因为我讲的是3.2第一题，所以首先把正则转NFA的过程捋一下。

参考资料 中国大学MOOC 形式语言与自动机理论 视频3.2.2 从7分10秒开始观看、

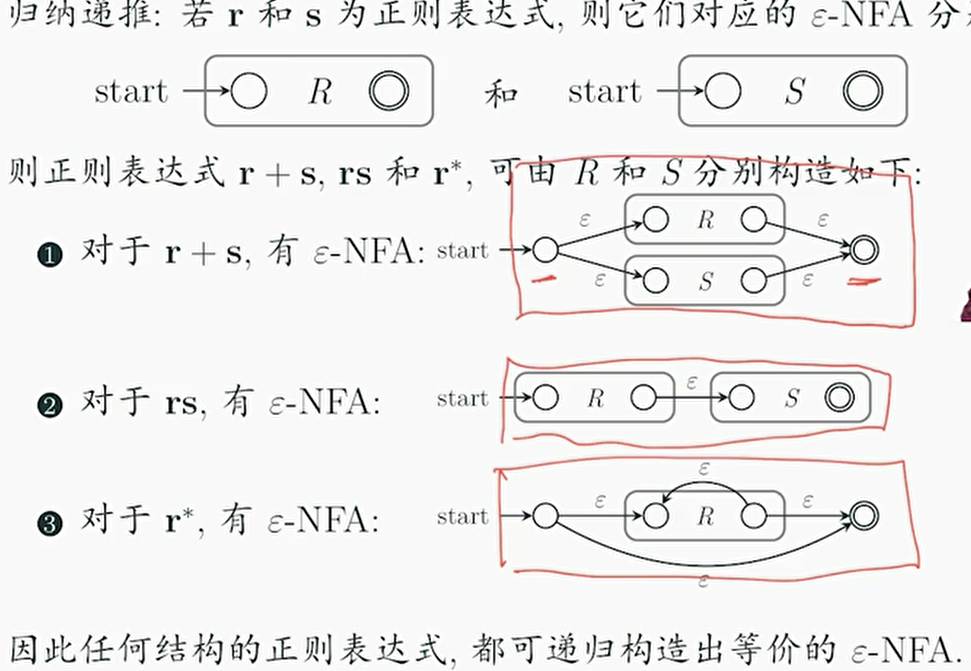
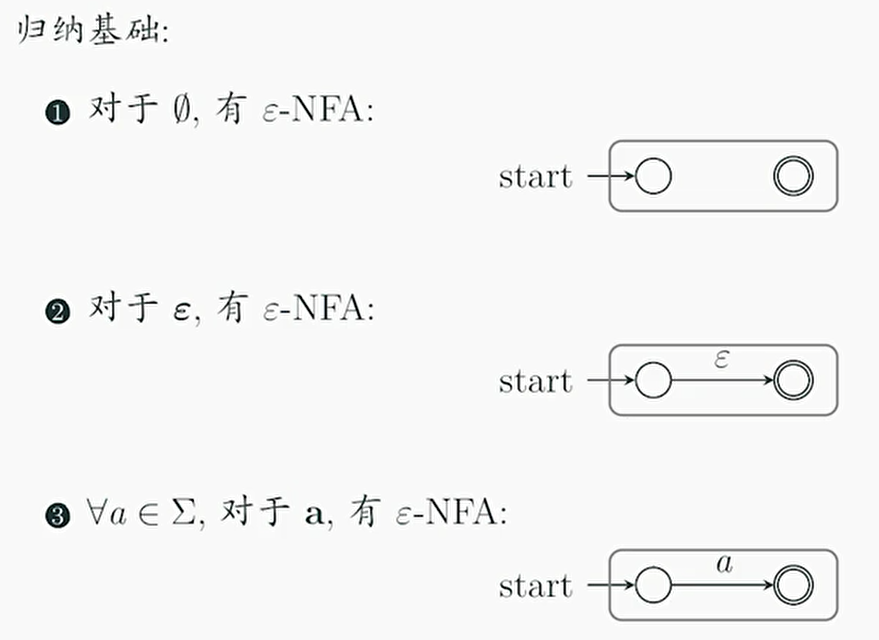
把证明去看一下，跟老师探讨一下，空串是否有必要、三个条件的问题（周二去）

