编译原理上机报告

题目：由正则表达式转NFA在转为DFA最后最小化形成最小化DFA

思路：本题一共可以分为五个部分来实现

1. 由正则式转化为NFA
2. 由NFA转化为DFA
3. 由DFA转化为最小DFA

第一部分又可以分为下面两个部分：

1. 添加连接字符

在实现这一部分的时候要注意什么时候加连接付&什么时候不加，经过多次试验我发现当前字符和接下来的字符满足下列条件时要加连接符。

currentChar != '(' && currentChar != '|' && nextChar != '\*' && nextChar != '|' && nextChar != ')'

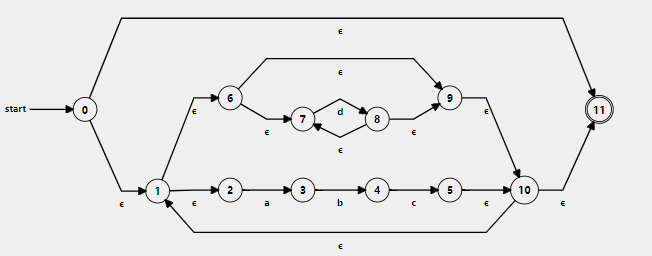
1. 由中缀表达式转化为后缀表达式

这一部分数据结构课设的时候做过，中缀转后缀最重要的就是优先级的确立，经过我反复的试验推到，确定\*的优先级最高，其次为|最后为&，这一部分比较坑，最初我以为\*是一元运算符，所以运算符最高，然后其他的两种应该是一样的，但过不了某些例子，所以才确定优先级

1. 由汤普孙算法构造出这个图，然后在利用后缀表达式的计算得到NFA,这一部分要注意，其实这些转换过程国外早有对应的算法，而且实现的方式不止一种，所以我们要做的就是把别人的算法实现一遍，我认为一个本科生自己可能也想不出一种新的算法出来，我们能做的就是学会别人的算法，变成自己的东西

第二部分：

1. 求0号态元素的closure集，然后进行迭代，看他经过输入字符后变到哪一个集合，然后检测这个集合是否在记录D中，如果在，就不用加入到D中，只需要在二维表中记录就可以了，这里注意如果某一个集合经过字符x变换到[],则不应该记录到D中，但是要记录在二维表中
2. 这部分的数据结构是用一个队列来时先遍历的，其实就是数据结构里学的队列实现层次遍历，不然使用递归的话代码可能不好写
3. 需要注意的部分：这一部分中秋closure集是使用递归的方法，也就是说必须避免成环的情况，如下面这种状态一定要及时判别，不然递归到死



这里0-1-6-9-10-1形成了一个环

第三部分：

使用myhill nerode theorem这个算法来实现我觉得比课堂上讲的方法简单，具体这个实现的方法我会给一份参考资料，在这里不做细说，这是一种填表法，能够快速找到需要合并的状态，但是一定要注意如果集合中一个是[]，一个有元素的话这种情况是不能合并的。

总结：经过这次实习我觉得不管实习有多难，只要坚持写总是可以解决问题的，如果遇到不会的问题一定要和老师同学交流，尽量不要卡在某一个地方太久。另外我有一个建议，就是实习的时候应该尽量鼓励同学们使用新的方法解决问题，如果大家都使用一个方法解决问题的话，可能限制了解决方案的多样性，另外最重要经过本次实习我认识到学好英语是多么重要啊，国外的很多资料真的要比国内好很多。