

Modern Construction of Industrial Strength Compilers

Современные принципы построения промышленных компиляторов

Vadim Zaytsev

Raincode Labs, Brussels, Belgium

vadim@grammarware.net

Зайцев Вадим Валерьевич

Raincode Labs, Брюссель, Бельгия

vadim@grammarware.net

Compiler construction is one of the oldest and the most mature fields of computer science. Despite existing since the 1950s, it has been continuously evolving, advancing and changing. Old classic study books are no longer enough to sufficiently prepare students and practitioners to face the research world nor the industrial reality. These lectures are a short course meant to strengthen the classic compiler construction / software evolution courses given at Bachelor and Master levels, by highlighting problems encountered in the field of compilation of fourth generation and domain-specific languages, legacy modernisation, software reverse engineering and re-engineering, and revisiting the standard points of interest from the modern standpoint. The lectures will be given based on previous personal experience in research at several leading European institutions, teaching MSc students and working in the industry at the biggest independent compiler company in the world.

There will be examples from modern toolkits and recent academic articles, as well as example stories taken from real industrial projects in language development, support and migration. Our company designs, develops and sells compilers and code analysers for many languages, ranging from legacy behemoths like COBOL or PL/I to customer-specific domain-focused little languages.

Approximate outline of the course goes as follows (for bibliographic references head to a dedicated webpage for this course at <http://grammarware.github.io/cssr/>). During the first lecture will have look at the world around us and pinpoint places where software languages and compilers exist, emerge and operate, shifting the focus from traditional compilers to domain-specific languages and language workbenches. We will also have a brief overview of tools that people who are not necessarily system programmers can use to automate tedious and error-prone tasks of language processing like configuration files handling or dealing with XML and JSON. The second lecture will be very technical and will go deeper into the actual techniques that have either emerged or have been improved during the last decade, including grammar extraction, convergence, adaptation, bidirectional transformation, testing, deployment, etc. The last lecture will highlight a few areas that still deserve much attention, and will be mostly interesting for BSc/MSc/PhD students searching for an inspiring research topic.

Построение компиляторов — одна из самых старых и зрелых областей информатики. Несмотря на долгую историю, начавшуюся в 1950-е годы, эта область постоянно изменяется и развивается. Классические учебники устаревают и не справляются с вопросом подготовки студентов и практикующих специалистов к решению научно-исследовательских и промышленных задач. Эти лекции представляют собой краткий курс углублённого изучения компиляторов, дополняющий традиционные бакалаврские или магистерские лекции по компиляции и эволюции программного обеспечения примерами задач в области компиляции языков четвёртого поколения и предметно-ориентированных языков, модернизации устаревших систем, обратной разработки и т. п. Лекции основаны на опыте преподавания в ведущих европейских вузах, научном руководстве магистрами и работе на крупнейшего в мире независимого производителя промышленных компиляторов (Raincode Labs).

Примеры будут братья из современных программных пакетов и свежих академических статей, а также из ежедневной практики промышленных проектов языковой разработки, поддержки и миграции. Наша компания проектирует, разрабатывает и продаёт компиляторы и анализаторы кода для разных языков от устаревших гигантов вроде КОБОЛа и ПЛ/1 до небольших языков, заточенных под конкретного клиента. Структура курса объяснена ниже (список литературы — на сайте курса <http://grammarware.github.io/cssr/>).

На первой лекции мы поищем примеры окружающих нас программных языков и компиляторов, сместив акцент с традиционного системного программного обеспечения на предметно-ориентированные пакеты и «языковые верстаки». Также мы кратко обсудим конкретные инструменты, облегчающие жизнь несистемным программистам, желающим просто автоматизировать рутинные и хрупкие задачи обработки конфигурационных файлов или манипуляции данными в XML, JSON и подобных форматах. Вторая лекция будет технической, мы углубимся в изучение конкретных техник и подходов, появившихся или развившихся за последнее десятилетие: извлечения, сходимости и адаптивирования грамматик, двунаправленных преобразований, тестирования, внедрения и т. д. На последней лекции будут рассмотрены несколько недостаточно проработанных вопросов, которые будут интересны студентам и аспирантам, ищущим тему для исследований.

