



Chapter 4

Syntax Analysis (LL)

王朝坤

llse@Tsinghua



Outline

THSS

44100593

2019 / XS-301

- ✧ 语法分析的基本思想*
- ✧ 带回溯的自顶向下分析
- ✧ 预测分析
- ✧ **LL(1)文法****
 - ✧ **First集; Follow集**
- ✧ 递归下降**LL(1)**分析*
- ✧ 表驱动**LL(1)**分析*
- ✧ 文法变换*
- ✧ 错误处理



基本思想

THSS

44100593

2019 / XS-301

✧ 语法分析程序（*Parser*）的作用

- 分析源程序的单词流是否符合语言的语法规则
- 报告语法错误
- 产生源程序的语法分析结果，
以语法分析树或与之等价的形式体现出来

✧ 语法规则描述工具

- 通常是一种上下文无关文法
- 语法分析的核心即为针对上下文无关文法的句型分析



◇ 语法分析

— 句型分析

对任意上下文无关文法 $G = (V, T, P, S)$ 和任意 $w \in T^*$ ，是否有 $w \in L(G)$ ？若成立，则给出分析树或（最左）推导步骤；否则，进行报错处理。

— 三种实现途径

通用分析（Cocke-Younger-Kasami 算法）

自顶向下（*top-down*）分析

自底向上（*bottom-up*）分析



◇ 自顶向下分析思想

- 从文法开始符号出发进行推导；每一步推导都获得文法的一个句型；直到产生出一个句子，恰好是所期望的终结符串
- 每一步推导是对当前句型中剩余的某个非终结符进行扩展，即用该非终结符的一个产生式的右部替换该非终结符
- 如果不存在任何一个可以产生出所期望的终结符串的推导，则表明存在语法错误



基本思想 (例)

THSS

44100593

2019 / XS-301

◇ 自顶向下分析举例

– 单词序列 aaab 的一个自顶向下分析过程

文法 $G(S)$:	S	$(S \rightarrow AB)$
$S \rightarrow AB$	$\Rightarrow AB$	$(A \rightarrow aA)$
$A \rightarrow aA \mid \varepsilon$	$\Rightarrow aAB$	$(B \rightarrow b)$
$B \rightarrow b \mid bB$	$\Rightarrow aAb$	$(A \rightarrow aA)$
	$\Rightarrow aaAb$	$(A \rightarrow aA)$
	$\Rightarrow aaaAb$	$(A \rightarrow \varepsilon)$
	$\Rightarrow aaab$	



带回溯的自顶向下分析

THSS

44100593

2019 / XS-301

◇ 一般方法

— 两类非确定性

在每一步推导中，选择哪一个非终结符、
哪一个产生式都可能是非确定的

分析成功的结果：得到一个推导



带回溯的自顶向下分析

THSS

44100593

2019 / XS-301

✧ 举例

– 单词序列 aaab 的一个自顶向下分析过程

文法 $G(S)$:

(1) $S \rightarrow AB$

(2) $A \rightarrow aA$

(3) $A \rightarrow \varepsilon$

(4) $B \rightarrow b$

(5) $B \rightarrow bB$

S (1)

$\Rightarrow AB$ (2)

$\Rightarrow aAB$ (5)

$\Rightarrow aAbB$ (2)

$\Rightarrow aaAbB$ (2)

$\Rightarrow aaaAbB$ (3)

$\Rightarrow aaabB$ (回溯)

.....

复杂度很高
失败条件较复杂



带回溯的自顶向下分析

THSS

44100593

2019 / XS-301

◇ 改进的方法

– 仅有产生式选择是非确定的

在每一步推导中，总是对最左边的非终结符进行展开，但选择哪一个产生式是非确定的
分析成功的结果：得到一个最左推导

原理：每个合法的句子都存在至少一个起始于开始符号的最左推导；一个终结字符串，只要存在一个起始于开始符号的最左推导，它就是一个合法的句子

从左向右扫描输入单词，失败条件较简单



带回溯的自顶向下分析

THSS

44100593

2019 / XS-301

◇ 改进的方法举例

– 单词序列 aaab 的一个自顶向下分析过程

文法 $G(S)$:	S	(1)
	$\Rightarrow AB$	(2)
(1) $S \rightarrow AB$	$\Rightarrow aAB$	(3)
(2) $A \rightarrow aA$	$\Rightarrow aB$	(回溯)
(3) $A \rightarrow \varepsilon$	$\Rightarrow aAB$	(2)
(4) $B \rightarrow b$	$\Rightarrow aaAB$	(2)
(5) $B \rightarrow bB$	$\Rightarrow aaaAB$	(3)
	$\Rightarrow aaaB$	(5)
复杂度降低	$\Rightarrow aaabB$	(回溯)
失败条件简化	$\Rightarrow aaaB$	(4)
	$\Rightarrow aaab$	(成功)



