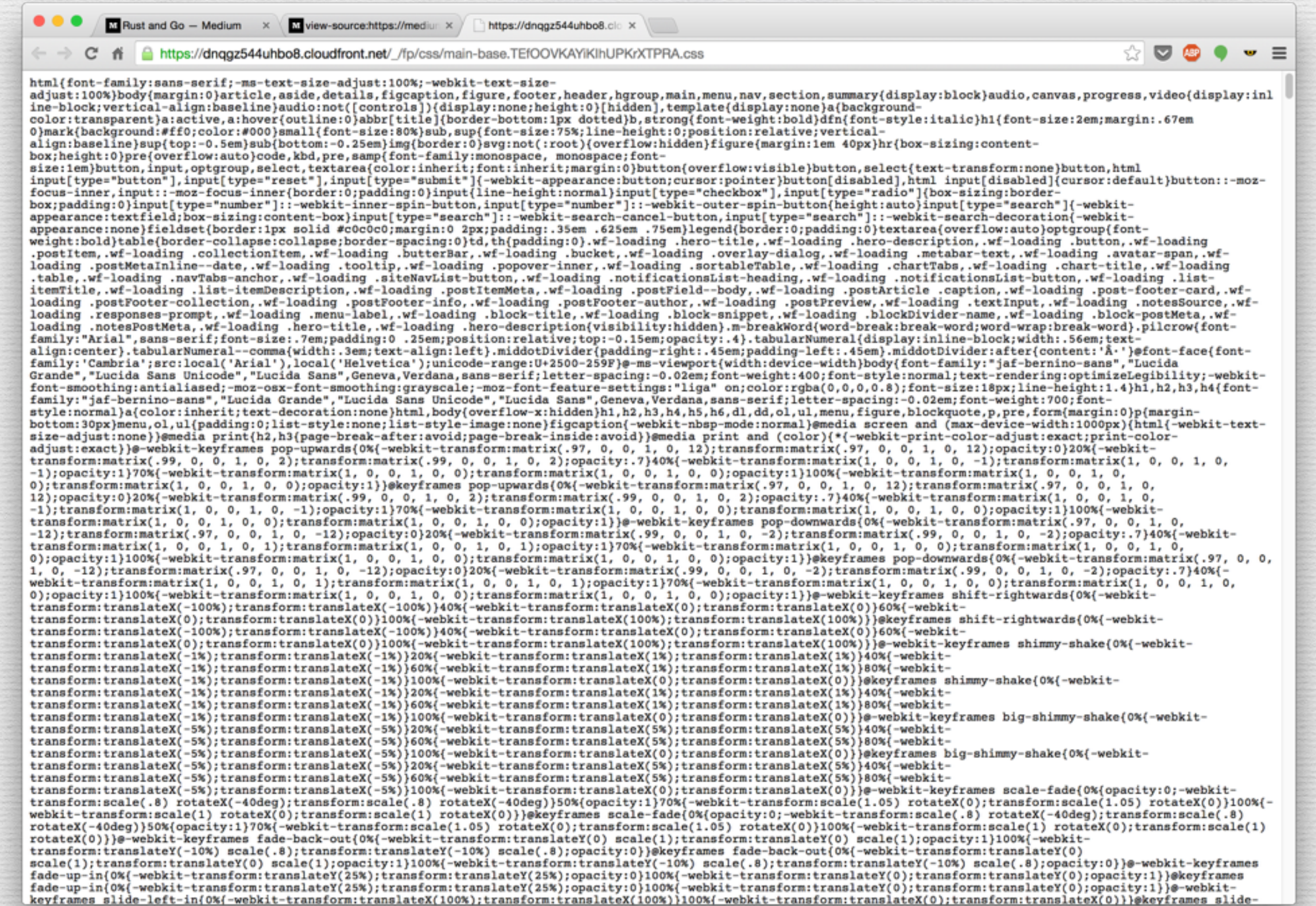
So — I decided to write a little Rust and, because everyone in my world seems swoony over it, Go.

jdon framework
whipuptitude? what's mean?

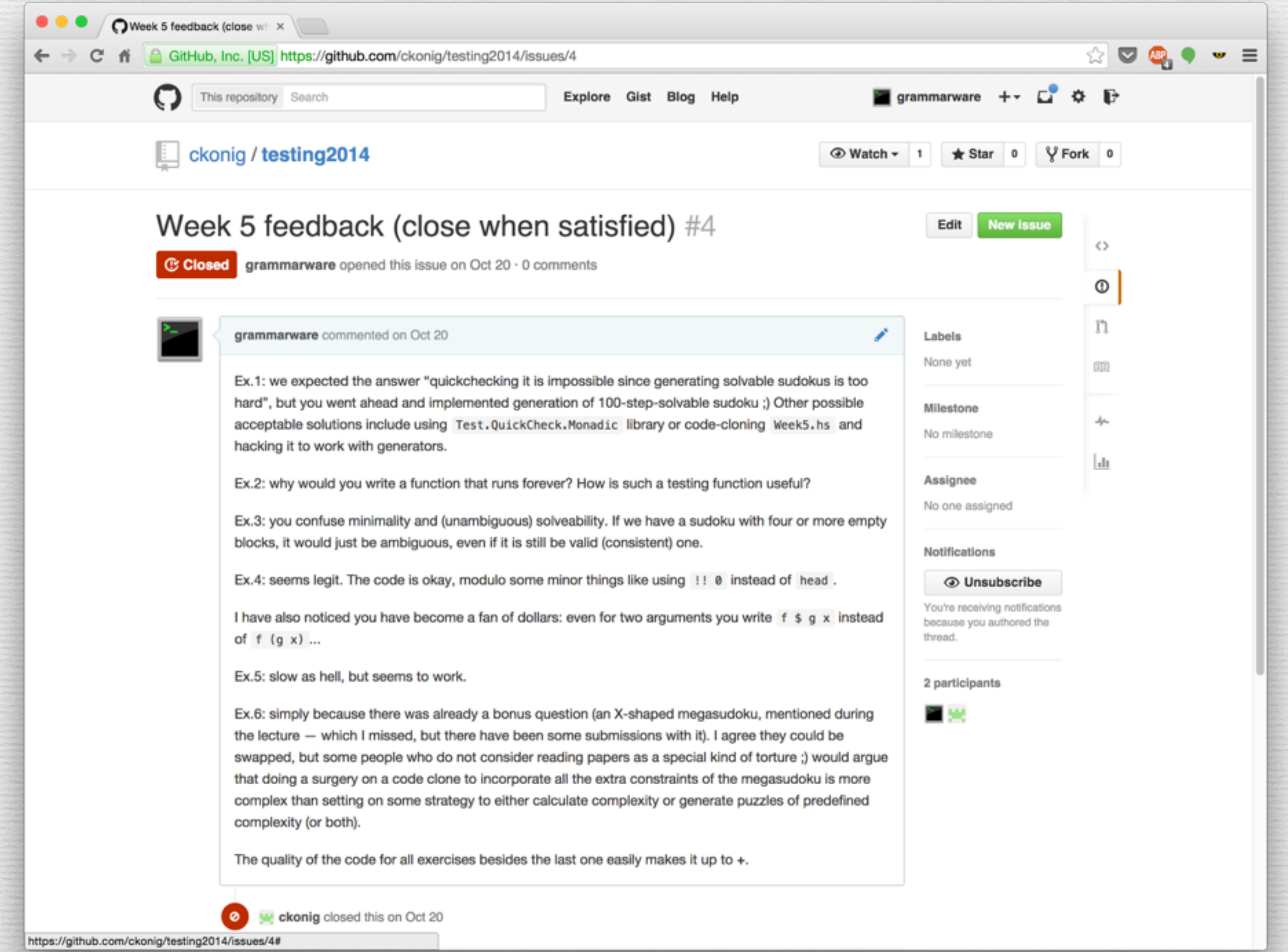
Adam Jacob
Larry Wall created the phrase. http://www.shlomifish.org/humour/fortunes/show.cgi?id=larry-wall-big-divide — it's one of my favorites.


Rust and Go — Medium
view-source:https://mediu
view-source:https://medium.com/@adamhjk/rust-and-go-e18d511fbd95

```
<!DOCTYPE html><html><head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# medium-com: http://ogp.me/ns/fb/medium-com#"><meta http-equiv="Content-Type"
content="text/html; charset=utf-8"><meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" user-scalable="no"><title>Rust and Go —
Medium</title><link rel="canonical" href="https://medium.com/@adamhjk/rust-and-go-e18d511fbd95"><meta name="title" content="Rust and Go"><meta name="description"
content="I've been spending a bit of my time playing around with new languages — in particular, Rust has captured my imagination.…"><meta property="og:site_name"
content="Medium"><meta property="og:title" content="Rust and Go"><meta property="og:url" content="https://medium.com/@adamhjk/rust-and-go-e18d511fbd95"><meta
property="og:image" content="https://d262ilb51hltx0.cloudfront.net/max/1600/1*CYNYuDhVDNZul9lLjhUR2w.jpeg"><meta property="fb:app_id" content="542599432471018"><meta
property="og:description" content="I've been spending a bit of my time playing around with new languages — in particular, Rust has captured my imagination.…"><meta
name="twitter:site" content="@Medium"><meta name="twitter:image:src" content="https://d262ilb51hltx0.cloudfront.net/max/1600/1*CYNYuDhVDNZul9lLjhUR2w.jpeg"><link
rel="publisher" href="https://plus.google.com/103654360130207659246"><link rel="author" href="https://medium.com/@adamhjk"><meta property="og:type" content="article">
<meta name="twitter:card" content="summary_large_image"><meta property="article:publisher" content="https://www.facebook.com/medium"><meta property="article:author"
content="https://medium.com/@adamhjk"><meta property="article:published_time" content="2014-11-07T18:44:07.069Z"><meta name="twitter:creator" content="@adamhjk"><meta
name="twitter:app:name:iphone" content="Medium"><meta name="twitter:app:id:iphone" content="828256236"><meta name="twitter:app:url:iphone"
content="medium://p/e18d511fbd95"><meta property="al:ios:app_name" content="Medium"><meta property="al:ios:app_store_id" content="828256236"><meta
property="al:ios:url" content="medium://p/e18d511fbd95"><meta property="al:web:url" content="https://medium.com/@adamhjk/rust-and-go-e18d511fbd95"><script>if
(window.top !== window.self) window.top.location = window.self.location.href;var OB_startTime = new Date().getTime(); var OB_fontLoaded = 0; var OB_loadErrors = [];
function _onerror(e) { OB_loadErrors.push(e) }; if (document.addEventListener) document.addEventListener('error', _onerror, true); else if (document.attachEvent)
document.attachEvent('onerror', _onerror); function _asyncScript(u) {var d = document, f = d.getElementsByTagName('script')[0], s = d.createElement('script'); s.type
= 'text/javascript'; s.async = true; s.src = u; f.parentNode.insertBefore(s, f);}function _asyncStyles(u) {var d = document, f = d.getElementsByTagName('script')[0],
s = d.createElement('link'); s.rel = 'stylesheet'; s.href = u; f.parentNode.insertBefore(s, f); return s}(function() {var h = document.getElementsByTagName("html")
[0]; function clearWfLoading() {h.className = h.className.replace(/( |^)wf-loading( |$)/g,"");}var config = {kitId: "dta5koc", scriptTimeout: 3000, active: function()
{window.requestAnimationFrame && window.requestAnimationFrame(function(){OB_fontLoaded = new Date().getTime(); clearTimeout(t); clearWfLoading();
window._onWebfontLoad&&window._onWebfontLoad();})}, inactive: function(){window._onWebfontError&&window._onWebfontError()}}; h.className += " wf-loading"; var t =
setTimeout(function(){clearWfLoading(); h.className += " wf-inactive"; window._onWebfontError && window._onWebfontError()}, config.scriptTimeout); var tk =
document.createElement("script"); tk.src = "//use.typekit.net/" + config.kitId + ".js"; tk.type = "text/javascript"; tk.async = "true"; tk.onload =
tk.onreadystatechange = function(){var a = this.readyState; if (a && a != "complete" && a != "loaded" ) return; try { Typekit.load(config) } catch (b) {};}; var s =
document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(tk,s)})();var _gaq = _gaq || []; _gaq.push(['_setAccount', 'UA-24232453-2']);
_gaq.push(['_trackPageview']); _asyncScript(('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js');</script>
<script>_asyncStyles('https:\/\/dnqgz544uhbo8.cloudfront.net\/_\/fp\/css\/main-sprites.1B2M2Y8AsgTpgAmY7PhCfg.css')</script><link rel="stylesheet"
href="https://dnqgz544uhbo8.cloudfront.net/_/fp/css/main-base.TEfOOVKAYiKIhUPKrXTPRA.css"><!--[if lt IE 9]><script charset="UTF-8"
src="https://dnqgz544uhbo8.cloudfront.net/_/fp/js/shiv.RI2ePTZ5gFmMgLzG5bEVAA.js"></script><![endif]--><link rel="apple-touch-icon-precomposed" sizes="152x152"
href="/apple-touch-icon-precomposed-152.png"><link rel="apple-touch-icon-precomposed" sizes="120x120" href="/apple-touch-icon-precomposed-120.png"><link rel="apple-
touch-icon-precomposed" sizes="76x76" href="/apple-touch-icon-precomposed-76.png"><link rel="apple-touch-icon-precomposed" href="/apple-touch-icon-precomposed.png">
</head><body itemscope itemtype="http://schema.org/Article" class="template-flex-article browser-chrome tier-1 os-mac js-loading "><nav class="siteNav" tabindex="-1">
<div class="siteNav-scrollableContainer"><ul class="siteNavList"><li class="siteNavList-item navigable-list-item siteNavList-item--home"><a class="siteNavList-button"
title="Go home" href="/" tabindex="-1" ><span class="icon icon--logoM"></span>Home</a><a class="siteNavList-button--secondary" title="Search Medium" href="/search"
tabindex="-1"><span class="icon icon--search"><span class="u-screenReaderText">search</span></span></a></li><li class="siteNavList-item navigable-list-item"><a
class="siteNavList-button" title="View your profile" href="/@grammarware" tabindex="-1"><span  class="avatar avatar--micro"><img
src="https://d262ilb51hltx0.cloudfront.net/fit/c/48/48/0*zLaB004GxPZF4lJm.png" class="avatar-image avatar-image--micro" title="Vadim Zaytsev"></span>Vadim Zaytsev</a>
<a class="siteNavList-button--secondary" title="View your settings" href="/me/settings" tabindex="-1"><span class="icon icon--settings"><span class="u-
screenReaderText">settings</span></span></a></li><li class="siteNavList-item siteNavList-item-new-post navigable-list-item"><a class="button siteNavList-button"
href="/p/new-post" data-action="open-new-post"tabindex="-1"><span class="icon icon--newPost"></span>New story</a></li><li class="siteNavList-item navigable-list-
item"><a class="siteNavList-button" href="/p/import" tabindex="-1"><span class="icon icon--importPost icon--import"></span>Import story</a></li><li
class="siteNavList-item siteNavList-item-drafts navigable-list-item"><a class="siteNavList-button" href="/me/stories/drafts" tabindex="-1"><span class="icon icon--
draft"></span>Your drafts and stories</a></li><li class="siteNavList-item navigable-list-item"><a class="siteNavList-button" href="/collections" tabindex="-1"><span
class="icon icon--collections"></span>Collections</a></li></ul><ul class="notificationsList js-notificationsList"></ul><div class="siteNav-footer"><footer
class="footer footer--light layoutSingleColumn"><a class="button button--link" title="Read Medium's policies" href="//medium.com/policy/f03bf92035c9">Privacy
Policy</a><a class="button button--link" title="Read Medium's terms of service" href="//medium.com/policy/9db0094a1e0f">Terms of Service</a><a class="button button--
link" title="View open job positions at Medium" href="//medium.com/jobs">Work at Medium</a><a class="button button--link" title="Visit Medium's help center"
href="https://medium.com/help-center/66f4ca0ede55">Help</a><a class="button button--link" title="Visit Medium's blog" href="https://medium.com/the-story">Blog</a>
</footer></div></div></nav><div class="siteNav-overlay"></div><div class="site-main" id="container"><div class="butterBar butterBar--error"></div><div
class="surface"><div id="prerendered" class="screenContent"><canvas class="canvas-renderer"></canvas><div class="listingEditorOverlay"></div><div class="listingEditor
js-listingEditor"><div class="listingEditor-inner u-backgroundWhite"><div class="listingEditor-content"><div class="listingEditor-header u-textAlignCenter">Ready to
publish?</div><div class="listingEditor-description u-textAlignCenter js-listingEditorDescription">Change the story's title, subtitle, and visibility as needed</div>
<div class="listingEditor-section listingEditor-section--highlightOnHover"><div class="block block--list js-block"><div class="block-image js-blockImage"></div><div
class="block-firefoxPositioningContainerHack"><div class="block-content"><div class="block-title js-blockTitle u-hideOutline"></div><div class="block-snippet block-
snippet--subtitle js-blockSnippet u-hideOutline"></div><div class="block-postMetaWrap u-clearfix"><div class="block-postMeta u-inlineBlock"><div class="postMetaInline
postMetaInline--author">Adam Jacob</div><div class="postMetaInline js-readingTime"><span class="readingTime">8 min read</span></div></div></div></div></div></div>
</div><div class="listingEditor-section listingEditor-section--controls"><div class="listingEditor-controlsLeft u-alignLeft"><button class="button js-
listingEditorCancelButton"  data-action="close-listing-editor">Close</button></div><div class="listingEditor-controlsRight u-alignRight"><button class="button button-
-chromeless js-selectVisibility"  data-action="show-visibility-popover">Visibility</button><button class="button button--chromeless js-selectFeatured"  data-
action="show-featured-popover">Featured</button><button class="button button--primary js-publishButton"  data-action="publish">Publish changes</button></div></div>
</div></div></div><div class="metabar u-clearfix js-metabar metabar--top is-onImagelessCoverPost metabar--postArticle v-collectionManagement "><div class="metabar-
bar"></div><div class="metabar-block u-alignLeft"><button class="siteNav-logo" data-action="open-nav"><span class="icon icon--logoM"><span class="u-
screenReaderText">Medium site navigation</span></span><div class="siteNav-activityFlag"></div></button><span class="postMetaInline postMetaInline--authorDateline"><a
href="/@adamhjk" class="avatar avatar--iconWithText avatar--inline link link--secondary" title="Go to the profile of Adam Jacob"><img
src="https://d262ilb51hltx0.cloudfront.net/fit/c/64/64/0*5w2VBqjCvu2niPVr.jpg" class="avatar-image avatar-image--iconWithText" title="Adam Jacob"><span class="avatar-
span avatar-span--iconWithText">Adam Jacob</span></a><span class="postMetaInline postMetaInline--date" data-tooltip="Updated Nov 8"><span class="u-hideOnMobile"> on
</span><time class="post-date">Nov 7</time></span></span></div><div class="metabar-block u-alignRight"><div class="voteWidget"></div><div class="metabar-text">8
```


