```
def getItemGroup(): Unit = {
        val ldx = ( relations(0)._1, "·" + relations(0)._2, "#" )
        val I0 = getClosure( ArrayBuffer(ldx) )
        val wholeCharacters = allCharacters
        var tot = 0
        itemGroup(I0) = tot
        var appendFlag = true
        while (appendFlag == true) {
            var originalAns = Map[ ArrayBuffer[ (String, String, String) ], Int ]()
            originalAns = itemGroup.clone()
            //为什么用 I 作为遍历变量不行？！
            for(item <- itemGroup.keys) {
                for (ch <- wholeCharacters) {
                    val newItem = go(item, ch.toString).sorted
                    if (newItem.isEmpty == false && itemGroup.contains(newItem) == false) {
                        tot += 1
                        itemGroup(newItem) = tot
                    }
                }
            }
            if( originalAns.equals(itemGroup) == true ) {
                appendFlag = false
            }
            else {
                originalAns.clear()
                originalAns = itemGroup.clone()
            }
        }
    }
```

Scala 实现 createMatrix 函数：

```
def createMatrix(): Array[ Array[String] ] = {
        val result = initiateMatrix()
        val localVT = VT
        val localVN = VN
        case class getColumn( ch: String ) {
            val matrix = initiateMatrix()
            var ans = -1
            for( j <- 0 to (columnLength - 1) ) {
                if( matrix(0)(j) == ch ) {
                    ans = j
                }
            }
        }
```