Atitit 词法分析最佳实践bp v4 s529

目录

[1.1. 使用状态表比较简单，dfa比较麻烦 状态转移表 1](#_Toc5935)

[1.2. 递归的方法来进行词法分析 1](#_Toc13484)

[1.3. 双case算法 每个case拆分更具toke 和state组合 1](#_Toc25178)

[1.4. 内部分支switch替换if else 2](#_Toc28945)

[1.5. error  出错处理的函数，处理出现的词法错误。 2](#_Toc16153)

[1.6. Case state命名法规范tokenXXXCase tokenXX\_statXX模式 2](#_Toc25135)

[1.7. 单独的db que状态，内部可以容易包含single quo 2](#_Toc7761)

[1.8. 单词分类 保留字、标识符、、运算符，，，、分隔符 ，，，常数 2](#_Toc21229)

[1.9. Token结果规范 3](#_Toc11907)

[1.10. 分词主要流程 3](#_Toc21045)

[1.11. 词法分析的第一阶段即扫描器 第二阶段评估器（Evaluator） 3](#_Toc7086)

## 使用状态表比较简单，dfa比较麻烦 状态转移表

## 递归的方法来进行词法分析

优先递归 替换循环

递归可读性更好。。

## 双case算法 每个case拆分更具toke 和state组合

**public** List<Token> TokenizeProcess(List<String> tokens) {

**switch** (cur\_token.trim()) {

**case** "new":

TokenNew\_cash();

**break**;

**case** ".":

TokenPeriod\_case();

**break**;

**default**:

tokenNormalchar\_case();

**break**;

}

token\_index++;

**return** TokenizeProcess(tokens);

}

## 内部分支switch替换if else

## error  出错处理的函数，处理出现的词法错误。

1. 有一类词法错误可以在词法分析时发现，如出现字母表以外的非法字符、不合规则的常数、标识符的前缀为保留字等。但还有一类词法错误，例如，把if写成fi，词法分析会将fi当作标识符处理，le ngth中间多了一个空格，词法分析会将le和ngth当作两个标识符处理。这类词法错误往往要推迟到语法分析时才能发现，不属于函数处理的范畴。

词法分析就是取出一个个词，然后给词归类、给个种别码什么的。所以遇到不认识的词或符号，一般就会报错。

## Case state命名法规范tokenXXXCase tokenXX\_statXX模式

## 单独的db que状态，内部可以容易包含single quo

## 单词分类 保留字、标识符、、运算符，，，、分隔符 ，，，常数

1. 关键字：具有特殊含义的标识符。
2. （1）标识符：用户自定义的变量名、函数名等字符串。

如何区别关键字和标识符呢？

因为在很多语言中关键字和标识符是有交集的，但从词法分析上看，关键字是标识符的一部分

第一种，我们需要将上面的转移图稍加修改：（以识别if为例）

　　　　此处还应注意一个问题，假如输入ifxy呢？并不是关键字，因此这里还需判断。

再看第二种，使用关键字表，即Hash表。对语言中所有的关键字构建哈希表，对所有的关键字和标识符，先统一按标识符的转移表进行识别，识别完成后，再查哈希表看是否为关键字。通过构造合理的哈希表，可以再O(1)时间内完成。

## Token结果规范

重要三属性 "tokenIndex":2, "text":"<EOF>", "type":-1

{

"tokenIndex":2,

"text":"<EOF>",

"type":-1

"channel":0,

"charPositionInLine":15,

"inputStream":{"$ref":"$.tokenSource.charStream"},

"line":1,

"startIndex":15,

"stopIndex":15,

}

参考Atitit.antlr实现词法分析Atitit.docx

## 分词主要流程

字符切分，对比字典词典得到分词，并查阅词典，标注词性。完善其他属性，输出

当词法分析器发现了一个标识符的词素时，要将这个词素添加到符号表中。

## 词法分析的第一阶段即扫描器 第二阶段评估器（Evaluator）

词法分析的第一阶段即扫描器，通常基于有限状态自动机

词法分析的第二阶段评估器（Evaluator）

，语法分析器需要第二阶段的评估器（Evaluator）。评估器根据语素中的字符序列生成一个“值”，这个“值”和语素的类型便构成了可以送入语法分析器的单词。一些诸如括号的语素并没有“值”，评估器函数便可以什么都不返回。整数、标识符、字符串的评估器则要复杂的多。评估器有时会抑制语素，被抑制的语素（例如空白语素和注释语素）随后不会被送入语法分析器。