

УДК 004.43(075.8)
М 219

Рецензенты: *А.В. Гунько*, канд. техн. наук, доц.;
Е.Л. Романов, канд. техн. наук, доц.

Работа подготовлена на кафедре
вычислительной техники для студентов IV курса АВТФ
(направление 230100 – Информатика и вычислительная техника)

Малявко А.А.

М 219 Системное программное обеспечение. Формальные языки и методы трансляции : учеб. пособие : в 3 ч. / А.А. Малявко. – Новосибирск: Изд-во НГТУ, 2010. – Ч. 1. – 102 с.

ISBN 978-5-7782-1429-3

В первой части рассмотрены процедурная и автоматная модели лексического анализа, изложены теоретические основы аппарата определения лексики (регулярные выражения) языков программирования, элементы теории конечных автоматов без памяти и методы ее практического применения для автоматизированного преобразования системы регулярных определений в лексический анализатор, способы организации информационных таблиц трансляторов, алгоритмы поиска в таблицах и пополнения таблиц.

Адресовано студентам старших курсов и аспирантам, а также преподавателям смежных дисциплин. Может быть полезно студентам и аспирантам ряда других технических специальностей, связанных с разработкой и использованием программного обеспечения.

УДК 004.43(075.8)

ISBN 978-5-7782-1429-3

© Малявко А.А., 2010
© Новосибирский государственный
технический университет, 2010

ВВЕДЕНИЕ

Организация трансляторов – одно из направлений программирования, широко использующих формальные математические модели. Поэтому основная цель настоящего издания – дать сжатое изложение формальных методов описания лексики и синтаксиса языков программирования, сформулировать представление об основных алгоритмах лексического и синтаксического анализа.

Первая часть учебного пособия вводит в теорию формальных языков. Эта часть содержит описание основных механизмов процесса трансляции и раздел лексического анализа, включающий:

- определения основных понятий формальных языков и метаязыков, постановку задач акцепта и анализа;
- элементы теории конечных автоматов без памяти;
- методы преобразования совокупности правил описания лексики языка в детерминированный оптимальный распознаватель слов;
- методы организации таблиц слов и соответствующие алгоритмы поиска в таблицах / пополнения таблиц.

Во второй части пособия рассмотрены вопросы синтаксического анализа.

Третья часть посвящена семантическому анализу, генерации объектного кода и интерпретации.

Трансляторы: компиляторы и интерпретаторы

Все новые программы для компьютеров создаются с использованием уже существующих программ: текстовых редакторов, макропроцессоров (или препроцессоров), трансляторов, компоновщиков, отладчиков. На рис. В.1 показана более или менее типичная последовательность технологических операций по созданию и исполнению новой программы,

выполняемых с помощью уже существующих программ. Исполняемые программы показаны в виде блоков с двойными рамками, обрабатываемые ими данные – с одинарными.

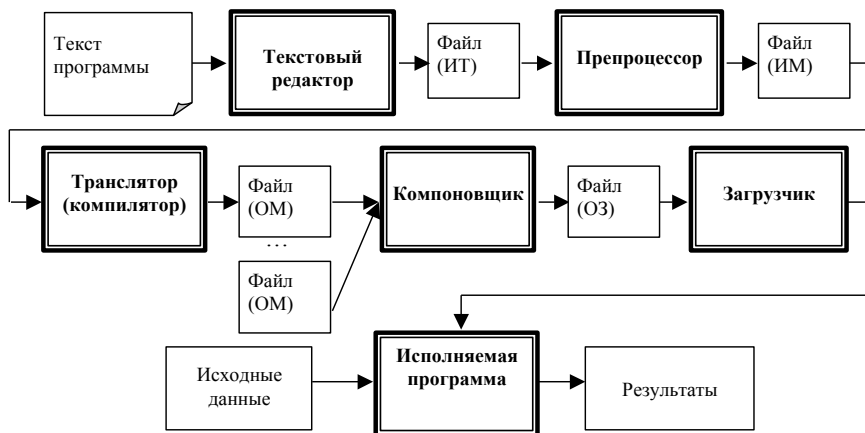


Рис. В.1. Технология компиляции: создание и исполнение программы

В показанной технологии каждый шаг выполняется однократно. На самом деле при создании любой реальной программы большинство из этих шагов, вероятнее всего, будет выполнено неоднократно. Кроме того, на рис. 1.1 не показан упомянутый ранее отладчик, без которого создание сколько-нибудь сложной программы трудно представить. Для удобства использования все программы, участвующие в процессе создания новых программ, обычно включаются в состав так называемых интегрированных оболочек (IDE – integrated development environment), облегчающих программисту взаимодействие с ними. Оболочка на рисунке тоже не показана. Естественно, и сама эта оболочка и все включенные в ее состав программы работают под управлением операционной системы компьютера.

