Syntax Parser Programming

57121210潘梓昕

(a) Motivation/Aim

本实验旨在实现一个基于上下文无关文法的语法分析器。我们将通过自顶向下和自底向上的分析方法，解析输入的字符流，输出相应的推导过程或归约过程。通过该实验，我们希望加深对语法分析的理解，掌握两种常见的语法分析技术，并学习如何应用它们来处理编程语言的解析。

(b) Content Description

本报告描述了一个简单的上下文无关文法分析器的实现。通过定义一个简单的文法（如S -> A B、A -> a、B -> b等），实现了两种类型的语法分析：自顶向下分析和自底向上分析。报告中还将详细描述实验的过程、核心算法、数据结构以及遇到的问题和解决方案。

(c) Ideas/Methods

自顶向下分析（Top-Down Parsing）：从起始符号出发，尝试匹配输入流中的字符。如果匹配成功，则记录推导过程。

自底向上分析（Bottom-Up Parsing）：从输入流的字符出发，逐步尝试归约为文法中的非终结符。

错误处理：在无法匹配的情况下，分析器会返回错误信息，提示输入流无法根据当前文法进行推导或归约。

(d) Assumptions

假设输入的字符流符合文法规则，且文法是上下文无关文法。

该实验仅支持有限的文法规则和简单的字符流，不处理复杂的错误处理和优化问题。

(e) Related FA Descriptions

虽然本实验涉及上下文无关文法的语法分析，但其核心原理与有限自动机（FA）相似。自顶向下分析可以类比为有限自动机的状态转移，而自底向上分析则类似于通过状态的归约来完成解析。

(f) Description of Important Data Structures

文法规则：采用字典存储文法规则，其中每个非终结符对应一个生成式的列表。

推导过程（自顶向下）：存储每一步的推导过程，记录每个非终结符如何被展开。

归约过程（自底向上）：存储每一步的归约过程，记录如何将终结符逐步归约为非终结符。

(g) Description of Core Algorithms

自顶向下推导：

从起始符号开始，逐步展开并尝试匹配输入流。

如果成功匹配，记录推导过程。

如果无法匹配，回退并尝试其他推导规则。

自底向上归约：

从输入流开始，逐步识别可以归约的子串。

如果匹配到某个生成式的右侧，则进行归约，替换为对应的非终结符。

继续归约直到输入流只剩下起始符号。

(h) Use Cases on Running（运行中的用例）

输入流：aabb

S -> AB

A -> a

A -> a

B -> b

B -> b

选择分析方法：自顶向下分析

输出结果：

Reduce: a -> A

Reduce: a -> A

Reduce: b -> B

Reduce: b -> B

(i) Problems Occurred and Related Solutions（遇到的问题及相关解决方案）

问题：如何处理输入流无法匹配任何规则的情况？

解决方案：在无法匹配时，分析器会返回错误信息，提示输入无法解析。

问题：如何优化归约过程？

解决方案：可以通过增加回溯机制和优化文法规则来提高归约效率。

(j) Your Feelings and Comments（个人感受与评论）

通过本实验，我对上下文无关文法的语法分析有了更深的理解，尤其是对自顶向下和自底向上分析方法的实现过程有了更加清晰的认识。虽然实现过程中遇到了一些问题，但通过不断调试和优化，最终能够正确地处理输入流。未来我将进一步学习更复杂的语法分析技术，并尝试将其应用到实际的编译器中。