1

第5章 句法分析(I)

目录

3

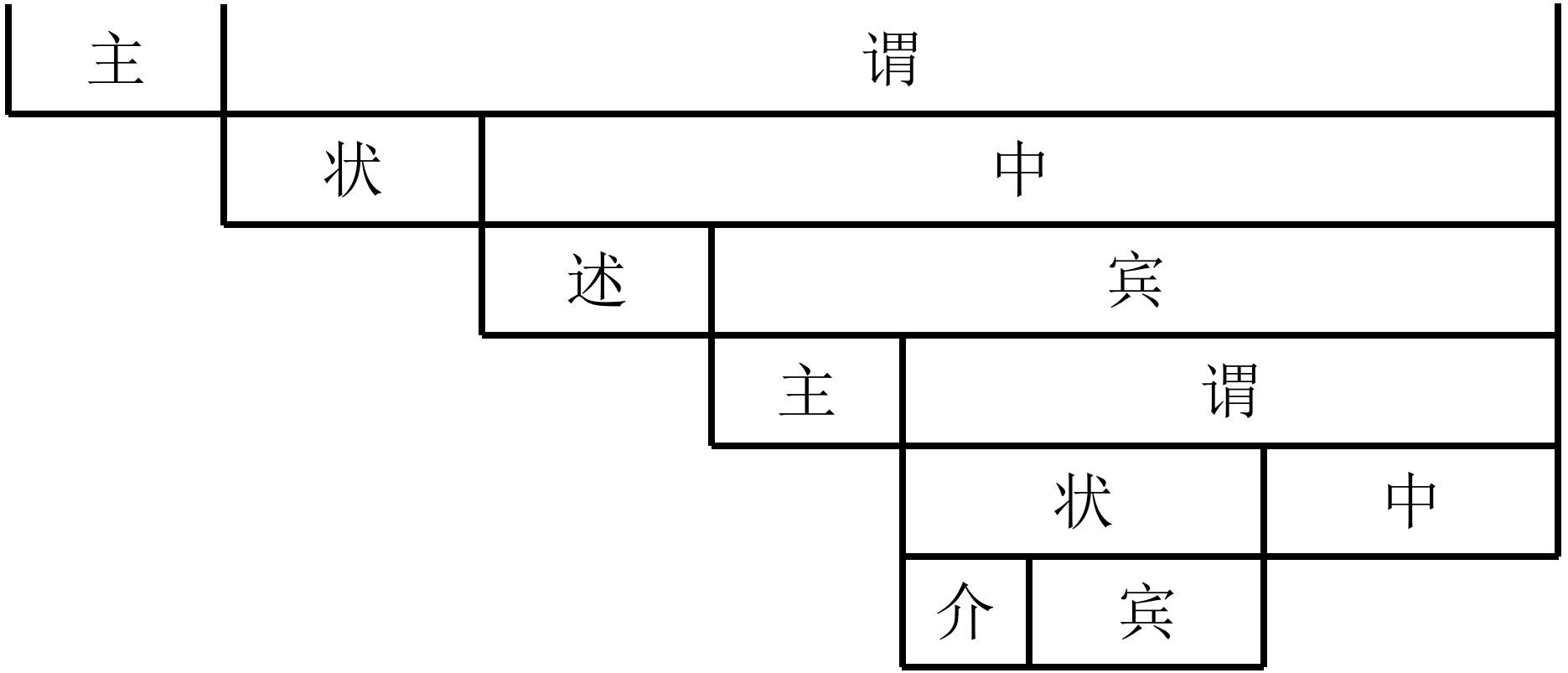
句法结构分析概述

句法分析(syntactic parsing)

- 识别句子的句法结构或句子中词汇之间的依存关系。
- 类型
  - 句法结构分析/短语结构分析
    - 分析出合乎语法的句子结构（句法分析树）
    - 完全句法分析：获得整个句子的语法结构
    - 局部句法分析：以获得局部成分为目标
  - 依存句法分析
    - 词与词之间的支配与被支配的关系
    - 需要分析识别句子中的“主谓宾”、“定状补”等语法成分

