Курсовая работа

Методические указания по выполнению

Курсовая работа должна выполняться после изучения всего теоретического материала и выполнения лабораторных работ. Курсовая работа состоит в написании программы в соответствии с заданием. В случае наличия ошибок в программе она возвращается на доработку.

**При выполнении работы над ошибками необходимо сохранять замечания преподавателя, а изменения в отчёт вносить другим цветом.**

Выполнение задания включает разработку программного средства, тестирование его на наборе данных и написание отчёта по работе.

Отчёт должен содержать:

1)      титульный лист;

2)      номер варианта и текст задания;

3)      описание алгоритма решения задачи с иллюстрацией его на своём примере;

4)      описание основных блоков программы;

5)      текст программы;

6)      результаты тестирования программы;

7)      распечатку файла результатов – 2–3 примера.

Высылаемый на проверку преподавателю архив должен содержать отчёт и все файлы программы – все модули, exe-модуль, файлы данных, файлы результатов.

**Внимание!!!**

**Необходимо предусмотреть обработку ошибок.
Никакие входные данные не должны нарушать работу программы!! Программа не должна «зависать» или прекращать выполнение по неизвестной причине – обязательна выдача соответствующей диагностики.**

При выполнении задания следует использовать средства объектно-ориентированного программирования. Выбор конкретного средства разработки оставляется за студентом. При наличии определённых требований к программному обеспечению студенту следует уведомлять об этом преподавателя.

Рекомендуется при разработке программного средства использовать материалы лабораторных работ (в зависимости от темы задания).

Программа должна управляться посредством меню, в котором должны присутствовать следующие пункты: "Автор", "Тема" (с полной информацией о разработчике и теме задания), "Данные" (выбор способа задания исходных данных – чтение из файла или ввод с клавиатуры), "Расчёты", "Запись результатов в файл" – и другие, определяемые конкретным заданием. При вводе данных с клавиатуры необходимо использовать соответствующую форму, а также предусмотреть возможность вызова справки с примером формата данных. При чтении из файла – должна открываться своя папка. Все результаты расчётов должны отображаться на экране и выводиться в файл (по требованию пользователя). При введении автором каких-либо ограничений (размер алфавита и т.п.) они должны быть описаны в пояснительной записке и в соответствующем пункте меню **Вариант 7**

Написать программу для автоматического приведения заданной контекстно-свободной грамматики (КС-грамматики) к нормальной форме Хомского (БНФ).

Вход программы: терминальный и нетерминальный алфавиты грамматики, целевой символ, правила грамматики, 2 числа – диапазон длин для генерации цепочек.

Выход: построенная грамматика в БНФ (все 4 элемента), результат генерации цепочек по обеим грамматикам.

Подробно:

Язык задан КС-грамматикой, причём для приведения к БНФ она должна находиться в каноническом виде (раздел лекций 3.2.2). Приводить её к этому виду не требуется, достаточно только проверить корректность задания – действительно ли исходная грамматика находится в каноническом виде – и при отрицательном результате выдать соответствующее сообщение. Причём в этом сообщении должны быть конкретно указаны причины, почему именно грамматика не имеет канонического вида (например: «в грамматике присутствуют цепные правила: А→С», или «в грамматике присутствует недостижимый символ: В»). Для того чтобы в исходной грамматике можно было использовать пустое правило, необходимо либо предусмотреть поле ввода для символа, которым обозначается пустая цепочка, либо дать пояснения пользователю, как именно ему следует задавать пустое правило.

Программа должна:

1.            проверить заданную КС-грамматику – находится ли она в каноническом виде и при отрицательном результате выдать сообщение;

2.            привести заданную КС-грамматику к нормальной форме Хомского (раздел 3.3.1);

3.            проверить построенную грамматику (БНФ) на эквивалентность исходной.

Для проверки построенной грамматики в БНФ на эквивалентность исходной по обеим грамматикам следует сгенерировать множества всех цепочек в заданном пользователем диапазоне длин и проверить эти множества на идентичность. Генерация цепочек осуществляется в соответствии с лабораторной работой №1. При обнаружении несовпадения должна выдаваться диагностика различий – где именно несовпадения и в чём они состоят. Для удобства сравнения множества цепочек необходимо упорядочить, цепочки перенумеровать, для генерации цепочек по каждой грамматике сделать отдельную кнопку. Следует предусмотреть возможность многократного изменения диапазона длин цепочек и повторной их генерации для новых значений длины.

Приведение КС-грамматики к нормальной форме Хомского следует осуществлять согласно алгоритму, изложенному в лекционном материале (раздел 3.3.1).

.