

中期报告(语法分析器)

项目模块

1. 根据上下文无关语法计算 First 集合和 Follow 集合
2. 根据上下文无关语法构建 NFA, 生成状态转移表
3. 基于先前预处理结果, 对终结符序列进行移入规约操作, 输出移入规约结果和步骤
4. 构造测试数据, 使用测试程序测试语法分析器正确性

模块接口

使用全局变量存储上下文无关语法:

```
// 上下文无关语法
int number_of_symb; /* number of symbols */
int number_of_prod; /* number of productions */
struct prod grammar[MAX_NUMBER_OF_PROD];
```

使用全局变量存储 First 集合和 Follow 集合, 使用预处理函数生成:

```
bool follow[MAX_NUMBER_OF_SYMB][MAX_NUMBER_OF_SYMB];
bool first[MAX_NUMBER_OF_SYMB][MAX_NUMBER_OF_SYMB];

void get_first();
void get_follow();
```

根据上下文无关语法生成状态转移表, 根据状态转移表判断扫描线左侧是否为可行结构:

```
void pre_trans(); // 预处理出状态转移表以及所有结点上的状态信息

bool check_feasible(struct array a); // 用于检查给定的字符序列 a 是否是可行结构，会用到状态转移表
```

在 main 函数中实现移入规约算法，将输出算法每步的具体操作。

已实现部分

- 基本读入函数和调试函数，实现从文件读入上下文无关语法并存储在 `struct prod grammar` 中
- 已实现生成状态转移表的模块。完成 `pre_trans` 和 `check_feasible` 函数
- 实现了简单的测试程序，构造了部分测试数据

实现中的部分

- 完成 First 集合 和 Follow 集合的计算
- 完成在获取到预处理结果后，移入规约算法的逻辑
- 增加测试数据，完善测试程序，实现多数据测试