Rust and Go — Medium
view-source:https://mediu
https://dnqgz544uhbo8.clo
https://dnqgz544uhbo8.cloudfront.net/_/fp/css/main-base.TEfOOVKAYiKlhUPKrXTPRA.css

```
html{font-family:sans-serif;-ms-text-size-adjust:100%;-webkit-text-size-
adjust:100%}body{margin:0}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inl
ine-block;vertical-align:baseline}audio:not([controls]){display:none;height:0}[hidden],template{display:none}a{background-
color:transparent}a:active,a:hover{outline:0}abbr[title]{border-bottom:1px dotted}b,strong{font-weight:bold}dfn{font-style:italic}h1{font-size:2em;margin:.67em
0}mark{background:#ff0;color:#000}small{font-size:80%}sub,sup{font-size:75%;line-height:0;position:relative;vertical-
align:baseline}sup{top:-0.5em}sub{bottom:-0.25em}img{border:0}svg:not(:root){overflow:hidden}figure{margin:1em 40px}hr{box-sizing:content-
box;height:0}pre{overflow:auto}code,kbd,pre,samp{font-family:monospace, monospace;font-
size:1em}button,input,optgroup,select,textarea{color:inherit;font:inherit;margin:0}button{overflow:visible}button,select{text-transform:none}button,html
input[type="button"],input[type="reset"],input[type="submit"]{-webkit-appearance:button;cursor:pointer}button[disabled],html input[disabled]{cursor:default}button::-moz-
focus-inner,input::-moz-focus-inner{border:0;padding:0}input{line-height:normal}input[type="checkbox"],input[type="radio"]{box-sizing:border-
box;padding:0}input[type="number"]::-webkit-inner-spin-button,input[type="number"]::-webkit-outer-spin-button{height:auto}input[type="search"]{-webkit-
appearance:textfield;box-sizing:content-box}input[type="search"]::-webkit-search-cancel-button,input[type="search"]::-webkit-search-decoration{-webkit-
appearance:none}fieldset{border:1px solid #c0c0c0;margin:0 2px;padding:.35em .625em .75em}legend{border:0;padding:0}textarea{overflow:auto}optgroup{font-
weight:bold}table{border-collapse:collapse;border-spacing:0}td,th{padding:0}.wf-loading .hero-title,.wf-loading .hero-description,.wf-loading .button,.wf-loading
.postItem,.wf-loading .collectionItem,.wf-loading .butterBar,.wf-loading .bucket,.wf-loading .overlay-dialog,.wf-loading .metabar-text,.wf-loading .avatar-span,.wf-
loading .postMetaInline--date,.wf-loading .tooltip,.wf-loading .popover-inner,.wf-loading .sortableTable,.wf-loading .chartTabs,.wf-loading .chart-title,.wf-loading
.table,.wf-loading .navTabs-anchor,.wf-loading .siteNavList-button,.wf-loading .notificationsList-heading,.wf-loading .notificationsList-button,.wf-loading .list-
itemTitle,.wf-loading .list-itemDescription,.wf-loading .postItemMeta,.wf-loading .postField--body,.wf-loading .postArticle .caption,.wf-loading .post-footer-card,.wf-
loading .postFooter-collection,.wf-loading .postFooter-info,.wf-loading .postFooter-author,.wf-loading .postPreview,.wf-loading .textInput,.wf-loading .notesSource,.wf-
loading .responses-prompt,.wf-loading .menu-label,.wf-loading .block-title,.wf-loading .block-snippet,.wf-loading .blockDivider-name,.wf-loading .block-postMeta,.wf-
loading .notesPostMeta,.wf-loading .hero-title,.wf-loading .hero-description{visibility:hidden}.m-breakWord{word-break:break-word;word-wrap:break-word}.pilcrow{font-
family:"Arial",sans-serif;font-size:.7em;padding:0 .25em;position:relative;top:-0.15em;opacity:.4}.tabularNumeral{display:inline-block;width:.56em;text-
align:center}.tabularNumeral--comma{width:.3em;text-align:left}.middotDivider{padding-right:.45em;padding-left:.45em}.middotDivider:after{content:'Â·'}@font-face{font-
family:'Cambria';src:local('Arial'),local('Helvetica');unicode-range:U+2500-259F}@-ms-viewport{width:device-width}body{font-family:"jaf-bernino-sans","Lucida
Grande","Lucida Sans Unicode","Lucida Sans",Geneva,Verdana,sans-serif;letter-spacing:-0.02em;font-weight:400;font-style:normal;text-rendering:optimizeLegibility;-webkit-
font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;-moz-font-feature-settings:"liga" on;color:rgba(0,0,0,0.8);font-size:18px;line-height:1.4}h1,h2,h3,h4{font-
family:"jaf-bernino-sans","Lucida Grande","Lucida Sans Unicode","Lucida Sans",Geneva,Verdana,sans-serif;letter-spacing:-0.02em;font-weight:700;font-
style:normal}a{color:inherit;text-decoration:none}html,body{overflow-x:hidden}h1,h2,h3,h4,h5,h6,dl,dd,ol,ul,menu,figure,blockquote,p,pre,form{margin:0}p{margin-
bottom:30px}menu,ol,ul{padding:0;list-style:none;list-style-image:none}figcaption{-webkit-nbsp-mode:normal}@media screen and (max-device-width:1000px){html{-webkit-text-
size-adjust:none}}@media print{h2,h3{page-break-after:avoid;page-break-inside:avoid}}@media print and (color){*{-webkit-print-color-adjust:exact;print-color-
adjust:exact}}@-webkit-keyframes pop-upwards{0%{-webkit-transform:matrix(.97, 0, 0, 1, 0, 12);transform:matrix(.97, 0, 0, 1, 0, 12);opacity:0}20%{-webkit-
transform:matrix(.99, 0, 0, 1, 0, 2);transform:matrix(.99, 0, 0, 1, 0, 2);opacity:.7}40%{-webkit-transform:matrix(1, 0, 0, 1, 0, -1);transform:matrix(1, 0, 0, 1, 0,
-1);opacity:1}70%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}100%{-webkit-transform:matrix(1, 0, 0, 1, 0,
0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}}@keyframes pop-upwards{0%{-webkit-transform:matrix(.97, 0, 0, 1, 0, 12);transform:matrix(.97, 0, 0, 1, 0,
12);opacity:0}20%{-webkit-transform:matrix(.99, 0, 0, 1, 0, 2);transform:matrix(.99, 0, 0, 1, 0, 2);opacity:.7}40%{-webkit-transform:matrix(1, 0, 0, 1, 0,
-1);transform:matrix(1, 0, 0, 1, 0, -1);opacity:1}70%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}100%{-webkit-
transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}}@-webkit-keyframes pop-downwards{0%{-webkit-transform:matrix(.97, 0, 0, 1, 0,
-12);transform:matrix(.97, 0, 0, 1, 0, -12);opacity:0}20%{-webkit-transform:matrix(.99, 0, 0, 1, 0, -2);transform:matrix(.99, 0, 0, 1, 0, -2);opacity:.7}40%{-webkit-
transform:matrix(1, 0, 0, 1, 0, 1);transform:matrix(1, 0, 0, 1, 0, 1);opacity:1}70%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0,
0);opacity:1}100%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}}@keyframes pop-downwards{0%{-webkit-transform:matrix(.97, 0, 0,
1, 0, -12);transform:matrix(.97, 0, 0, 1, 0, -12);opacity:0}20%{-webkit-transform:matrix(.99, 0, 0, 1, 0, -2);transform:matrix(.99, 0, 0, 1, 0, -2);opacity:.7}40%{-
webkit-transform:matrix(1, 0, 0, 1, 0, 1);transform:matrix(1, 0, 0, 1, 0, 1);opacity:1}70%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0,
0);opacity:1}100%{-webkit-transform:matrix(1, 0, 0, 1, 0, 0);transform:matrix(1, 0, 0, 1, 0, 0);opacity:1}}@-webkit-keyframes shift-rightwards{0%{-webkit-
transform:translateX(-100%);transform:translateX(-100%)}40%{-webkit-transform:translateX(0);transform:translateX(0)}60%{-webkit-
transform:translateX(0);transform:translateX(0)}100%{-webkit-transform:translateX(100%);transform:translateX(100%)}}@keyframes shift-rightwards{0%{-webkit-
transform:translateX(-100%);transform:translateX(-100%)}40%{-webkit-transform:translateX(0);transform:translateX(0)}60%{-webkit-
transform:translateX(0);transform:translateX(0)}100%{-webkit-transform:translateX(100%);transform:translateX(100%)}}@-webkit-keyframes shimmy-shake{0%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}20%{-webkit-transform:translateX(1%);transform:translateX(1%)}40%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}60%{-webkit-transform:translateX(1%);transform:translateX(1%)}80%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}100%{-webkit-transform:translateX(0);transform:translateX(0)}}@keyframes shimmy-shake{0%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}20%{-webkit-transform:translateX(1%);transform:translateX(1%)}40%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}60%{-webkit-transform:translateX(1%);transform:translateX(1%)}80%{-webkit-
transform:translateX(-1%);transform:translateX(-1%)}100%{-webkit-transform:translateX(0);transform:translateX(0)}}@-webkit-keyframes big-shimmy-shake{0%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}20%{-webkit-transform:translateX(5%);transform:translateX(5%)}40%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}60%{-webkit-transform:translateX(5%);transform:translateX(5%)}80%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}100%{-webkit-transform:translateX(0);transform:translateX(0)}}@keyframes big-shimmy-shake{0%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}20%{-webkit-transform:translateX(5%);transform:translateX(5%)}40%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}60%{-webkit-transform:translateX(5%);transform:translateX(5%)}80%{-webkit-
transform:translateX(-5%);transform:translateX(-5%)}100%{-webkit-transform:translateX(0);transform:translateX(0)}}@-webkit-keyframes scale-fade{0%{opacity:0;-webkit-
transform:scale(.8) rotateX(-40deg);transform:scale(.8) rotateX(-40deg)}50%{opacity:1}70%{-webkit-transform:scale(1.05) rotateX(0);transform:scale(1.05) rotateX(0)}100%{-
webkit-transform:scale(1) rotateX(0);transform:scale(1) rotateX(0)}}@keyframes scale-fade{0%{opacity:0;-webkit-transform:scale(.8) rotateX(-40deg);transform:scale(.8)
rotateX(-40deg)}50%{opacity:1}70%{-webkit-transform:scale(1.05) rotateX(0);transform:scale(1.05) rotateX(0)}100%{-webkit-transform:scale(1) rotateX(0);transform:scale(1)
rotateX(0)}}@-webkit-keyframes fade-back-out{0%{-webkit-transform:translateY(0) scale(1);transform:translateY(0) scale(1);opacity:1}100%{-webkit-
transform:translateY(-10%) scale(.8);transform:translateY(-10%) scale(.8);opacity:0}}@keyframes fade-back-out{0%{-webkit-transform:translateY(0)
scale(1);transform:translateY(0) scale(1);opacity:1}100%{-webkit-transform:translateY(-10%) scale(.8);transform:translateY(-10%) scale(.8);opacity:0}}@-webkit-keyframes
fade-up-in{0%{-webkit-transform:translateY(25%);transform:translateY(25%);opacity:0}100%{-webkit-transform:translateY(0);transform:translateY(0);opacity:1}}@keyframes
fade-up-in{0%{-webkit-transform:translateY(25%);transform:translateY(25%);opacity:0}100%{-webkit-transform:translateY(0);transform:translateY(0);opacity:1}}@-webkit-
keyframes slide-left-in{0%{-webkit-transform:translateX(100%);transform:translateX(100%)}100%{-webkit-transform:translateX(0);transform:translateX(0)}}@keyframes slide-
```



Week 5 feedback (close when satisfied) #4

Closed **grammarware** opened this issue on Oct 20 · 0 comments

grammarware commented on Oct 20

Ex.1: we expected the answer "quickchecking it is impossible since generating solvable sudokus is too hard", but you went ahead and implemented generation of 100-step-solvable sudoku ;) Other possible acceptable solutions include using `Test.QuickCheck.Monadic` library or code-cloning `Week5.hs` and hacking it to work with generators.

Ex.2: why would you write a function that runs forever? How is such a testing function useful?

Ex.3: you confuse minimality and (unambiguous) solveability. If we have a sudoku with four or more empty blocks, it would just be ambiguous, even if it is still be valid (consistent) one.

Ex.4: seems legit. The code is okay, modulo some minor things like using `!! 0` instead of `head`.

I have also noticed you have become a fan of dollars: even for two arguments you write `f $ g x` instead of `f (g x)` ...