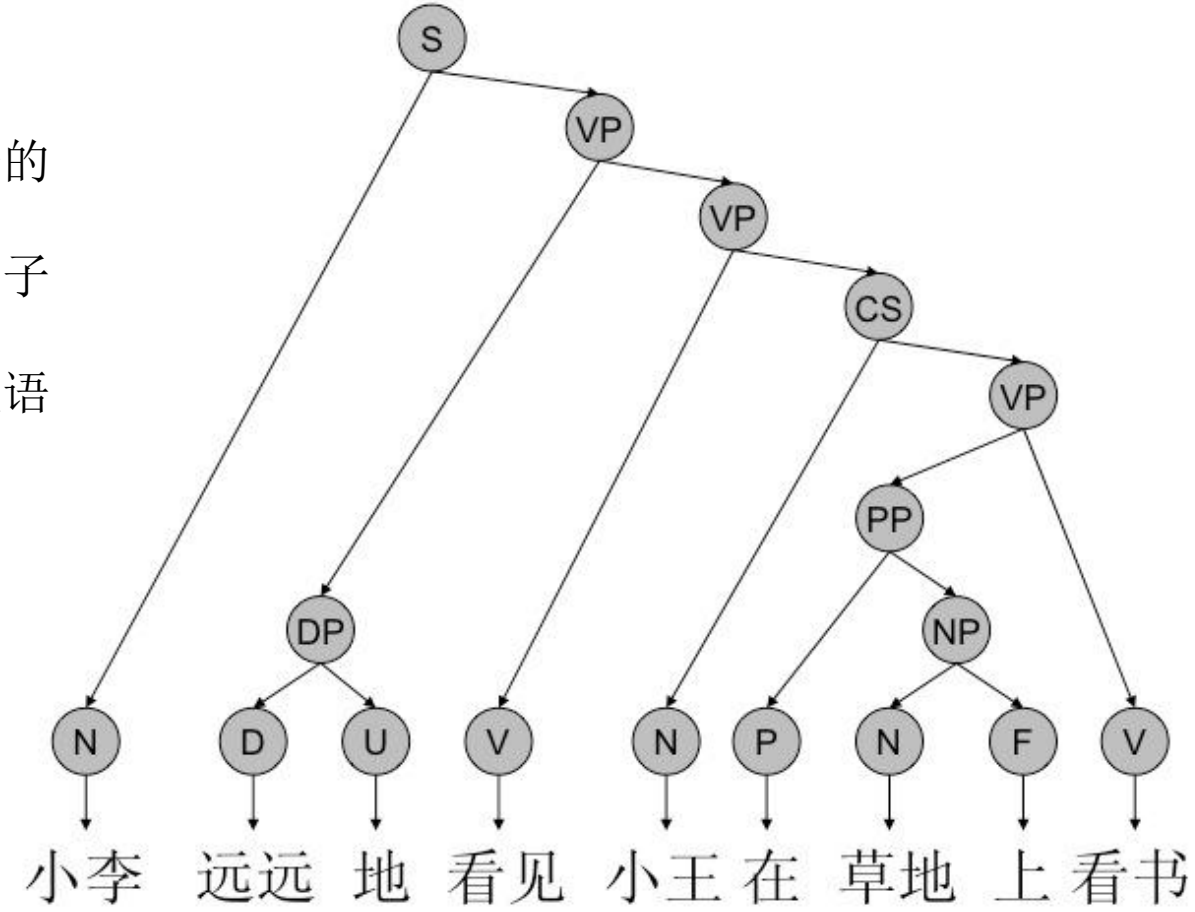
词语到句子的组合顺序

小李　远远地　看见　小王　在草地上　看书。

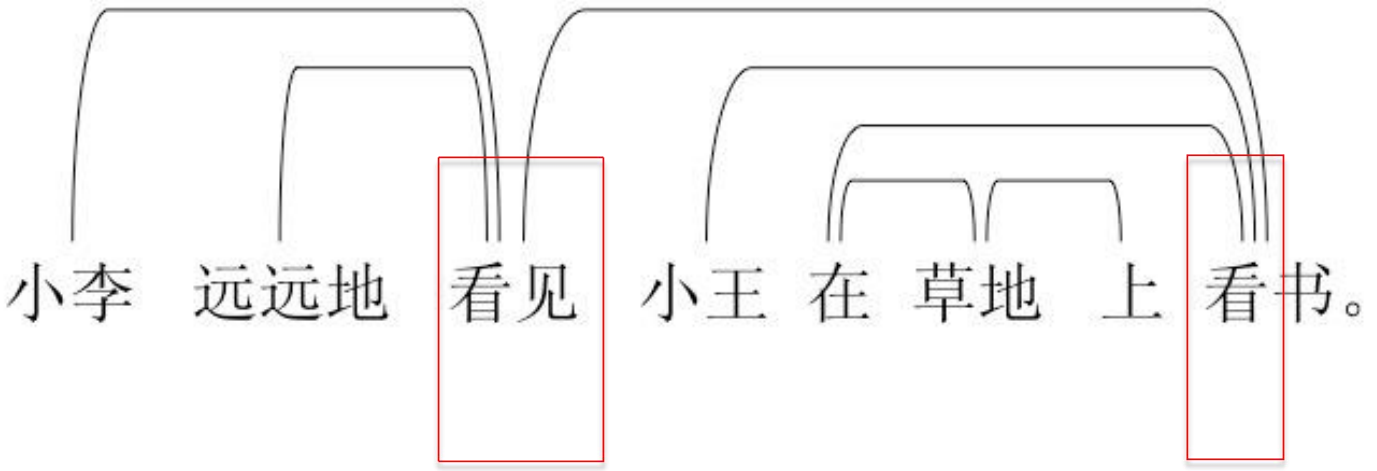


短语结构树

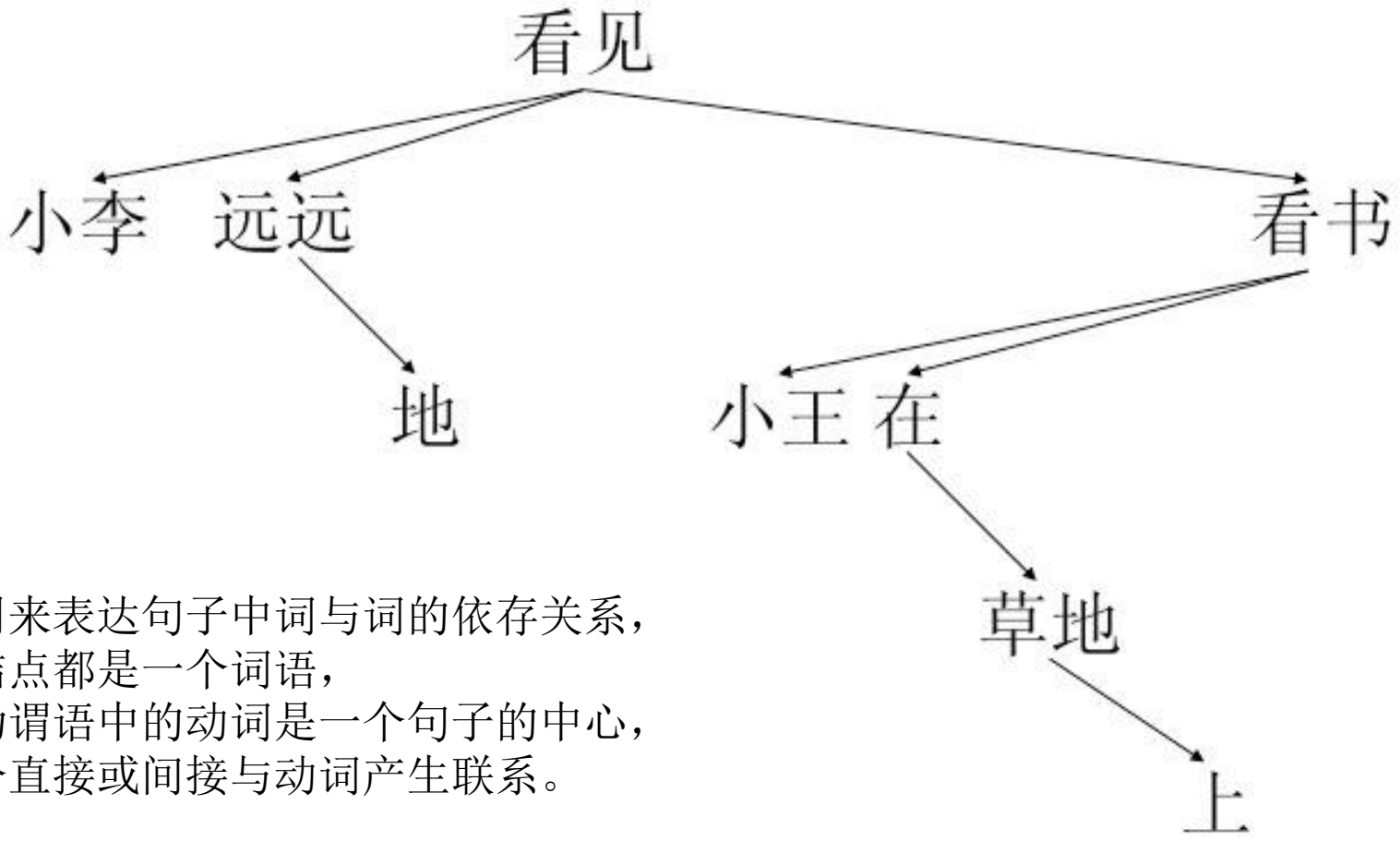
- 短语结构树用来表达句子的句法结构，
- 其只有叶子结点与输入句子中的词语相关联，
- 其他中间结点都是标记短语成分。



词语之间的依赖关系



依存结构树



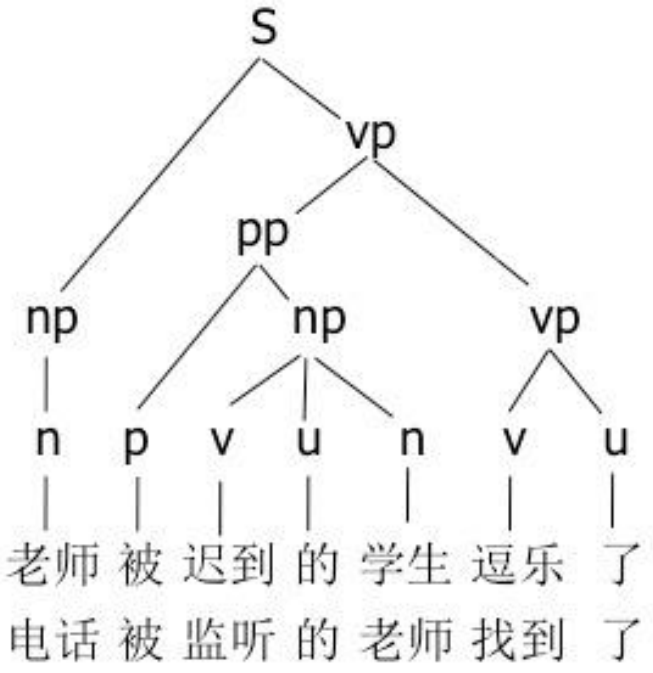
- 依存树用来表达句子中词与词的依存关系，
- 其每个结点都是一个词语，
- 通常认为谓语中的动词是一个句子的中心，
- 其他成分直接或间接与动词产生联系。

句法分析概述

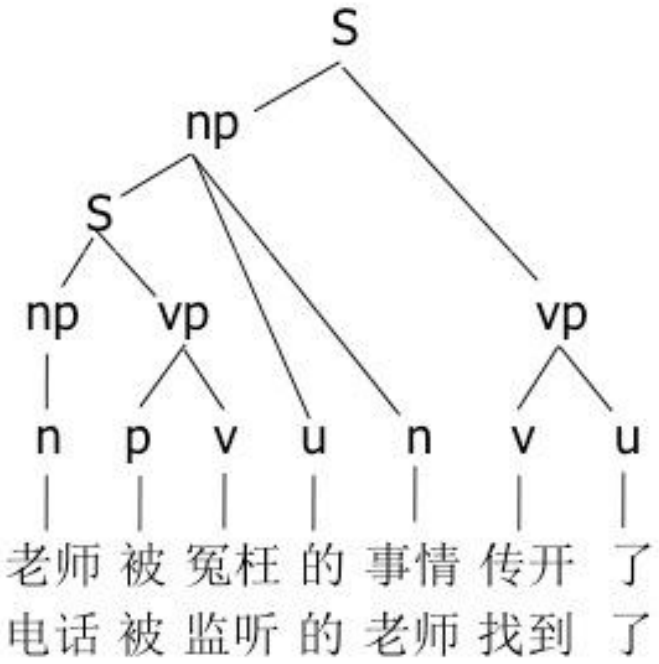
- 句法：词是如何组成句子的？
- 例：下面这些句子有何不同？

	甲	乙
1	老师 被 迟到 的 学生 逗乐 了	a 【迟到 的 学生】把【老师】逗乐了 b 【老师 被 迟到 的 学生】 被 逗乐了
2	老师 被 冤枉 的事情 传开 了	a 【冤枉 的 事情】把【老师】传开了 b 【老师 被 冤枉 的 事情】 被 传开 了
3	电话 被 监听 的 老师 找到 了	a 【监听 的 老师】把【电话】找到了 b 【电话 被 监听 的 老师】 被 找到了