◇ 确定的自顶向下分析

- 非终结符选择和产生式选择都是确定的

在每一步推导中，总是对最左边的非终结符进行展开，且选择哪一个产生式是确定的，因此是一种无回溯的方法

从左向右扫描，可能向前查看（lookahead）
确定数目的单词

分析成功的结果：得到唯一的最左推导

分析条件：对文法需要有一定的限制



预测分析

THSS

44100593

2019 / XS-301

◇ 举例（向前查看 2 个单词）

– 单词序列 $a^n b^m$ ($n \geq 0, m > 0$) 的预测分析过程

文法 $G(S)$:

(1) $S \rightarrow AB$

(2) $A \rightarrow aA$

(3) $A \rightarrow \varepsilon$

(4) $B \rightarrow b$

(5) $B \rightarrow bB$

只要向前查看 2 个
单词，就可预测分
析 $L(G)$ 中所有句子

S (1)

$\Rightarrow AB$ (2)

$\Rightarrow aAB$ (2)

.....

$\Rightarrow a^n AB$ (3)

$\Rightarrow a^n B$ (5)

$\Rightarrow a^n bB$ (5)

.....

$\Rightarrow a^n b^{m-1} B$ (4)

$\Rightarrow a^n b^m$ (成功)



预测分析

THSS

44100593

2019 / XS-301

◇ 左递归带来的问题

– 考虑下列文法识别 ba^n 的分析过程

文法 $G(S)$:	S	(1)
(1) $S \rightarrow Sa$	$\Rightarrow Sa$	(1)
(2) $S \rightarrow b$	$\Rightarrow Saa$	(1)
	$\Rightarrow Saaa$	(1)

需要向前查看 $n+1$ 个单词， 才能确定这样的推导序列	$\Rightarrow Sa^n$	(2)
	$\Rightarrow ba^n$	

但是：无论向前查看的单词数确定为多少，
都无法满足预测分析 $L(G)$ 中所有句子的需求



预测分析

THSS

44100593

2019 / XS-301

◇ 要求文法不含左递归

— 例：直接左递归

$$P \rightarrow Pa$$
$$P \rightarrow b$$

.....

— 例：间接左递归

$$P \rightarrow Aa$$
$$A \rightarrow Pb$$

.....

— 可以通过文法变换消除左递归
专门讨论



预测分析

THSS

44100593

2019 / XS-301

◇ 左公因子带来的问题

- 如下文法需要向前查看 3 个单词来预测分析 $L(G)$ 中的句子

文法 $G(S) : S \rightarrow abA \mid abB$
 $A \rightarrow a$
 $B \rightarrow b$

文法 $G'(S) : S \rightarrow aAb \mid aAc$
 $A \rightarrow a \mid aA$

- 对于文法 G' **无法** 确定需要向前查看多少个单词来预测分析 $L(G)$ 中的句子



预测分析

THSS

44100593

2019 / XS-301

- ◇ 通常要求文法不含左公因子
 - 可以通过文法变换消除左公因子
- 专门讨论



预测分析

THSS

44100593

2019 / XS-301

- ✧ 应用较普遍的预测分析是
LL(1) 分析
 - 要求文法一定是LL(1) 文法
专门讨论
 - LL(1) 分析程序既可以手工构造，
也可以自动构造



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ LL (1) 的含义

- 第一个 “L”, 代表从左 (Left) 向右扫描单词
- 第二个 “L”, 代表产生的是最左 (Leftmost) 推导
- “1”代表向前查看 (lookahead) 一个单词



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 对文法的限制

- 要求文法是LL (1) 的
- 什么是LL (1) 文法?