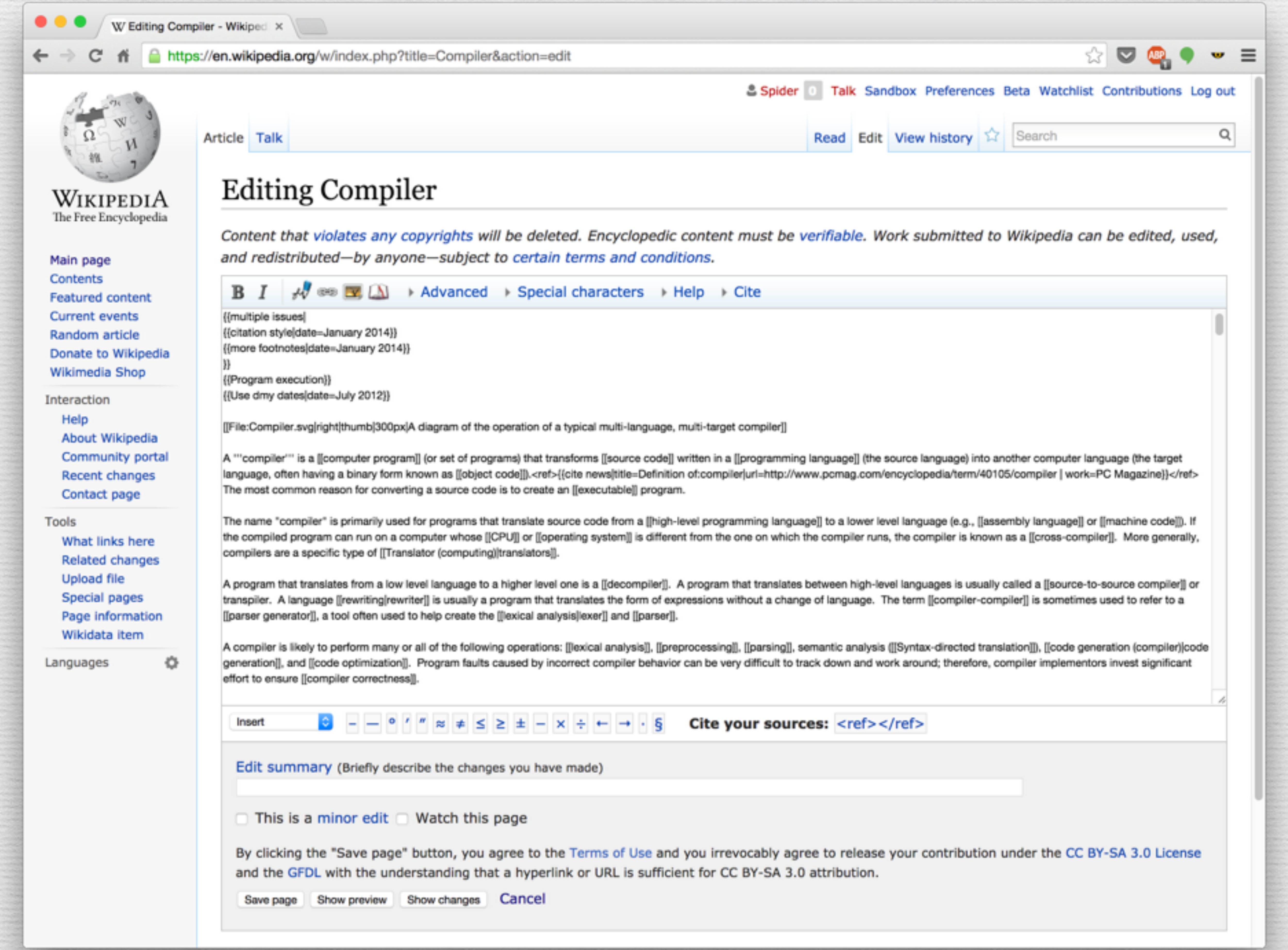
Ex.5: slow as hell, but seems to work.

Ex.6: simply because there was already a bonus question (an X-shaped megasudoku, mentioned during the lecture — which I missed, but there have been some submissions with it). I agree they could be swapped, but some people who do not consider reading papers as a special kind of torture ;) would argue that doing a surgery on a code clone to incorporate all the extra constraints of the megasudoku is more complex than setting on some strategy to either calculate complexity or generate puzzles of predefined complexity (or both).

The quality of the code for all exercises besides the last one easily makes it up to +.

ckonig closed this on Oct 20

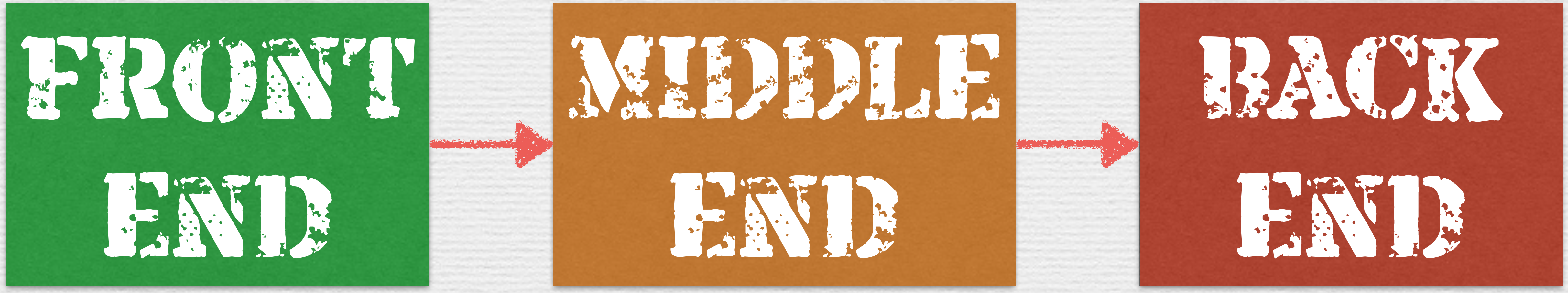
Labels
None yet
Milestone
No milestone
Assignee
No one assigned
Notifications
Unsubscribe
You're receiving notifications because you authored the thread.
2 participants
https://github.com/ckonig/testing2014/issues/4#



Language processing

- Internal structures
  - databases, configurations, tables, ...
- External structures
  - protocols, interfaces, bytecode, ...
- Software language
  - programming, modelling, markup, ...

Compiler

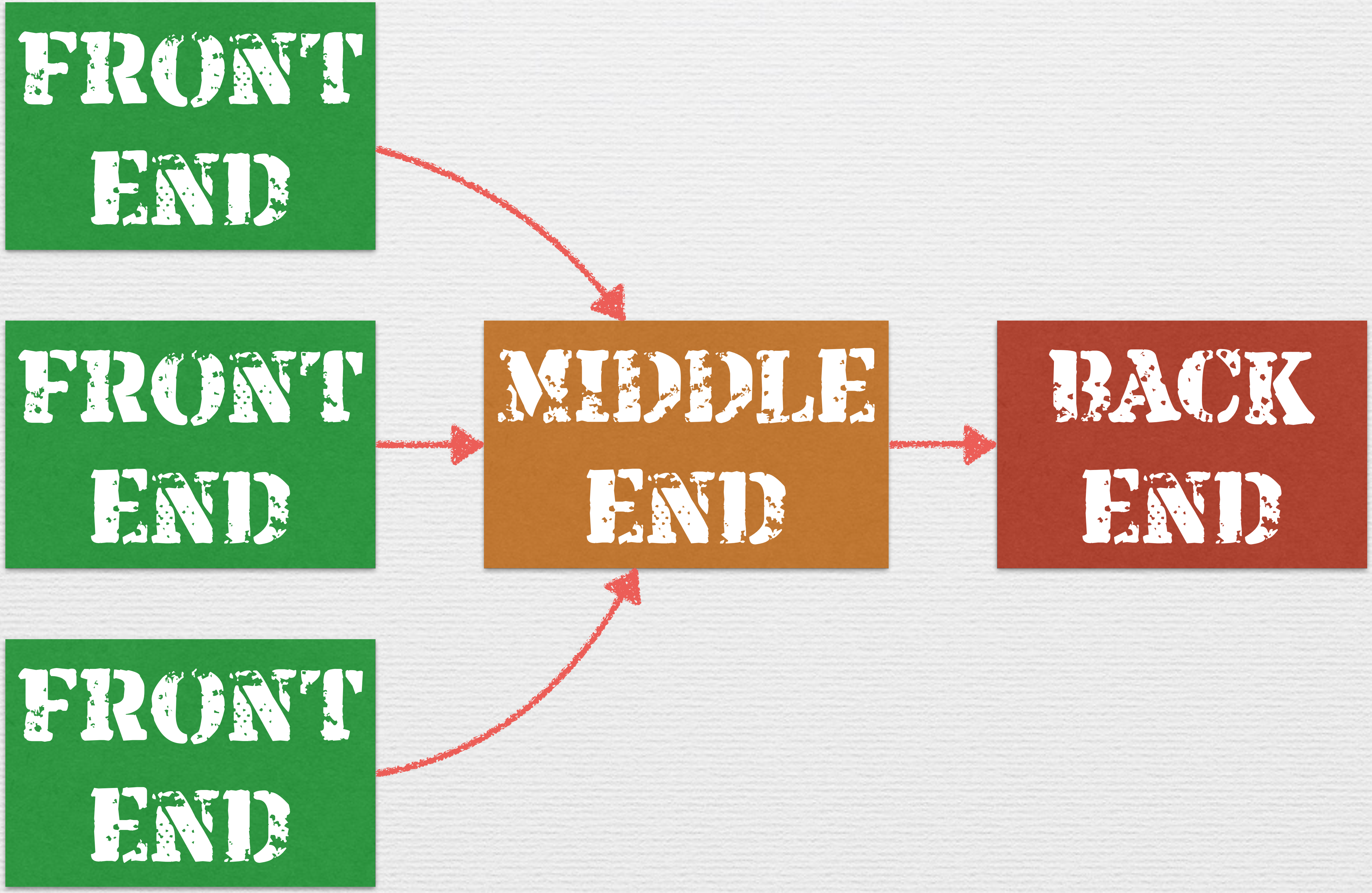


FRONT END

MIDDLE END

BACK END

Multi-language compiler



FRONT END

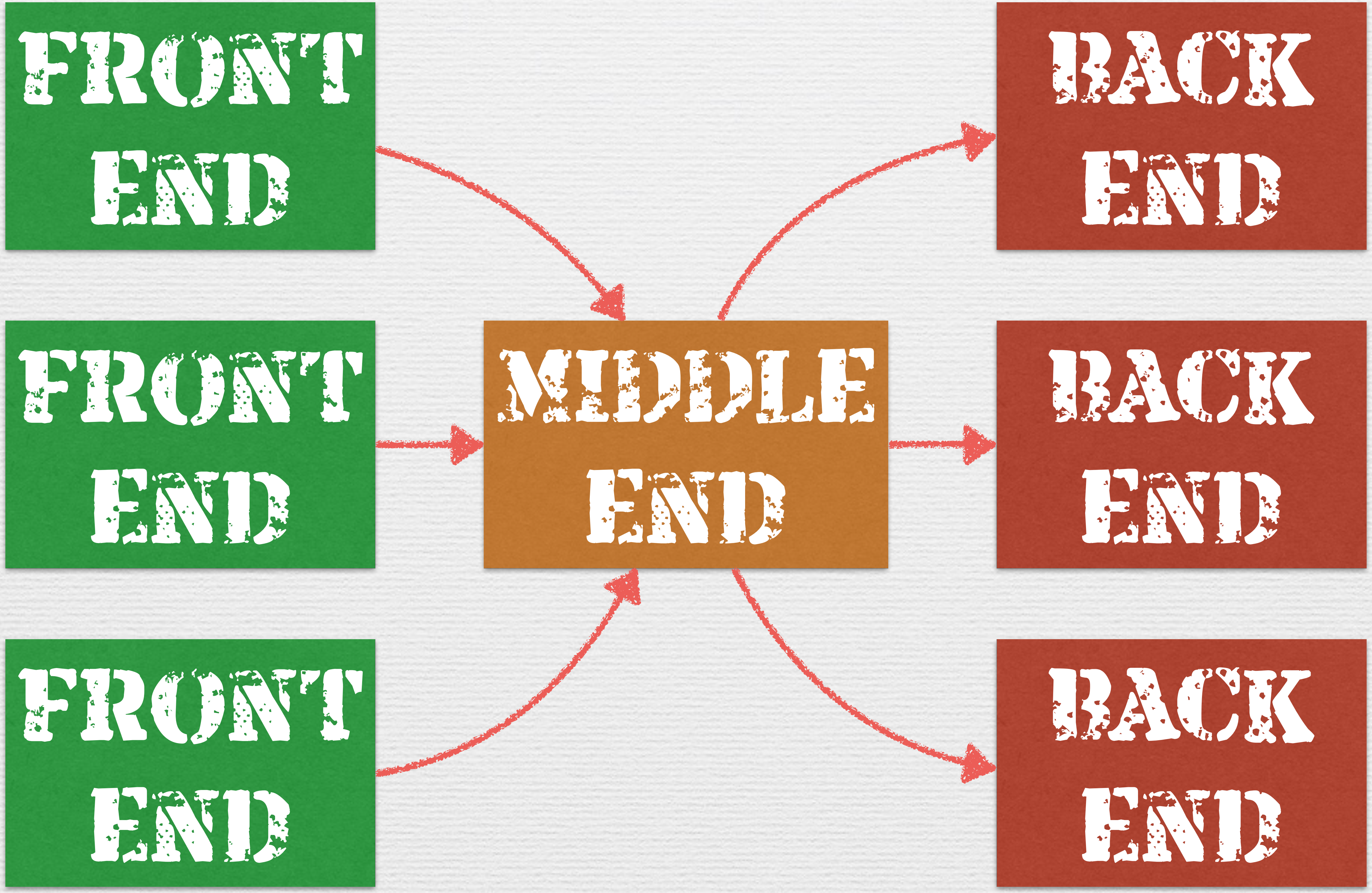
FRONT END

FRONT END

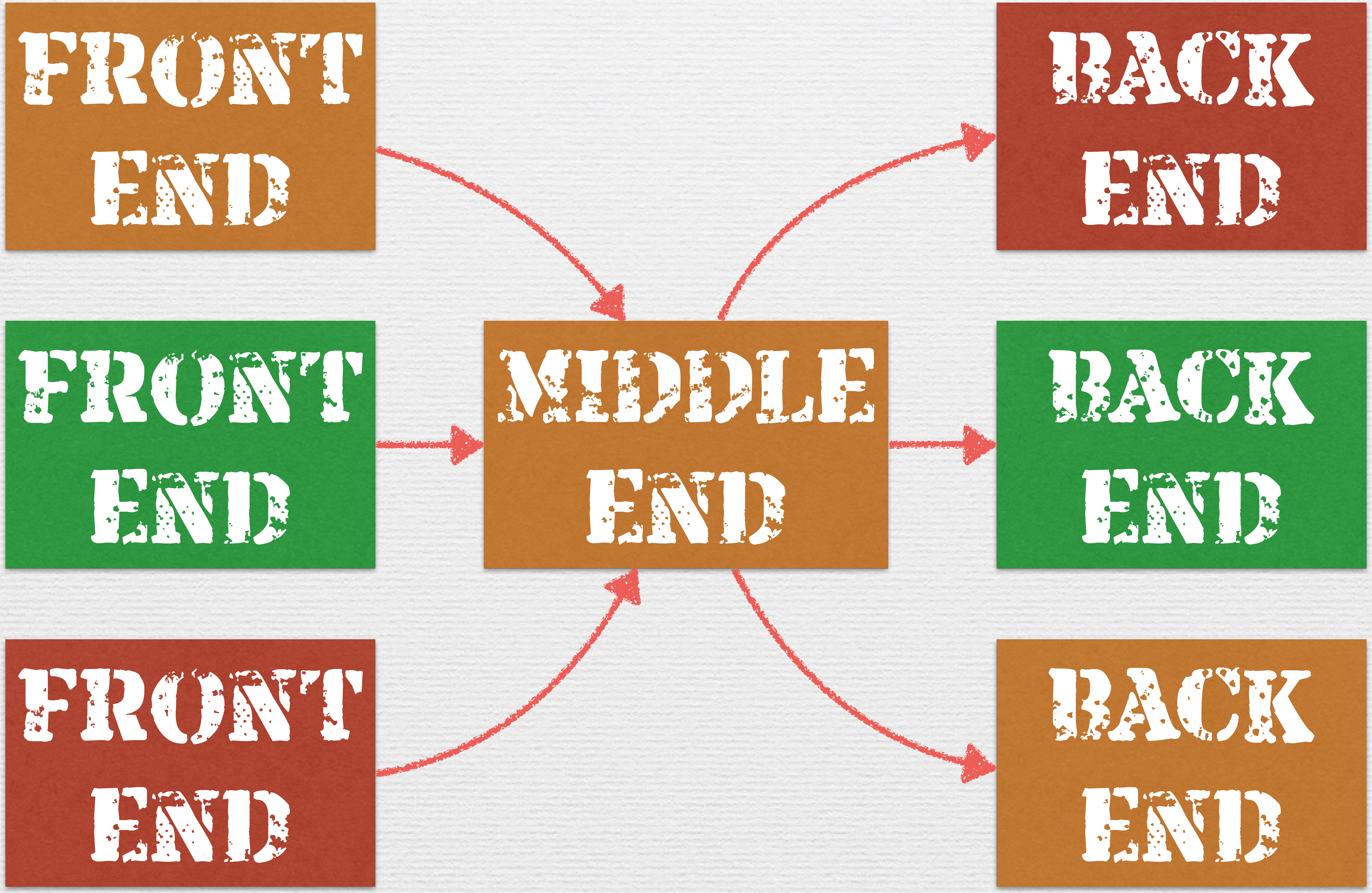
MIDDLE END

BACK END

Multi-target compiler



Grammarware



FRONT END

FRONT END

FRONT END

MIDDLE END

BACK END

BACK END

BACK END

Compilers transform between languages

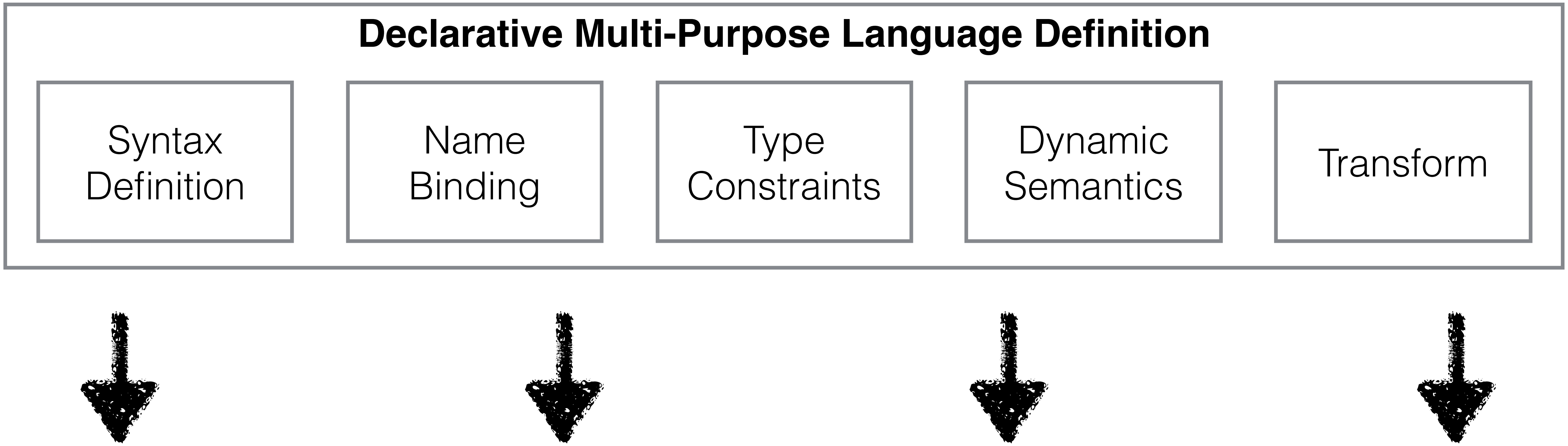
Grammarware commits to grammatical structure