句子内部结构的树图表示



【监听 的 老师】把【电话】找到了



【电话 被 监听 的 老师】 被 找到了

自然语言的层次结构特性示例

1 听说服装设计很吃香　听说那套服装设计得很有品味

2 听说孩子丢了　　听说孩子丢了鞋子

3 听说北京队大败　　听说北京队大败上海队

同一个线性字符串，根据所处上下文环境的不同而解释为不同的树结构！

如何进行句法结构分析

- 句法结构分析：从“线性串”到“树结构”的映射
- 需要做两件事：
  - 形式文法：形式语法理论的任务
  - 搜索算法：计算技术的任务

形式文法

语言是按照一定规律构成的句子和符号串的有限或无限集合

文法描述：语言中的每一个句子用严格定义的规则来构造，利用规则生成语言中合法的句子

形式文法-无穷字符序列的有穷表示法

非终结符号字母表 $\cap$ 终结符号字母表 $= \phi$

$$G = \{V_N, V_T, S, P\}$$

S：开始符号

P：变换式集合

$$\alpha \rightarrow \beta$$

产生式规则，
也称为重写规则

产生式规则的条件

- α 可以是 V_N和 V_T上的任意字符串
- β 可以是 V_N和 V_T上的任意字符串
- P中至少有一个产生式中的 α 得由S来充当

Chomsky Hierarchy

分级	名称	产生式规则的形式限制
0	PSG	$\alpha \rightarrow \beta \ with \ \alpha \in (V_T \cup V_N)^+ \ and \ \beta \in (V_T \cup V_N)^*$
1	CSG	$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2 \ with \ A \in V_N \ and \ \alpha_1, \alpha_2 \in (V_T \cup V_N)^* \ and \ \beta \in (V_T \cup V_N)^+$
2	CFG	$A \rightarrow B \ with \ A \in V_N \ and \ B \in (V_T \cup V_N)^*$
3	RG	$A \rightarrow \beta B \ or \ A \rightarrow B \ with \ A, B \in V_N \ and \ \beta \in V_T^*$

正则文法

- 如果文法 G=(N, Σ, P, S) 的 P 中的规则满足如下形式：A→Bx，或 A→x，其中A, B∈N，x∈ Σ，则称该文法为**正则文法**或称**3型文法**。(左线性正则文法)
  (如果 A → x B，则该文法称为右线性正则文法。)

- 例：G = (N, Σ, P, S), N = {S, A, B}, Σ = {a, b},
  P: (a) S → a A； (b) A → a A； (c) A → b b B；
  (e) B → b

- $L(G) = \{a^n b^m\}, n \geq 1, m \geq 3$

上下文无关文法(CFG)

- 如果 P 中的规则满足如下形式：A → α，其中 A ∈ N，α ∈ (N∪Σ)*，则称该文法为上下文无关文法(CFG)或称2 型文法。

- 例： G = (N, Σ, P, S), N = {S, A, B, C}, Σ = {a, b, c},
- P: (a)S→ABC； (b)A→aA|a； (c) B → bB|b； (d) C → BA|c

- $L(G)=\{a^n b^m a^k c^\alpha\}$, $n\geq 1$, $m\geq 1$, $k\geq 0$, $\alpha \in \{0,1\}$

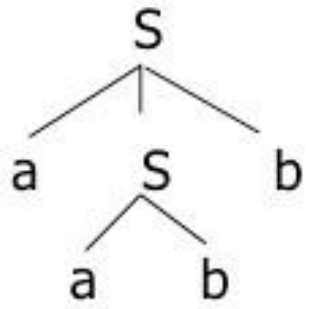
用CFG来描述语言

- 思考

对于语言L={ab，aabb，aaabbb，...，a^nb^n，...}，n是自然数。请写出L的上下文无关文法。

(1) S→a S b

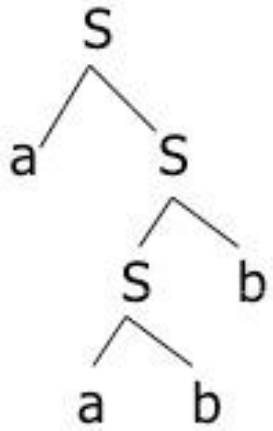
(2) S→a b



(1) S→a S

(2) S→S b

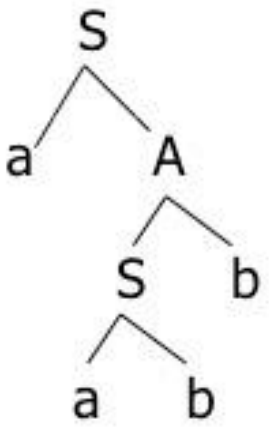
(3) S→a b



(1) S→a A

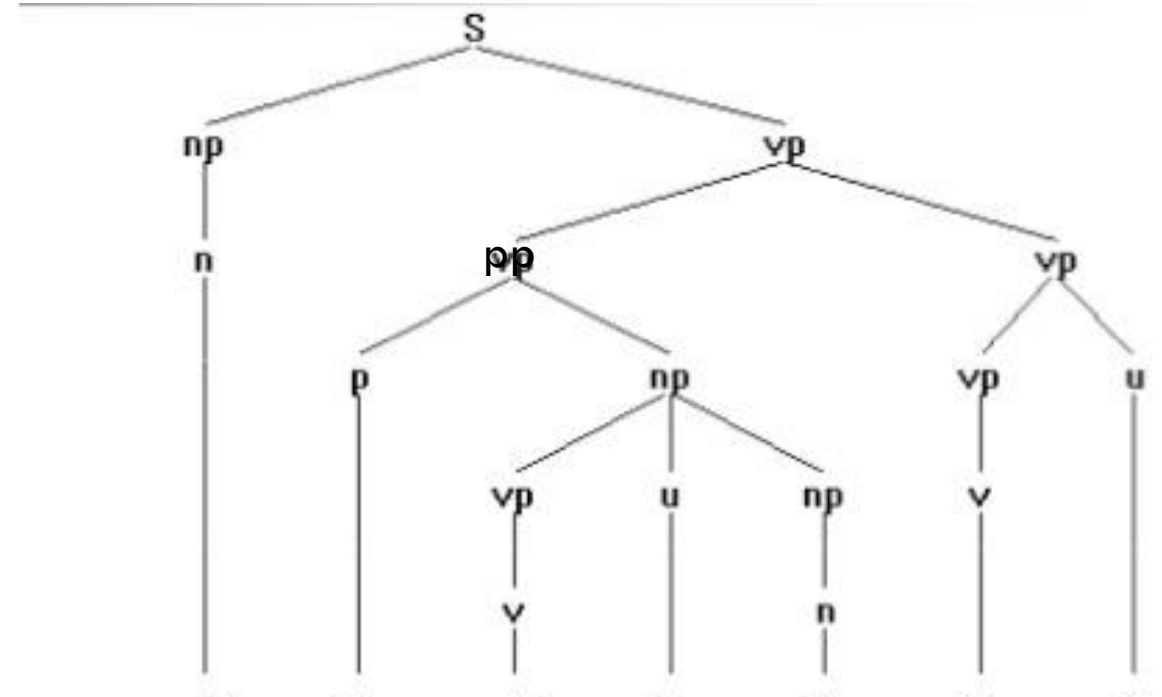
(2) A→S b

(3) S→a b



用CFG描述自然语言

1. S → np vp
2. np → vp u np
3. vp → pp vp
4. vp → vp u
5. pp → p np
6. np → n
7. vp → v
8. n → 老师 | 学生 …
9. v → 迟到 | 逗乐 …
10. p → 被 …
11. u → 的| 了 …



