

3. <http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html> — Intel Architecture Software Developer's Manual.
4. <http://ref.x86asm.net/geek32.html> — X86 Opcode and Instruction Reference.

УДК 004.4'41

АНАЛИЗ НЕКОТОРЫХ СЕМАНТИЧЕСКИХ АСПЕКТОВ ИСХОДНЫХ ТЕКСТОВ ПРОГРАММ НА ОСНОВЕ ФОРМАЛЬНЫХ СПЕЦИФИКАЦИЙ СИНТАКСИСА И СЕМАНТИКИ

А.М. Саух

Целью работы является создание инструментального средства, позволяющего осуществлять семантический анализ исходных текстов программ, написанных на различных языках программирования, на основе формальных спецификаций лексики, синтаксиса и семантики целевого языка программирования. Рассматривается задача анализа некоторых семантических аспектов исходных текстов однопоточных программ на императивных языках программирования, таких, как объявление и использование идентификаторов, а также выявления областей их использования на основе анализа потоков исполнения программы. Целью такого анализа является извлечение семантически значимых конструкций и алгоритмов работы из текста программы и представление их в независимом от конкретного языка виде.

Анализ проводится в несколько этапов. Сначала производится лексический и синтаксический анализ, сопровождающийся построением абстрактного синтаксического дерева по исходному тексту программы. В зависимости от конкретного языка построение таблицы идентификаторов и дерева пространств имён происходит либо в один этап — сразу при построении синтаксического дерева, либо в два этапа, первый из которых осуществляется при построении дерева, а второй — при вторичном обходе построенного дерева.

В качестве отправной точки для решения задачи построения синтаксического дерева используется система генераторов компиляторов Lex и Yacc. Генераторы модифицированы так, что вместо исходного кода анализирующих конечных автоматов, который необходимо сначала скомпилировать, сразу создаются готовые к работе структуры анализатора. В качестве формата файлов спецификаций используются оригинальные форматы файлов спецификаций лексики и синтаксиса Lex и Yacc, модифицированные в части описания действий, исполняемых при свёртке по синтаксическим правилам. Лексические правила описываются в виде набора регулярных выражений, которым сопоставлены управляющие конструкции, регулирующие возврат обнаруженных лексем в синтаксический анализатор. Синтаксические правила представляют собой описания структур языка, которым сопоставлены управляющие конструкции, регулирующие построение абстрактного синтаксического дерева, таблиц идентификаторов, дерева пространств имён и других элементов (такой подход описан в [1]). Синтаксическим анализатором поддерживаются контекстно-свободные грамматики, разбираемые по алгоритму LALR.

Семантический анализ производится на основе нескольких обходов абстрактного синтаксического дерева с построением конструкций, описывающих объявление и использование идентификаторов (типов, переменных, функций, классов, методов и т. д.), организацию пространств имён и ограниченно описывающих потоки исполнения команд, что позволяет анализировать использование тех или иных участков кода.

Результатом данной работы является инструмент, способный анализировать исходные тексты программ на языках программирования, описанных с помощью формальных спецификаций лексики, синтаксиса и семантики, и создавать абстрактное представление разбираемой программы, доступное для семантического анализа. В дальнейшем предполагается расширить список формально описанных в терминах системы языков (на данный момент рассматривается только язык C#), а также, возможно, добавить функцию генерации исходных текстов из абстрактного представления.

ЛИТЕРАТУРА

1. Ахо А. В., Лам М. С., Сети Р., Ульман Д. Д. Компиляторы. Принципы, технологии и инструментарий. М.: Вильямс, 2008. 1184 с.