

任务3.1 上下文无关文法左递归消去算法的实现

一、实验目的

- 理解上下文无关文法中左递归的概念及其对语法分析的影响。
- 掌握消去上下文无关文法中直接和间接左递归的算法。
- 培养运用编程语言实现文法变换的能力。

二、实验内容

实现消去上下文无关文法中所有左递归的算法。具体步骤包括：

1. 对非终结符集合进行排序。
2. 按顺序遍历每个非终结符，检查其候选式是否以排在其前面的非终结符开头，并进行代换。
3. 消去直接左递归。

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。【为了跟后续实验贯穿，建议仔细设计良好的数据结构来表示文法】
2. **输出：**消去左递归后的文法。
3. **算法：**应处理直接和间接左递归，确保输出文法与输入文法等价。
4. **测试：**提供测试用例，验证算法的正确性。
5. **文档：**编写文档，说明如何使用你的程序，包括输入格式和输出解释。

四、实验指南

算法流程概述：【详细描述参考理论课课件或教材】

1. **非终结符排序：**对非终结符进行字典顺序排序（或者按某个其他特性排序，如出现顺序等）。
2. **遍历非终结符：**
 - 对于每个非终结符 A_i ：
 - 查找其候选式是否有其他非终结符（排在 A_i 前面的非终结符）作为开头：
 - 如果有，将候选式中该非终结符为首用该非终结符的产生式替代，从而将可能的间接左递归转化为直接左递归。
 - 如果没有，跳过该候选式。
 - 对所有直接左递归的候选式进行处理，生成新非终结符，并转换产生式，使其无左递归。

算法描述

消除左递归的算法通常包括以下步骤：

1. **识别直接左递归**：文法规则形式为 $A \rightarrow A\alpha \mid \beta$ ，其中 A 是非终结符， α 和 β 是任意的文法符号序列（终结符或非终结符），并且至少有一个 β 不以 A 开头。要消除直接左递归，可以引入一个新的非终结符 A' 来替换原产生式的左递归部分，并重新定义 A 的产生式如下：
 - $A \rightarrow \beta A'$
 - $A' \rightarrow \alpha A' \mid \epsilon$
2. **识别间接左递归**：间接左递归的文法规则可能没有直接的形式 $A \rightarrow A\alpha$ ，但是通过一系列的推导， A 可以推导出以 A 开头的字符串。消除间接左递归通常需要将间接左递归转换为直接左递归，然后应用直接左递归的消除方法。
3. **转换产生式**：对于每个存在左递归的非终结符，引入新的非终结符作为辅助符号，并重新定义产生式以消除左递归。

输入输出样例

输入：

```
1 S -> S+T | T
2 T -> T*F | F
3 F -> (E) | id
```

输出：

```
1 S -> T S'
2 S' -> +T S' | ε
3 T -> F T'
4 T' -> *F T' | ε
5 F -> (E) | id
```

重点难点

- 设计文法的数据结构，特别是产生式规则的左部和右部如何表示，才能更方便进行替换
- 文法输入形式，并转换为相应的数据结构
- 如何验证转换出来的文法与原始文法是等价的？【可探讨】