文法的三个作用

- 生成：产生语言L中所有的句子；
- 判定：一个字符串是否属于语言L；
- 分析：得到L中句子的结构树。

课后练习

1. 写出可以产生汉语自然数表达式的CFG
2. 用你写的CFG,画出下列数字的分析树：
   - 一亿零三百万　三万六千五百八十一

句法结构分析算法

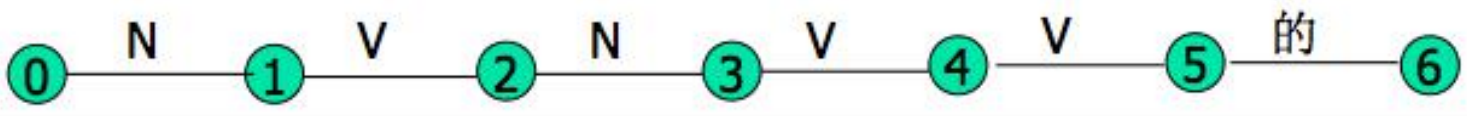
句法结构分析算法

- **自底向上：基于规约的方法**
  - 从待分析字符串开始，用待分析字符串去匹配CFG规则箭头的右部字符，匹配成功后替换为左部字符，直到S。
- **自顶向下：基于预测的方法**
  - 从CFG规则中的S规则开始，将CFG规则箭头左部的符号展开，直到形成以终结符开始的序列，用该序列去匹配待分析字符串，直到完全匹配上。

自底向上和自顶向下分析示例

Earley算法

- Top-down(为主)与bottom-up(为辅)相结合
- 预测能力+数据驱动
- 张三是县长派来的 -- N V N V V 的



基本概念：状态

- 上下文无关文法规则
- 圆点
- 状态的起止位置
  - 整数i：状态起点（已分析子串的起点）
  - 整数j：状态终点（已分析子串的终点）
- **例：<S→NP· VP [0,4]>**

基本操作/算子

- 预测（Predicator）
- 扫描（Scanner）
- 归约（Completer）

算子的形式定义

- Predict：对于状态Z→α·X β [j,k]，其中X是非终结符，对于语法中的每条形如X→γ的规则，都可以形成一个新状态：X→·γ[k,k]
- Scan：对于状态Z→α·X β [j,k]，其中X是终结符。如果X与输入字符串中第k个字符匹配，就形成一个新状态 Z→αX·β [j,k+1]
- Complete：对于一个已经“完成”的状态Z→γ· [j,k]，如果已有状态集合中有形如X→α· Zβ[i,j]的状态，形成一个新状态：X→αZ· β [i,k]

Earley算法: 算法描述

- 设输入字符串长度为n，字符间隔可记做 $0, 1, 2, \ldots, n$
- 算法执行过程如下页所示。

1. 将文法规则中形如 S→α的规则形成为状态：<S→ • α [0,0]>加入到状态集合中(种子状态/seed state)

2. 对状态集中的每个状态,依次进行循环：

   (1) 如果当前状态是[未完成状态],且点后不是终结符,则 执行Predicator;

   (2)如果当前状态是[未完成状态],且点后是终结符,则执行Scanner;

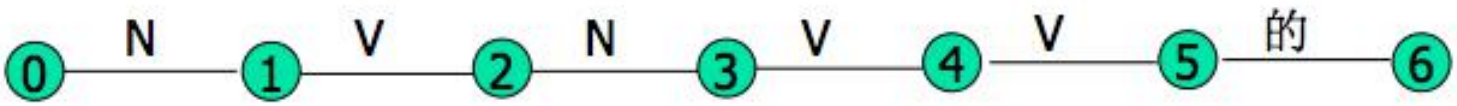
   (3)如果当前状态是[完成状态],则执行Completer;

3. 如果最后得到形如<S →α • [0, n]>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

Earley算法过程示例

- 张三是县长派来的
- 老虎是瞎子打死的
- 主意是董永想出来的
- ……
- N V N V V 的

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V



Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

S [0,0] Seed State

Predict the rule S → • NP VP

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

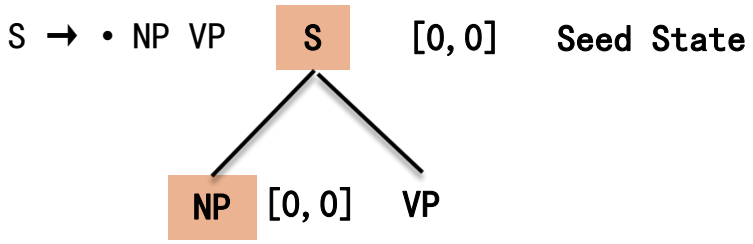
- Predict：对于状态Z→ α •X β [j,k]，其中X是非终结符，对于语法中的每条形如X→ γ 的规则，都可以形成一个新状态：X→ • γ [k,k]

自顶向下基于预测的方法：

- 从CFG规则中的起始符S规则开始，将CFG规则箭头左部的符号展开预测，直到形成以终结符开始的序列
- 然后用该序列去匹配待分析字符串，直到完全匹配上
- 依次进行循环

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5



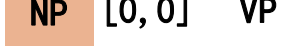
- Predict：对于状态Z→ α •X β [j,k]，其中X是非终结符，对于语法中的每条形如X→ γ 的规则，都可以形成一个新状态：X→ • γ [k,k]

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

S → • NP VP　S　[0,0]　Seed State



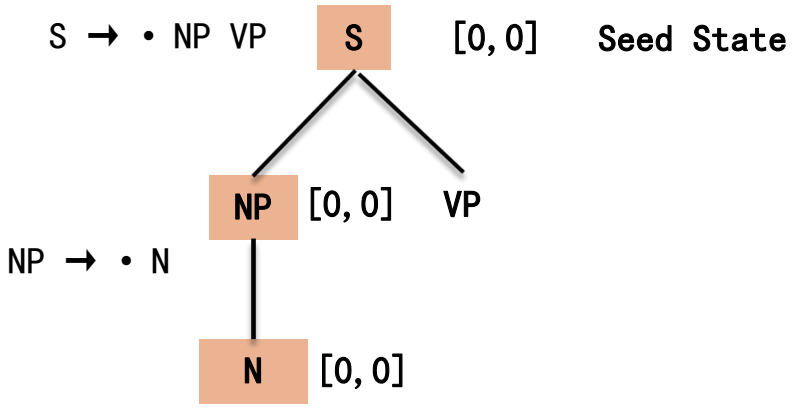
Predict the rule NP → • N

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 因为当前状态是未完成状态，NP为非终止符，继续执行Predictor，预测规则NP → • N

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

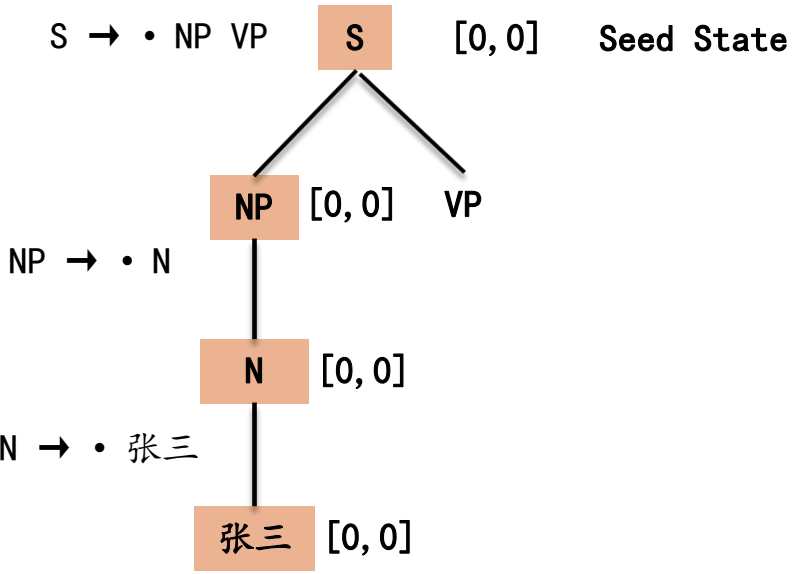


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 接下来，因为当前状态是未完成状态，N为非终止符，继续执行Predictor，预测规则N → • 张三