Kinds of grammarware

- Parser
- Compiler
- Interpreter
- Prettyprinter
- Scanner
- Browser
- Static checker
- Struct.editor
- IDE
- DSL
- Preprocessor
- Postprocessor
- Validator
- Model checker
- Refactorer
- Code slicer
- API
- XMLware
- Modelware
- Lang.
- RE
- Benchmark
- Recommender
- Renovation tool

Klint, Lämmel, Verhoef, Toward an Engineering Discipline for Grammarware

Languages vs. grammars



Declarative Multi-Purpose Language Definition

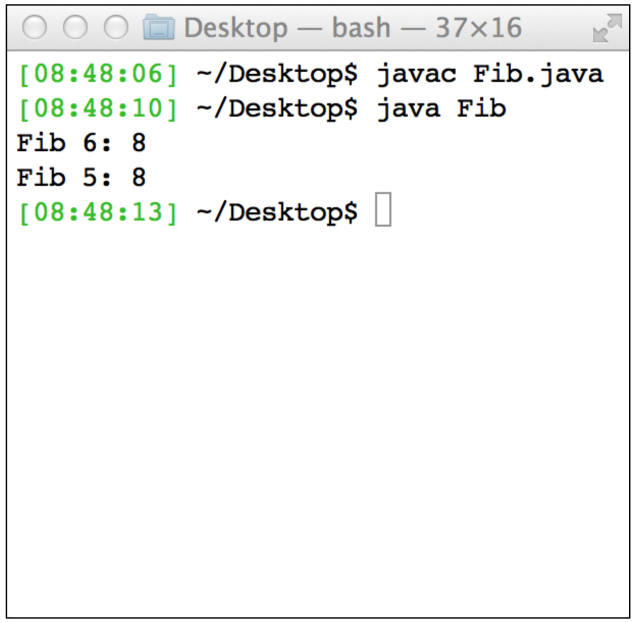
Syntax Definition

Name Binding

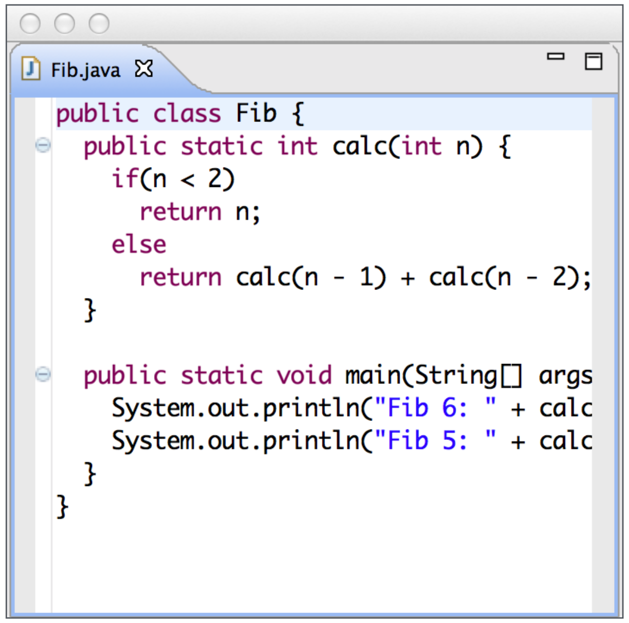
Type Constraints

Dynamic Semantics

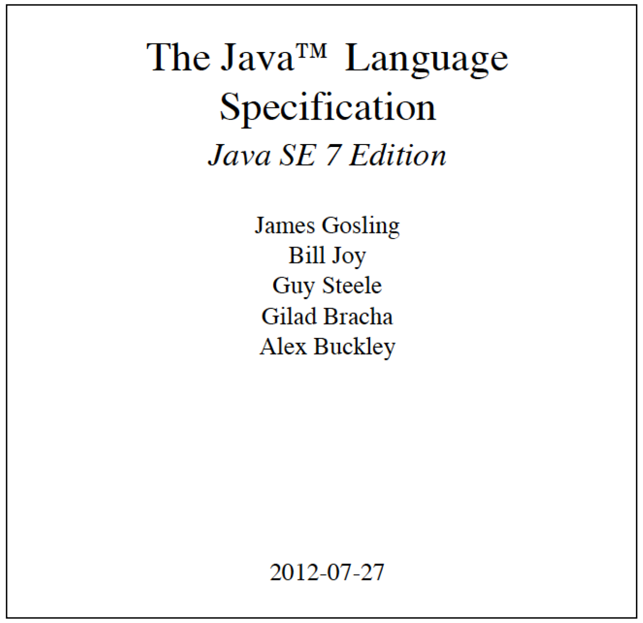
Transform



```
[08:48:06] ~/Desktop$ javac Fib.java
[08:48:10] ~/Desktop$ java Fib
Fib 6: 8
Fib 5: 8
[08:48:13] ~/Desktop$ 
```



```
public class Fib {
  public static int calc(int n) {
    if(n < 2)
      return n;
    else
      return calc(n - 1) + calc(n - 2);
  }

  public static void main(String[] args
    System.out.println("Fib 6: " + calc
    System.out.println("Fib 5: " + calc
  }
}
```

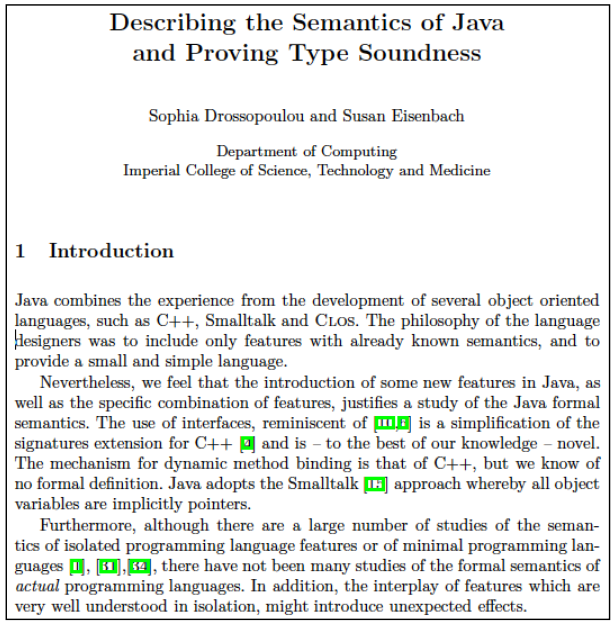


The Java™ Language Specification

Java SE 7 Edition

James Gosling
Bill Joy
Guy Steele
Gilad Bracha
Alex Buckley

2012-07-27



Describing the Semantics of Java and Proving Type Soundness

Sophia Drossopoulou and Susan Eisenbach

Department of Computing
Imperial College of Science, Technology and Medicine

1 Introduction

Java combines the experience from the development of several object oriented languages, such as C++, Smalltalk and CLOS. The philosophy of the language designers was to include only features with already known semantics, and to provide a small and simple language.

Nevertheless, we feel that the introduction of some new features in Java, as well as the specific combination of features, justifies a study of the Java formal semantics. The use of interfaces, reminiscent of [11,6] is a simplification of the signatures extension for C++ [4] and is – to the best of our knowledge – novel. The mechanism for dynamic method binding is that of C++, but we know of no formal definition. Java adopts the Smalltalk [15] approach whereby all object variables are implicitly pointers.

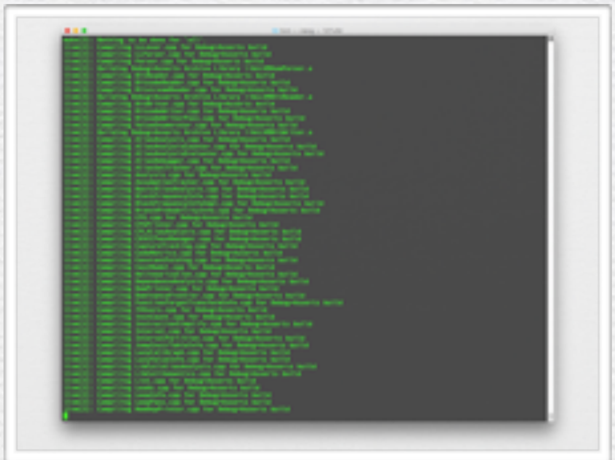
Furthermore, although there are a large number of studies of the semantics of isolated programming language features or of minimal programming languages [1], [31],[34], there have not been many studies of the formal semantics of *actual* programming languages. In addition, the interplay of features which are very well understood in isolation, might introduce unexpected effects.

From Compilers

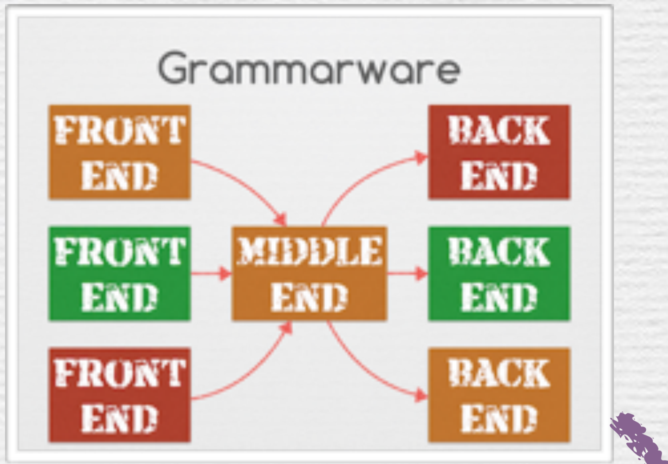
to Grammarware

Dr. Vadim Zaytsev

Introduction



Compilers



Grammarware

Grammarware

Negotiating the result

Transformation

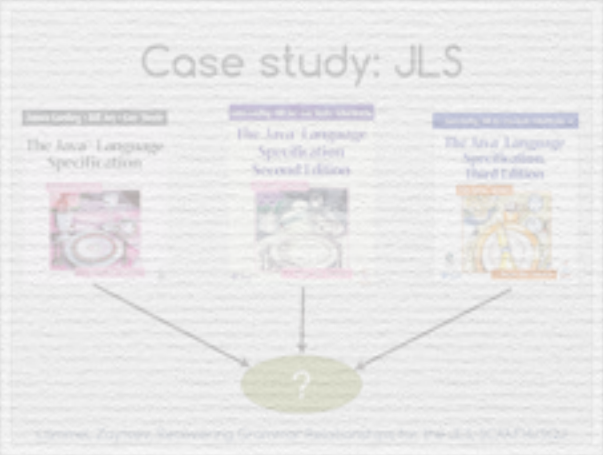
Grammar Zoo

- 974 fetched grammars
- 588 extracted
- 79 connected
- 9 adapted

+metadata

http://slebok.github.io/zoo

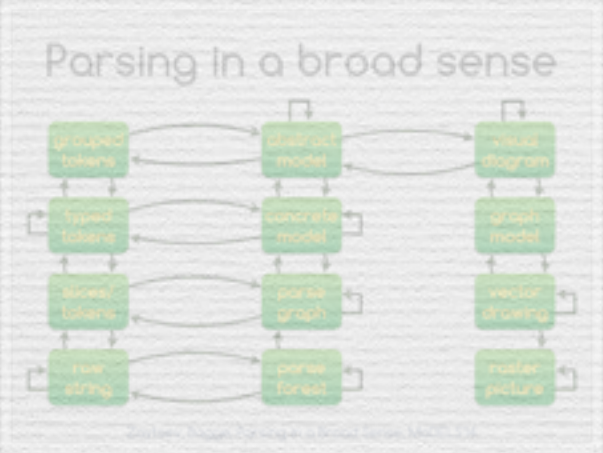
Maturity



Case study: JLS

The Java Language Specification
The Java Language Specification Second Edition
The Java Language Specification Third Edition
?

Consistency



Parsing in a broad sense

Understanding

Reality vs. specification

- Obtain a grammar
  - Construct as an oracle
  - Extract from the tool
  - Infer from the codebase
- Converge/diff.test

Testing

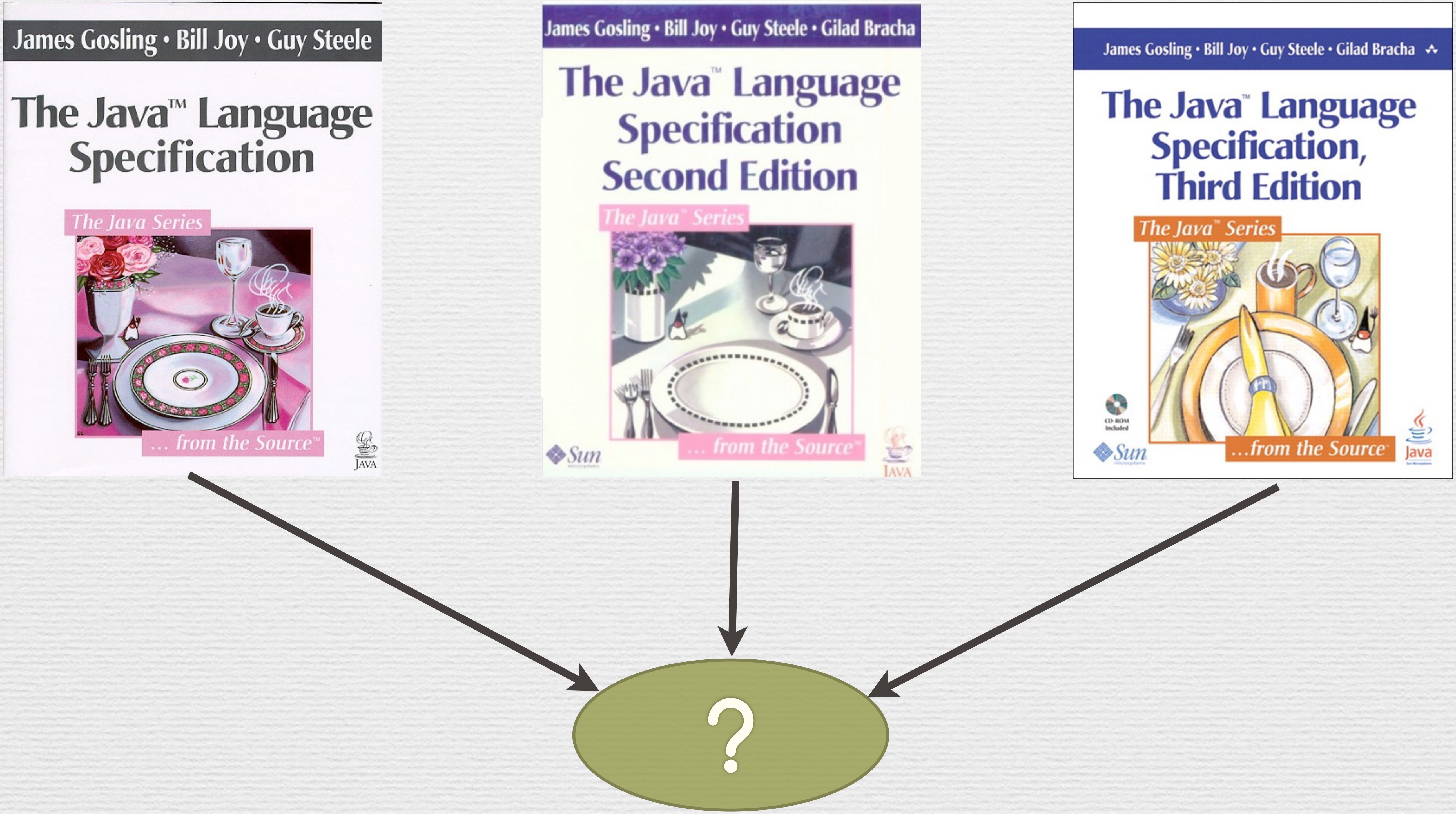
Conclusion

- Grammarware is more than just compilers
- Borrow methods from other domains
- Automate whenever possible
- Compare & combine
- Advance taxonomies & formalisms
- Bet on robust/tolerant methods

Conclusion

What is good grammarware?

