

在Earley句法分析中增加合一约束

詹卫东

<http://ccl.pku.edu.cn/doubtfire>

冯志伟、孙乐译《自然语言处理综论》电子工业出版社2005年版。第11章。

Daniel Jurafsky & James H. Martin, 2006, *Speech and Language Processing*, Pearson Education, Inc., Prentice Hall.
Chapter 16. Features and Unification

对Earley算法进行改进

- (1) 根据重写规后所附的合一约束为一个节点生成特征结构
- (2) 对chart中的状态进行改进
- (3) 对Predicator, Scanner, Completer等操作进行改进
- (4) 生成新状态时对特征结构的包孕关系进行检查

带合一约束的CFG规则

- 将重写规则中所附的约束转写为节点的特征结构

$S \rightarrow NP VP$

$\langle NP \text{ HEAD AGREEMENT} \rangle = \langle VP \text{ HEAD AGREEMENT} \rangle$

$\langle S \text{ HEAD} \rangle = \langle VP \text{ HEAD} \rangle$

$$\left[\begin{array}{l} S \quad [\text{HEAD} \quad \boxed{1}] \\ NP \quad [\text{HEAD} \quad [\text{AGREEMENT} \quad \boxed{2}]] \\ VP \quad [\text{HEAD} \quad \boxed{1} [\text{AGREEMENT} \quad \boxed{2}]] \end{array} \right]$$

- 特征结构可以实现为有向无环图(DAG)

为Earley分析法中的状态 增加特征结构字段

- 改进后的状态包含4部分：
 - (1) 上下文无关文法规则，代表分析子树
 - (2) 点标记 \cdot ，表示子树分析的完成状况
 - (3) 状态对应的起止位置， $[i,j]$
 - (4) 特征结构

$$(1) \quad \underline{S \rightarrow \cdot NP VP} \quad , \quad \underline{[0,0]} \quad , \quad \underline{DAG}$$

(2) (3) (4)

Predictor操作

- 每当一个状态被Predictor操作加入状态表时，该规则对应的特征结构也作为状态的一个字段加入。

Predictor: 对于状态 $Z \rightarrow \alpha \cdot X \beta \ [j, k] \ \text{DAG}_Z$ 其中X是非终结符
对于语法中每条形如 $X \rightarrow \gamma \ \text{DAG}_X$ 的规则，都可以形成一个新状态： $X \rightarrow \cdot \gamma \ [k, k] \ \text{DAG}_X$

Scanner操作

Scanner: 对于状态 $Z \rightarrow \alpha \cdot X \beta \ [j, k] \text{ DAG}_Z$ 其中 X 是终结符
如果 X 与输入字符串中第 k 个字符匹配，就将词典中
 X 的特征结构 DAG_x 跟 DAG_Z 合一，若成功，则形成
一个新状态：

$$Z \rightarrow \alpha X \cdot \beta \ [j, k+1] \text{ New-DAG}$$

否则，不改变当前状态集。

Completer操作

Completer : 对于一个已经“完成”的状态 $Z \rightarrow \gamma \cdot [j, k] \text{ DAG}_z$

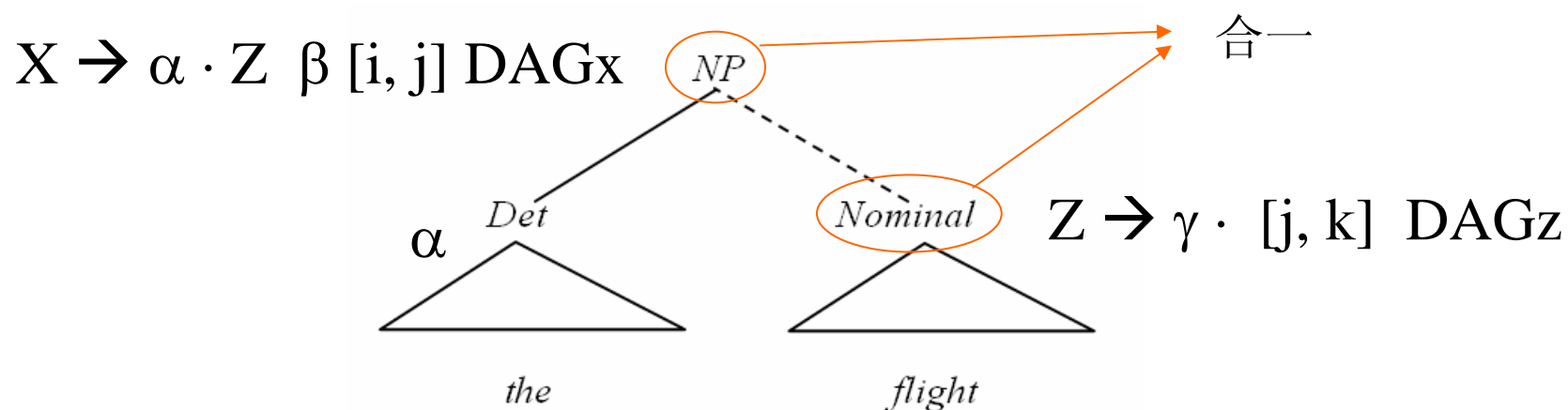
如果已有状态集合中有形如 $X \rightarrow \alpha \cdot Z \beta [i, j] \text{ DAG}_x$ 这样的状态，就将 DAG_z 跟 DAG_x 进行合一运算，二者合一的结果（记作 **New-DAG**）若为成功，则形成一个新状态：

