Лабораторная работа № 1 Генерация цепочек языка

Пусть язык задан контекстно-свободной грамматикой. Написать программу, которая по заданной грамматике будет генерировать ВСЕ цепочки языка в некотором диапазоне длин. Использовать только левосторонний или правосторонний вывод! Диапазон длин генерируемых цепочек должен задаваться пользователем при запуске программы.

Предусмотреть возможность выбора пользователю – использовать заданную в программе грамматику или вводить свою с клавиатуры.

На вход программы подаётся КС-грамматика (терминальный и нетерминальный алфавиты, целевой символ, правила вывода), задаётся диапазон длин цепочек, указывается тип вывода (левосторонний или правосторонний).

Рекомендуется для ввода исходных данных использовать соответствующую форму. При вводе правил грамматики не рекомендуется предоставлять пользователю излишнюю свободу действий, например, удобнее запретить ввод в левой части правил более чем одного нетерминального символа, чем проверять введённые правила на соответствие контекстно-свободному типу.

На выходе должен быть список построенных цепочек. Все цепочки в списке должны быть различны. При получении одинаковых цепочек (разными способами) их следует сохранять в списке выводимых цепочек только в случае выполнения дополнительного задания.

Проверить программу на примерах из лекционного курса и на заданиях из контрольных вопросов к теоретической части.

Дополнительно: Дополнить предыдущую программу таким образом, чтобы для одной или нескольких цепочек (цепочки выбирает пользователь из числа построенных на предыдущем этапе работы) строилось дерево вывода, т.е. была структурно представлена последовательность правил, использованных при построении цепочки.

***Контрольные вопросы:***

1. Как поведёт себя программа, если при вводе правил грамматики сделать ошибку – ввести правила грамматики, не соответствующей КС-типу, или не использовать какие-то символы исходных алфавитов, или поместить в левой части правил терминальный символ?
2. Используется ли при генерации цепочек рекурсия? Если да, то как ограничена её глубина?
3. Какой язык порождается грамматикой G({0},{S,A},{S→AAA, A→AAA |0}, S)? Правильно ли Ваша программа работает на этом примере? Если нет – доработайте программу.
4. Разрешает ли Ваша программа использовать в исходной грамматике пустые правила? Если нет – доработайте программу.