Текст новой программы, написанной на языке программирования высокого уровня (например, Java или С#, или С++ ...) для решения некоторой задачи, вводится с клавиатуры с использованием текстового редактора. Результат работы текстового редактора – исходный текст (ИТ) – запоминается в виде файла на диске. Этот файл затем может обрабатываться макропроцессором, реализующим макроподстановки и включение заголовочных файлов. Результатом этой обработки являет-

ся так называемый исходный модуль (ИМ). Далее транслятор осуществляет преобразование исходного модуля в так называемый объектный модуль (ОМ). Объектный модуль представляет собой файл программы, эквивалентной исходной, но содержащей смесь машинных команд с инструкциями компоновщику по «стыковке» с другими объектными модулями. Несколько объектных модулей, полученных в результате трансляции разных исходных модулей (выполнявшейся, возможно, на различных компьютерах в разных частях света и в разное время), преобразуется компоновщиком (другие названия – сборщик, линкер, редактор связей ...) в файл образа задачи (ОЗ) и сохраняется им на диске. Теперь этот файл может быть использован независимо от тех программ, которые принимали участие в его создании, причем вполне вероятно – на других компьютерах. Загрузчик операционной системы при каждом запуске преобразует его в исполняемую программу, работающую под управлением и во взаимодействии с операционной системой. Исполняемая программа обрабатывает исходные данные (ИД) и формирует результаты решения задачи (Р).

Технология, показанная на рис. В.1 и называемая **компиляцией** (от слова «компилировать», т. е. собирать), отнюдь не является единственно возможной. Транслятор, реализующий в этой технологической цепочке перевод с одного языка на другой, называется, соответственно, компилятором. Показанный на рис. В.2 другой возможный способ получения результатов решения задачи путем прямого выполнения описанных в тексте исходной программы операций преобразования исходных данных, без создания и сохранения образа задачи называется **интерпретацией**.

В этой технологии транслятор называется интерпретатором и выполняет функции как ввода/редактирования, макрообработки, собственно трансляции, так и пошагового исполнения (интерпретации) операций

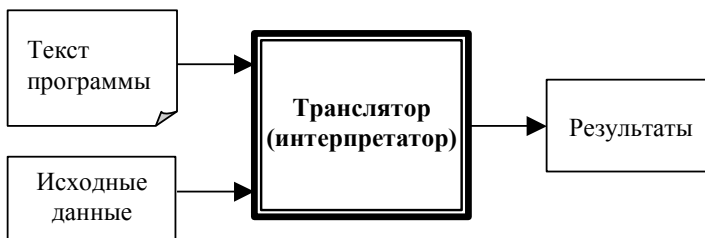


Рис. В.2. Технология интерпретации программы

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Трансляторы: компиляторы и интерпретаторы	3
Элементарные понятия формальных языков	6
Этапы процесса трансляции	9
Проектирование трансляторов	14
1. ЛЕКСИЧЕСКИЙ АНАЛИЗ	16
1.1. Постановка задачи	16
1.2. Структура и функции лексического анализатора	20
1.3. Процедурная реализация лексического акцептора	22
1.4. Автоматная модель лексического акцептора	39
1.4.1 Конечные автоматы без памяти	42
1.4.2. Способы определения лексики формальных языков	56
1.4.3. Преобразование системы регулярных определений в ко- нечный автомат	61
1.5. Взаимодействие лексического анализатора с синтаксическим и семантическим анализаторами	85
1.5.1. Учет областей видимости	87
1.5.2. Пространства имен	88
1.6. Поиск в таблицах / пополнение таблиц	90
1.6.1. Последовательная организация	91
1.6.2. Упорядоченная организация	92
1.6.3. Рандомизированная организация	95
1.7. Формирование лексем	99
ЛИТЕРАТУРА	100