$$X \rightarrow \alpha Z \cdot \beta [i, k] \text{ New-DAG}$$

否则，不改变当前状态集。

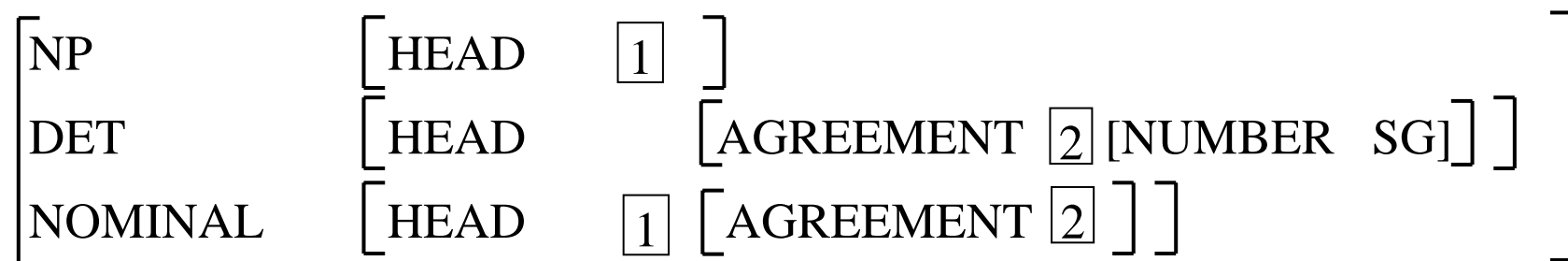
Completer操作

- 每当一个子树(s_1)完成分析时，触发Completer操作
- 检查当前状态集中是否有子树(s_2)等待子树(s_1)来完成分析
- 将 s_1 对应的特征结构与 s_2 对应的特征结构进行合一
- 若合一失败，不产生新的状态
- 若合一成功，则产生新状态，并把合一结果作为新状态的特征结构

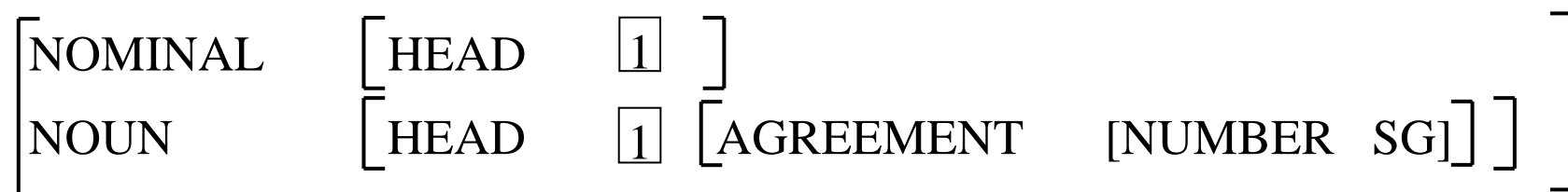


Completer示例

- $NP \rightarrow Det \cdot Nominal, [0,1], DAG_1$



- $Nominal \rightarrow Noun \cdot, [1,2], DAG_2$



- 将 DAG_1 中的 $Nominal$ 同 DAG_2 中的 $Nominal$ 进行合一

Completer操作中检查DAG包孕关系

- 为了保证不重复分析子树，在Earley算法中，如果新产生的状态和状态集合中的某个状态相同，新状态将不被加入状态集
- 在基于合一的Earley算法中，同时还要检查状态所附的特征结构是否具有包孕关系（subsumption），若新状态的特征结构被状态集中的状态的特征结构包孕（subsumed），则不被加入状态集。

例如： $NP \rightarrow \cdot Det NP, [i,i], DAG$

若状态集中的已有状态对 Det 没有约束，而新产生的状态要求 Det 必须是单数的，则新状态不必加入。

基于合一的Earley分析算法

设输入字符串长度为 n , 字符间隔可记做 $0,1,2,\dots,n$

(1) 将语法规则中形如 $S \rightarrow \alpha$ DAG_s 的规则形成为状态:

$\langle S \rightarrow \cdot \alpha \ [0, 0] \text{DAG}_s \rangle$ 加入到状态集合中 (种子状态/seed state)

(2) 对当前分析句子的每个词, 依次进行循环:

