

Лабораторная работа № 1.

Цели и задачи работы: Построение КС-грамматики языка программирования и конечного автомата, реализующего лексический уровень.

Теоретические сведения о работе приведены в источниках [1,3,4] и в конспекте лекций по дисциплине.

Задание к работе:

1. Для языка программирования, указанного в Вашем задании, определить синтаксическую конструкцию, соответствующую главной программе. Построить правила КС-грамматики, определяющие программу в целом и место в ней описаний и операторов. Построить правила КС-грамматики, определяющие синтаксис отдельных операторов и описаний.

2. Проанализировать список операций (арифметических, логических, сравнения и т.п.), которые должны использоваться в языке программирования. Построить таблицу приоритетов операций. Построить правила КС-грамматики для выражений. При определении понятия элементарного выражения учесть все типы констант и простых операндов (из числа простых переменных, элементов массивов, вызовов функций, полей структур и т.п.).

3. Определить лексический и синтаксический уровень Вашей грамматики. Для этого в каждом правиле КС-грамматики выделить все лексемы.

4. Построить таблицу лексем языка программирования на основе КС-грамматики. В таблице отметить все признаки окончания лексем. Ввести обозначения типов для каждой лексемы. Ввести дополнительные обозначения для специальных лексем окончания исходного модуля и ошибочного типа лексемы.

5. Построить конечный автомат, допускающий соответствующие лексемы, и выполнить его детерминизацию. Разметить все заключительные состояния автомата, указав в них соответствующие типы выделяемых лексем.

6. Построить список игнорируемых символов (например, знаков пробелов, табуляции и т.п.). Построить конечный автомат для комментариев, которые допускаются в языке программирования Вашего задания.

Описание используемых средств для выполнения работы : текстовый редактор.

Методика выполнения работы:

1. Изучить построение правил КС-грамматики для языков программирования.

2. Выполнить задание к работе.
3. Результат работы оформить в виде файла.

Требования к отчету:

Отчет должен содержать титульный лист, индивидуальное задание по языковым конструкциям, распечатку файла.

Примеры индивидуальных заданий :

В задании описан усеченный вариант языков C и Паскаль. Предполагается, что допустимо использование пустого и составного операторов, помимо указанных.

Индивидуальный вариант.

Программа: функции языка C, включая главную функцию, с параметрами, возвращающие значения допустимых в программе типов.

Типы данных: int, float

Составные типы данных: одномерные массивы

Операторы: присваивания, if

Операции: арифметические, логические

Лабораторная работа № 2.

Цели и задачи работы: Программирование лексического анализатора (сканера).

Теоретические сведения о работе приведены в источниках [3,5] и в конспекте лекций по дисциплине.

Задание к работе:

1. Определить в файле defs.h типы данных, соответствующие лексемам.
2. Написать программу сканера в соответствии с конечным автоматом, построенным в лабораторной работе 3. Предусмотреть выдачу сообщения о лексических ошибках. Предусмотреть ограничение длины выделяемой лексемы, если конечный автомат может допускать лексемы бесконечной длины.
3. Написать процедуру выдачи сообщения об ошибке PrintErr().
4. Написать главную программу, в которой необходимо предусмотреть:
 - чтение файла с исходным модулем и вставку символа-маркера конца;
 - вызов сканера до тех пор, пока не достигнут конец исходного модуля;
 - вывод отсканированной лексемы и ее типа.
5. Построить проект, содержащий главный модуль, сканер, опи-

сание глобальных данных и типов, функцию PrintErr().

6. Отладить программу на правильных и неправильных лексических конструкциях в исходном модуле.

Описание используемых средств для выполнения работы : язык программирования высокого уровня (C, C++, C#, Java), среда разработки (Visual Studio, Eclipse, NetBeans).

Методика выполнения работы:

1. Изучить типовую структуру программы-сканера.
2. Изучить программирование сканера в соответствии со структурой конечного автомата.
3. Выполнить задание к работе.

Требования к отчету:

Отчет должен содержать титульный лист, текст разработанного программного обеспечения, тесты.

Индивидуальные задания берутся из лабораторной № 1.

Лабораторная работа № 3.

Цели и задачи работы: Построение синтаксических диаграмм и их разметка для программирования.

Теоретические сведения о работе приведены в источнике [3,5] и в конспекте лекций по дисциплине.

Задание к работе:

1. Построить синтаксические диаграммы для КС-грамматики.
2. Преобразовать построенные синтаксические диаграммы:
 - устранить левую рекурсию;
 - вынести левые и правые множители;
 - выполнить подстановку диаграммы в диаграмму, если в результате такой подстановки уменьшится общая сложность конструкции.
3. При устранении рекурсии учитывать семантическую нагрузку, которую несут отдельные лексемы и конструкции.
4. Выполнить разметку ветвей синтаксических диаграмм. Для этого
 - построить функции *first* для всех нетерминалов Вашей грамматики;
 - построить функции *follow*;
 - разметить ветви синтаксических диаграмм с использованием построенных функций.

5. Проверить однозначность переходов в диаграммах по точкам ветвления. Если переходы неоднозначны, то выбрать один из вариантов разрешения каждого конфликта:

- преобразовать грамматику и синтаксические диаграммы;
- увеличить длину анализируемого контекста в точках ветвления;
- принять некоторые соглашения, однозначно определяющие действия в точке ветвления (например, по аналогии с условием, при котором else всегда соответствует последнему if).

Описание используемых средств для выполнения работы: текстовый и графический редактор, операционная система Windows. Для чернового варианта работы – бумага, карандаш, ластик.

Требования к отчету:

Отчет должен содержать титульный лист, индивидуальное задание по языковым конструкциям, распечатку файла с диаграммами, размеченными с помощью функций first и follow, распечатку таблицы со значениями first и follow для нетерминалов грамматики.

Индивидуальные задания берутся из лабораторной № 1.

Лабораторная работа № 4.

Цели и задачи работы: Программирование синтаксического анализатора методом рекурсивного спуска.

Теоретические сведения о работе приведены в источниках [3,5] и в конспекте лекций по дисциплине.

Задание к работе:

1. Каждой синтаксической диаграмме, полученной в лабораторной работе 5, поставить в соответствие процедуру (функцию) без параметров. В силу рекурсивного характера совокупности синтаксических диаграмм предусмотреть предварительное объявление каждой функции, например, в модуле Diagram.h.
2. Написать тело каждой функции, соответствующей синтаксической диаграмме.
3. В главной программе предусмотреть вызов функции, соответствующей аксиоме.
4. Дополнить проект лабораторной работы 4 модулем, в котором содержится реализация функций всех синтаксических диаграмм.

5. Дополнить набор сообщений о лексических ошибках, которые выдаются программой PrintErr(), сообщениями о синтаксических ошибках.

6. Отладить программу на правильных и неправильных синтаксических конструкциях в исходном модуле.

Описание используемых средств для выполнения работы : язык программирования высокого уровня (C, C++, C#, Java), среда разработки (Visual Studio, Eclipse, NetBeans).

Требования к отчету:

Отчет должен содержать титульный лист, текст разработанного про-

Индивидуальные задания берутся из лабораторной № 1.