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

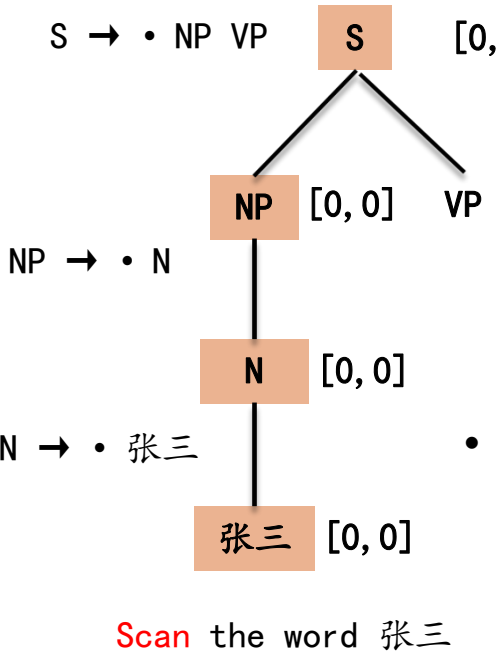


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 预测出单词“张三”

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5



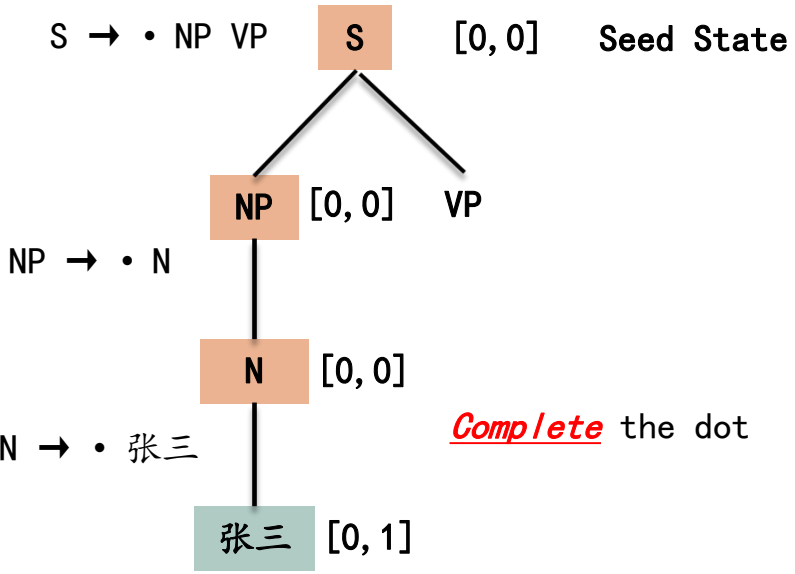
(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 因为当前状态是未完成状态，且点后是终结符，所以执行Scanner，扫描单词“张三”与输入句子是否匹配

- Scan：对于状态Z→ α • X β [j,k]，其中X是终结符。如果X与输入字符串中第k个字符匹配，就形成一个新状态
  Z→ α X • β [j,k+1]

Earley算法过程示例

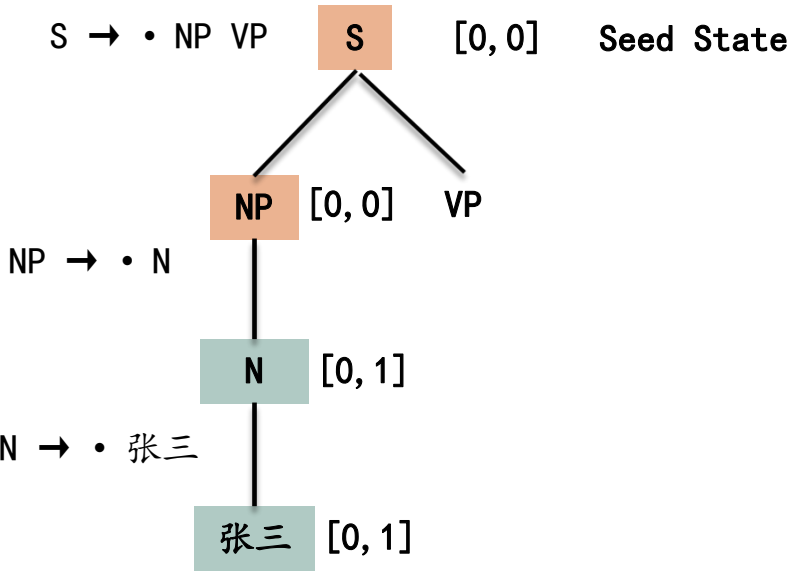
0 张三 1 是 2 县长 3 派来 4 的 5



(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

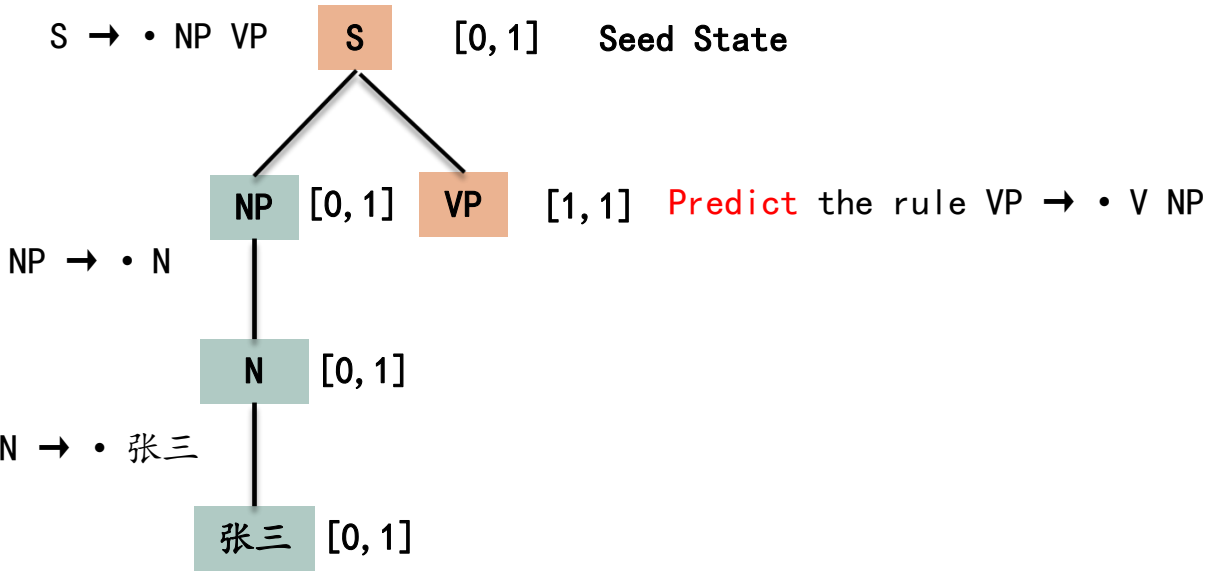


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 预测规则完成，并更新点为[0，1]。

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

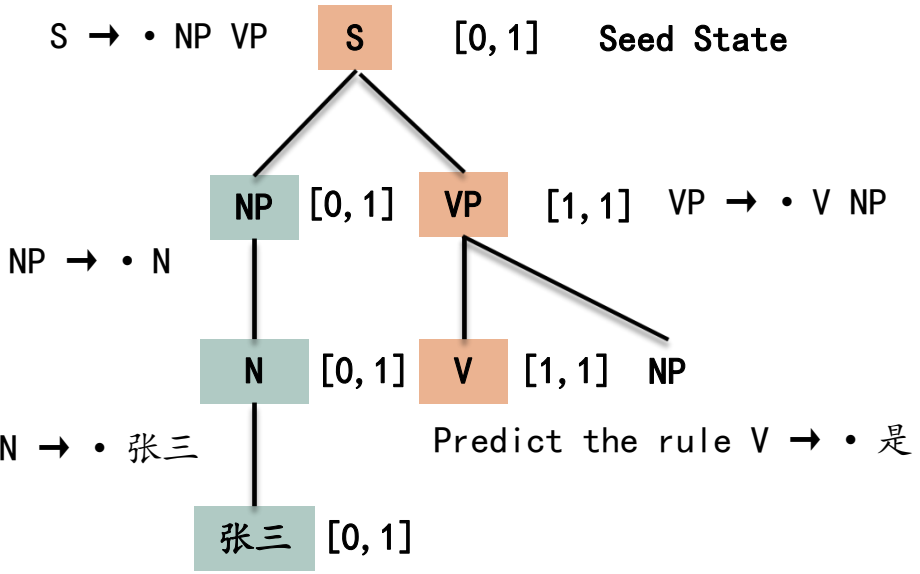


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

继续对VP进行解析，首先由于当前状态是未完成状态，VP为非终止符，执行Predictor，预测规则VP → • V NP，那么现在V与NP就是VP下的左右子结点。

