

Top-down Parsing 简介



什么是top-down parsing?

Top-down parsing（自顶向下解析）是一种语法解析技术，它从**文法的起始符号**（Start Symbol）开始，逐步尝试匹配输入字符串，按照**语法规则**推导出终结符序列。如果推导出的终结符序列与输入匹配，则解析成功，否则解析失败。

主要特点

1. **从起始符号出发**，按规则推导输入字符串，而不是从输入字符串反向构造语法树。
2. **使用递归或迭代**，按照语法规则展开非终结符。
3. **适用于LL文法**（Left-to-right, Leftmost derivation，即从左到右扫描输入，采用最左推导）。

常见的 Top-down 解析方法

1. **递归下降解析（Recursive Descent Parsing）**
 - 使用递归函数实现。
 - 解析器针对每个非终结符编写一个递归函数。
 - 适用于**简单 LL(1) 文法**（即文法是确定的，每个非终结符的产生式可以通过一个符号决定）。
2. **预测分析（Predictive Parsing）**
 - 主要用于**LL(1) 文法**，避免了回溯（Backtracking）。
 - 通过****预测分析表（Predictive Parsing Table）****实现解析。

示例

假设有一个简单的文法：

```
r
E → T + E | T
T → int | (E)
```

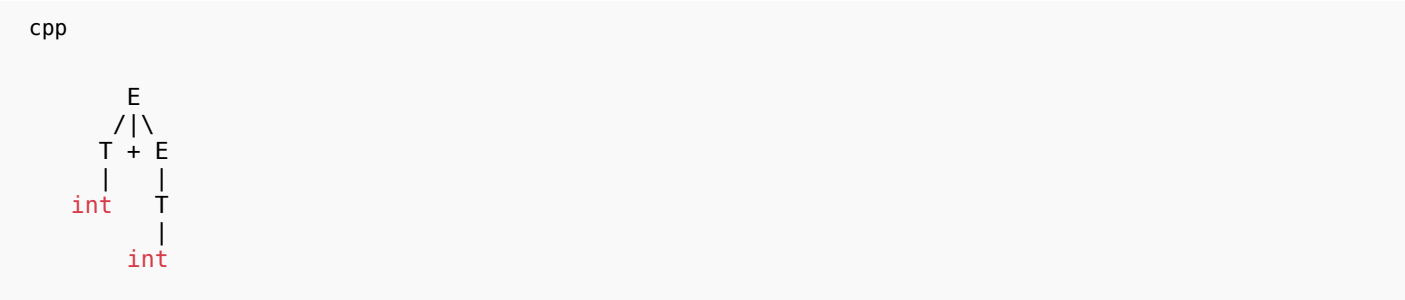
输入字符串：`int + int`

递归下降解析的步骤：

1. 从 `E` 开始：
 - `E → T + E`
2. `T → int`
3. 读取 `+`
4. 解析 `E`：
 - `E → T`

5. `T → int`

这样，构建出的**语法树**大致如下：



最终匹配 `int + int`，解析成功。

Top-down Parsing vs Bottom-up Parsing

比较点	自顶向下解析 (Top-down)	自底向上解析 (Bottom-up)
解析方式	从起始符号开始推导	从输入串归约回起始符号
适用文法	LL(k) 文法	LR(k) 文法
解析器类型	递归下降、预测分析	移进-归约 (Shift-Reduce)、LR 解析
是否回溯	可能需要回溯 (LL(1) 无需回溯)	无需回溯
复杂度	适用于简单文法	适用于更复杂的文法

总结

Top-down parsing 是一种**从文法的起始符号**出发的解析技术，递归下降和预测分析是其两种主要方法。它适用于 **LL 文法**，而不像 Bottom-up parsing 那样适用于更复杂的 **LR 文法**。