

编译原理实践第7次课

自下而上的语法分析-1

张昊 1927405160

[概述](#)

[编程环境说明](#)

[运行](#)

概述

使用 Python3 实现了概率上下文无关文法（PCFG），以及**CYK算法**。

PCFG中定义一棵句法树的概率为所有用到的规则概率的乘积，一般来说，概率值大的更可能是正确的句法树。

CYK算法是一个基于“动态规划”算法设计思想，用于测试串 w 对于一个上下文无关文法 L 的成员性的一个算法。CYK算法可以在 $O(n^3)$ 的时间内得出结果。CYK算法是由三个独立发现同样思想本质的人（J. Cocke、D. Younger和T. Kasami）来命名的。

给定文法：

```
S -> NP VP 0.9
S -> VP 0.1
VP -> V NP 0.5
VP -> V 0.1
VP -> V @VP_V 0.3
VP -> V PP 0.1
@VP_V -> NP PP 1.0
NP -> NP NP 0.1
NP -> NP PP 0.2
NP -> N 0.7
PP -> P NP 1.0
N -> people 0.5
N -> fish 0.2
N -> tanks 0.2
N -> rods 0.1
V -> people 0.1
V -> fish 0.6
```

```
V -> tanks 0.3
P -> with 1.0
```

基于上述文法和CYK算法，编程求句子 `fish people fish tanks` 的最优分析树。

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

项目源文件为： `main.py` 。

主程序接受一个参数，是保存有配置项的 json 文件，其内容如下：

```
{
  "grammar_file": "grammar.txt", // 文法定义
  "non_terminals": [ // 非终结符
    "S",
    "NP",
    "VP",
    "N",
    "V",
    "PP",
    "P",
    "@VP_V"
  ],
  "start_symbol": "S" // 开始符号
}
```

其中文法定义如下：

产生式头部 -> 产生式尾部 概率

程序从标准输入读取一行字符串，进行语法树的构建。运行方法如下：

```
$ python3 main.py grammar.json
```

或

```
$ cat input.txt | python3 main.py grammar.json
```

输入如下（保存至 `input.txt` 文件）：

```
fish people fish tanks
```

输出如下：

```
句子: fish people fish tanks
```

```
最优分析树: [S [NP [NP [N[fish]]] [NP [N[people]]]] [VP [V[fish]] [NP [N[tanks]]]]]
```

```
最优分析树的概率: 0.00018522
```

```
/Users/holger/codes/pcfg/venv/bin/python main.py grammar.json
```

```
fish people fish tanks
```

```
句子: fish people fish tanks
```

```
最优分析树: [S [NP [NP [N[fish]]] [NP [N[people]]]] [VP [V[fish]] [NP [N[tanks]]]]]
```

```
最优分析树的概率: 0.00018522
```

将该字符串形式生成树状图如下：