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

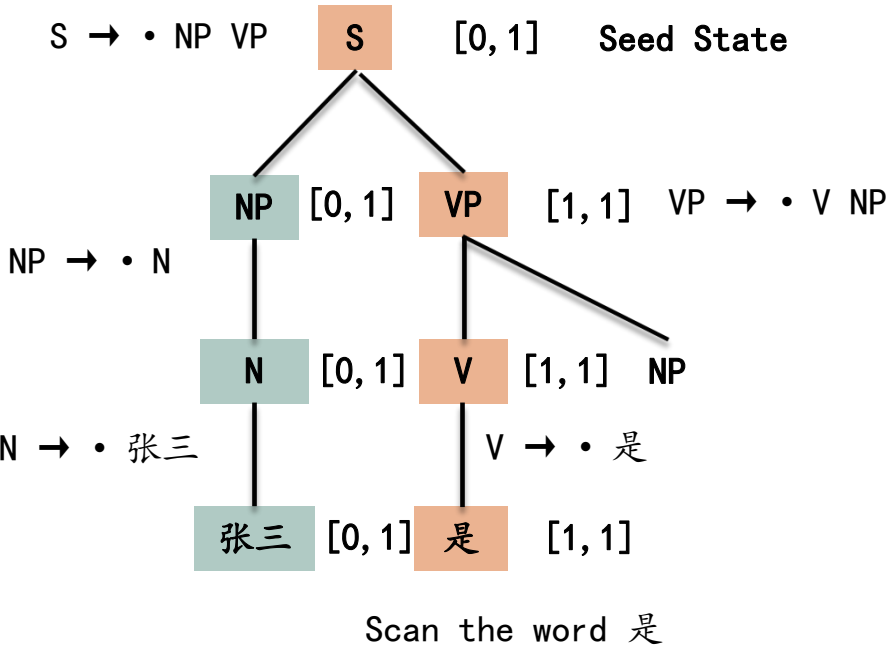


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 对于这个VP，同理是运用自顶向下基于预测的方法
- 先从其左侧子节点开始，因为当前状态是未完成状态，且V是非终结符，执行Predictor，预测规则 V → • 是

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5



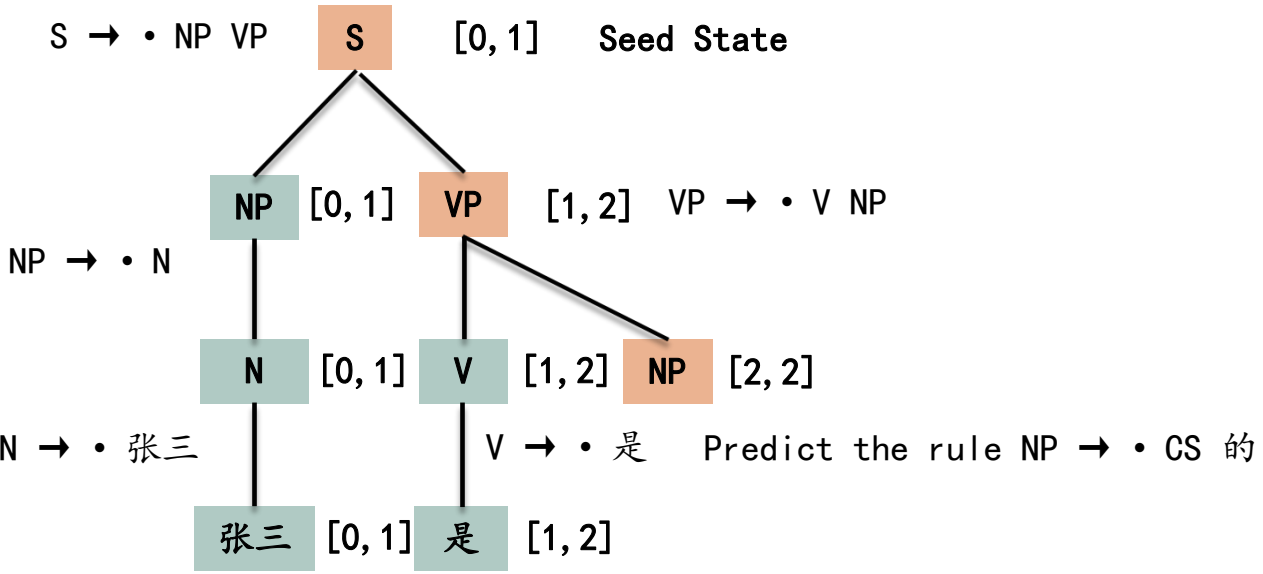
Scan the word 是

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 预测出单词“是”，因为当前状态是未完成状态，且点后是终结符，所以执行Scanner，扫描单词“是”与输入句子是否匹配。