对状态集中的每个状态, 依次进行循环:

i) 如果当前状态是[未完成状态], 且点后不是终结符, 则

执行Predicator;

ii) 如果当前状态是[未完成状态], 且点后是终结符, 则

执行Scanner;

iii) 如果当前状态是[完成状态], 则

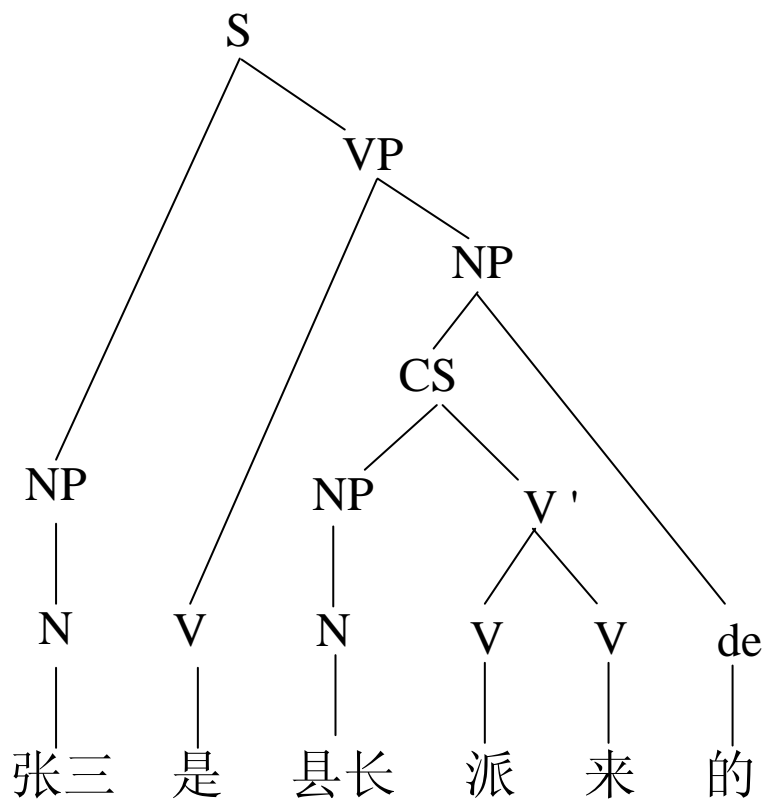
执行Completer;

(3) 如果最后得到形如 $\langle S \rightarrow \alpha \cdot \ [0, n] \text{DAG}_s \rangle$ 这样的状态, 那么输入字符串被接受为合法的句子, 否则分析失败

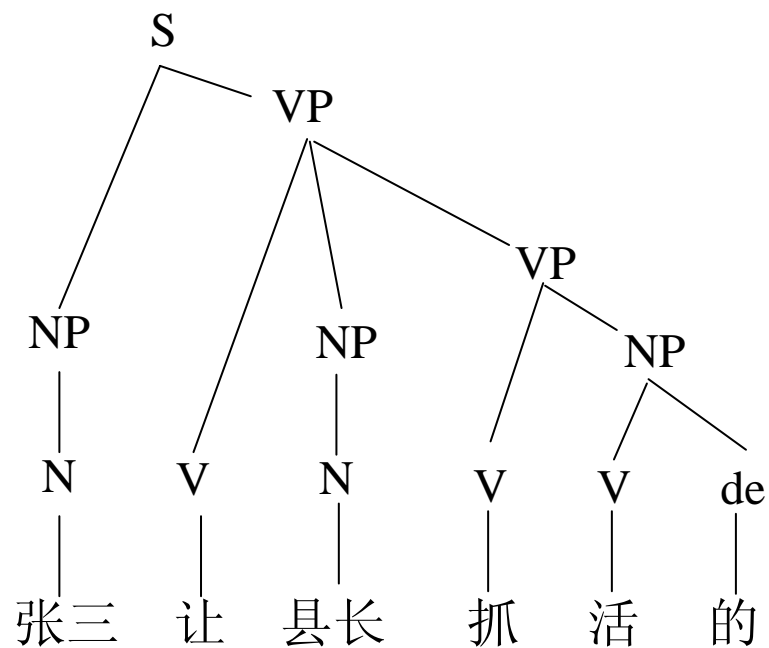
基于合一的Earley分析法示例

- (1) $S \rightarrow NP \ VP$
- (2) $NP \rightarrow N$
- (3) $NP \rightarrow CS \text{ 的}$
- (4) $CS \rightarrow NP \ V'$
- (5) $VP \rightarrow V \ NP$
- (6) $V' \rightarrow V \ V$

- (7) $NP \rightarrow V \text{ 的}$
- (8) $VP \rightarrow V \ NP \ VP$



(例1)