Course Outline

- Grammarware as the future of compilers
- Language workbenches and other advanced tools
- Domain-specific and fourth generation languages
- Software language engineering
- Language design with intent
- Modern parsing algorithms: GLL, ALL(*), SGLR, Packrat
- Semi-parsing, error recovery, graceful degradation
- Parsing in a broad sense
- Grammar extraction and recovery
- Grammar quality and grammar smells
- Grammar equivalence and convergence
- Code generation with frameworks
- Bidirectional and coupled transformations
- Semantics and interpretation
- Meta-programming, analysis and comprehension
- Between interpretation and compilation
- Compiler testing and fuzzing
- Language documentation
- Compiler deployment and integration
- Machine learning in compiler construction
- Open problems of the future

Speaker Biography

Dr. Vadim Zaytsev is the Chief Science Officer of Raincode Labs, a Belgian company providing compiler services and specialising in modernisation of software legacy systems. He has acquired PhD in 2010 at the Vrije Universiteit Amsterdam in the field of software language engineering with a focus on grammar(ware) technology. He has done research in that direction at Universität Koblenz-Landau in Germany (2008–2010), Centrum Wiskunde & Informatica in the Netherlands (2010–2013), as well as at University of Amsterdam in the Netherlands (2013–2016) where he also was extensively involved in teaching many graduate-level courses, supervising student projects, maintaining and expanding the network of university relations with the industry. Prior to that, he received MSc cum laude degrees from Rostov State University in Russia (applied mathematics, model checking) and from Universiteit Twente in the Netherlands (telematics, grammar-based testing) while working part-time as a research technician in electro-technical and power-supply labs of Rostov State Transport University (electrical engineering, railway engineering). His interests and research activities tend to invade such topics as software quality assessment, source code analysis and transformation, modelling, metamodeling and megamodeling, programming paradigms, declarative and functional programming, maintenance and renovation of legacy systems and others. More information on his publications can be found at <http://grammarware.net> and on his projects at <http://grammarware.github.io>.

Программа курса

- Будущее компиляторов и грамматическое обеспечение
- Языковые «верстаки» и прочий мощный инструментарий
- Предметно-ориентированные языки и четвёртое поколение
- Проектирование программных языков как новая область
- Умышленная разработка языков
- Современные алгоритмы разбора: GLL, ALL(*), SGLR, Packrat
- Полуразбор, восстановление от ошибок, изящная деградация
- Разбор в широком смысле
- Извлечение и восстановление грамматик
- Качество грамматик и грамматики с душком
- Эквивалентность и сходимости грамматик
- Кодогенерация с помощью соответствующих библиотек
- Двунправленные и связанные преобразования
- Семантики и интерпретация
- Метапрограммирование, анализ и понимание
- Между интерпретацией и компиляцией
- Тестирование компиляторов и техника размытия
- Языковая документация
- Внедрение и интеграция компиляторов
- Место машинного обучения в построении компиляторов
- Нерешённые проблемы будущего

Краткая биография лектора

В. В. Зайцев — заместитель директора по науке в бельгийской компании Raincode Labs, занимающейся предоставлением услуг по разработке и поддержке компиляторов, а также консультациями по связанными с этим вопросам. Он защитил кандидатскую диссертацию в 2010 году в Свободном университете Амстердама по направлению языковой инженерии и грамматических технологий. Над этой темой он также работал в немецком Университете Кобленца-Ландау (2008–2010), нидерландском Центре математики и информатики (2010–2013) и в Университете Амстердама (2013–2016), где также был активно вовлечён в учебный процесс, читал курсы магистрам и бакалаврам, осуществлял научное руководство, поддерживал и расширял связи университета с индустрией. В самом начале своей карьеры он получил диплом специалиста Ростовского государственного университета (ныне ЮФУ) и диплом магистра Технического университета Твенте (в Нидерландах), оба с отличием. Параллельно он участвовал в проектировании разнообразных программных пакетов железнодорожной направленности и написании научных статей в качестве техника научно-исследовательского сектора энергетического факультета Ростовского государственного университета путей сообщения. Его интересы и научная активность сосредоточена вокруг таких тем, как оценка качества программного обеспечения, анализ и трансформации программного кода, моделирование, метамоделирование и мегамоделирование, парадигмы программирования, декларативное и функциональное программирование, поддержка и обновление устаревших систем. Полное резюме и список публикаций доступен по адресу <http://grammarware.net>, а список активных проектов — <http://grammarware.github.io>.