УДК 519.685.3

Улитин К.А. (5 курс, каф. СП, мат-мех, СПбГУ),

Кириленко Я.А., ст.преп. СПбГУ

МОДУЛЬНЫЙ ГЕНЕРАТОР СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ

Задача генератора синтаксических анализаторов состоит в преобразовании грамматики, заданной на определенном языке, в код синтаксического анализатора или транслятора. Каждое приложение имеет свой язык описания грамматики, алгоритм генерации, порождает транслятор, работающий по строго предопределённому алгоритму. Но в процессе развития грамматики возможен выход за рамки возможностей генератора.

Разрабатываемый инструмент позволяет пользователю выбирать парсер входной грамматики и генератор из списка поддерживаемых, что делает инструмент более гибким и охватывающим более широкий спектр задач. Инструмент построен вокруг унифицированного представления трансляции и оснащен библиотекой распространённых преобразований над грамматикой. Для достижения желаемой гибкости применяется модульная архитектура, позволяющая создавать инструмент под задачу из базовых элементов. Кратко структуру приложения можно изобразить так:



Такая архитектура дает возможность интегрировать сторонние приложения в рассматриваемый инструмент с минимальными трудозатратами, что позволяет переиспользовать наработки других инструментов, в частности, описанные грамматики различных языков и алгоритмы генерации. Достаточно лишь реализовать преобразование внутреннего представления грамматики нашего инструмента и стороннего, что чаще всего возможно. Это достигается широтой набора конструкций внутреннего представления грамматики нашего инструмента.

Удобные возможности для построения такого инструмента предоставляет платформа .NET, такие, как возможность реализовывать разные модули на более удобных языках, подходящих для задачи, богатый набор различных библиотек, удобную среду разработки и билд-систему. Также, имеется немалое число готовых генераторов синтаксических анализаторов, которые можно использовать, интегрировав их в проект.

Описание компонентов путем реализации интерфейсов позволяет организовать модульное тестирование, для чего используется система NUnit. Также проводится и автоматизированное тестирование всего продукта – по описанию входной грамматики строится инструмент и проверяется правильность его работы.

Исходный код и дополнительную информацию по проекту можно найти на сайте http://code.google.com/p/recursive-ascent/.