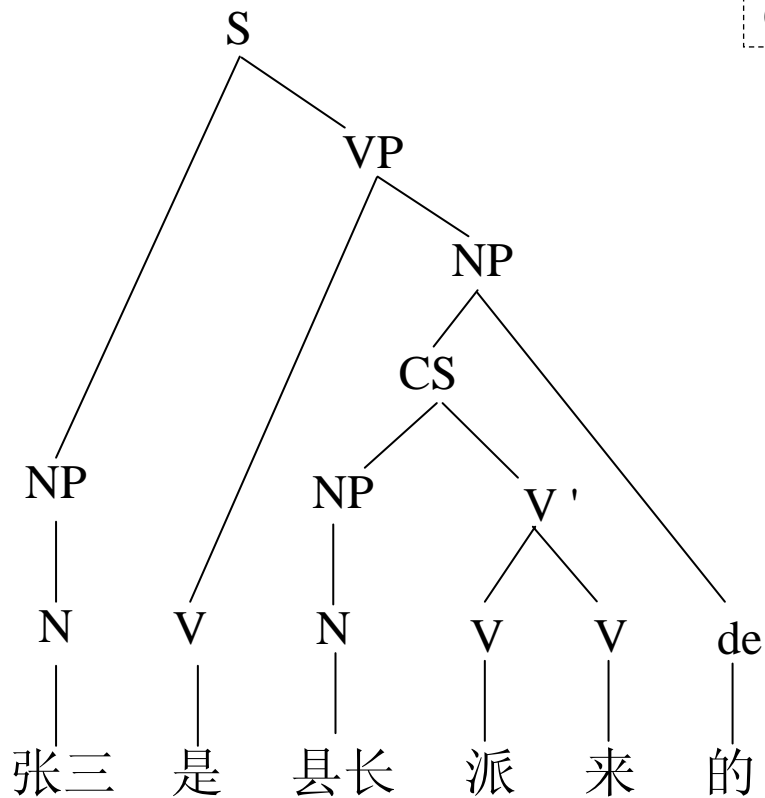


(例2)

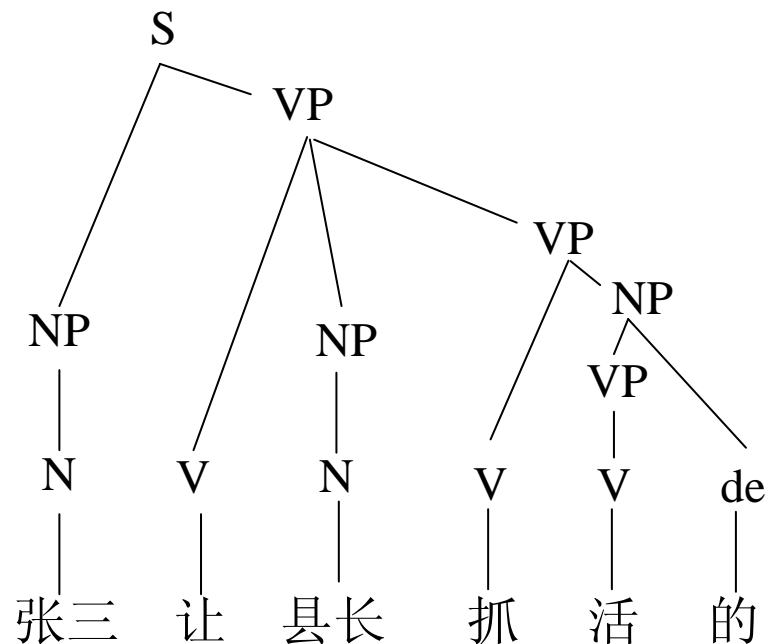
基于合一的Earley分析法示例

- ① 是 V [subcat:a]
- ② 派 V [subcat:b]
- ③ 来 V [subcat:c]
- ④ 让 V [subcat:b]
- ⑤ 抓 V [subcat:a]
- ⑥ 活 V [subcat:d]

- (1) $S \rightarrow NP \ VP$
- (2) $NP \rightarrow N$
- (3) $NP \rightarrow CS \ 的$
- (4) $CS \rightarrow NP \ V'$
- (5) $VP \rightarrow V \ NP :: \%V.subcat=a$
- (6) $V' \rightarrow V \ V :: \% \%V.subcat=c$
- (7) $NP \rightarrow V \ 的$
- (8) $VP \rightarrow V \ NP \ VP :: \%V.subcat=b$



(例1)



(例2)

6	$S \rightarrow NP VP \cdot$	$VP \rightarrow V NP \cdot \text{dag}_{①-(5)}$	$NP \rightarrow CS \text{ 的} \cdot$				归约 扫描
5			$NP \rightarrow CS \cdot \text{ 的}$ $CS \rightarrow NP V' \cdot$	$V' \rightarrow V V \cdot \text{dag}_{②-(6)-③}$			归约 扫描
4				$V' \rightarrow V \cdot V \text{ dag}_{②-(6)}$			扫描
3	$S \rightarrow NP VP \cdot$	$VP \rightarrow V NP \cdot \text{dag}_{①-(5)}$	$CS \rightarrow NP \cdot V'$ $NP \rightarrow N \cdot$	$V' \rightarrow \cdot V V \text{ dag}_{(6)}$			预测 归约 扫描
2		$VP \rightarrow V \cdot NP VP \text{ dag}_{①-(8)}$ $VP \rightarrow V \cdot NP \text{ dag}_{①-(5)}$ $V' \rightarrow V \cdot V \text{ dag}_{①-(6)}$	$CS \rightarrow \cdot NP V'$ $NP \rightarrow \cdot V \text{ 的}$ $NP \rightarrow \cdot N$ $NP \rightarrow \cdot CS \text{ 的}$				预测 扫描
1	$S \rightarrow NP \cdot VP$ $CS \rightarrow NP \cdot V'$ $NP \rightarrow N \cdot$	$VP \rightarrow \cdot V NP VP \text{ dag}_{(8)}$ $VP \rightarrow \cdot V NP \text{ dag}_{(5)}$ $V' \rightarrow \cdot V V \text{ dag}_{(6)}$					预测 归约 扫描
0	$CS \rightarrow \cdot NP V'$ $NP \rightarrow \cdot V \text{ 的}$ $NP \rightarrow \cdot N$ $NP \rightarrow \cdot CS \text{ 的}$ $S \rightarrow \cdot NP VP$						预测 种子
	0	1	2	3	4	5	6