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

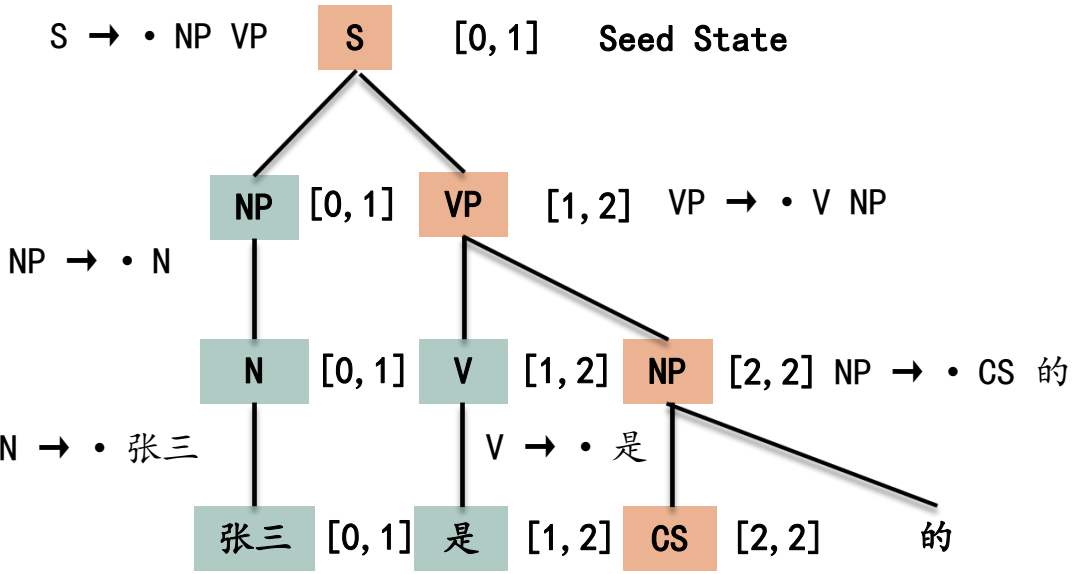


(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

- 单词“是”顺利匹配，预测规则完成，也就完成了这个点，并更新点为[1，2]。
- 然后处理VP的右侧子节点NP，应用同样的自顶向下的预测过程

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5



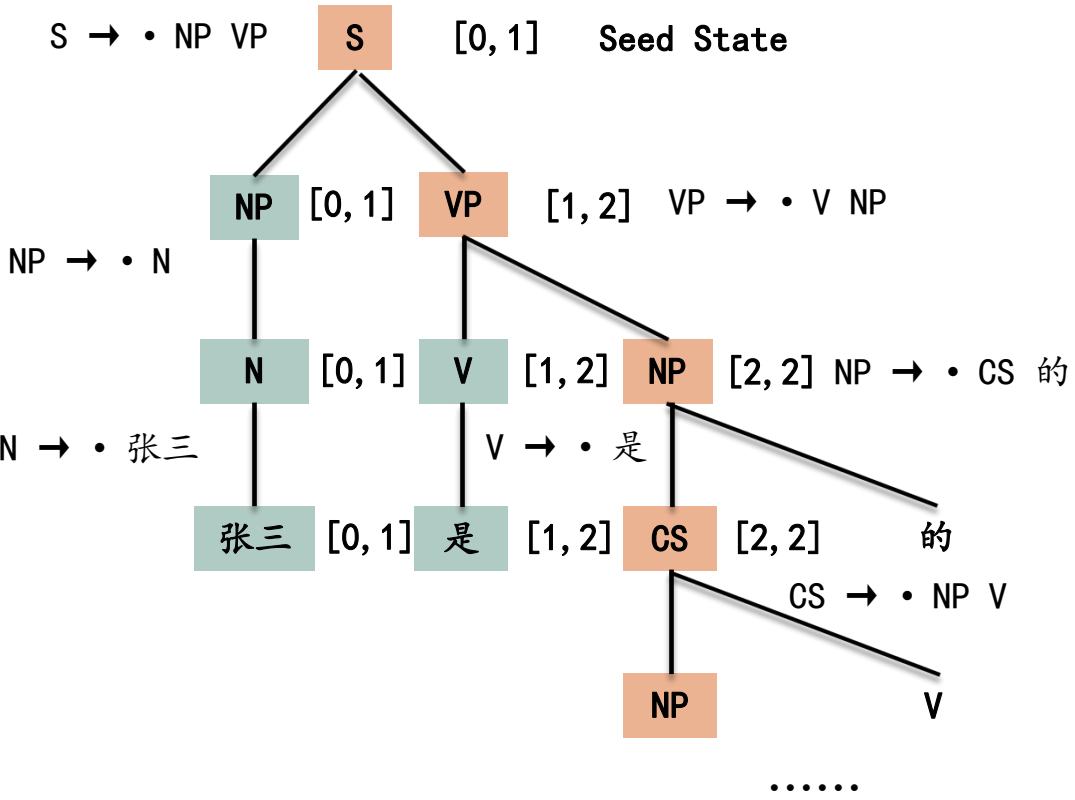
(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Predict the rule CS → • NP V

