**Теория языков программирования и методы трансляции**

Написать программу для автоматического построения грамматики, эквивалентной заданному регулярному выражению (РВ).

Вход программы: регулярное выражение в виде строки символов, 2 числа – диапазон длин для генерации цепочек.

Выход: построенная грамматика (все 4 элемента), результат генерации цепочек.

Подробно:

Язык задан регулярным выражением. При его записи могут быть использованы символы алфавита языка, а также: «+» (выбор одного из слагаемых), круглые скобки, «\*» для обозначения итерации.

Программа должна:

1. по предложенному регулярному выражению строить эквивалентную грамматику, генерирующую этот же язык, в том виде, как она рассматривалась в теории, раздел 1.3.1;
2. с помощью построенной грамматики генерировать все цепочки языка в заданном пользователем диапазоне длин.

Грамматика может строиться любая – контекстно-свободная или регулярная, по выбору разработчика. Отдельно следует указывать, какой нетерминальный символ является целевым. Если в грамматике используется пустое правило, то необходимо дать пояснение, каким именно символом обозначается пустая цепочка.

После построения грамматики пользователь может убедиться в её правильности путём генерации всех цепочек языка в том диапазоне длин, который он задаст. Генерацию каждой цепочки языка следует поэтапно отображать на экране в виде цепочки вывода (в соответствии с примерами раздела 1.4.1.). Генерация осуществляется в соответствии с лабораторной работой №1.

Рассмотрим пример построения КС-грамматики.

Задано регулярное выражение: ((0+1+b)\*a(0+1+b)\*a)\*(0+1+b)\*a(0+1+b)\*01a.

Для построения правил грамматики следует сделать разбор исходного регулярного выражения. Каждая скобка обозначается своим нетерминалом. Если на скобке стоит звёздочка (итерация), значит, на этом нетерминале будет явная рекурсия и пустое правило. Если в выражении стоит «+», то это означает альтернативу в правилах.

Обозначим первую большую скобку через A=((0+1+b)\*a(0+1+b)\*a)\*, вторую B=(0+1+b)\*. Само РВ должно порождаться из целевого символа грамматики. Тогда первое правило будет иметь вид: S→ABaB01a. В правиле для A будет присутствовать B: A→BaBa. Поскольку на скобке есть звёздочка, то надо добавить рекурсию и пустое правило: A→BaBaA|λ. Нетерминал B рекурсивно порождает любые символы, кроме ‘a’: B→0B|1B|bB|λ. Итак, грамматика построена, выпишем её полностью.

G({0,1,a,b},{S,A,B},P,S), где P: S→ABaB01a; A→BaBaA|λ, B→0B|1B|bB|λ.

Аналогично строится регулярная грамматика, только там следует учитывать, что в правой части правил может использоваться не более одного нетерминала, и располагаться во всех правилах грамматики он должен с одной стороны от цепочки терминальных символов.