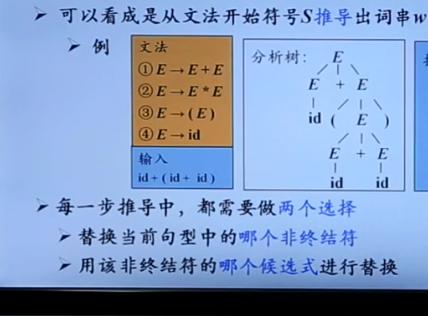
top-down parsing， 从符号S推导出词串w的过程



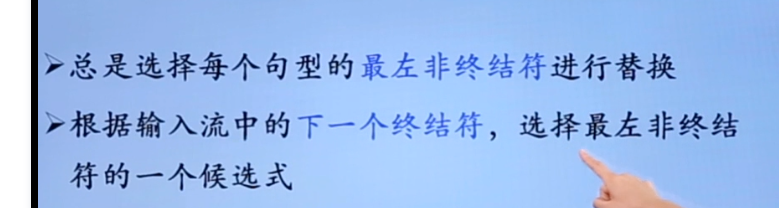
从E开始一步一步推到ID+(ID+ID)

每一次推导我们需要考虑两个问题

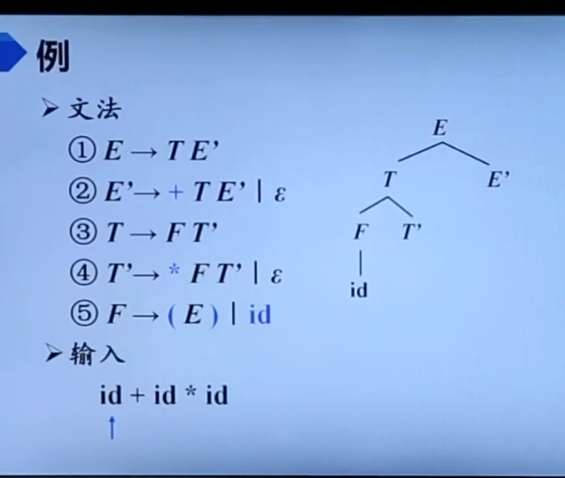
1.我们使用哪个non-teriminal进行替换

2.我们使用这个Non-terminal的哪条规则

LEFT-MOST,永远取最左的



选择候选式：根据下一个terminal

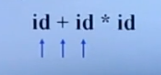
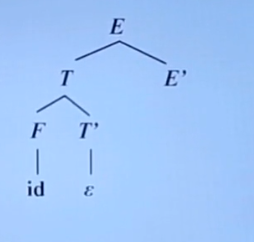


例如这里

E开头，我们按照最左规则，一直可以推到T->FT'

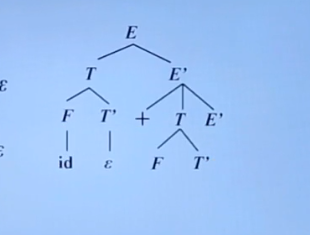
这时输入是id，

因此我们使用5中的第二候选式

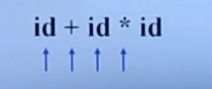
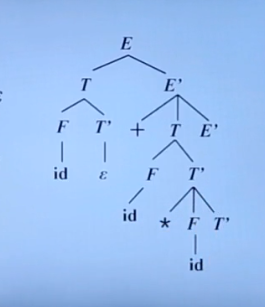