Case study: JLS


James Gosling • Bill Joy • Guy Steele
The Java™ Language Specification
The Java Series
... from the Source™
JAVA
James Gosling • Bill Joy • Guy Steele • Gilad Bracha
The Java™ Language Specification Second Edition
The Java™ Series
... from the Source™
Sun
JAVA
James Gosling • Bill Joy • Guy Steele • Gilad Bracha
The Java™ Language Specification, Third Edition
The Java™ Series
...from the Source™
CD-ROM Included
Sun
Java
?

Lämmel, Zaytsev, Recovering Grammar Relationships for the

What is good grammarware?

What is good software?

What is good software?

- functional
- reliable
- usable
- efficient
- maintainable
- portable

ISO/IEC 9126.

What is good grammarware?

- functional: commits to the language
- reliable: tolerant to errors
- usable: the language is learnable
- efficient: fast (live?) and responsive
- maintainable: can be tested and evolved
- portable

Certified
Language
Processor

Certified Language Engineer

Capability Maturity Model

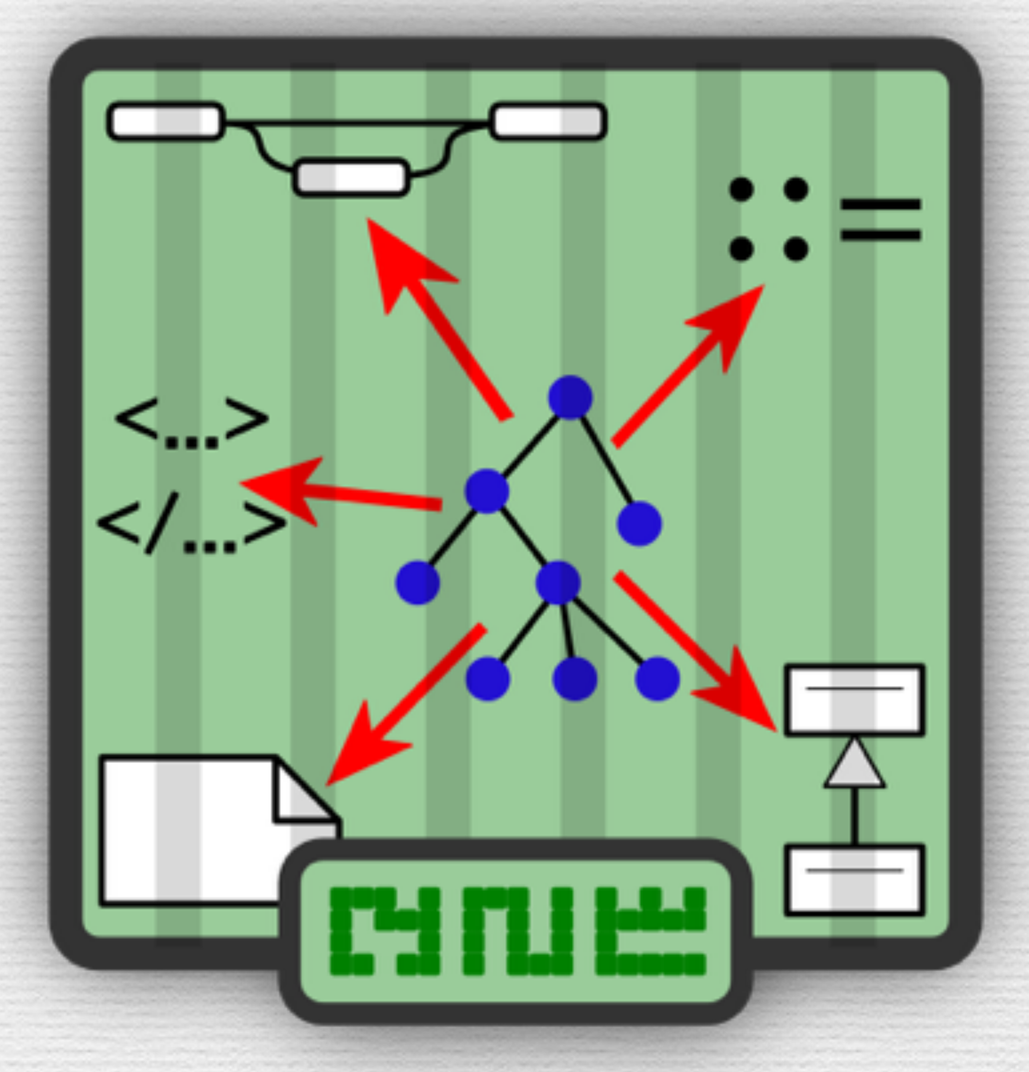
- Level 1 — Chaotic
- Level 2 — Repeatable
- Level 3 — Defined
- Level 4
- Level 5 — Optimising

Paulk, Weber, Curtis, Chrissis, Capability Maturity Model for Software

Grammar Zoo

- 974 fetched grammars
- 588 extracted
- 79 connected
- 9 adapted

+metadata



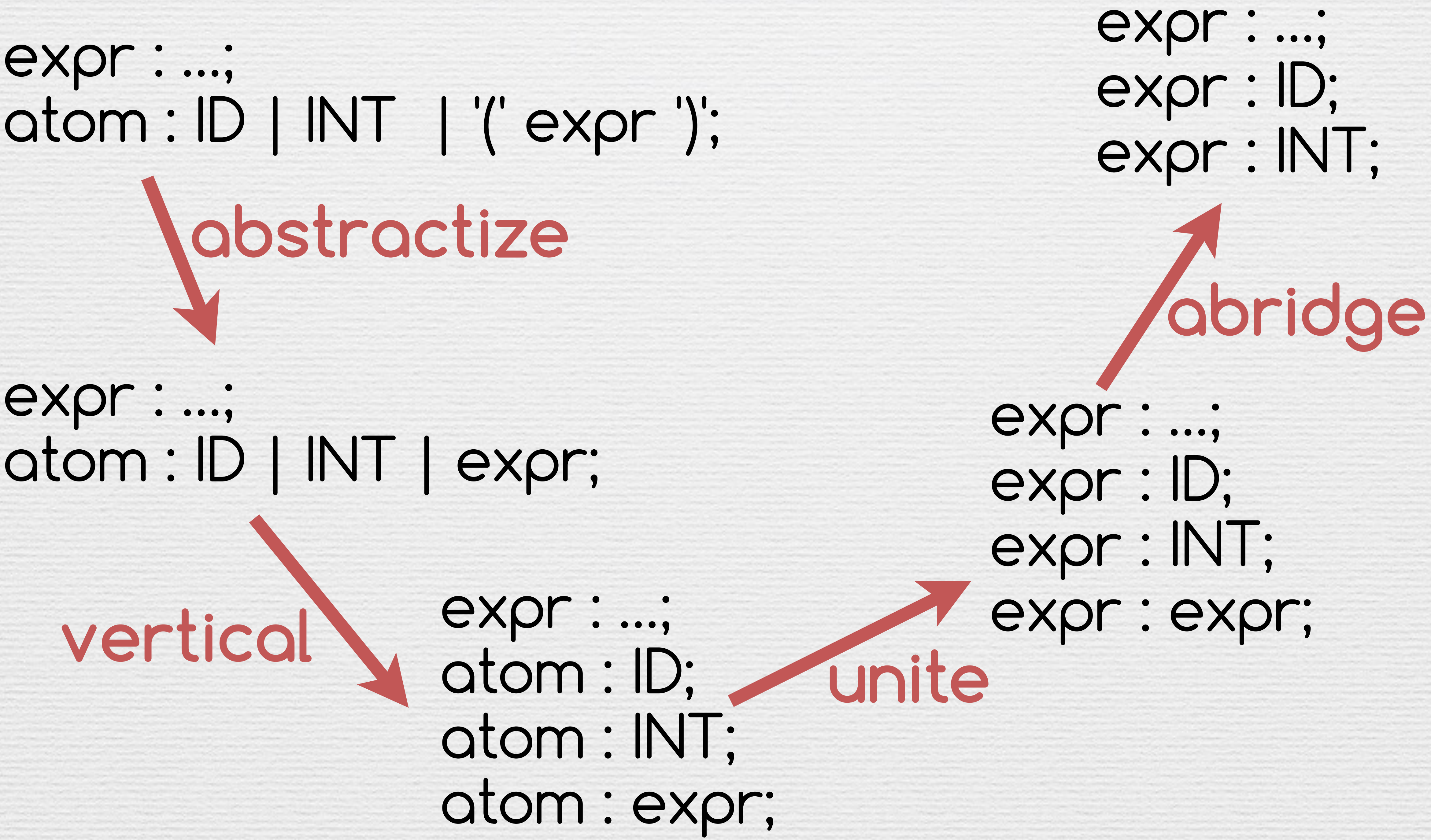
http://slebok.github.io/zoo

Zaytsev, Grammar Maturity Model
Zaytsev, Grammar Zoo: A Corpus of Experimental Grammarware

Improving quality

- Manual inline editing
- Refactorings
- Programmed transformations
  - +Differs
- Grammar mutations
- Inference of transformation/mutation steps

How to transform



```
expr : ...;
atom : ID | INT  | '(' expr ')';
```

abstractize

```
expr : ...;
atom : ID | INT | expr;
```

vertical

```
expr : ...;
atom : ID;
atom : INT;
atom : expr;
```

unite

```
expr : ...;
expr : ID;
expr : INT;
expr : expr;
```

abridge

```
expr : ...;
expr : ID;
expr : INT;
```

Lämmel, Zaytsev, An Introduction to Grammar Convergence, IFM’

How to mutate

- Grammar has no starting symbol?
  - Reroot2top
- Need abstract syntax from concrete syntax?
  - RetireTs
- Grammar productions written in an
  - DeyaccifyAll
- Change naming convention?
  - RenameAllNLower2Camel

How to be guided

- Equality & algebraic equivalence
- Prodsig-equivalence
  - signatures based on nonterminal patterns
  - tolerant to permutations
  - weak equivalence tolerant to iteration kinds
- Abstract Normal Form
  - no terminals, labels, markers
  - consistent disjunctive style

Zaytsev, Guided Grammar Convergence,

How to be guided

$$
\begin{array}{llclcl}
 & p_{master} & = & \mathbf{p}(\varepsilon, & expr, & expr \cdot operator \cdot expr) \\
\bigstar & p_{antlr} & = & \mathbf{p}(\varepsilon, & binary, & \mathbf{s}(l, atom) \cdot *(\mathbf{s}(o, ops) \cdot \mathbf{s}(r, atom))) \\
\bigstar & p_{dcg} & = & \mathbf{p}(binary, & expr, & atom \cdot *(ops \cdot atom)) \\
 & p_{emf} & = & \mathbf{p}(\varepsilon, & Binary, & \mathbf{s}(ops, Ops) \cdot \mathbf{s}(left, Expr) \cdot \mathbf{s}(right, Expr)) \\
 & p_{jaxb} & = & \mathbf{p}(\varepsilon, & Binary, & \mathbf{s}(Ops, Ops) \cdot \mathbf{s}(Left, Expr) \cdot \mathbf{s}(Right, Expr)) \\
 & p_{om} & = & \mathbf{p}(\varepsilon, & Binary, & \mathbf{s}(ops, Ops) \cdot \mathbf{s}(left, Expr) \cdot \mathbf{s}(right, Expr)) \\
\bigstar & p_{python} & = & \mathbf{p}(\varepsilon, & binary, & atom \cdot *(operators \cdot atom)) \\
 & p_{adt} & = & \mathbf{p}(\varepsilon, & FLExpr, & \mathbf{s}(binary, \mathbf{s}(e1, FLExpr) \cdot \mathbf{s}(op, FLOp) \cdot \mathbf{s}(e2, FLExpr))) \\
 & p_{rascal} & = & \mathbf{p}(binary, & Expr, & \mathbf{s}(lexpr, Expr) \cdot \mathbf{s}(op, Ops) \cdot \mathbf{s}(rexpr, Expr)) \\
 & p_{sdf} & = & \mathbf{p}(binary, & Expr, & Expr \cdot Ops \cdot Expr) \\
 & p_{txl} & = & \mathbf{p}(\varepsilon, & expression, & expression \cdot op \cdot expression) \\
 & p_{xsd} & = & \mathbf{p}(\varepsilon, & Binary, & \mathbf{s}(ops, Ops) \cdot \mathbf{s}(left, Expr) \cdot \mathbf{s}(right, Expr))
\end{array}
$$