✧ 两个重要概念

- First 集合
- Follow 集合

An Example:
 $S \xRightarrow{*} abcAde;$
 $a \in \text{first}(S);$
 $d \in \text{follow}(A).$



LL (1) 分析

THSS

44100593

2019 / XS-301

◇ First 集合

– 定义

设 $G = (V_T, V_N, P, S)$ 是上下文无关文法

对 $\alpha \in (V_T \cup V_N)^*$,

$\text{First}(\alpha) = \{ a \mid \alpha \xRightarrow{*} a\beta, a \in V_T, \beta \in (V_T \cup V_N)^* \}$

若 $\alpha \xRightarrow{*} \epsilon$ 则规定 $\epsilon \in \text{First}(\alpha)$



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 计算 First 集合

设 $\alpha = X \in V_N \cup V_T$, 则 $\text{First}(X)$ 可按如下步骤计算:

- 若 $X \in V_T$, 则 $\text{First}(X) = \{X\}$
- 若 $X \rightarrow \varepsilon$ 也是一个产生式, 则把 ε 也加到 $\text{First}(X)$ 中;
- 若 $X \in V_N$, 且有产生式 $X \rightarrow a...$, $a \in V_T$, 则把 a 加入到 $\text{First}(X)$ 中;

若 $X \rightarrow Y_1 Y_2 \dots Y_k$ 是一个产生式, $Y_1, Y_2, \dots, Y_k \in V_N \cup V_T$,

- 1) 对于任何 $j: 1 \leq j \leq i-1, 1 \leq i \leq k$, $\text{First}(Y_j)$ 都含有 ε , 但 $\text{First}(Y_i)$ 不含 ε , 则把 $\text{First}(Y_j)$ 中的所有非 ε 元素和 $\text{First}(Y_i)$ 中的所有元素都加到 $\text{First}(X)$ 中;
- 2) 特别是, 对于任何 $j: 1 \leq j \leq k$, $\text{First}(Y_j)$ 都含 ε , 则除 $\text{First}(Y_j)$ 中的非 ε 元素外, 把 ε 也加到 $\text{First}(X)$ 中.



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 计算 First 集合

设 $\alpha = X_1X_2\dots X_n$, 则 $\text{First}(\alpha)$ 可按如下步骤计算:

– 若对于任何 $j: 1 \leq j \leq i-1 < n$, 有

$$\varepsilon \in \text{First}(X_j) \wedge \varepsilon \notin \text{First}(X_i),$$

$$\text{则 } \text{First}(\alpha) = \bigcup_{j=1}^{i-1} \text{First}(X_j) \cup \text{First}(X_i) - \{\varepsilon\} = \bigcup_{j=1}^i \text{First}(X_j) - \{\varepsilon\}$$

– 若所有的 $j, 1 \leq j \leq n$, 都有 $\varepsilon \in \text{First}(X_j)$,

$$\text{则 } \text{First}(\alpha) = \bigcup_{j=1}^n \text{First}(X_j)$$

$S \rightarrow ES'$

$S' \rightarrow \varepsilon \mid +S$

$E \rightarrow \text{number} \mid (S)$

$\text{FIRST}(S) = \{\text{number}, (\}$

$\text{FIRST}(S') = \{\varepsilon, + \}$

$\text{FIRST}(E) = \{\text{number}, (\}$



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ Follow 集合

– 定义

设 $G = (V_T, V_N, P, S)$ 是上下文无关文法
对 $A \in V_N$,

$$\text{Follow}(A) = \{a \mid S \xRightarrow{*} \alpha A \beta \text{ 且 } a \in \text{First}(\beta), \\ \alpha \in (V_T \cup V_N)^*, \beta \in (V_T \cup V_N)^+\}$$

若 $S \xRightarrow{*} \alpha A \beta$, 且 $\beta \xRightarrow{*} \epsilon$, 则规定
 $\# \in \text{Follow}(A)$

(# 代表输入单词序列右边的结束符)



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 计算 Follow 集合

- 对于文法的开始符号 S , 置 $\#$ 于 $\text{Follow}(S)$ 中;
- 若 $A \rightarrow \alpha B \beta$ 是一个产生式, 则把 $\text{First}(\beta) - \{\epsilon\}$ 加至 $\text{Follow}(B)$ 中;
- 若 $A \rightarrow \alpha B$ 是一个产生式, 或 $A \rightarrow \alpha B \beta$ 是一个产生式而 $\beta \xRightarrow{*} \epsilon$ (即 $\epsilon \in \text{First}(\beta)$), 则把 $\text{Follow}(A)$ 加至 $\text{Follow}(B)$ 中.