然后下一个符号是+号

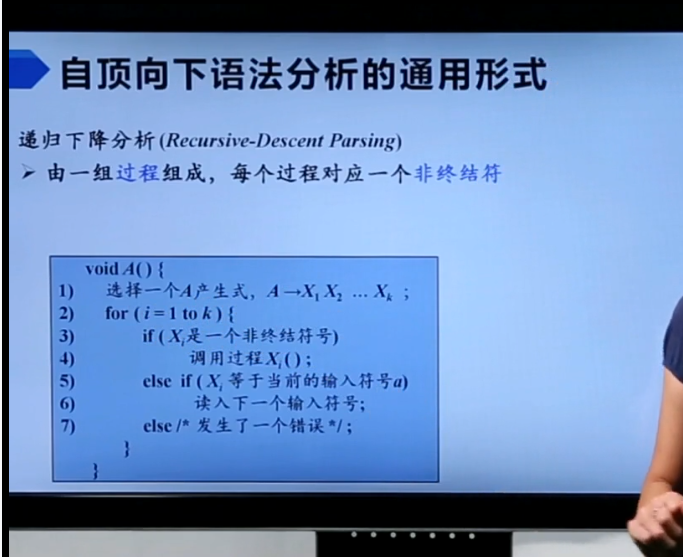
我们要处理T'， 没有对应的，所以我们只能把T'替换成空

然后处理E'，他有加号，所以是+TE'  


F处理成id， T'处理成\*FT'

.3

\*后面是id，然后我们就无所谓了，因为全都转化完毕，T'与E'都是空串



过程：derivation

一个derivation对应一个非终结符

例如A的function

首先选择一个产生式

如果Xi是一个非终结符号 //想想树，Xi就是展开以后左下角第一个

那么展开这个xi //就是继续展开，也就是recursive call

如果是一个terminal

载入下一个输入符号 //就是假设有一个变量为当前符号，这时原来的就可以翻篇了

如果是不同的terminal就是·return error，说明这个树不行

这个有一个限制

A->BC

A->CD 我们选哪一个呢，因此我们引入新规则，把他转换

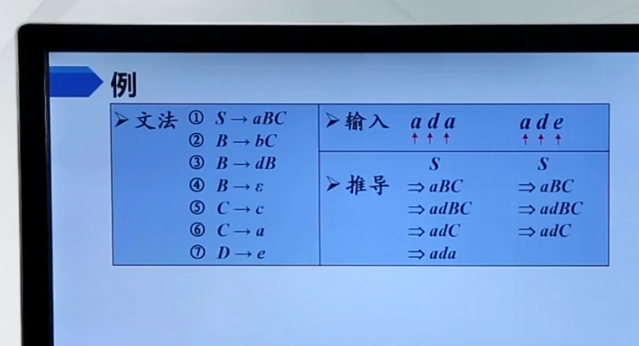
A->Bc|BD|BE

转换成A->BA'

