УНИВЕРСАЛЬНЫЙ СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР

А.И. Толкачёв

Универсальный синтаксический анализатор - это пакет программ для построения лексических и синтаксических анализаторов. Их основная задача - первичная обработка текстов на каком-либо формальном языке, например, языке программирования. Обработка текста заключается в проверке его на корректность и построении структур данных, необходимых для дальнейшей его обработки. Лексический и синтаксический анализаторы являются составной частью любого компилятора. Опишем схему их работы:

Исходный текст поступает на вход лексического анализатора. Он разбивает текст на лексемы, т.е. мелкие составные части, например отдельные слова, знаки пунктуации и т.д. В результате получается последовательность лексем, которая анализируется синтаксическим анализатором. Синтаксический анализатор последовательно читает лексемы и одновременно строит другое представление текста, которое будет использоваться в дальнейшем. Это обычно таблицы идентификаторов и синтаксическое дерево. Если провести аналогию с русским языком, то это список встретившихся слов и структура текста, отдельных предложений и т.д.

Заметим, что таблицы и дерево - только характерный пример, на практике могут использоваться разные структуры в зависимости от задачи.

Для описания грамматики языка используются расширенные формулы Бэкуса-Наура (РБНФ). Для описания действий, которые анализатор должен выполнять во время разбора, в РБНФ вводятся символы действия (СД). Они являются основным механизмом взаимодействия анализатора с внешней программой. Формально СД можно определить как символы грамматики, при

распознавании которых происходит некоторое действие, например, передача информации внешней программе (т.е. программе, использующей анализатор), построение синтаксического дерева или исполнение некоторого кода, описанного в грамматике. С помощью СД реализуются семантические правила и передача информации о типе прочитанной лексемы из лексического анализатора.

Для построения анализатора используется разработанный мной алгоритм, позволяющий выделить из альтернативных правил одинаковые префиксы конечной или бесконечной длины, что позволяет в большинстве случаев уменьшить k для LL(k)-грамматик. Этот алгоритм используется в сочетании с алгоритмом линейной аппроксимации аванцепочек (linear approximate lookahead [3]), позволяющим строить эффективные распознаватели для LL(k), k>1 грамматик, имеющих практическое значение.

Реализация. Пакет состоит из двух частей - компилятора РБНФ и самого анализатора, реализованного в виде COM-объекта. РБНФ описываются в текстовом файле, после его компиляции получается файл с описанием грамматики во внутреннем формате анализатора. Далее программа, использующая анализатор, сообщает ему имя файла с нужным откомпилированным описанием грамматики и анализатор загружает его. Взаимодействие программы с анализатором происходит через COM-интерфейс.

Описание грамматики состоит из описания лексем и описания синтаксиса. В описании синтаксиса в качестве терминалов могут использоваться только лексемы. В описании синтаксиса могут использоваться СД следующих типов:

– передать команду внешней программе, команда - это числовая константа, при компиляции РБНФ создаётся исходный файл на языке C++ или Pascal, содержащий описания всех констант;

– передать последнюю прочитанную лексему внешней программе;

– вызвать подпрограмму, работает аналогично передаче команды с той разницей, что используется другой метод интерфейса, предоставляющий возможность управления работой анализатора из внешней программы. СД этого типа фактически является семантическим правилом.

Возможные применения – работа с файлами конфигурации, интерпретаторы, подсветка синтаксиса в редакторах. В настоящий момент планируется его использование в системе HLCCAD для поддержки моделей на Vhdl, а так же для обработки специального языка описания тестов.

Литература

1. А.Ахо, Дж.Ульман Теория синтаксического анализа, перевода и компиляции Т.1,2 М.: Мир, 1978
2. Толкачёв А.И. Универсальный синтаксический анализатор: Материалы XXVI научной конференции по есетственным, техническим и гуманитарным наукам. Гомель, 1997

3. Terence J. Parr "Obtaining Practical Variants of LL(k) and LR(k) for k>1 by Splitting the Atomic k-Tuple" PhD thesis, Purdue University, West Lafayette, Indiana, August 1993.