

对于剩余部分，首先用两个循环对生成式进行遍历，将转换规则中(q,a,A) -> (p,BC...)提取出来存储到q, a, A, state中，若state大小不等于1，说明转化后的生成式右侧不止一个非终结符，首先把生成式左侧以及右侧的输入字符a储存好，对剩余部分我们进行如下讨论：

对自动机中若有

而S可以推导出,和,

则有：

,

,

即生成式右侧的第一个非终结符的左侧需要与生成式左侧的非终结符的左侧状态相同，而生成式右侧的最后一个非终结符的右侧的状态需要与生成式左侧非终结符的右侧状态相同，而生成式右侧的非终结符的其他状态则需要对所有状态进行遍历填空并满足前后相等即可

对上面的例子为：

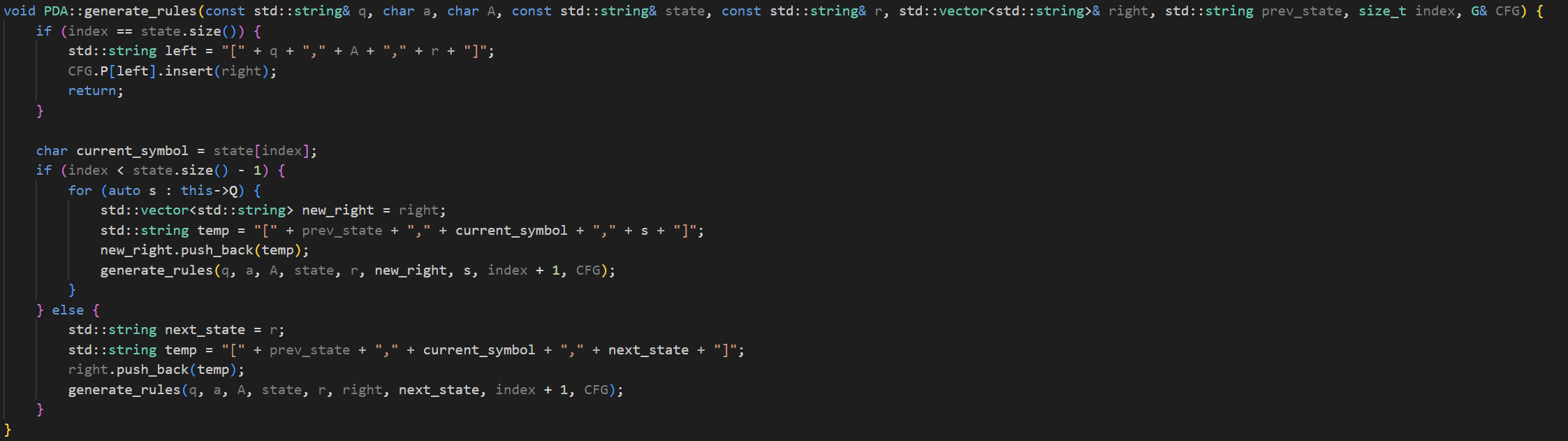
,

,

,

,

所以若Q中有n个状态，一条生成式则以上的中间中间部分有n种可能，若对栈操作有m个字符，则有种情况，因此我们采用递归函数来进行计算



利用递归对每一对空遍历所有状态，由此完成对一般PDA转CFG的一般情况的讨论，若state大小等于1，则直接插入即可，不需要对Q中所有状态进行遍历。