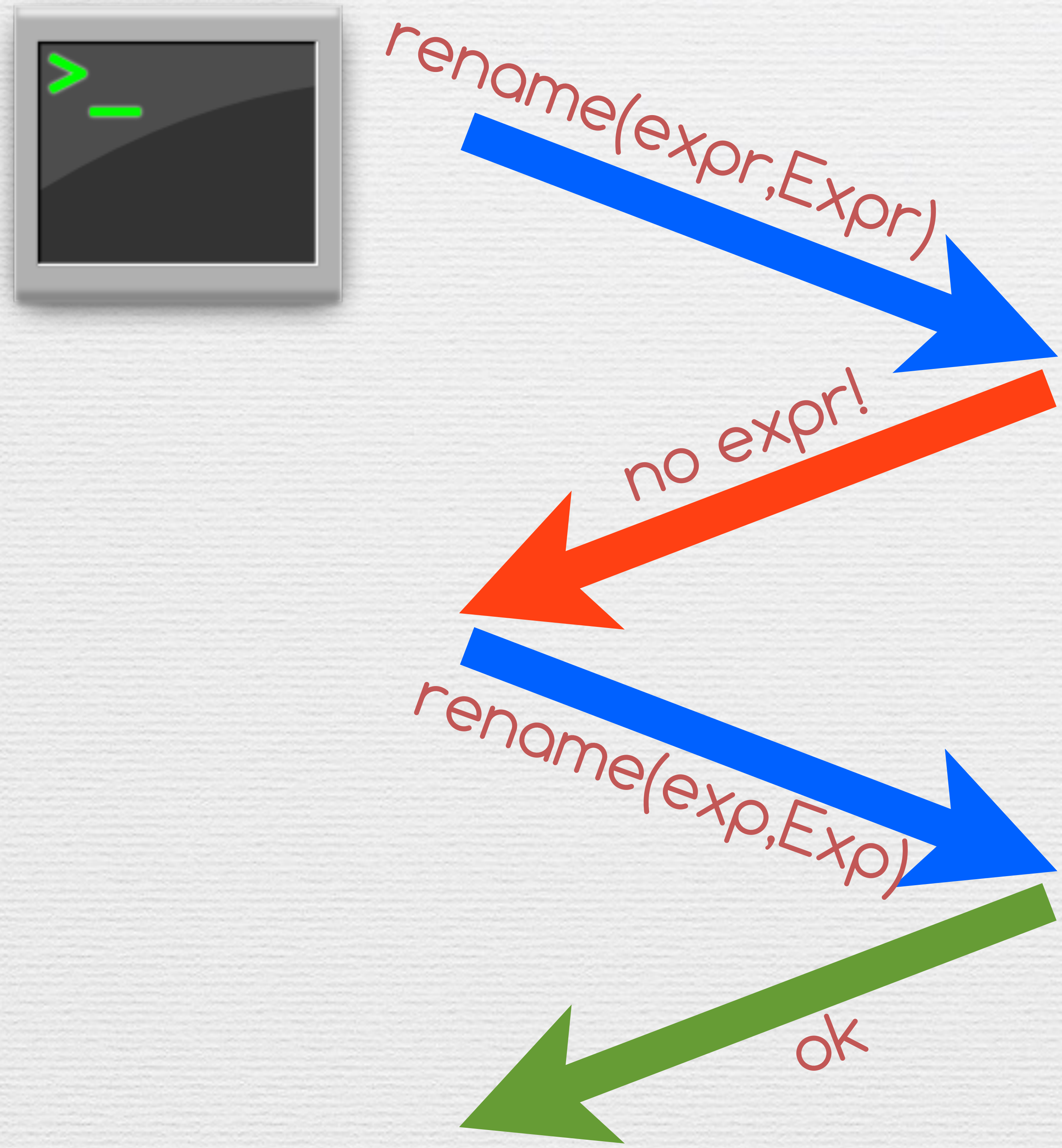
Zaytsev, Guided Grammar Convergence,

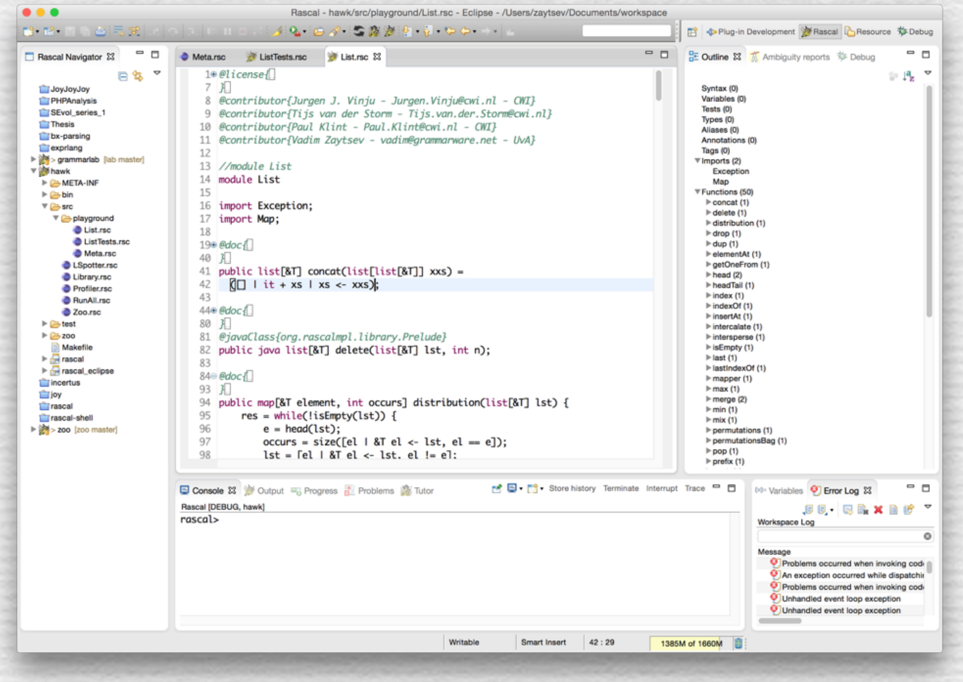
What we want in general

- Maintenance assistants
  - infer whatever possible
  - provide advice on the rest
- Not necessarily “request => result or fail”
  - pending
  - negotiated

Zaytsev, Pending Evolution of Grammars
Zaytsev, Negotiated Grammar Evolution

Negotiating the result





```
@license{
}
@contributor{Jurgen J. Vinju - Jurgen.Vinju@cwi.nl - CWI}
@contributor{Tijs van der Storm - Tijs.van.der.Storm@cwi.nl}
@contributor{Paul Klint - Paul.Klint@cwi.nl - CWI}
@contributor{Vadim Zaytsev - vadim@grammarware.net - UvA}

//module List
module List

import Exception;
import Map;

@doc{
}
public list[&T] concat(list[list[&T]] xxs) =
  ([] | it + xs | xs <- xxs);

@doc{
}
@javaClass{org.rascalmpl.library.Prelude}
public java list[&T] delete(list[&T] lst, int n);

@doc{
}
public map[&T element, int occurs] distribution(list[&T] lst) {
    res = while(!isEmpty(lst)) {
        e = head(lst);
        occurs = size([el | &T el <- lst, el == e]);
        lst = [el | &T el <- lst, el != e];
```

```
Rascal [DEBUG, hawk]
rascal>
```

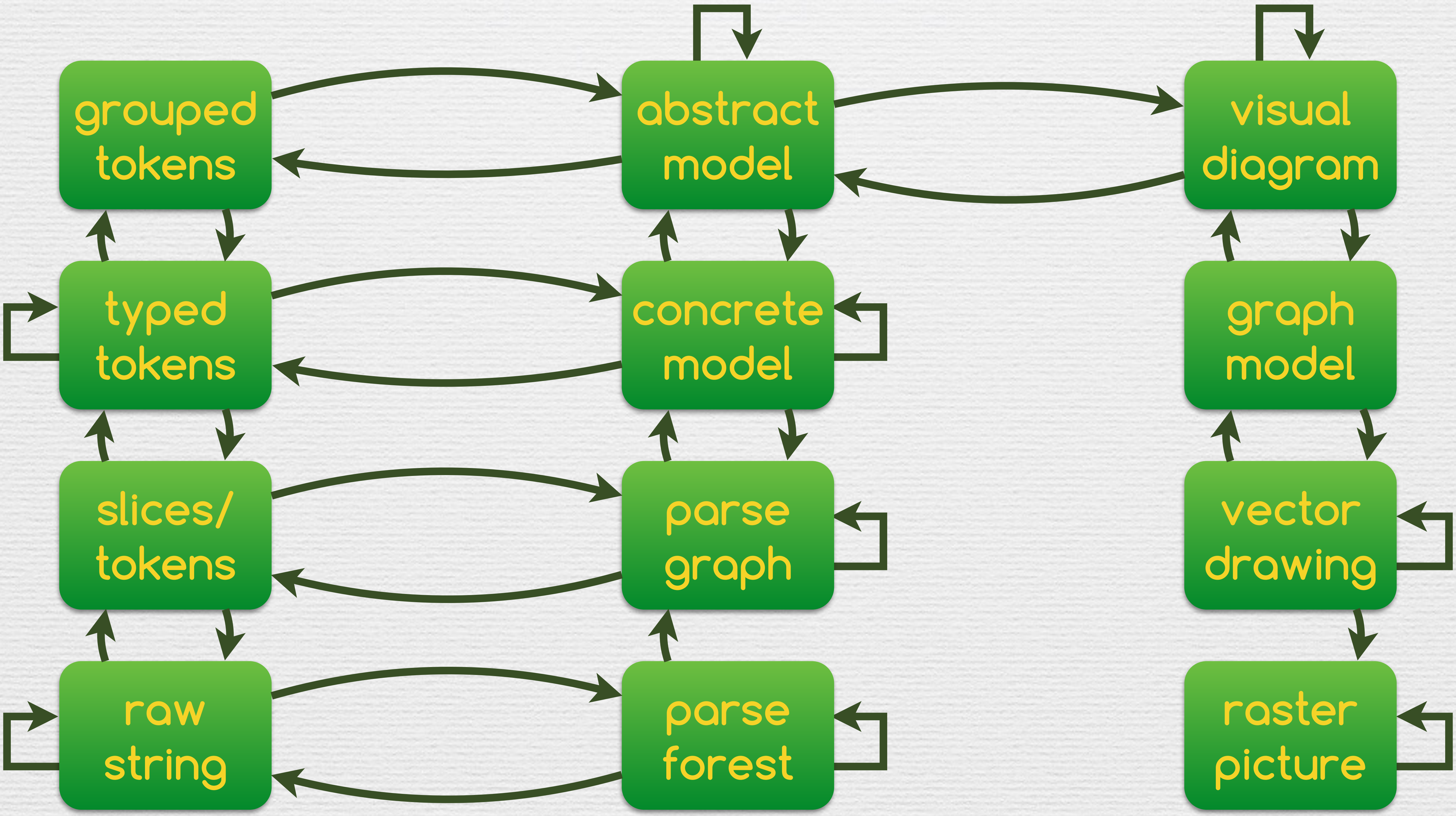
```
Workspace Log
Message
Problems occurred when invoking cod
An exception occurred while dispatchi
Problems occurred when invoking cod
Unhandled event loop exception
Unhandled event loop exception
```

Zaytsev, Negotiated Grammar Evolution

Key points

- For grammarware, we need
    - consistency
    - a clear quality model
    - improvement processes
    - automation
- Also,
    - understanding user scenarios

Parsing in a broad sense



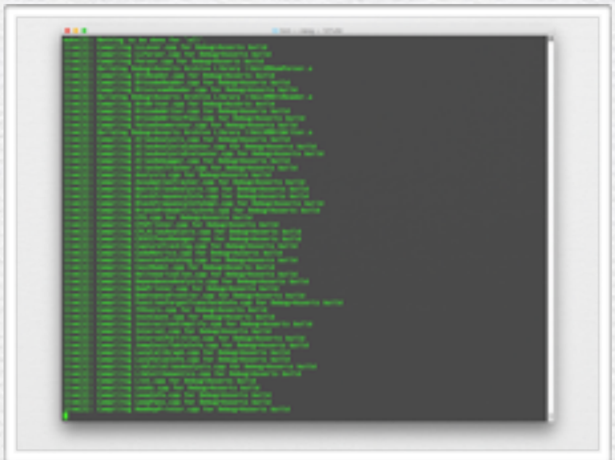
Zaytsev, Bagge, Parsing in a Broad Sense, MoDELS'

From Compilers

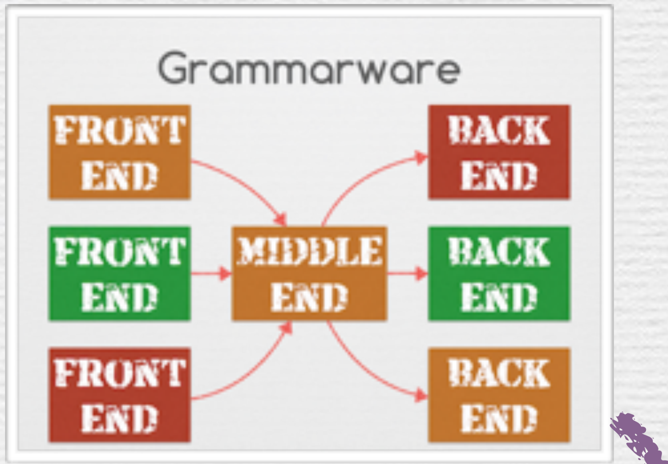
to Grammarware

Dr. Vadim Zaytsev

Introduction

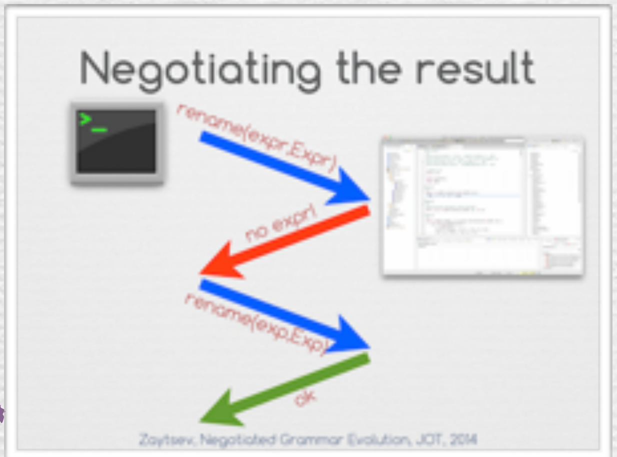


Compilers



Grammarware

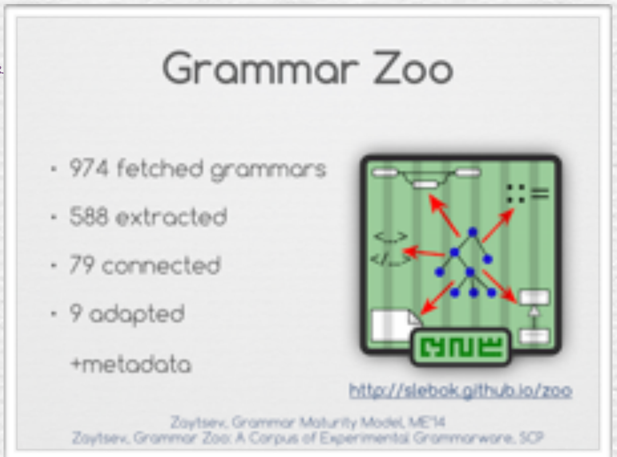
Grammarware



Negotiating the result

Zaytsev, Negotiated Grammar Evolution, JOT, 2014

Transformation



Grammar Zoo

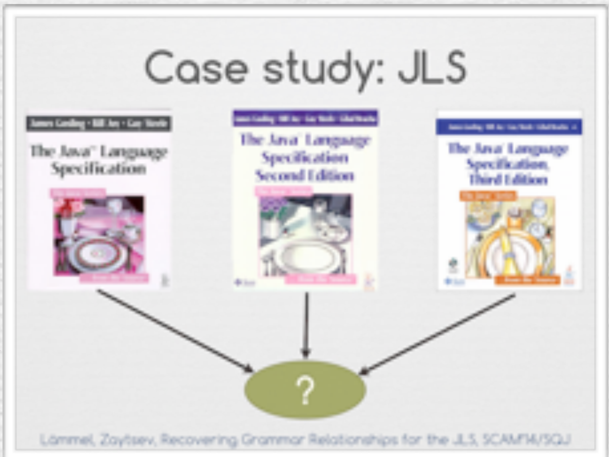
- 974 fetched grammars
- 588 extracted
- 79 connected
- 9 adapted

+metadata

http://slebok.github.io/zoo

Zaytsev, Grammar Maturity Model, ME'14
Zaytsev, Grammar Zoo: A Corpus of Experimental Grammarware, SCP

Maturity

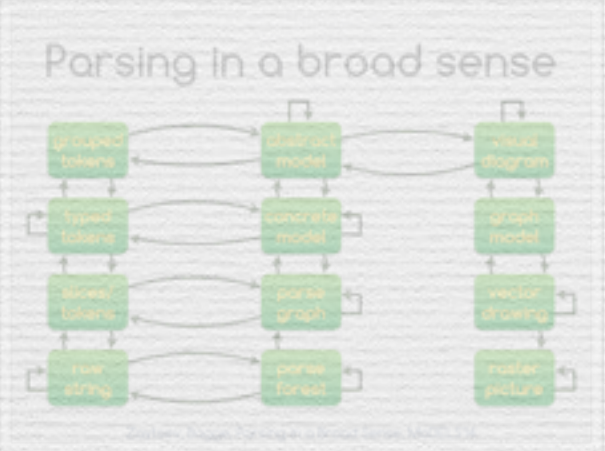


Case study: JLS

The Java Language Specification
The Java Language Specification Second Edition
The Java Language Specification, Third Edition
?

Lämmel, Zaytsev, Recovering Grammar Relationships for the JLS, SCAM'11/SQJ

Consistency



Parsing in a broad sense

[illegible]

Understanding

Reality vs. specification

- Obtain a grammar
  - Construct as an oracle
  - Extract from the tool
  - Infer from the codebase
- Converge/diff.test

[illegible]

Testing

Conclusion

- Grammarware is more than just compilers
- Borrow methods from other domains
- Automate whenever possible
- Compare & combine
- Advance taxonomies & formalisms
- Bet on robust/tolerant methods

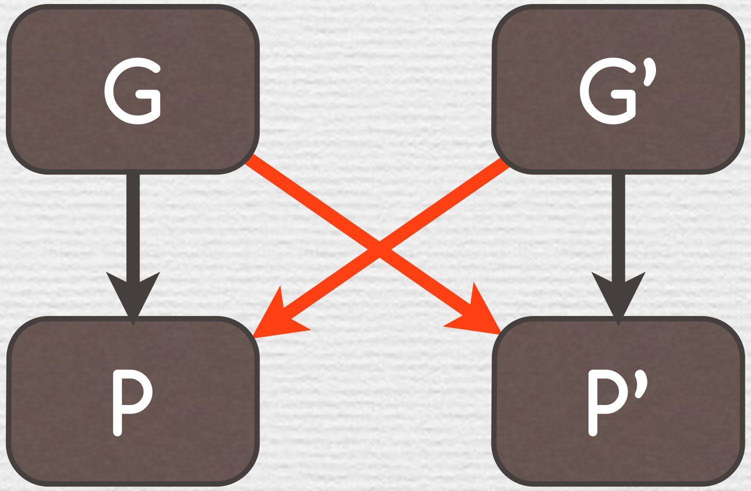
Conclusion

So, grammarware is based on grammars…

…can we test/validate it based on grammars?