N
张三

V
是

N
县长

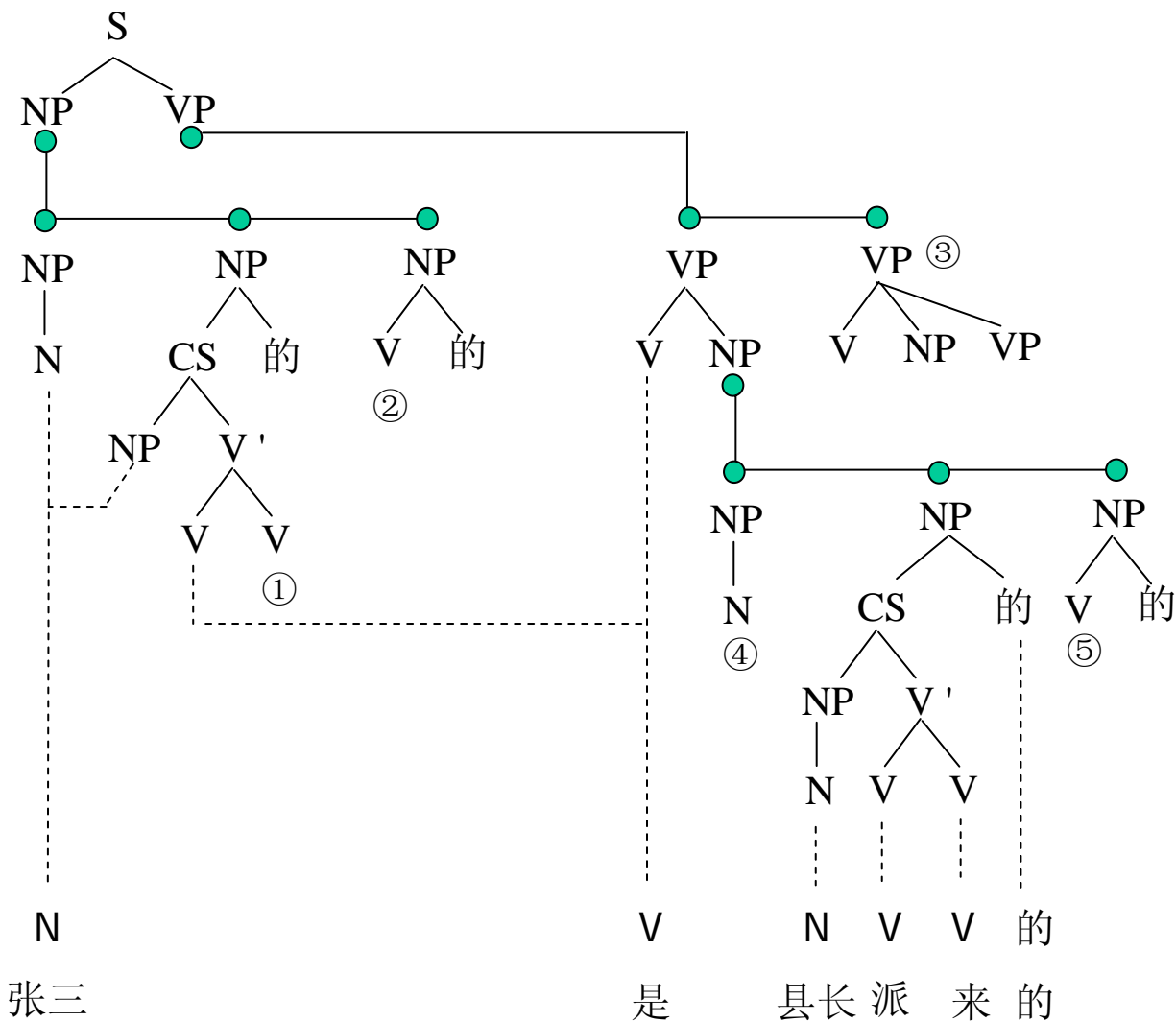
V
派

V
来

的
的

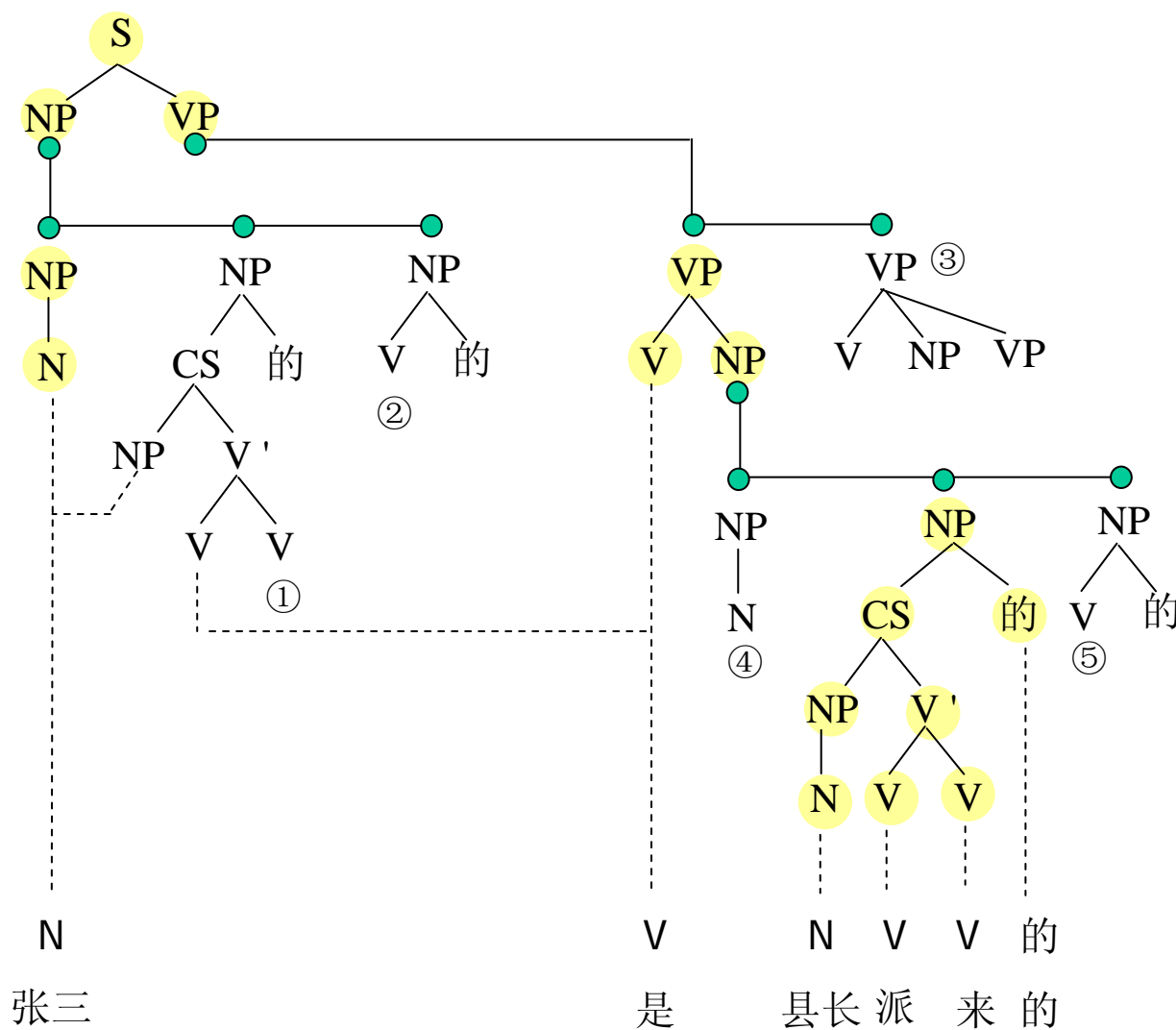
6	$S \rightarrow NP VP \cdot$	$VP \rightarrow V NP VP \cdot \text{dag}_{④-(8)}$		$VP \rightarrow V NP \cdot \text{dag}_{⑤-(5)}$ $VP \rightarrow V NP \cdot VP \text{dag}_{⑤-(8)}$	$CS \rightarrow NP \cdot V'$ $NP \rightarrow V \text{ 的} \cdot$		归约 扫描
5				$V' \rightarrow V V \cdot \text{dag}_{⑤-(6)-(6)}$	$NP \rightarrow V \cdot \text{ 的}$		扫描
4				$V' \rightarrow V \cdot V \text{dag}_{⑤-(6)}$ $VP \rightarrow V \cdot NP \text{dag}_{⑤-(5)}$ $VP \rightarrow V \cdot NP VP \text{dag}_{⑤-(8)}$	$CS \rightarrow \cdot NP V'$ $NP \rightarrow \cdot V \text{ 的}$ $NP \rightarrow \cdot N$ $NP \rightarrow \cdot CS \text{ 的}$		预测 扫描