$S \rightarrow ES'$

$\text{FOLLOW}(S) = \{ \#,) \}$

$S' \rightarrow \epsilon \mid +S$

$\text{FOLLOW}(S') = \text{FOLLOW}(S) = \{ \#,) \}$

$E \rightarrow \text{number} \mid (S)$

$\text{FOLLOW}(E) = (\text{FIRST}(S') - \{\epsilon\}) \cup \text{FOLLOW}(S)$
 $= \{ +, \#,) \}$



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 例：计算 First 和 Follow 集合

文法 **G (S)** :

(1) **S** \rightarrow **AB**

(2) **A** \rightarrow **Da** $\mid \epsilon$

(3) **B** \rightarrow **cC**

(4) **C** \rightarrow **aADC** $\mid \epsilon$

(5) **D** \rightarrow **b** $\mid \epsilon$

First(D) = {**b**, ϵ }

First(C) = {**a**, ϵ }

First(B) = {**c**}

First(A) = {**b**, **a**, ϵ }

First(S) = {**b**, **a**, **c**}

Follow(S) = {**#**}

Follow(C) = {**#**}

Follow(A) = {**c**, **b**, **a**, **#**}

Follow(B) = {**#**}

Follow(D) = {**a**, **#**}



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 定义： LL (1) 文法

文法 **G** 是 **LL (1)** 的，当且仅当对于 **G** 的每个非终结符 **A** 的任何两个不同产生式 $A \rightarrow \alpha \mid \beta$ ，下面的条件成立：

- $\text{First}(\alpha) \cap \text{First}(\beta) = \phi$ ，即 α 和 β 推导不出以同一个单词为首的符号串，也不会同时推导出 ε
- 假若 $\beta^* \Rightarrow \varepsilon$ ，那么 $\text{First}(\alpha) \cap \text{Follow}(A) = \phi$ ，即 α 所能推出的串的首符号不应在 $\text{Follow}(A)$ 中。



LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 举例：判断如下文法 $G(S)$ 是否是 LL(1) 文法

文法 $G(S)$:

(1) $S \rightarrow AB$

(2) $A \rightarrow Da \mid \varepsilon$

(3) $B \rightarrow cC$

(4) $C \rightarrow aADC \mid \varepsilon$

(5) $D \rightarrow b \mid \varepsilon$

$\text{First}(D) = \{b, \varepsilon\}$

$\text{First}(C) = \{a, \varepsilon\}$

$\text{First}(B) = \{c\}$

$\text{First}(A) = \{b, a, \varepsilon\}$

$\text{First}(S) = \{b, a, c\}$

$\text{Follow}(S) = \{\#\}$

$\text{Follow}(C) = \{\#\}$

$\text{Follow}(A) = \{c, b, a, \#\}$

$\text{Follow}(B) = \{\#\}$

$\text{Follow}(D) = \{a, \#\}$

For A: $\text{first}(Da) \cap \text{follow}(A) = \{b, a\} \cap \{c, b, a, \#\} = \{b, a\}$

$\therefore G(S)$ 不是 LL(1) 文法



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ LL(1)文法的性质

LL(1)文法是无二义的

LL(1)文法是无左递归的

LL(1)文法是无左公因子的

除了利用定义外，有时可以利用这些性质判定某些文法不是LL(1)的



LL (1) 分析

THSS

44100593

2019 / XS-301

✧ LL (1) 分析的实现

- 递归下降 LL (1) 分析
(递归下降分析: 非终结符 \Leftrightarrow 子程序)
- 表驱动 LL (1) 分析
借助于预测分析表和一个下推栈



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 递归下降LL(1)分析程序

— 工作原理

每个非终结符都对应一个子程序。该子程序的行为根据语法描述来明确：

- 每遇到一个终结符，则判断当前读入的单词是否与该终结符相匹配，若匹配，再读取下一个单词继续分析；不匹配，则进行出错处理
- 每遇到一个非终结符，则调用相应的子程序



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 非终结符对应的递归下降子程序

- 例 对于下列文法 （其中 function, identifier, parameter_list 和 statement 是非终结符）

function \rightarrow FUNC identifier (parameter_list) statement

```
void ParseFunction( )
{
    MatchToken(T_FUNC);    //匹配FUNC
    ParseIdentifier( );
    MatchToken(T_LPAREN); // 匹配 (
    ParseParameterList( );
    MatchToken(T_RPAREN); // 匹配 )
    ParseStatement( );
}
```



