

任务3.3 上下文无关文法FIRST集和FOLLOW集求解及实现

一、实验目的

- 理解上下文无关文法中FIRST集和FOLLOW集的概念及其在语法分析中的重要性。
- 掌握计算文法中FIRST集和FOLLOW集的算法及其实现。
- 培养分析和解决文法问题的能力。

二、实验内容

实现求解上下文无关文法的FIRST集和FOLLOW集的算法。具体步骤包括：

1. 输入上下文无关文法。
2. 计算每个非终结符的FIRST集。
3. 计算每个非终结符的FOLLOW集。

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. 输出每个非终结符的FIRST集和FOLLOW集。
3. 算法应考虑文法的各种情况，确保输出结果准确。

四、实验指南

FIRST集计算

1. 基本规则：
 - 对于每个终结符 a ，将 $FIRST(a)$ 设置为 $\{a\}$ 。
 - 对于每个非终结符 A ，根据其产生式进行计算：
 - 如果产生式为 $A \rightarrow a\alpha$ ，则将 a 加入 $FIRST(A)$ 。
 - 如果产生式为 $A \rightarrow B_1B_2 \dots B_n$ ，则依次计算 B_1 的 $FIRST$ 集。若 B_1 的 $FIRST$ 集中包含空串 ϵ ，则继续计算 B_2 以此类推。

FOLLOW集计算

1. 基本规则：
 - 对于开始符号 S ，将 $\$$ 加入 $FOLLOW(S)$ 。
 - 初始化每个非终结符的 $FOLLOW(S)$ 集为空。
2. 迭代计算过程：
 - 设定一个标志变量 `changed` 用于检测 $FOLLOW$ 集是否发生变化。
 - 反复进行以下步骤，直到没有任何 $FOLLOW$ 集发生变化为止：
 - 遍历每个产生式 $A \rightarrow \alpha B \beta$ ：

- 将 $FIRST(\beta)$ 中的所有符号（去除空串）加入 $FOLLOW(B)$ 。
- 如果 β 可以推导出空串 ε ，则将 $FOLLOW(A)$ 中的所有符号加入 $FOLLOW(B)$ 。

3. 具体步骤：

- 初始化：对每个非终结符 X ， $FOLLOW(X) = \{\}$ 。
- 开始符号处理：将 $\$$ 加入 $FOLLOW(S)$ ，其中 S 是开始符号。
- 迭代更新：

```

1  REPEAT
2      changed = false
3      FOR each production  $A \rightarrow \alpha B \beta$ :
4          IF  $\beta$  is not empty THEN
5               $FIRST(\beta) = \{\text{取出} FIRST(\beta) \text{ 的所有符号 (排除空串)}\}$ 
6               $FOLLOW(B) += FIRST(\beta)$  (excluding  $\varepsilon$ )
7          IF  $\varepsilon$  in  $FIRST(\beta)$  THEN
8               $FOLLOW(B) += FOLLOW(A)$ 
9          IF  $FOLLOW(B)$  has changed THEN
10             changed = true
11 UNTIL changed is false

```

示例

给定文法【复杂的例子见课件或教材】：

```

1  1.  $S \rightarrow AB$ 
2  2.  $A \rightarrow a \mid \varepsilon$ 
3  3.  $B \rightarrow b$ 

```

计算结果：

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

五、难点分析

- 在计算 $FIRST$ 集时，特别是处理递归产生式和空串的情况，需要确保算法的准确性和收敛性。
- 在计算 $FOLLOW$ 集时，需要正确处理符号间的关系，确保所有符号的 $FOLLOW$ 集能准确反映文法的推导规则。