

Проекты квалификационных работ 2011

Пеленицын А. М.

1. Метапрограммирование в C++

Изначально механизм шаблонов C++ создавался для обеспечения базовых потребностей статического полиморфизма, однако в середине 90-х стало ясно, что «the C++ template facility goes far beyond simple “containers of T”» (B. Eckel, TIS++, vol. II). Как выяснилось, шаблоны представляют собой Тьюринг-полный язык в языке, программы на котором выполняются во время компиляции (процесс инстанцирования шаблонов). Сочетание этого языка совместно с самим C++ предоставляет интересные возможности решения задач проектирования ПО. Этот вопрос рассмотрен в ряде книг (А. Александреску, «Современное проектирование на C++», D. Abrahams, A. Gurtovoy, «C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond»), однако хороших практик применения метапрограммирования на шаблонах C++ всё ещё не так много.

Одной из интересных задач, для решения которых техники метапрограммирования могут оказаться полезными, мне представляется проектирование библиотеки арифметики конечных полей. Я ожидаю существенного преимущества от использования таких техник по сравнению с мейнстримовыми подходами, обладающими намного более скромными полиморфными возможностями (ООП в духе Java/C# или процедурное программирование C). Выявление этих преимуществ на основе собственной реализации такой библиотеки и сравнении её с известными библиотеками (например, [NTL](#)) или «очевидными» ОО-подходами и является основной целью работы.

Имеются также более специальные вопросы использования и создания новых паттернов метапрограммирования, например, обработка рекурсивных метапрограммных структур данных.

2. Компонентная модель OSGi

Идея модульности является одной из главных идей, лежащих в основе создания больших (программных) систем. Был предпринят ряд попыток достаточно формально описать общую архитектуру расширяемой за счёт модулей системы или платформы. Один из примеров это [OSGi](#). Имеется довольно популярная реализация модели OSGi, которая называется Equinox и на которой построена Eclipse IDE. Целью работы является изучение и описание инфраструктуры OSGi и создание на основе этого разнообразных плагинов к Eclipse. Примерами простых заданий являются обычные редакторы с подсветкой синтаксиса (например, для PascalABC.NET или Haskell). Более сложные задачи состоят в создании минимальной IDE для языков PascalABC.NET и Haskell. Здесь неизбежно возникнет ряд интересных вопросов интероперабельности различных платформ и оболочек (Java ↔ .NET/Mono, взаимодействие с интерпретатором Haskell). Дальнейшее развитие полученных плагинов можно про-

должать практически бесконечно, организуя взаимодействие с другими плагинами Eclipse (например, плагинами для работы с репозиториями, такими как MercurialEclipse). Для выполнения работы необходимо будет ознакомиться с имеющейся литературой на английском языке, посвящённой описанию OSGi и созданию плагинов Eclipse.

3. Комбинаторика на словах

Относительно молодой раздел дискретной математики, изучающий свойства формальных языков и отдельных слов средствами комбинаторики. Основополагающим трудом в этой области является серия из трёх книг М. Lothaire. Здесь встречается значительное число интересных алгоритмов. Например, алгоритмы для взвешенных автоматов (аналогичные алгоритмам для обычных конечных автоматов) и их использование для статистической обработки естественных языков (М. Lothaire, *Applied Combinatorics On Words*, chap. 4). Кроме реализации отдельных алгоритмов важной частью работы стало бы написание обзора развития дисциплины по первой книге М. Lothaire, так как эта область, насколько мне известно, плохо представлена в русскоязычной литературе.

4. Прикладная алгебра

В настоящее время существует ряд программных пакетов (в том числе, с открытыми кодами), относящихся к так называемым системам компьютерной алгебры, а также отдельных библиотек для алгебраических вычислений. Не все из них содержат достаточно полный набор последних алгоритмов алгебры и алгебраической геометрии. Задача состоит в расширении какого-либо пакета реализацией таких алгоритмов. Например, автор библиотеки [NTL](#) предлагает реализовать в структуре его библиотеки алгоритмы редукции на решётках ([LLL алгоритм](#)), структуры данных и алгоритмы для [базисов Грёбнера](#). Меня интересует также возможность написания расширений для платформы [Sage](#) (язык программирования — Python). Основным источником базовых алгоритмов вычислительной алгебраической геометрии может стать книга D. A. Cox, J. Little, D. O'Shea, «Using Algebraic Geometry» (Springer, 2005), а также ряд интересных статей одного из её авторов Джона Литтла.