Atiit 如何手写词法解析器

[1.1. 通过编程直接从正则->nfa->dfa->表驱动词法解析一条龙自动生成。那是用程序自动生成是需要这样的，自己手写完全不必要这么复杂 1](#_Toc4785)

[1.2. 状态转移表。使用状态表比较简单，dfa比较麻烦。Dfa其实就是比较高级的状态表。。 1](#_Toc32450)

[1.3. 然后给了你代码框架（这里以nested case statement 为例）： 2](#_Toc16610)

[1.4. 源码实现 2](#_Toc18229)

## 通过编程直接从正则->nfa->dfa->表驱动词法解析一条龙自动生成。那是用程序自动生成是需要这样的，自己手写完全不必要这么复杂

尤其是scanner 的时候一上来就看各种自动机。  
直接回答你的疑问就是：在实际中手写词法分析器时，你所说的“RE -> NFA -> DFA -> Scanning Table” 一个都不会出现。原因有二：

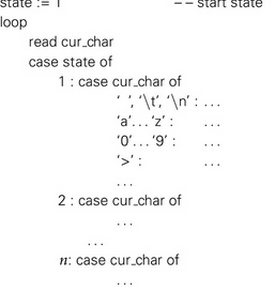
书上说的这么复杂的一系列计算都是为了做scanner generator（比如flex）。自动生成的scanner 一般有两部分，一部分是固定的一段代码，相当于一个interpreter，它读入scanning table 和源程序，生成一系列的token；另一部分就是scanning table，它直接对应你给的词法规则，而要通过“程序”生成这个table 就需要你说的那一长串计算。然而你手写scanner 的时候根本不用考虑这些  
  
 自己写解析器，正则什么的都不需要了解的。。

第一步：：做个状态转换表，就是当前状态什么，当前字符是什么，下一状态是什么就可以了。

## 状态转移表。使用状态表比较简单，dfa比较麻烦。Dfa其实就是比较高级的状态表。。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cur\_dbquo\_stat | 当前状态 | 当前字符 | 要即将转换到的下一状态 |  |
|  |  | \ |  |  |
|  | “ dbQuo\_start | ‘ | <none> |  |
|  | Not sQuo start  Not dbQuo\_start | ‘ | sQuo start |  |
|  | sQuo start | ‘ | sQuo end |  |
| Dbquo end or <non>  Not Dbquo start |  | “ | Dbquo start |  |
| Dbquo start |  | “ | Dbquo end |  |
|  | Non sQuo dbquo start | , |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 然后给了你代码框架（这里以nested case statement 为例）：



然后告诉你最外层case覆盖特定字符，内层每个case覆盖这个状态的所有转换。特别清晰简单有木有！！！编程时候直接填空就行了！

## 源码实现

**public** List<Token> **getTokens**(String codeStr) {

List<Token> **li** = Lists.*newArrayList*();

code\_char\_arr = codeStr.toCharArray();

**while** (**true**) {

Object **tk**;

**try** {

tk = nextTokens();

} **catch** (TokenEndEx **e**) {

**break**;

}

**if** (tk **instanceof** Token)

li.add((Token) tk);

**else** **if** (tk **instanceof** List)

li.addAll((Collection<? **extends** Token>) tk);

**else**

**throw** **new** RuntimeException("token type err,curchar:" + **this**.cur\_char + ",colidx:" + **this**.gColumn);

}

**return** li;

}

**public** Object **nextTokens**() **throws** TokenEndEx {

// code\_char\_arr = code.toCharArray();

gColumn++;

**if** (gColumn > code\_char\_arr.length - 1)

**throw** **new** TokenEndEx(**new** String(code\_char\_arr));

cur\_char = code\_char\_arr[gColumn];

// cur\_char=cur\_char;

**if** (**this**.curTokenTxt.equals("1598"))

System.***out***.println("dbg");

**if** (**this**.gColumn == 30)

System.***out***.println("dbg");

// get next char,,then change stat

// jude cur char and cur stat...then if or not chage stat

**switch** (cur\_char) {

**case** '(':

**return** BrkStartEvt();

// break;

**case** ')':

**return** brkEndEvt();

**case** '\'':

**return** sQuoEvt();

**case** '\"':

**return** dbQuoEvt();

**case** ':':

**return** colonEvt();

**case** ',':

**return** commaEvt();

**default**:

**return** normalCharEvt();

// break;

}

}

**private** Object **BrkStartEvt**() {

**char** **c** = **this**.cur\_char;

**if** (c == '(' && !**this**.curStat.equals("strStart")) { // && cur stta=ini

List<Token> **li** = Lists.*newArrayList*();

Token **tk** = **new** Token(**this**.curTokenTxt).setType("var");

li.add(tk);

Token **tk2** = **new** Token("(").setType("op");

li.add(tk2);

**this**.curTokenTxt = "";

**this**.curStat = "brkStart";

**return** li;

}

**throw** **new** RuntimeException("BrkStartEvt");

}

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ）

汉字名：艾提拉（艾龙），   EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

Atiend