

# 任务3.4 LL(1)文法判定与预测分析器设计及实现

## 一、实验目的

- 理解LL(1)文法的概念及其在语法分析中的应用。
- 掌握判定文法是否为LL(1)的方法。
- 学习设计和实现LL(1)预测分析器的过程。
- 培养运用编程语言实现自顶向下语法分析的能力。

## 二、实验内容

实现LL(1)文法的判定算法和预测分析器的设计与实现。具体步骤包括：

1. 输入上下文无关文法。
2. 判断文法是否为LL(1)。
3. 构造预测分析表。
4. 实现预测分析器，能够根据输入串进行语法分析。

## 三、实验要求

1. 输入一个上下文无关文法，包括非终结符、终结符和产生式。
2. 在任务3.1-3.3基础上来实现判断
3. 输出文法是否为LL(1)的判断结果。
4. 输出预测分析表。
5. 输入一个字符串，输出语法分析结果（是否成功以及分析过程）。【可以结合课程的实际例子运行，比如表达式文法】

## 四、实验指南

### LL(1)文法判定

1. 定义：
  - 文法为LL(1)的条件是对于每个非终结符A及其产生式 $A \rightarrow \alpha \mid \beta$ ，满足以下条件：
    - $FIRST(\alpha) \cap FIRST(\beta) = \emptyset$
    - 如果  $\epsilon$  在  $FIRST(\alpha)$  或  $FIRST(\beta)$  中，则需满足  $FOLLOW(A) \cap FIRST(\beta) = \emptyset$  或  $FOLLOW(A) \cap FIRST(\alpha) = \emptyset$ 。
2. 算法流程：
  - 计算文法的FIRST集和FOLLOW集。
  - 对于每个非终结符的所有产生式，检查是否满足LL(1)条件。

### 预测分析器设计

### 1. 构造预测分析表：

- 对于每个产生式  $A \rightarrow \alpha$ ，将其加入分析表  $M[A, a]$  中，其中  $a$  是  $FIRST(\alpha)$  中的第一个终结符。
- 如果  $\epsilon$  在  $FIRST(\alpha)$  中，则还需将  $A \rightarrow \alpha$  加入  $M[A, b]$  中，其中  $b$  是  $FOLLOW(A)$  中的每个终结符。

### 2. 实现预测分析器：

- 使用栈结构实现分析器，初始时将文法的开始符号压入栈中。
- 持续进行如下操作，直到栈为空：
  - 如果栈顶符号是终结符且与输入符号匹配，弹出栈顶并移动输入符号。
  - 如果栈顶符号是非终结符，从预测分析表中查找对应的产生式，替换栈顶符号为产生式右侧的符号。
  - 如果无法进行匹配，则语法分析失败。

## 示例

给定文法：

```
1  1. S → AB
2  2. A → aA | ε
3  3. B → b
```

#### • 计算得到：

- $FIRST(S) = \{a, b\}$
- $FIRST(A) = \{a, \epsilon\}$
- $FIRST(B) = \{b\}$
- $FOLLOW(S) = \{\$ \}$
- $FOLLOW(A) = \{b\}$
- $FOLLOW(B) = \{\$ \}$

#### • 预测分析表如下：

非终结符	a	b	\$
S	$S \rightarrow AB$	$S \rightarrow AB$	
A	$A \rightarrow aA$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
B		$B \rightarrow b$	

## 五、难点分析

- 判定文法是否为LL(1)时，需准确计算FIRST集和FOLLOW集，且确保文法的各个产生式满足LL(1)条件。
- 预测分析器的实现需要处理输入串的逐步匹配，确保栈的操作正确无误。

## 伪代码示例【仅供参考，总体流程参考课件和教材】

以下是LL(1)文法判定和预测分析器的伪代码示例：

```
1  FUNCTION is_LL1(grammar):
2      FIRST_sets = compute_FIRST(grammar)
3      FOLLOW_sets = compute_FOLLOW(grammar)
4      FOR each non-terminal A in grammar:
5          FOR each production  $A \rightarrow \alpha \mid \beta$ :
6              IF  $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) \neq \emptyset$  THEN
7                  RETURN false
8              IF  $\epsilon$  in  $\text{FIRST}(\alpha)$  THEN
9                  IF  $\text{FOLLOW}(A) \cap \text{FIRST}(\beta) \neq \emptyset$  THEN
10                     RETURN false
11      RETURN true
12
13 FUNCTION LL1_parser(input_string, grammar):
14     stack = [grammar.start_symbol]
15     input_buffer = input_string + "$"
16     WHILE stack is not empty:
17         top = stack.pop()
18         current_symbol = input_buffer[0]
19
20         IF top is terminal THEN
21             IF top == current_symbol THEN
22                 input_buffer = input_buffer[1:]
23             ELSE
24                 RETURN "Syntax Error"
25         ELSE IF top is non-terminal THEN
26             production = M[top][current_symbol] // 从预测分析表中查找
27             IF production is not empty THEN
28                 FOR each symbol in reversed production:
29                     stack.push(symbol)
30             ELSE
31                 RETURN "Syntax Error"
32     RETURN "Parsing Successful"
```