

Тема: Синтаксически управляемая трансляция.

В любом языке программирования есть набор правил составления программ, которые не охватывает грамматика.

Например, в языке МИКРОЛИСП грамматика не может описать такие правила:

- 1. Каждая процедура, вызванная в программе, должна быть определена, причем только один раз.**
- 2. Количество аргументов вызова процедуры должно совпадать с количеством ее параметров.**

Все правила, которые не охватывает грамматика, относятся к правилам семантики, их проверкой занимается семантический анализатор.

Еще один набор правил должен быть формально описан и реализован в компиляторе. Это правила перевода исходного текста в эквивалентный текст на целевом языке. Перевод выполняет генератор кода.

Во всех реально применяемых технологиях конструирования компиляторов алгоритмы семантического анализа и перевода разрабатываются вручную и записываются на каком-то языке высокого уровня. Чаще всего используются языки С или С++, Но есть очень интересный опыт реализации языка Паскаль, выполненной Никлаусом Виртом. Полное описание компилятора он составил на самом Паскале, а затем применил метод «раскрутки». Сначала он вручную написал примитивный промежуточный компилятор. Затем он пропустил текст полного компилятора через промежуточный и получил окончательную реализацию языка. С тех пор язык считается полноценным языком программирования, если на нем можно описать компилятор для самого себя.

Метод синтаксически управляемой трансляции позволяет структурировать программный код компилятора, разбив его на небольшие функции с четко прописанными интерфейсами.

Суть метода в следующем.

Каждому синтаксическому классу сопоставляется набор атрибутов - чисел, строк, булевских флагов и т.п. Сигнатуру и назначение атрибутов разработчик выбирает с учетом информационных потребностей конкретного компилятора.

Атрибуты характеризуют свойства синтаксических классов, существенные для компилятора. Например, определение процедуры имеет такие существенные для семантического анализатора свойства, как имя процедуры и количество параметров. Для генератора кода существенным будет такое свойство, как фрагмент эквивалентной целевой программы.

Пометка узла дерева разбора нетерминалом грамматики обозначает одну цепочку терминалов – экземпляр синтаксического класса. Для этой цепочки фиксируются значения атрибутов. Например, имя процедуры – “f”, количество параметров – 2.

Каждой продукции грамматики сопоставляется функция преобразования атрибутов. Эта функция применяется в момент свертки основы сентенциальной формы к нетерминалу левой части продукции. Функция манипулирует значениями атрибутов дочерних узлов и вычисляет значения атрибутов родительского узла, помеченного нетерминалом. Этот процесс называется синтезом атрибутов. Он начинается от листьев дерева и постепенно распространяется к корню.

Функцию преобразования атрибутов удобно называть продукцией атрибутов, подчеркивая ее неразрывную связь с продукцией грамматики.

Таким образом, продукции грамматики $A \rightarrow a_1 a_2 \dots a_n$ сопоставляется продукция атрибутов $p(Sa_1, Sa_2, \dots, Sa_n) \rightarrow SA$, символ S обозначает набор атрибутов.

Алгоритмы семантического анализа и перевода разделяются на небольшие части, инкапсулированные в продукции атрибутов.

**Синтаксический анализатор создает поток управления
продукциями и обеспечивает передачу значений
атрибутов.**