A'->C|D|E

总之一次只能check一个符号

选空串的问题



ada选空串是可以成功的，

但是ade

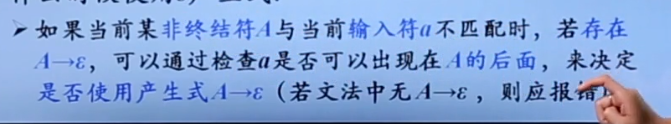
S->aBC

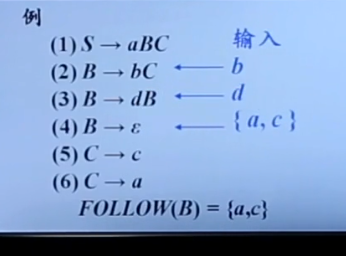
有d，所以选aBC->adBC, 然后B空串adC,怎么也推不出ade

什么时候选空串呢

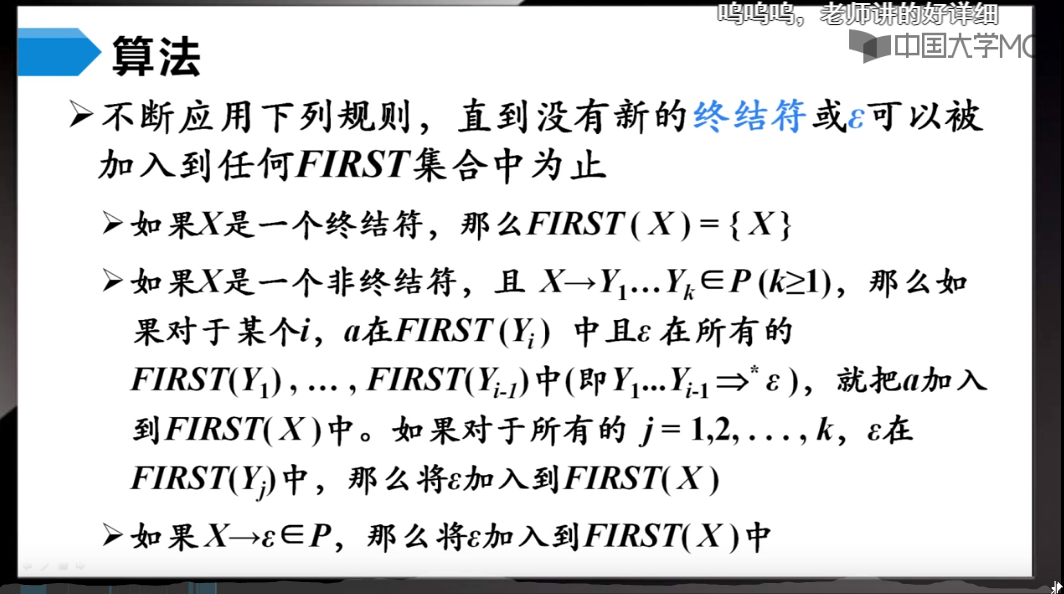
B与a不匹配，但是紧跟着B的是大C也就是c或a,可以空串

而e不是c或a，不能空串，报错



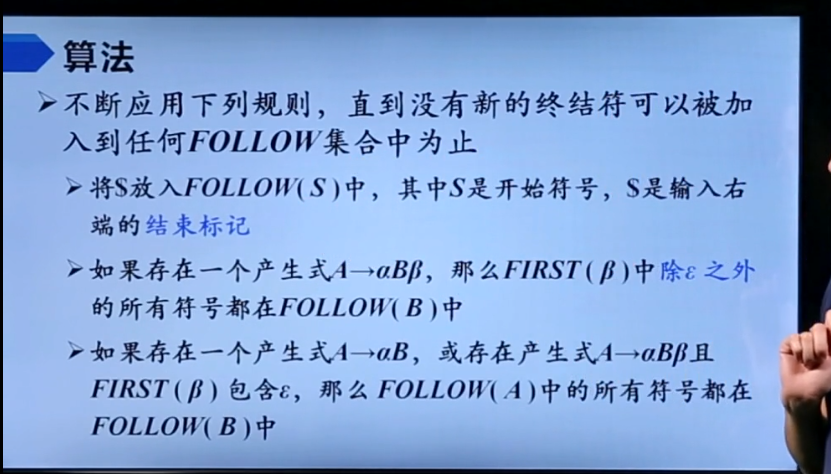


因此因此如果输入b，选2，如果输入d，选3，如果follow是a或c，选4



如果Y1能推倒出空串

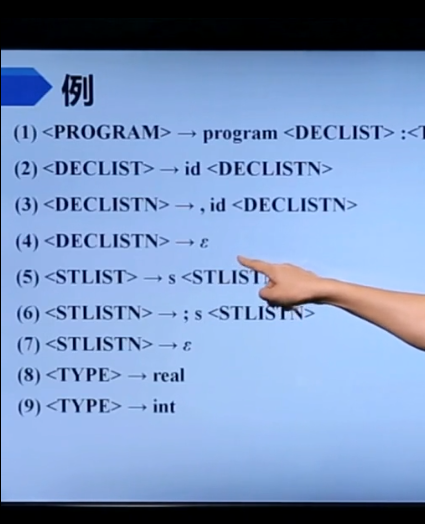
那么Y2的terminal也加入X FIRST



dollar符号是结束标记

你可以看做是一个树

S那一行只有一个S，所以要添加dollar(更多的是为了应用规则，这样可以直接推导出那些有dollar)



注意4和7

如果输入的terminal例如小写a

我们推到了4与7的地步

就可以看看4|7左侧的follow集也就是declistn的follow集