Семантический анализ

Полный разбор предложения языка не может провести компилятор. Полный распознаватель ЯП можно построить на основе распознавателя контекстно-свободного языка, но такой распознаватель имеет экспоненциальную зависимость требуемую для выполнения разбора цепочки вычислительных ресурсов от длины входной цепочки. Компилятор построенный на основе такого распознавателя будет не эффективным с точки зрения либо скорости работы, либо объема необходимой памяти. Поэтому такие компиляторы используются достаточно редко, а компиляторы на этапе разбора входных цепочек проверяют только синтаксические конструкции входного языка не учитывая его семантику. Но для того чтобы повысить эффективность компиляторов разбор цепочек входного языка выполняется в два этапа:

1. Синтаксический разбор на основе распознавателя одного из известных классов контекстно свободных языков.
2. Семантический анализ входной цепочки. Входными данными служат таблица идентификаторов являющаяся результатом лексического анализа и результаты разбора синтаксических конструкций входного языка.

Семантический анализатор выполняет следующие действия:

* Проверка соблюдения во входной программе семантических соглашений входного языка
* Пополнение внутреннего представления программы в компиляторе операторами и действиями не явно предусмотренными семантикой входного языка.
* Проверка элементарных семантических норм программирования на прямую не связанных с входным языком.

Проверка соблюдения во входной программе семантических соглашений заключается в сопоставлений входных цепочек программы с требованием семантики входного языка программирования. Каждый ЯП имеет четко заданные и специфицированные семантические соглашения, которые не могут быть проверены на этапе синтаксического разбора, именно их и проверяет семантический анализатор.

Контекстным условием, проверяемым синтаксическим анализатором относится следующее :

1. Каждая метка на которую есть ссылка должна один раз присутствовать в программе
2. Каждый идентификатор должен быть описан 1 раз. И ни один идентификатор не может быть описан более 1 раза.
3. При вызове функции число фактических параметров и их типы должны соответствовать числу и типам формальных параметров
4. Все операнды в выражениях и операциях должны иметь типы допустимые для данного выражения или операции
5. Типы переменных в выражениях должны быть согласованны между собой.

Если какой-либо из семантических требований входного языка не выполняется, то компилятор выдает сообщение об ошибке и процесс компиляции на этом может закончится. Дополнение внутреннего представления программы операторами и действиями неявно предусмотренными семантикой входного языка связанно с преобразование типов операндов в выражениях и при передаче параметров в процедуры и функции. Семантический анализ можно выполнять сразу после синтаксического анализа, некоторые требования можно контролировать во время генерации кода, а можно совместить синтаксическим анализом.

На фазе семантического анализа обрабатываются структуры распознанные синтаксическим анализатором. На этой стадии выполняется ряд важных функций таких как ведение таблицы символов, обнаружение большинства ошибок, макро обработка, выполнение инструкций отнесенных ко времени компиляции, включение неявной информации, т.е. информация заданная в программе неявно должна быть представлена в явном виде в программах низкого уровня. Большая часть информации данного типа называется соглашением по умолчанию и интерпретируется соответствующим образом, если не задана явная спецификация. При простых трансляциях семантический анализатор может на самом деле породить объектный код, но обычно выходом этой стадии служит некоторая внутренняя форма выполняемой программы с которой затем транслятор работает на стадии оптимизации перед генерацией кода.

Обычно семантический анализатор разделяется на ряд более мелких анализаторов, каждый из которых предназначен для конкретной программной конструкции, например описание массивов обрабатываются одним анализатором, а арифметические выражения другим. Соответствующий анализатор вызывается синтаксическим анализатором как только последний распознает синтаксическую единицу требующую обработки. Семантические анализаторы взаимодействуют между собой посредством информации хранящейся в различных структурах данных, в частности в центральной таблице символов. Например семантический анализатор обрабатывающий описание типов для простых переменных не делает ничего кроме занесения описанных типов в таблицу символов. Позже семантический анализатор обрабатывающий арифметические выражения может использовать описанные типы для включения соответствующих специфических типов арифметических операций в объектный код. Таким образом действия выполняемые семантическим анализатором существенным образом влияет на порождаемый компилятором код результирующей программы. Проверка элементарных смысловых норм языков программирования напрямую не связанных с входным языком это сервисная функция, которую предоставляют большинство современных компиляторов. Эта функция обеспечивает проверку компилятора некоторых соглашений применимых к большинству современных ЯП, выполнение которых связанно со смыслом как всей входной программы так и отдельных ее фрагментов