3		$VP \rightarrow V NP \cdot VP \text{dag}_{④-(8)}$	$CS \rightarrow NP \cdot V'$ $NP \rightarrow N \cdot$	$V' \rightarrow \cdot V V \text{dag}_{(6)}$ $VP \rightarrow \cdot V NP \text{dag}_{(5)}$ $VP \rightarrow \cdot V NP VP \text{dag}_{(8)}$			预测 归约 扫描
2		$VP \rightarrow V \cdot NP VP \text{dag}_{④-(8)}$ $VP \rightarrow V \cdot NP \text{dag}_{④-(5)}$ $V' \rightarrow V \cdot V \text{dag}_{④-(6)}$	$CS \rightarrow \cdot NP V'$ $NP \rightarrow \cdot V \text{ 的}$ $NP \rightarrow \cdot N$ $NP \rightarrow \cdot CS \text{ 的}$				预测 扫描
1	$S \rightarrow NP \cdot VP$ $CS \rightarrow NP \cdot V'$ $NP \rightarrow N \cdot$	$VP \rightarrow \cdot V NP VP \text{dag}_{(8)}$ $VP \rightarrow \cdot V NP \text{dag}_{(5)}$ $V' \rightarrow \cdot V V \text{dag}_{(6)}$					预测 归约 扫描
0	$CS \rightarrow \cdot NP V'$ $NP \rightarrow \cdot V \text{ 的}$ $NP \rightarrow \cdot N$ $NP \rightarrow \cdot CS \text{ 的}$ $S \rightarrow \cdot NP VP$						预测 种子
	0	1	2	3	4	5	6
		N 张三	V 让	N 县长	V 抓	V 活	的 的