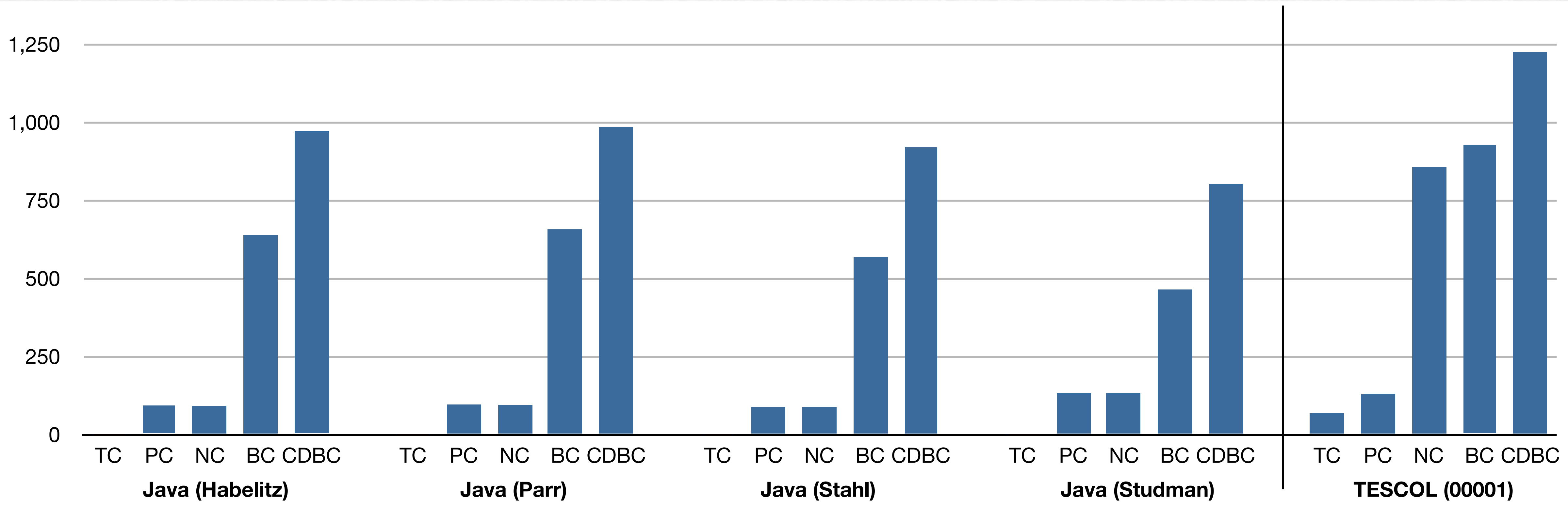
Grammar-based testing

- Purdom's generator
  - builds the shortest conforming term
- Maurer's generator
  - randomly selects alternatives
- Coverage criteria
  - TC, NC, PC, BC, UC, CDBC
- Negative cases?



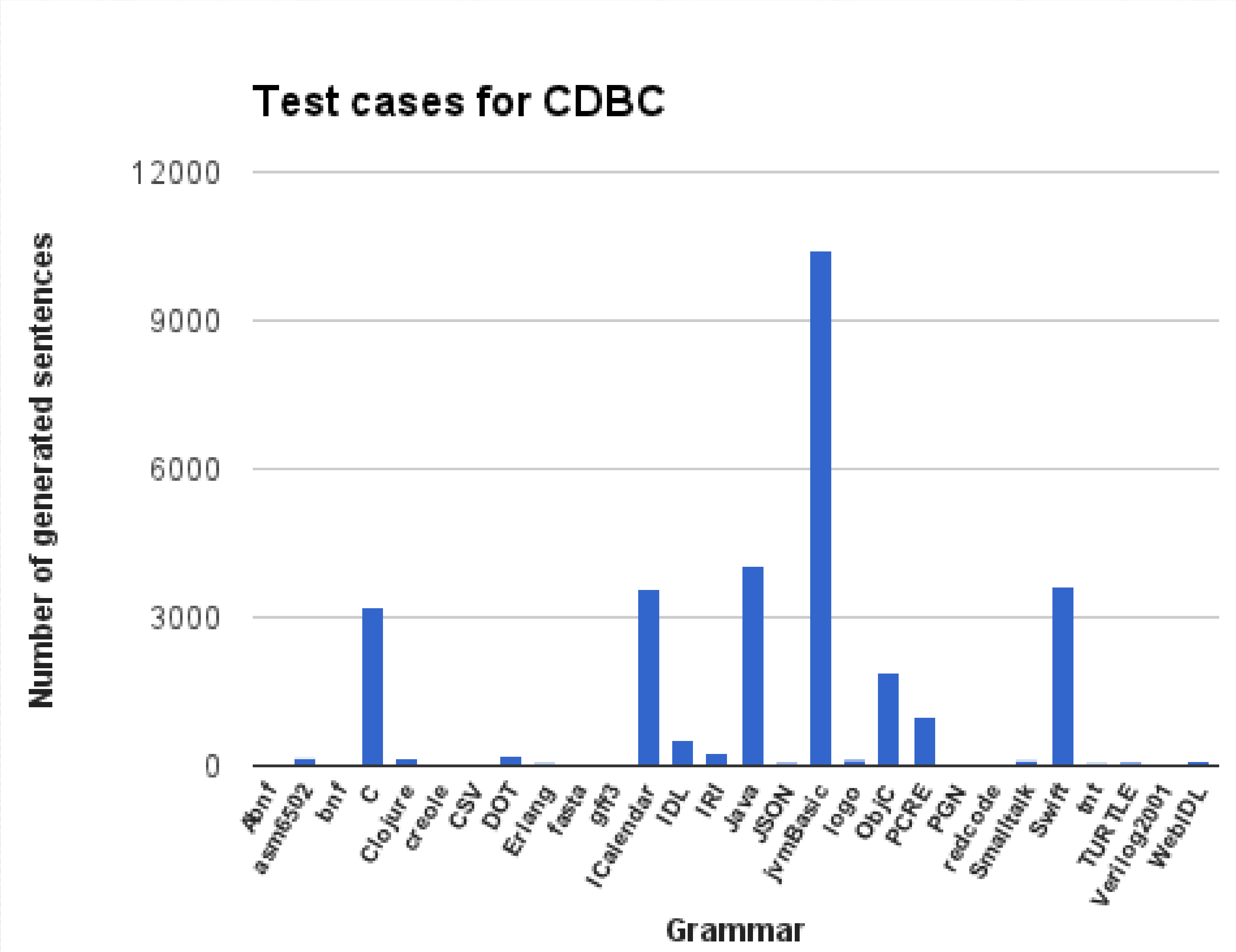
Fischer, Lämmel, Zaytsev, Comparison of CFGs Based on ... Test Data

Combinatorial explosion



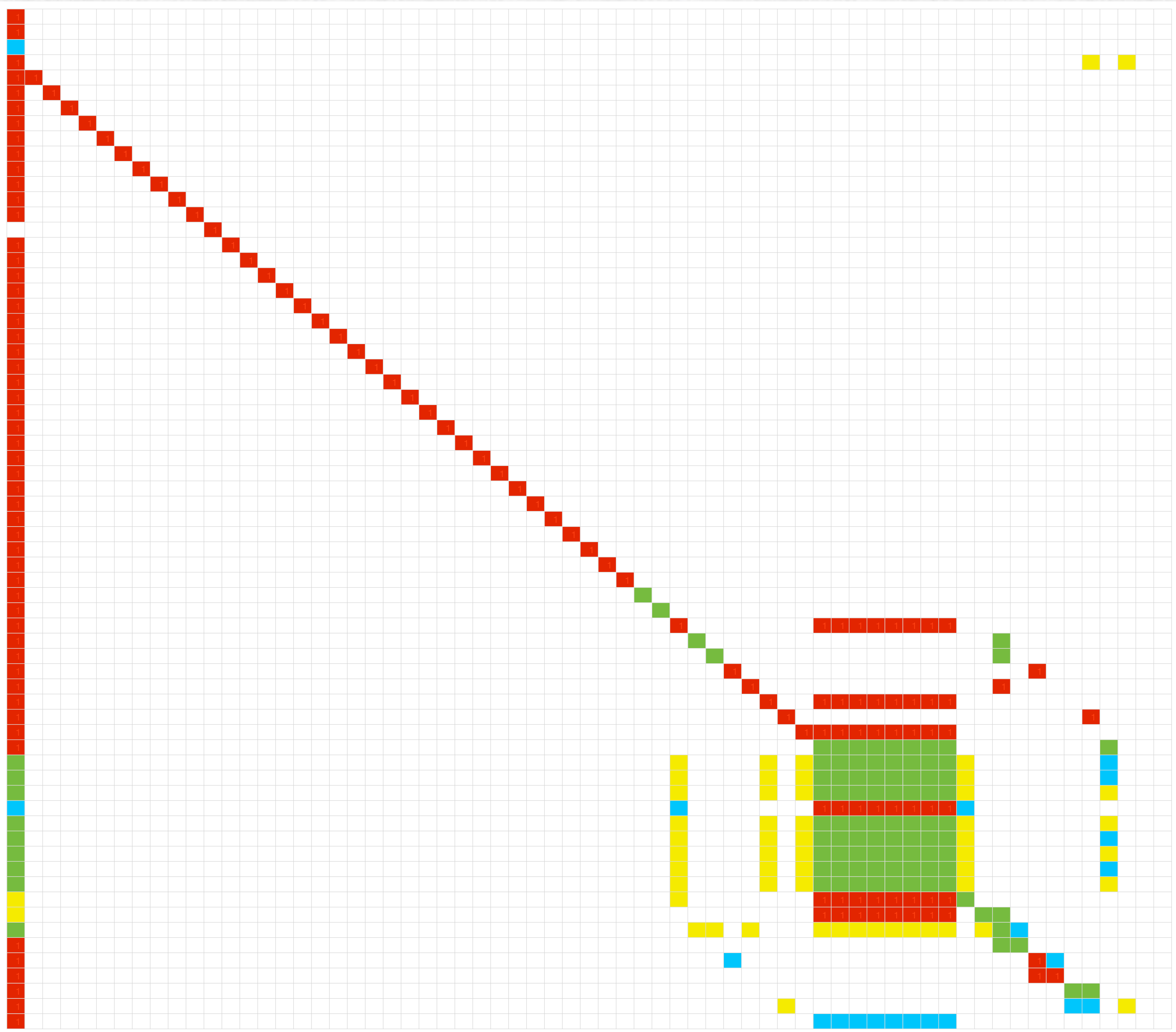
Fischer, Lämmel, Zaytsev, Comparison of CFGs Based on ... Test Data

Combinatorial explosion



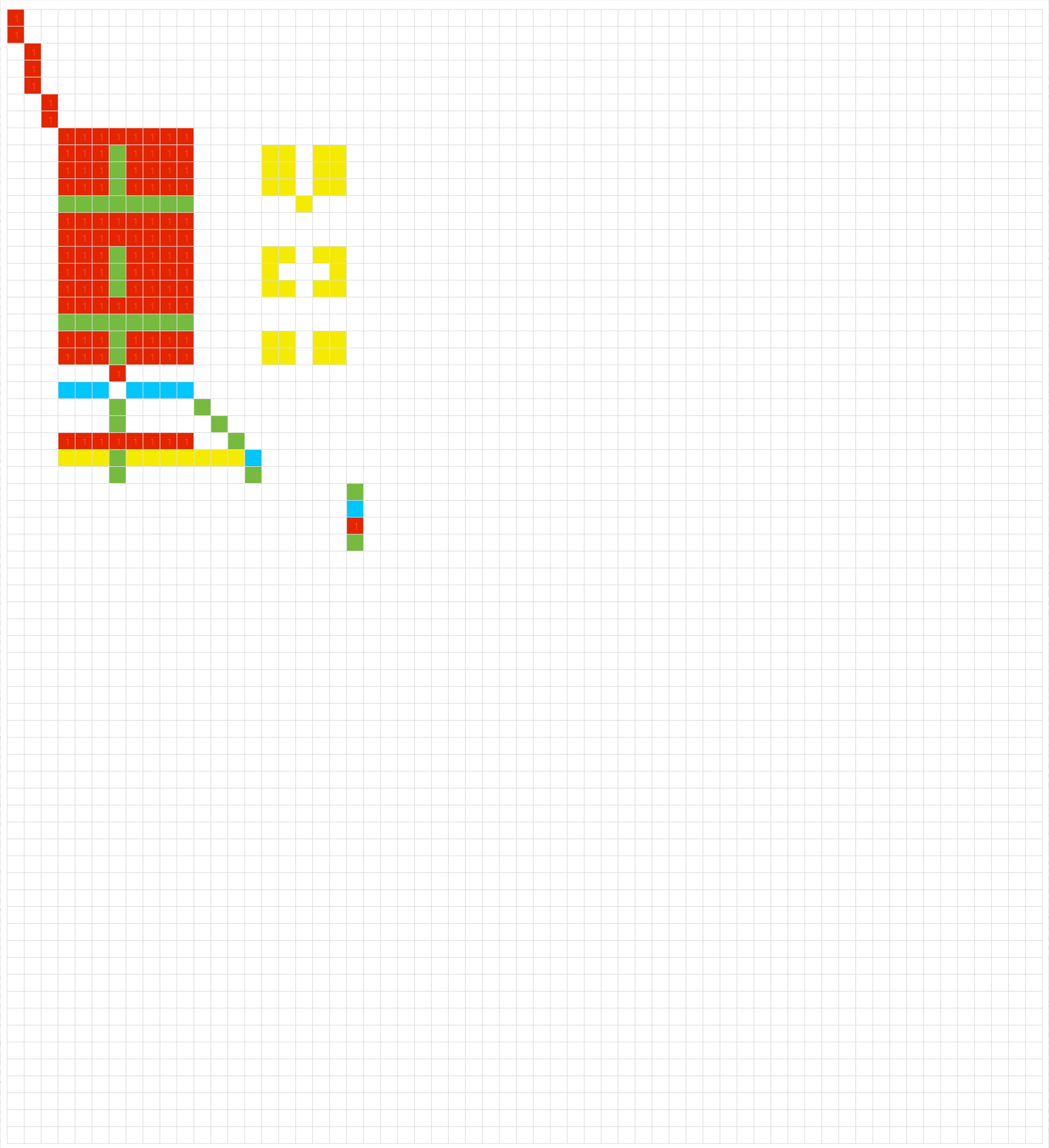
Butrus, Zaytsev, Grammar-based Testing Made Easy with Mutations

Nonterminal matching



Fischer, Lämmel, Zaytsev, Comparison of CFGs Based on ... Test Data

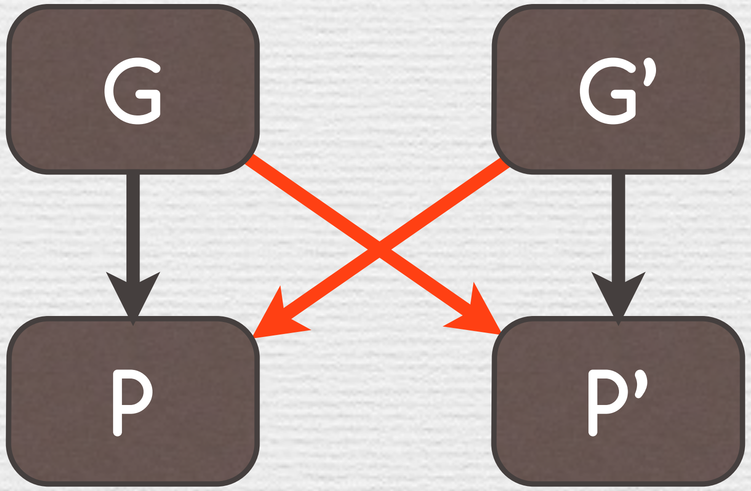
Badly matched:



Fischer, Lämmel, Zaytsev, Comparison of CFGs Based on … Test Data

Differential methods

- Oracles are unnecessary
- Comparing grammars
  - of varying structure, style, etc
  - across TSs
- Investigate disagreements



McKeeman, Differential Testing for Software
Spinellis. Differential Debugging. IEEE Software, 2013

What is a bug?