语法分析树的最左最右唯一、歧义指的是啥

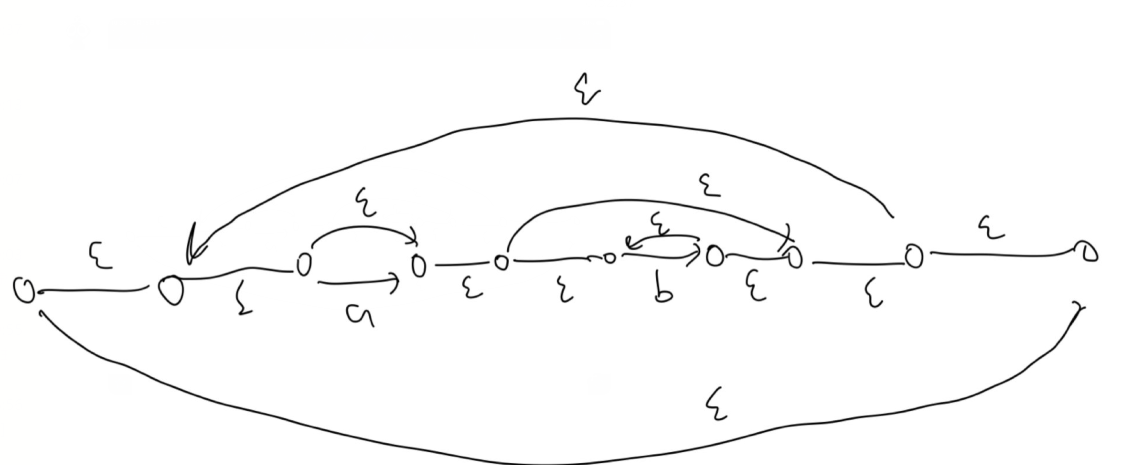


关于三个条件的详细阐述，可以回看一下mooc

但是关于



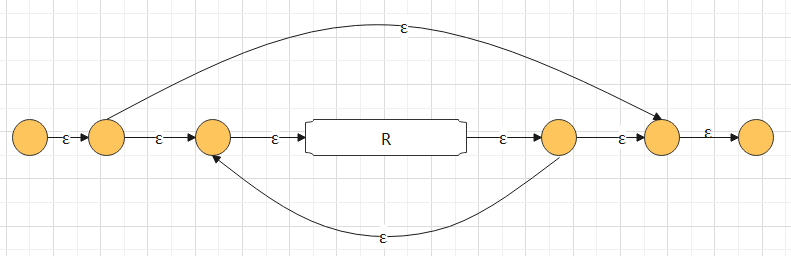
自顶向下地逐渐分解展开，和我们的第四章自顶向下分析有点类似



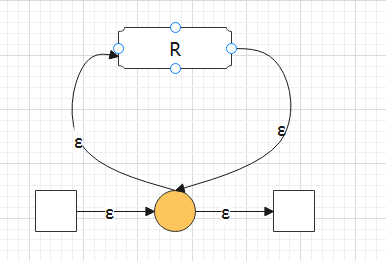
其实没有那么多的空转移也可以的，不影响结果

而在陈老师上课的时候给出了一个例题，除了空转移没有很多之外，有一个和形式语言与自动机课程不同的地方值得关注

我们主要关注关于闭包的构造方法

zx

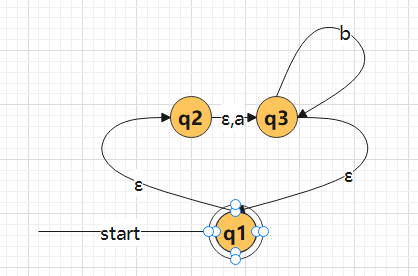
形式语言与自动机



编译原理

形式语言那个定义递归结构非常明显，适合计算机实现

编译原理这个更简洁一些



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | ε | Eclose() |
| Q1 | ∅ | ∅ | {q2} | {q1,q2,q3} |
| Q2 | {q3} | ∅ | {q3} | {q1,q2,q3} |
| Q3 | ∅ | {q3} | {q1} | {q1,q2,q3} |

这道题不适合作为解题演示，因为Eclose()集把所有状态全包含了

如果去年讲的忘了，推荐去看中国大学mooc形式语言与自动机的第三章第二节子集构造法的例子，这次我们几位同学讲的习题3.2后面几道题使用那里讲的子集构造法会解题更加清晰流畅，

学习编译原理的时候可以把去年形式语言学的东西拿出来，对比一下，其实有很多东西不太一样，比如这里正则转NFA的过程，首先编译原理提供了一个新的方法处理闭包，而且学习了编译原理课自顶向下分解的思想的话，可以以这种方法去理解转化的过程，对程序实现突然也有了头绪，但是去年讲这里的时候我肯定不会实现这个东西的。去年形式语言则对这个过程有着严格的形式化定义，比如这个过程的归纳基础和归纳递推，还讲了这个NFA的三个特性和必须遵守的原因。

编译原理的自动机为编译器设计服务，内容精炼，而且和我们所学的东西衔接密切，可以在编译原理对它的应用里更好地理解他们的原理，自动机讲的很严格，理论不止服务编译器，内容也很丰富，成系统，不少知识编译原理用不上。

我看了第四章，第四章自顶向下分析那块这种感觉尤其强烈，学习那块可以把去年形式语言的第五章看一看，尤其是文法的化简与范式，两门课文法这里的内容不互相包含，比如形式语言四种文法分类没有讲，但是那里讲的各种内容对编译原理非常有帮助，可以考虑把去年的讲义拿出来当做辅助资料学习编译原理。

