```
}
                }
                if (result != originalResult) {
                        originalResult.remove(0, originalResult.length)
                        originalResult.appendAll(result)
                        cnt += 1
                }
                else {
                        addFlag = false
                        cnt += 1
                }
        }
        result
    }
```

Scala 实现 go 函数：

```
def go( I: ArrayBuffer[ (String, String, String) ], X: String ): ArrayBuffer[ (String, String, String) ] = {
        //GO(I, X) = CLOSURE(J)
        //J = {任何形如[A->α X • β , a]的项目|[A->α • X β , a]∈I}
        val ans = new ArrayBuffer[ (String, String, String) ]()
        val items = new ArrayBuffer[ (String, String, String) ]()

        for( ex <- I ) {
                val pointPosition = ex._2.indexOf("·")
                //•  不在最右边
                if (pointPosition < ex._2.length - 1) {
                        val A = ex._1
                        val possibleX = ex._2( pointPosition + 1)
                        //    αXβ
                        val noPointExpressionPart2 = ex._2.replace("·", "")
                        if( X == possibleX.toString ) {
                                //    αX·β
                                val newPart2 = noPointExpressionPart2.substring(0, pointPosition + 1) + "·" +
                                        noPointExpressionPart2.substring(pointPosition + 1,
noPointExpressionPart2.length)
                                val a = ex._3
                                items += ( (A, newPart2, a) )
                        }
                }
        }
        ans.appendAll( getClosure(items) )
        ans
    }
```