```
def getClosure( items: ArrayBuffer[ (String, String, String) ] ): ArrayBuffer[ (String, String, String) ] = {
        val result = new ArrayBuffer[ (String, String, String) ]()
        result.appendAll(items)
        val localFIRST = FIRST()
        var addFlag = true
        var cnt = 1
        while (addFlag == true ) {
            val originalResult = new ArrayBuffer[(String, String, String)]()
            originalResult.appendAll(result)
            for (ex <- result) {

                val pointPosition = ex._2.indexOf("·")
                //·  不在最右边
                if (pointPosition < ex._2.length - 1) {
                    //B 在  ·  的右边
                    val B = ex._2(pointPosition + 1)
                    val a = ex._3

                    // case 1: β != Φ and a != # or
                    // case 2: β != Φ and a = #
                    if (pointPosition < ex._2.length - 2) {
                        val β = ex._2(pointPosition + 2)
                        //  ξ
                        val rightExpressionsOfB = getRightExpressions(B.toString)
                        val FIRST_Of_βa = localFIRST(β.toString)
                        for (b <- FIRST_Of_βa) {
                            for (ksi <- rightExpressionsOfB) {
                                val tmp = ((B.toString, "·" + ksi, b.toString))

                                if (result.contains(tmp) == false) {
                                    result += tmp
                                }
                            }
                        }
                    }
                    // case 3: β = Φ and a equals any character
                    if (pointPosition == ex._2.length - 2) {
                        val rightExpressionsOfB = getRightExpressions(B.toString)
                        val FIRST_Of_βa = localFIRST(a.toString)
                        for (b <- FIRST_Of_βa) {
                            for (ksi <- rightExpressionsOfB) {
                                val tmp = ((B.toString, "·" + ksi, b.toString))
                                if (result.contains(tmp) == false) {
                                    result += tmp
                                }
                            }
                        }
                    }
```