Earley算法构造分析树示意图（例1）



- ① 因为“是”的后面不是V，该节点“到此为止”。
- ② 起始位置不是“V”。
- ③ 该规则的约束要求V的子类为b，而“是”的子类特征值为a，合一失败。
- ④ N如果跟“县长”匹配成功，则分析完毕，但句子并未结束。
- ⑤ V跟“县长/N”不匹配

Earley算法构造分析树示意图（例1）



① 因为“是”的后面不是V，该节点“到此为止”。

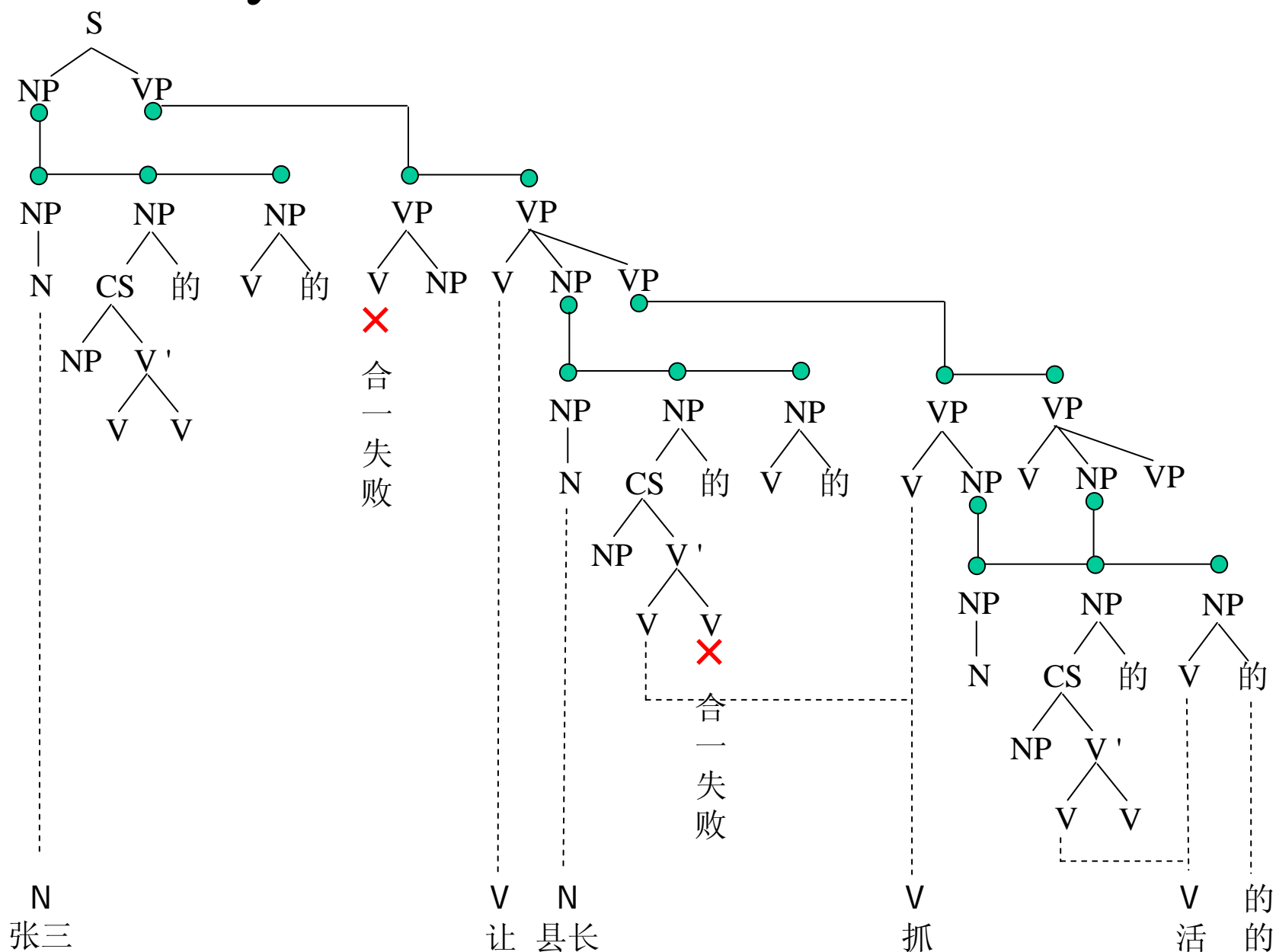
② 起始位置不是“V”。