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 非终结符对应的递归下降子程序

– 例 续上页

```
void MatchToken(int expected)
{
    if (lookahead != expected) //判别当前单词是否与
    {                           //期望的终结符匹配
        printf("syntax error \n");
        exit(0);
    }
    else // 若匹配,消费掉当前单词并读入下一个
        lookahead = nexttoken(); //调用词法分析程序
}
```




递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 非终结符对应的递归下降子程序

— 一般结构

设 A 的产生式:

$$A \rightarrow u_1 \mid u_2 \mid \dots,$$

相对于非终结符 A
的递归下降子程序
ParseA的一般结构如右图所示

```
void ParseA()
{
    switch (lookahead) {
        case First( $u_1$ ):
            /* code to recognize  $u_1$  */
            break;
        case First( $u_2$ ):
            /* code to recognize  $u_2$  */
            break;
        ...
        case Follow( $A$ ): /* when  $A \xRightarrow{*} \varepsilon$  */
            /* usually do nothing here */
            break;
        default:
            printf("syntax error \n");
            exit(0);
    }
}
```



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 递归下降LL(1)分析程序举例

– 例 对于下列文法

$G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

$\text{First}(S) = \{a, b, c, d\}$

$\text{First}(A) = \{a\}$

$\text{First}(B) = \{\varepsilon, c\}$

$\text{Follow}(S) = \{\#\}$

$\text{Follow}(A) = \{a\}$

$\text{Follow}(B) = \{b\}$

因为 $\text{First}(AaS) = \{a\}$, $\text{First}(BbS) = \{b, c\}$, 以及 $\text{First}(d) = \{d\}$ 之间两两互不相交, 同时 $\text{Follow}(B) = \{b\}$ 与 $\text{First}(c) = \{c\}$ 不相交, 所以, $G(S)$ 是 LL(1) 文法



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

– 接上例 针对文法G(S)构造的递归下降分析程序

G(S): $S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

First (S) = {a, b, c, d}

Follow (S) = {#}

```
void ParseS( )
```

```
{
```

```
    switch (lookahead) {
```

```
        case a:
```

```
            ParseA( );
```

```
            MatchToken(a);
```

```
            ParseS( );
```

```
            break;
```

```
        case b,c:
```

```
            ParseB( );
```

```
            MatchToken(b);
```

```
            ParseS( );
```

```
            break;
```

```
        case d:
```

```
            MatchToken(d);
```

```
            break;
```

```
        default:
```

```
            printf("syntax error \n");
```

```
            exit(0);
```

```
    }
```

```
}
```



递归下降 LL (1) 分析

THSS

44100593

2019 / XS-301

– 接上例 针对文法G(S) 构造的递归下降分析程序

G(S): $S \rightarrow AaS \mid BbS \mid d$
 $A \rightarrow a$
 $B \rightarrow \varepsilon \mid c$

```
void ParseB( )
{
    if (lookahead==c) {
        MatchToken(c);
    }
    else if (lookahead==b) {
    }
    else {
        printf("syntax error \n");
        exit(0);
    }
}
```

First (A) = {a}
First (B) = { ε , c}
Follow (S) = {#}
Follow (A) = {a}
Follow (B) = {b}

```
void ParseA( )
{
    if (lookahead==a) {
        MatchToken(a);
    }
    else {
        printf("syntax error \n");
        exit(0);
    }
}
```



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 表驱动LL(1)分析程序

– 工作原理 利用预测分析表和一个下推栈实现

- (0) 初始化，将符号#入栈；
- (1) 文法开始符号入栈；
- (2) 若栈顶为 **终结符**，则判断当前读入的单词是否与
该终结符相匹配，
 - (a) 若匹配，再读取下一 单词继续分析；
 - (b) 不匹配，则进行出错处理；
- (3) 若栈顶为**非终结符**，则根据该非终结符和当前
输入单词查预测分析表，
 - (a) 若相应表项中是产生式（唯一的），则将此非
终结符出栈，并把产生式右部符号**从右至左**入栈；
 - (b) 若表项为空，则进行出错处理；
- (4) 重复（2）和（3），直到栈顶为 # 同时输入也
遇到结束符 # 时，分析结束



