

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ТРАНСЛЯТОРОВ

А.И. Толкачёв

Необходимость обработки текстовой информации, вводимой человеком, возникает в самых разнообразных программных продуктах – от компьютерных игр до компиляторов. В большинстве случаев эта обработка осуществляется с использованием лексического и синтаксического анализаторов.

Поскольку написание лексического и синтаксического анализаторов на каком-либо алгоритмическом языке программирования – весьма трудоёмкий процесс и внутренняя структура анализаторов практически не зависит от синтаксиса обрабатываемого языка, то для построения анализаторов используются т.н. генераторы распознавателей.

Генератор распознавателей представляет собой универсальное программное обеспечение, позволяющее по заданному формальному описанию некоторого языка автоматически получить для него лексический и синтаксический анализаторы.

Основные требования, предъявляемые подобным системам – простота и наглядность формального описания грамматики, эффективность по скорости работы и требуемому объёму памяти как анализатора, так и генератора распознавателей, возможность использования системы в проектах, разрабатываемых с использованием различных языков программирования. Из конкретных применений генераторов распознавателей можно выделить разработку инструментария для разработки программного обеспечения встроённых систем, быстро развивающихся в последнее время. При создании или настройке компиляторов, интерпретаторов или ассемблеров для таких систем непременно возникает необходимость изменения синтаксиса входного языка при переходе с одной платформы на другую (добавление специфических ключевых слов, типов дан-

ных и др.). По этой причине класс грамматик, поддерживаемый генератором распознавателей, должен быть достаточно широким для того, что бы внесение изменений в описание грамматики не требовало значительных усилий со стороны программиста.

Кроме лексического и синтаксического анализаторов, важными составными частями компилятора являются генератор промежуточного представления программы и модуль, обеспечивающий сохранение отладочной информации. Эти части имеют много общего для различных компиляторов и, поэтому, желательно их наличие в универсальном средстве проектирования компиляторов.

Многие существующие программные средства не удовлетворяют перечисленным требованиям. Актуальной является и теоретическая разработка проблемы, поскольку методы, известные в теории грамматик, часто оказываются неприменимыми или недостаточно эффективными для задач, возникающих на практике.

Автором разработан программно – технологический комплекс (ПТК) Unisan. Распознаватель позволяет работать с $SLL^1(k)$, $k \geq 1$ - грамматиками, которые являются подмножеством $LL(k)$. Объём памяти и время, затрачиваемое для работы и построения распознавателя для $SLL^1(k)$ – грамматики, линейно зависит от k , что позволяет эффективно использовать грамматики при $k > 1$. Кроме этого, используется эвристический алгоритм факторизации синтаксических диаграмм, позволяющий в ряде случаев уменьшить k или даже привести грамматику к виду $SLL^1(k)$, если она не являлась такой в исходном виде. Из дополнительных возможностей отметим наличие механизма, позволяющего управлять потоком лексем во время разбора. Эта возможность упрощает реализацию, например, макроподстановок или включения файла в исходный текст (директива `#include` препроцессора языка C). ПТК Unisan реализован на языке C++. На данный момент имеется версия для платформы Win32, хотя, при необходимости, возможен его перенос на другие платформы.

Для генерации промежуточного представления программы в ПТК Unisan реализован механизм, позволяющий в описании грамматики включать конструкции, описывающие синтаксическое дерево.

Для сохранения отладочной информации разработан формат ESDI, обеспечивающий высокую скорость сохранения и загрузки программы и являющийся универсальным для различных платформ. Реализована библиотека для работы с этим форматом. Формат ESDI поддерживается средой проектирования и отладки программного обеспечения встроенных систем Winter и средой высокоуровневого проектирования цифровых устройств HLCCAD.

Литература:

1. 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.
2. Terence Parr, Russell Quong "LL and LR Translator Need k Tokens of Lookahead" // SIGPLAN Notices Volume 31 #2 February 1996.

3. А.Ахо, Дж.Ульман Теория синтаксического анализа, перевода и компиляции: Пер с.англ: // под ред. В. М. Курочкина. - М.: Мир, 1978 т.1,2.
4. Д. Грис Конструирование компиляторов для цифровых вычислительных машин М.,Мир 1976.
5. Ф. Льюис, Д. Розенкранц, Р. Стирнз "Теоретические основы проектирования компиляторов" М., Мир, 1976.