③ 该规则的约束要求V的子类为b，而“是”的子类特征值为a，合一失败。

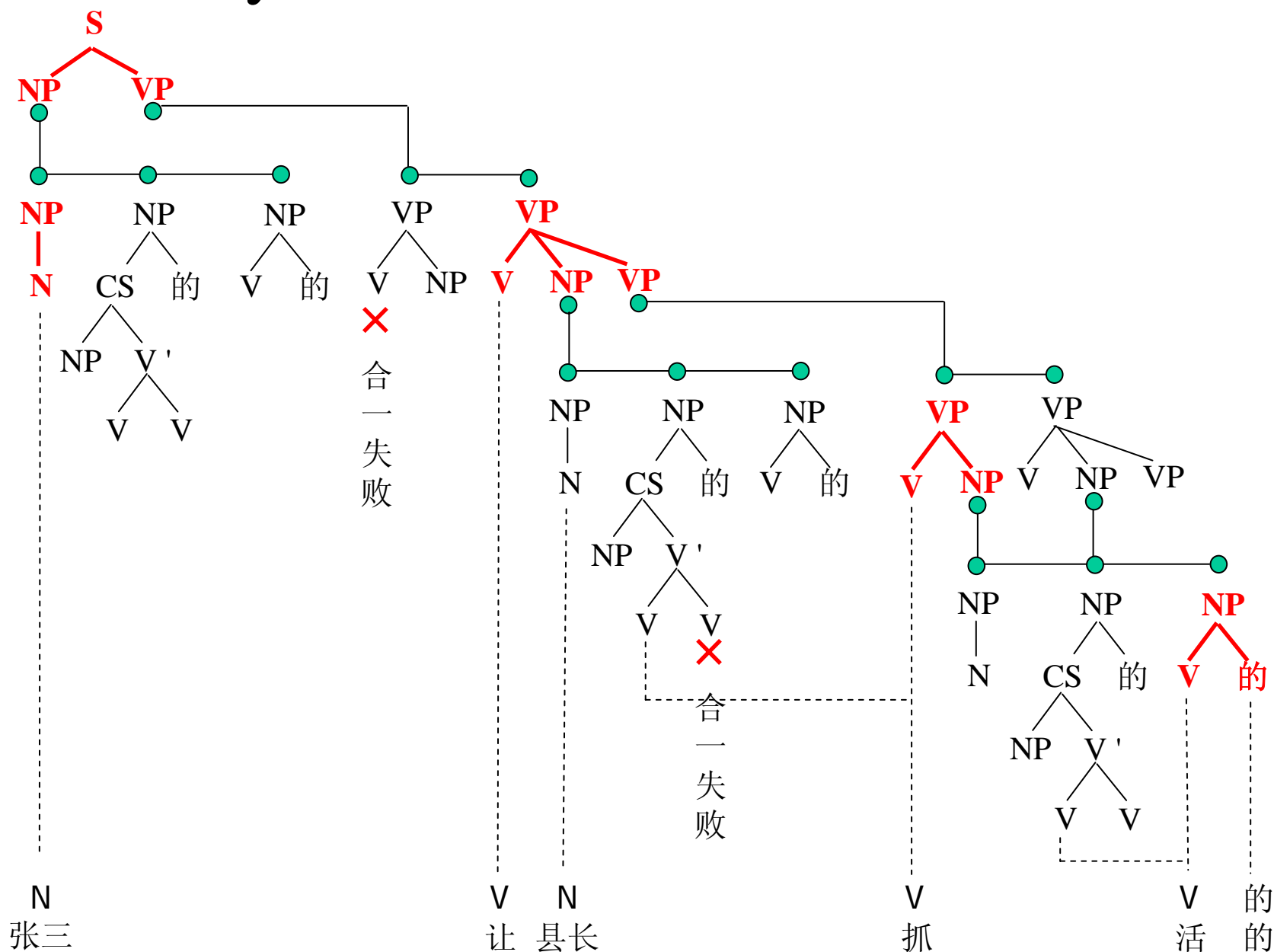
④ N如果跟“县长”匹配成功，则分析完毕，但句子并未结束。

⑤ V跟“县长/N”不匹配

Earley算法构造分析树示意图（例2）



Earley算法构造分析树示意图（例2）



练习

请构造带有合一描述的上下文无关语法规则集和词典，然后用Earley算法分析下面句子的结构：

1. 张三 让 李四 抓 活 的 回来
2. 张三 说 李四 是 要饭 的 装阔