- 后边依次类推，依次对状态集中的每个状态，进行predictor，scanner或completer操作

Earley算法过程示例

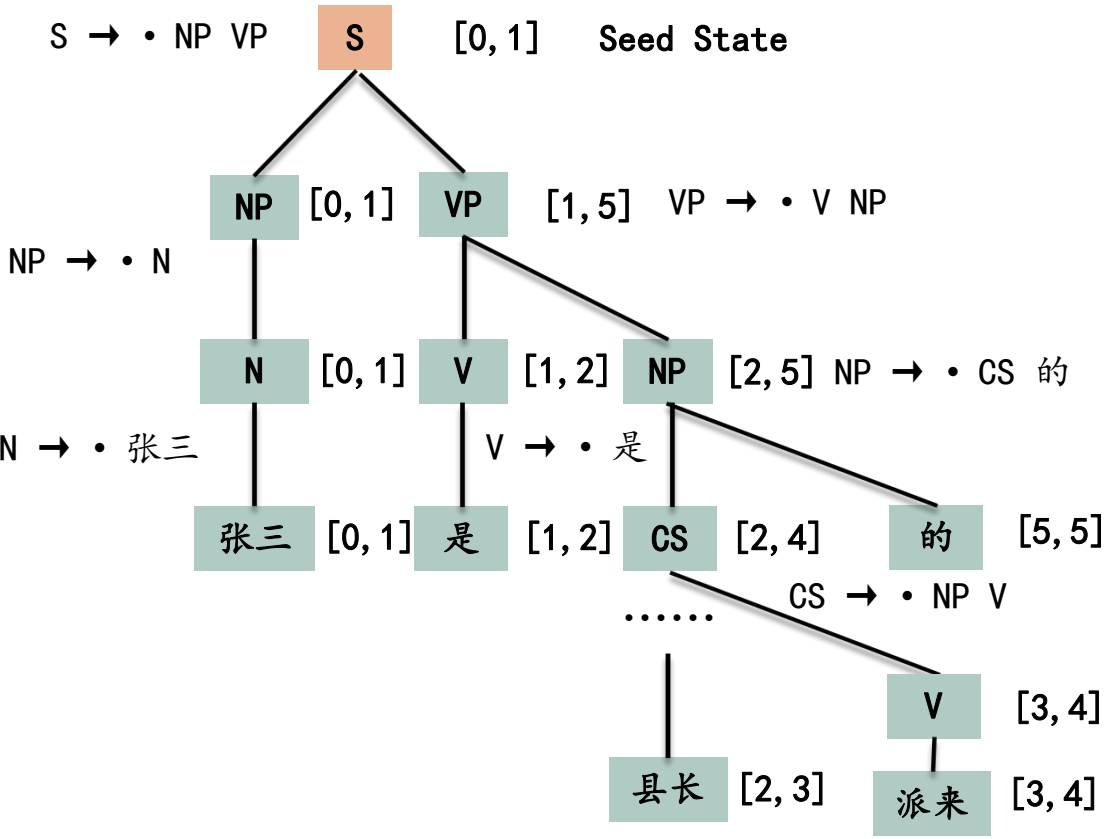
0 张三 1 是 2 县长 3 派来 4 的 5



(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Earley算法过程示例

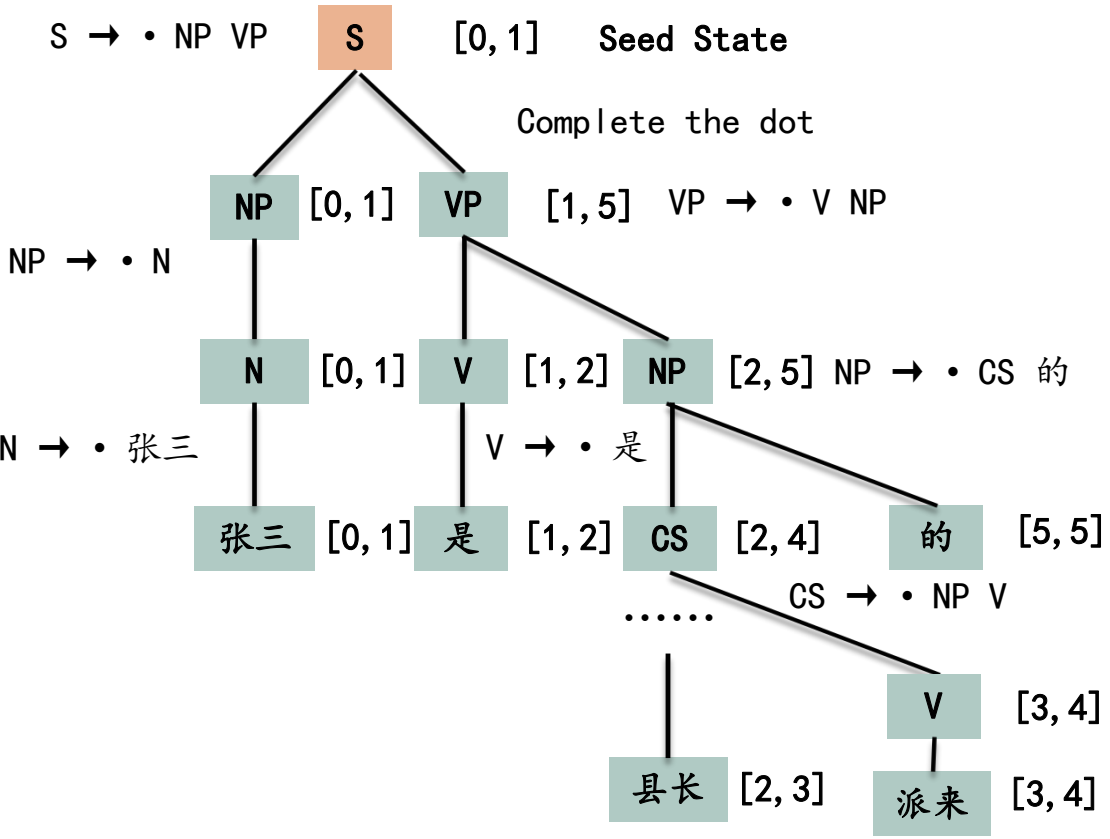
0 张三 1 是 2 县长 3 派来 4 的 5



(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Earley算法过程示例

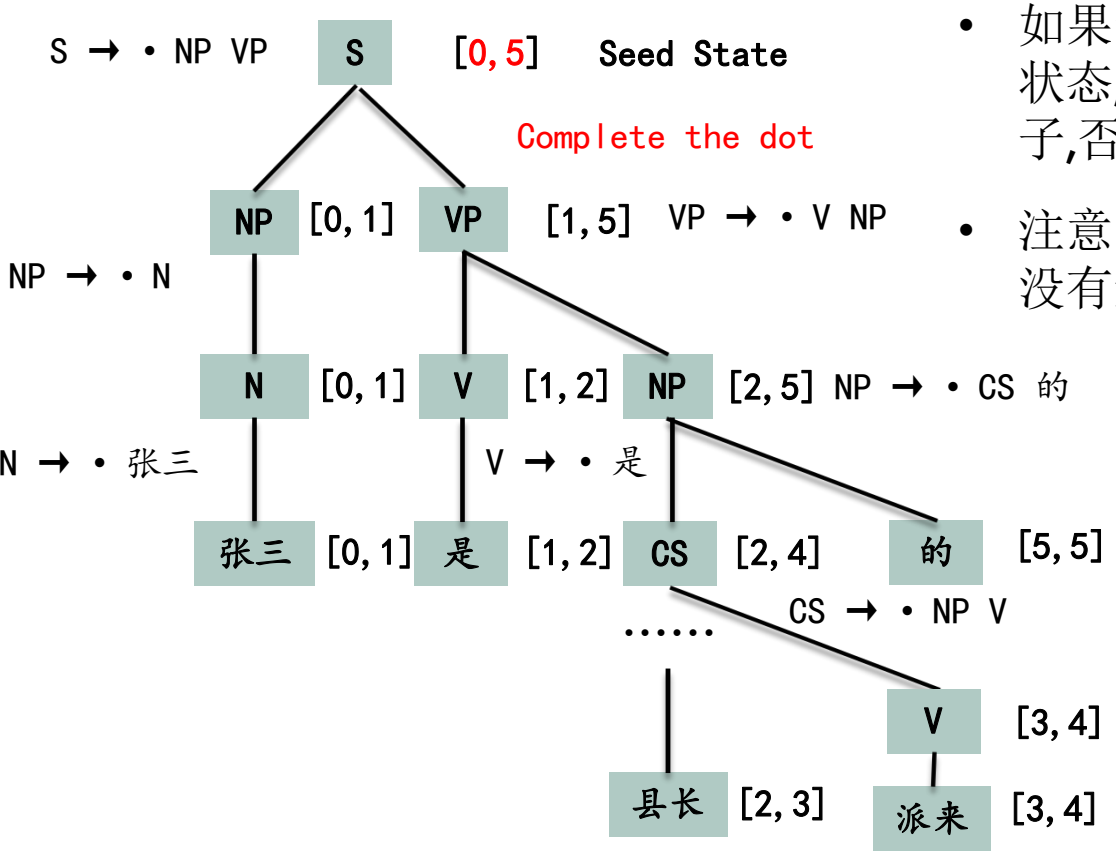
0 张三 1 是 2 县长 3 派来 4 的 5



(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

Earley算法过程示例

0 张三 1 是 2 县长 3 派来 4 的 5

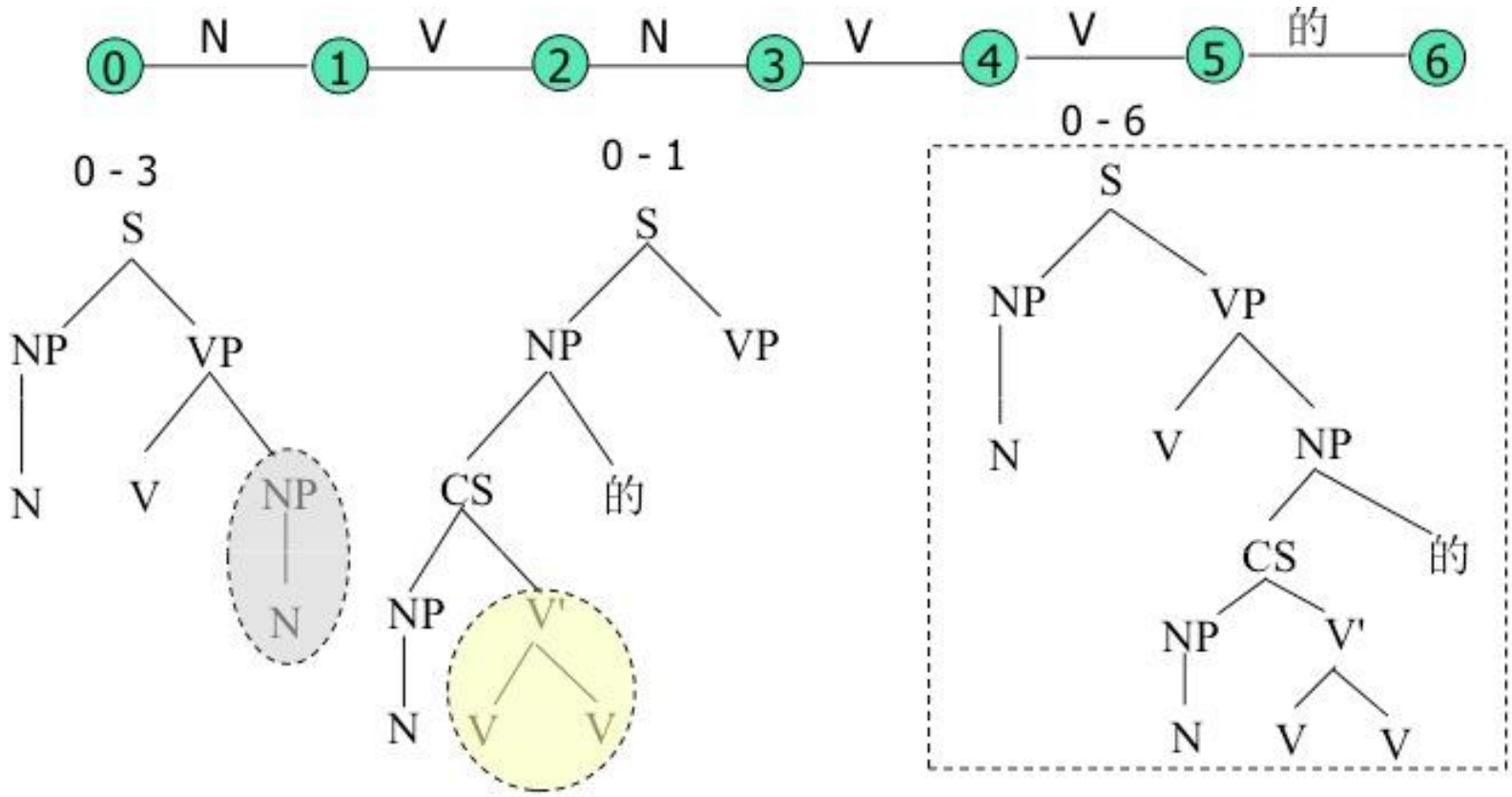


- 如果最后得到形如<S →α· [0, n]>这样的状态,那么输入字符串被接受为合法的句子,否则分析失败
- 注意：我们这里对派来这个词做了简化，没有进一步拆分。

Earley算法构造分析树示意

(1) S → NP VP
(2) NP → N
(3) NP → CS 的
(4) CS → NP V'
(5) VP → V NP
(6) V' → V V

注意：这里是对派来这个词做了进一步拆分的结果！



小结

- 语言模型：保证句法结构分析的准确 把事情做对
- 分析算法：保证句法结构分析的效率 把事情做好

TAG(树邻接语法)
HPSG(中心驱动的短语结构语法)
FUG(功能合一语法)
LFG(词汇功能语法)
PCFG(概率上下文无关文法)
Link Grammar(链语法)
Dependency Grammar(依存语法)
.....

CYK算法
ATN 算法
Probabilistic Earley算法
Chart-based Parsing算法
依存句法分析算法
链语法分析算法
....