Лексический анализатор

Словарь сокращений:

* ЛА – лексический анализатор
* СА – синтаксический анализатор
* ЯП – язык программирования

Лексема – структурная единица языка, которая состоит из элементарных символов языка и не содержит в своем составе других структурных единиц языка. Лексемами являются слова в естественных языках. В ЯП – ключевые слова, идентификаторы и тд. Лексема – терминальный символ.

*В лабораторных работах неправильно используется терминология. Лексема = терминальный символ.*

Любой язык описывается синтаксисом, как и ЯП.

Лексический анализатор – часть компилятора, которая читает исходную программу и выделяет в ее тексте лексемы входного языка. На вход поступает исходный тест программы. С лексического анализа начинается разбор программы для высокоуровневого ЯП. Определяется лексема, определяется ее принадлежность, создаются таблицы:

* Таблица с константами.
* Идентификаторы
* Ключевые слова, разделители

Это объединяется в таблицу лексем. Для каждой лексемы хранится ее тип (константа, идентификатор, ключевое слово). Лексическому анализатору не важно если описание для переменных или нет, главное определить, что идентификатор состоит из правильных символов. В результате работы ЛА является таблица лексем. Любая лексема может встречаться в таблице лексем любое количество раз.

С теоретической точкой зрения ЛА не является обязательной частью компилятора, но большинство компиляторов включают лексический анализ так как:

1. Применение ЛА сокращает объем информации, обрабатываемой на этапе синтаксического разбора.
2. Некоторые задачи, требующие использования сложных вычислительных методов на этапе синтаксического анализа могут быть решены более простыми методами на этапе лексического анализа.
3. Лексический анализатор отделяет сложные по конструкции синтаксический анализатор от работы непосредственно с текстом исходной программы. Структура которого может варьироваться в зависимости от архитектуры ВС, где выполняется компиляция, при такой конструкции компилятора. для перехода на другую вычислительную систему достаточно только перестроить относительно простой лексический анализатор
4. В современных системах программирования лексический анализатор может выполнять обработку теста исходной программы параллельно с его подготовкой пользователем.

ЛА и СА взаимодействуют между собой и существует 2 различных метода такого взаимодействия:

* Последовательный. ЛА просматривает весь текст исходной программы и преобразует в таблицу лексем. Полностью заполненная таблица лексем используется для последующих фаз компиляции.
* Параллельный. ЛА выделяет очередную лексему и передает СА синтаксический анализатор может подтвердить правильность найденной лексемы и обратиться к ЛА за следующей лексемой, либо отвергнуть найденную лексему. Во втором случае он может проинформировать ЛА о том, что надо вернуться назад к уже просмотренному ранее фрагменту исходного кода и сообщить ему дополнительную информацию о том лексему какого типа следует ожидать. Взаимодействуя между собой таким образом ЛА и СА могут перебрать несколько возможных вариантов лексем и если ни один из них не подойдет, будет считаться, что исходная программа содержит ошибку. После того как СА успешно выполнит разбор очередной конструкции исходного языка, ЛА помещает найденные лексемы в свои таблицы и разбор продолжается в том же порядке.

В конечном автомате, который лежит в основе ЛА действие по обнаружению лексемы могут трактоваться шире чем только порождение цепочки символов входного языка. Он должен выполнять следующие действия:

1. Помещать новую лексему в таблицу лексем.
2. Проверять наличие найденной лексемы в таблице идентификаторов, если нет, то добавлять.
3. Добавлять новую лексему в таблицу идентификаторов.
4. Выдавать сообщение пользователю о найденных ошибках и предупреждения об обнаруженных неточностях в программе.
5. Прерывать процесс компиляции.