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 预测分析表

- 表驱动分析程序需要的二维表 M
- 表的每一行 A 对应一个非终结符
- 表的每一列 a 对应某个终结符或输入结束符 $\#$
- 表中的项 $M(A, a)$ 表示栈顶为 A ，下一个输入符号为 a 时，可选的产生式集合
- 对于 LL (1) 文法，可以构造出一个 $M(A, a)$ 最多只包含一个产生式的预测分析表，可称之为 LL (1) 分析表
- $M(A, a)$ 不含产生式时，对应一个出错位置



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

◇ 预测分析表的构造算法

- 对文法 **G** 的每个产生式 $A \rightarrow \alpha$ 执行如下步骤:
 - (1) 对每个 $a \in \text{First}(\alpha)$, 把 $A \rightarrow \alpha$ 加入 $M[A, a]$
 - (2) 若 $\varepsilon \in \text{First}(\alpha)$, 则对任何 $b \in \text{Follow}(A)$, 把 $A \rightarrow \alpha$ 加至 $M[A, b]$ 中
- 把所有无定义的 $M[A, a]$ 标上“出错标志”

可以证明: 一个文法 **G** 的预测分析表不含多重入口, 当且仅当该文法是 **LL(1)** 的



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 预测分析表的构造举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

$\text{First}(S) = \{a, b, c, d\}$

$\text{First}(A) = \{a\}$

$\text{First}(B) = \{\varepsilon, c\}$

$\text{Follow}(S) = \{\#\}$

$\text{Follow}(A) = \{a\}$

$\text{Follow}(B) = \{b\}$

可构造如下预测分析表:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析程序分析算法

初始时 ‘#’入栈，然后文法开始符号入栈；首个输入符号读进 **a** ；

flag = TRUE;

while (flag) do {

 栈顶符号出栈并放在**X**中；

if ($X \in V_T$) {

if ($X == a$)

 把下一个输入符号读进**a**;

else ERROR;

}

else if ($X == \#$) {

if ($a == \#$) flag = FALSE;

else ERROR;

}

else if ($M[X, a] == \{X \rightarrow X_1 X_2 \dots X_k\}$) X_k, X_{k-1}, \dots, X_1 依次进栈;

else ERROR;

}

/*分析成功，过程完毕*/



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

$aabd\#$

S
 $\#$

分析输入串 **$aabd$** 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

aabd#

A
a
S
#

分析输入串 *aabd* 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

$aabd\#$

a
 a
 S
 $\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

$a b d \#$

a
 S
 $\#$

分析输入串 $a a b d$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$ 剩余的输入串

$A \rightarrow a$

$bd\#$

$B \rightarrow \varepsilon \mid c$

S
 $\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

$bd\#$

B
 b
 S
 $\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$

$A \rightarrow a$

$B \rightarrow \varepsilon \mid c$

剩余的输入串

$bd\#$

b
 S
 $\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$ 剩余的输入串

$A \rightarrow a$

$d\#$

$B \rightarrow \varepsilon \mid c$

S
 $\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$ 剩余的输入串

$A \rightarrow a$

$d\#$

$B \rightarrow \varepsilon \mid c$

$d\#$

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



表驱动 LL (1) 分析

THSS

44100593

2019 / XS-301

✧ 表驱动预测分析过程举例

– 对于下列文法 $G(S)$:

$S \rightarrow AaS \mid BbS \mid d$ 剩余的输入串

$A \rightarrow a$

#

$B \rightarrow \varepsilon \mid c$



#

分析输入串 $aabd$ 的过程:

	a	b	c	d	#
S	$S \rightarrow AaS$	$S \rightarrow BbS$	$S \rightarrow BbS$	$S \rightarrow d$	
A	$A \rightarrow a$				
B		$B \rightarrow \varepsilon$	$B \rightarrow c$		



Conclusions

THSS

44100593

2019 / XS-301

- ✧ 语法分析的基本思想
- ✧ 带回溯的自顶向下分析
- ✧ 预测分析
- ✧ **First/Follow**集
- ✧ **LL(1)**文法
- ✧ 递归下降**LL(1)**分析
- ✧ 表驱动**LL(1)**分析



推荐教学资料

THSS

44100593

2019 / XS-301

✧ § 4.1.1 The Role of the Parser

✧ § 4.3 Writing a Grammar

✧ § 4.4 Top-Down Parsing

✧ ANTLR



Thank you!