编译原理实践第4次课
自顶向下的语法分析

张昊　1927405160

概述

使用 Python3 实现了自顶向下的语法分析，使用**递归下降**法生成语法树，并将语法树打印成字符串形式。

项目只有一个源文件：`recursive_descent_analysis.py`

递归下降语法分析的文法为：

```
E → TE'
E' → +TE' | e
T → F T'
T' → * F T' | e
F → (E) | id
```

编程环境说明

- **语言**：Python3
- **文件编码**：UTF-8
- **测试环境**：Python 3.8.10

运行示例

文件从标准输入读取输入的词法单元，并将分析结果输出到标准输出。若出现错误，则向标准错误输出错误信息。

运行方法：

```
$ python3 ./recursive_descent_analysis.py
```

运行实例1

输入：

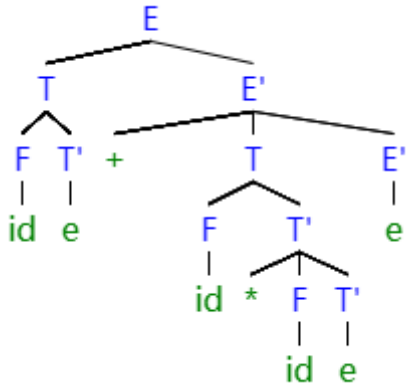
```
id + id * id
```

输出：

```
[E[T[F[id]][T'[e]]][E'[+][T[F[id]][T'[*][F[id]][T'[e]]]][E'[e]]]]
```

```
(base) holger@HOLGER-HONOR:/mnt/d/Code/compiler/recursive-descent$ python3 ./recursive_descent_analysis.py
Please input the lexical units: id + id * id
[E[T[F[id]][T'[e]]][E'[+][T[F[id]][T'[*][F[id]][T'[e]]]][E'[e]]]]
```

树状图：



运行实例2

输入：

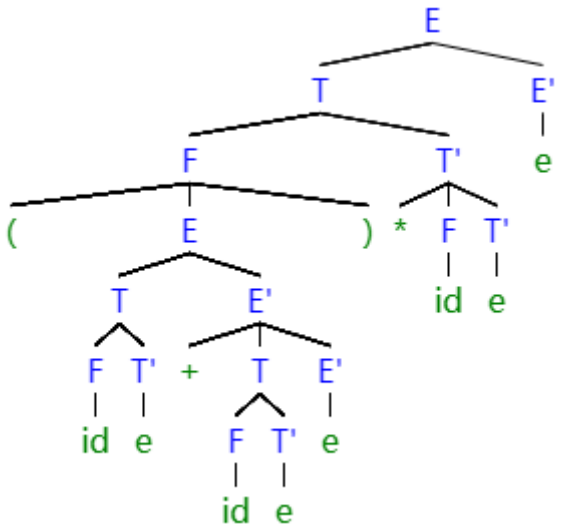
```
( id + id ) * id
```

输出：

```
[E[T[F[(][E[T[F[id]][T'[e]]][E'[+][T[F[id]][T'[e]]][E'[e]]]][)]][T'[*][F[id]]
[T'[e]]]][E'[e]]]
```

```
(base) holger@HOLGER-HONOR:/mnt/d/Code/compiler/recursive-descent$ python3 ./recursive_descent_analysis.py
Please input the lexical units: ( id + id ) * id
[E[T[F[(][E[T[F[id]][T'[e]]][E'[+][T[F[id]][T'[e]]][E'[e]]]][)]][T'[*][F[id]][T'[e]]]][E'[e]]]
```

树状图：



运行实例3

输入：

```
id *
```

输出：

```
Invalid Input at "$"!
```

```
(base) holger@HOLGER-HONOR:/mnt/d/Code/compiler/recursive-descent$ python3 ./recursive_descent_analysis.py
Please input the lexical units: id *
Invalid Input at "$"!
```

输入不合法，返回错误信息。