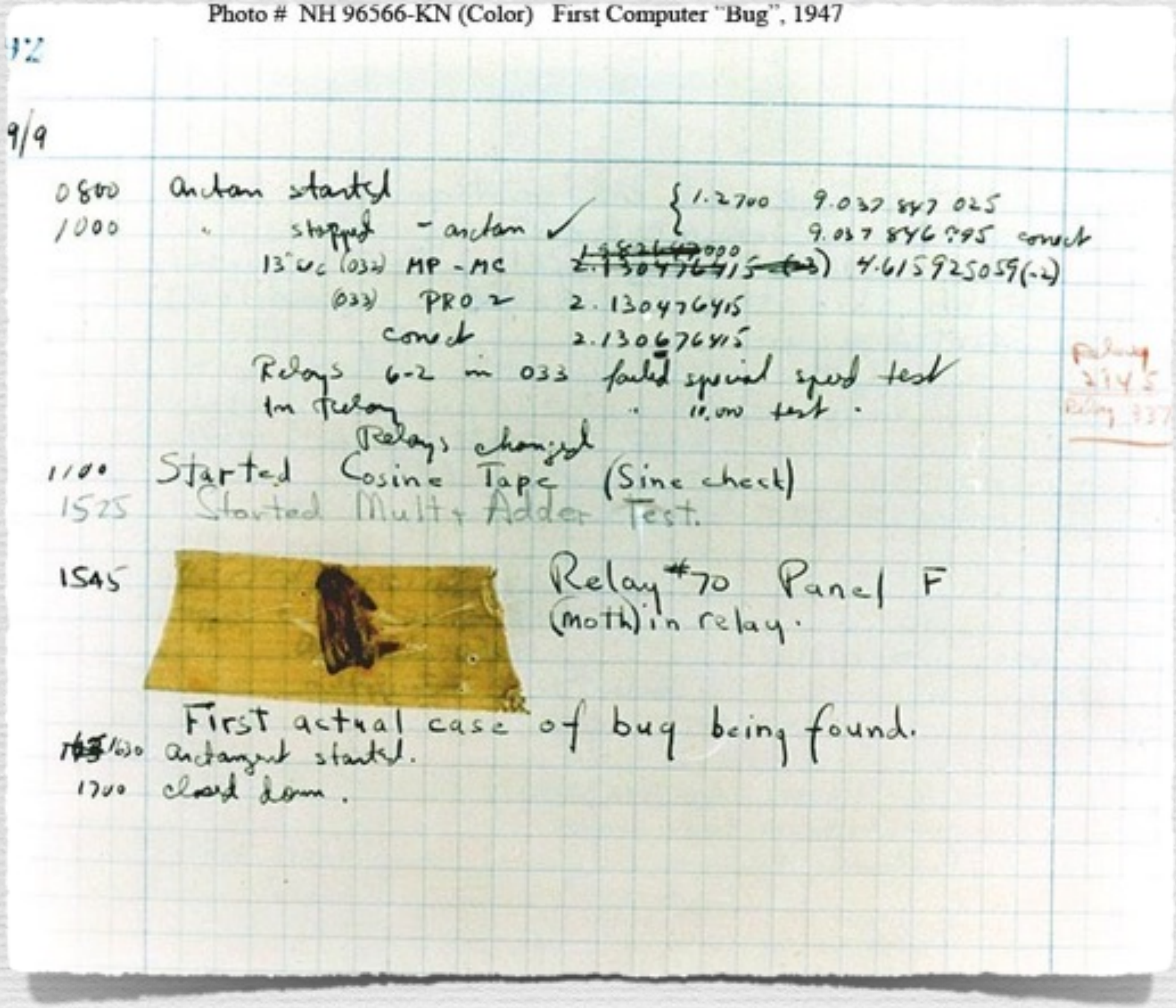


Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/9

0800 Arctan started

1000 " stopped - arctan ✓ {1.2700 9.037 847 025

9.037 846 995 const

13" uc (033) MP - MC ~~1.982147000~~ ~~2.130476415~~ ~~(-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

const 2.130676415

Relays 6-2 in 033 failed special speed test

in relay " 10,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

1630 Antangent started.

1700 closed down.

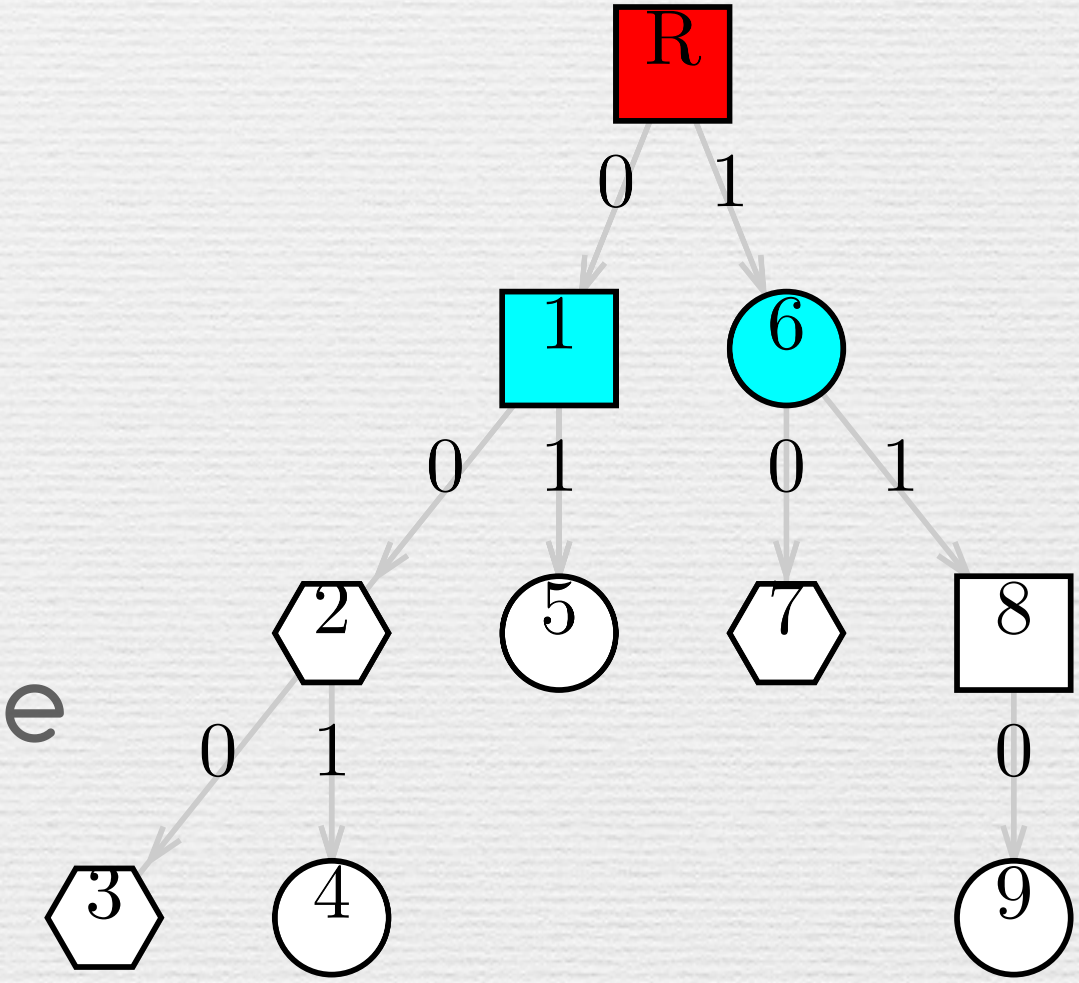
Relay
2145
Relay 337

The First Computer Bug
Photo #____________

- Grammarware processes languages
- A bug is a program
- Inspect programs to deal with bugs

Reality vs. specification

- Obtain a grammar
  - Construct as an oracle
  - Extract from the tool
  - Infer from the codebase
- Converge/diff.test



Stevenson, Cordy, A Survey of Grammatical Inference in Software Engineering

Roș

Antipatterns

- Some ways lead to bugs faster
- Detect them => predict defects
- Smells
  - left/right recursion
  - ambiguous x*?

Taba, Khomh, Zou, Hassan, Nagappan, Predicting Bugs Using Antipatterns, ICSM 2013
Sajnani, Saini, Lopes, A Comparative Study of Bug Patterns in Java, SCAM 2014
Trubiani, Di Marco, Cortellessa, Mani, Petriu, Exploring Synergies...

Process improvement

- find defects
- fix defects
- learn how to fix defects
- learn to tolerate defects
- learn to avoid

Semiparsing

- 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

Zaytsev, Formal Foundations for Semi-parsing, CSMR-WCRE’

Conclusion

- Grammarware is more than just compilers
- Borrow methods from other domains
- Automate whenever possible
- Compare & combine
- Advance taxonomies & formalisms
- Bet on robust/tolerant methods

Thank you!

Questions?

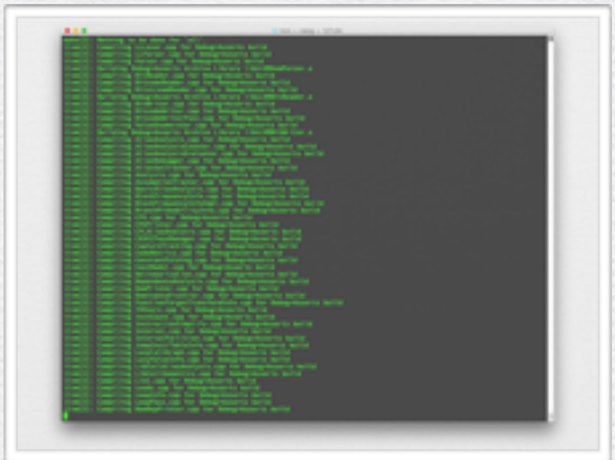
- Sources:
  - Figures used from own papers & talks
    - + Eelco Visser's keynote @ MODULARITY
    - + Tobias Baanders
    - + JLS book covers (Fair Use)
  - Self-made screenshots
  - All photos from public domain
  - Comfortaa: font

From Compilers

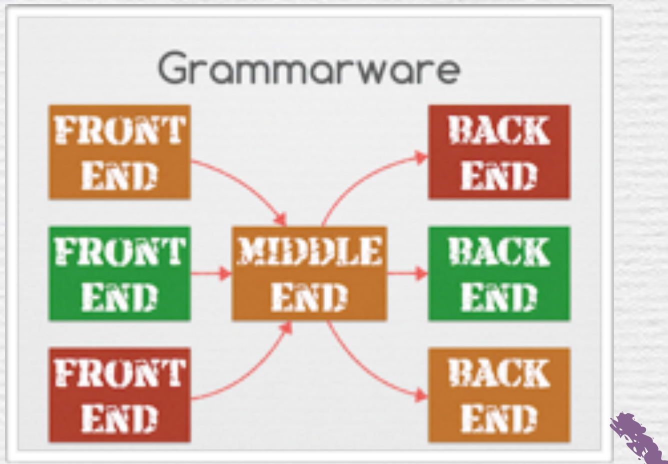
to Grammarware

Dr. Vadim Zaytsev

Introduction



Compilers



Grammarware

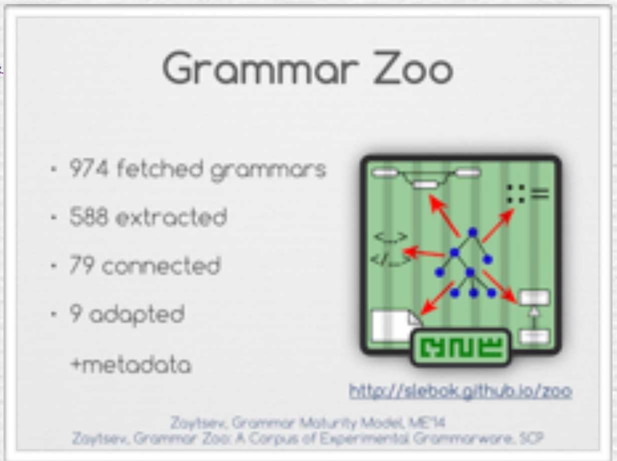
Grammarware

Case study: JLS

The Java Language Specification
The Java Language Specification Second Edition
The Java Language Specification, Third Edition
?

Lämmel, Zaytsev, Recovering Grammar Relationships for the JLS, SCAM'11/SQJ

Consistency



Grammar Zoo

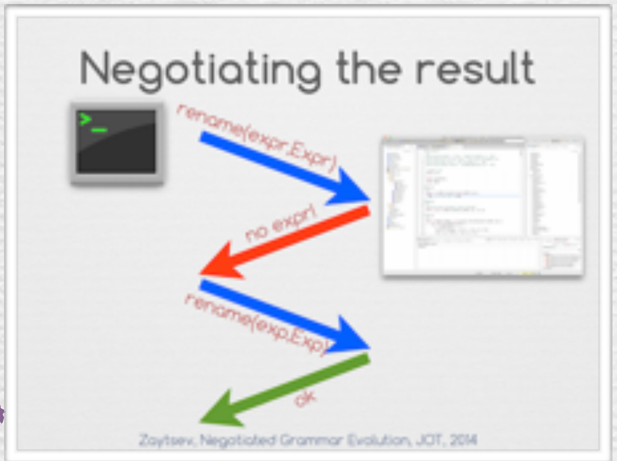
- 974 fetched grammars
- 588 extracted
- 79 connected
- 9 adapted

+metadata

http://slebok.github.io/zoo

Zaytsev, Grammar Maturity Model, ME'14
Zaytsev, Grammar Zoo: A Corpus of Experimental Grammarware, SCP

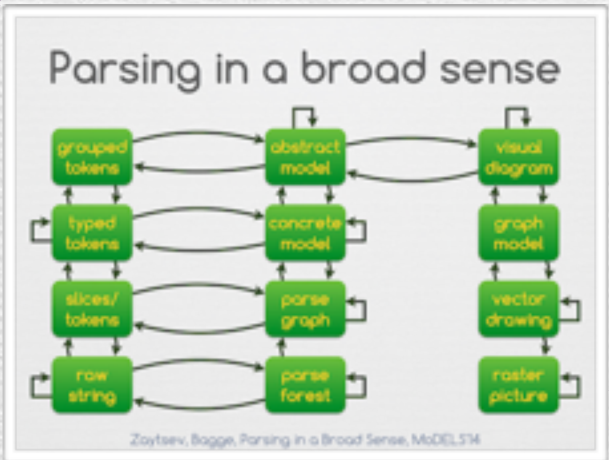
Maturity



Negotiating the result

Zaytsev, Negotiated Grammar Evolution, JOT, 2014

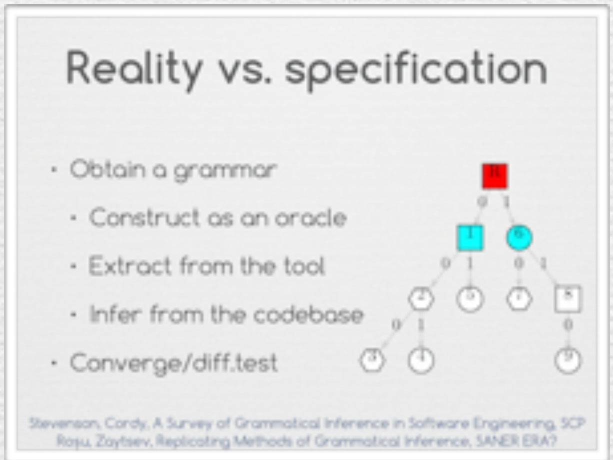
Transformation



Parsing in a broad sense

Zaytsev, Bagge, Parsing in a Broad Sense, MoDELS'14

Understanding



Reality vs. specification

- Obtain a grammar
  - Construct as an oracle
  - Extract from the tool
  - Infer from the codebase
- Converge/diff.test

Stevenson, Cordy, A Survey of Grammatical Inference in Software Engineering, SCP
Rosu, Zaytsev, Replicating Methods of Grammatical Inference, SANER ERA?

Testing

Conclusion

- Grammarware is more than just compilers
- Borrow methods from other domains
- Automate whenever possible
- Compare & combine
- Advance taxonomies & formalisms
- Bet on robust/tolerant methods

Conclusion