序列化方式

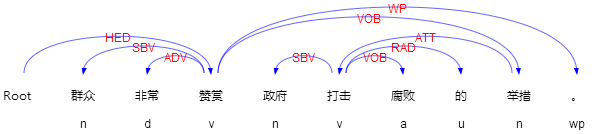
1.依存符号插入该节点（及其子节点）和其父节点之间。

2.从根结点开始，对树进行遍历。遍历到一个结点时，就把它和它的子树用括号括起来。

3.在最终的序列化结果里去掉了ROOT（因为有没有意义不大）。

例如：

群众非常赞赏政府打击腐败的举措。



HED ( 群众 SBV 非常 ADV 赞赏 VOB ( ( 政府 SBV 打击 VOB 腐败 RAD 的 ) ATT 举措 ) WP 。 )

对于赞赏而言，它和它的子树是整个句子。故

HED ( 群众 SBV 非常 ADV 赞赏 VOB ( ( 政府 SBV 打击 VOB 腐败 RAD 的 ) ATT 举措 ) WP 。 )

赞赏的子结点有4个，群众，非常，wp，举措。但前三个都是叶子结点，故不做操作。“举措”不是叶子结点，故把“举措”及其子树括起来：

HED ( 群众 SBV 非常 ADV 赞赏 VOB ( ( 政府 SBV 打击 VOB 腐败 RAD 的 ) ATT 举措 ) WP 。 )

举措的子结点只有一个，打击。打击不是叶子节点，故把它和它的子树括起来：

HED ( 群众 SBV 非常 ADV 赞赏 VOB ( ( 政府 SBV 打击 VOB 腐败 RAD 的 ) ATT 举措 ) WP 。 )

语法规则：

整个大的框架是：

'final\_sentence : SYN\_HED sentence'

'sentence : LEFT sentence\_content SYN\_WP POS\_WP RIGHT'

'sentence\_content : subject'

'sentence\_content : subject SYN\_SBV predicate'

'sentence\_content : subject SYN\_SBV predicate SYN\_VOB object'

'''subject : nphrase

| vphrase

| vobphrase

| aphrase

| sbv\_vob\_phrase'''

'''predicate : vphrase

| aphrase'''

'''object : nphrase

| aphrase

| vphrase

| vobphrase

| sbvphrase

| sbv\_vob\_phrase'''

最大的问题：移入-规约问题

yacc是lr（1）文法的。而序列化文本很难做到lr（1）。

比如对于下面的规则：

**rqphrase : POS\_R SYN\_ATT POS\_Q //POS\_R是代词，POS\_Q是量词。例：“这把”**

**attphrase : rqphrase**

**attphrase : POS\_R**

**npharase : attphrase SYN\_ATT nphrase**

**nphrase : POS\_N**

对于“我们国家”这个词，我们希望规约得到一个nphrase

“我们国家”的序列化结果是：

**POS\_R SYN\_ATT POS\_N**

我们希望的情景是：

**首先**，移入POS\_R。根据**attphrase : POS\_R**规约，得到attphrase。

**接下来**，经过若干步骤，根据**npharase : attphrase SYN\_ATT nphrase** 得到nphrase。

**但实际情况是：**

移入POS\_R之后，yacc有两个选择，

按照 **attphrase : POS\_R** 规约，

或者按照**rqphrase : POS\_R SYN\_ATT POS\_Q**继续移入。

而yacc默认是继续移入。

这么一来，按照规则**rqphrase : POS\_R SYN\_ATT POS\_Q** ，是不可能得到我们想要的结果的，因而报错。

采取新的序列化方式，也是因为遇到这个问题。（比如下面的例子）。采取新的序列化方式之后，解决了一部分问题，然而还有很多此类移入-规约问题没有得到解决。

在用原来的序列化方式时，加入两条规则：

'vphrase : vphrase vphrase SYN\_CMP'

'vphrase : LEFT vphrase RIGHT’

在分析下面的句子时会报错，”syntax error”：

( 群众 SBV 非常 ADV 赞赏 ( ( 政府 SBV 打击 腐败 VOB 的 RAD ) ATT 举措 ) VOB )HED

然而上面这个句子中都没有出现CMP，为何会报错？ply会生成一个parse.out的文件，记录了语法分析状态机的状态。在对照句子，仔细查看parse.tab之后，我找到了答案：

“赞赏”是一个POS\_V，在这里会被规约为vphrase。遇到左括号时，可以根据上面两条规则进行移入，又或者根据sbvphrase : nphrase SYN\_SBV vphrase进行规约。此时会产生一个移入-规约冲突。ply对这种问题的处理方式是：选择移入。而本来我们期待它在这里，是根据

sbvphrase : nphrase SYN\_SBV vphrase 进行规约的。

采取新的序列化方式之后，规则变成了：

vphrase : vphrase SYN\_CMP vphrase ，就不存在这个问题了。

目前能识别的句子：

**简单句**

1.只含有主语的句子。

例如：

多么漂亮的蝴蝶！

2.只含有主语谓语的句子。

例如：

我们不断前进。

3.含有主语谓语宾语的句子。

例如：

这是一本好书。

１９９７年，是中国发展历史上非常重要的很不平凡的一年。

实现祖国的完全统一，是海内外全体中国人的共同心愿。

4.把字句被字句

例如：

我们把老虎打死了。

**复合句**

1.主语不同的并列句

例如：

农业生产再次获得好的收成，企业改革继续深化，人民生活进一步改善。