1. All of the productions that exhibit left-recurision:
   * decl: string\_decl decl | var\_decl decl | ;
   * id\_tail: ',' id id\_tail | ;
   * param\_decl\_tail: ',' param\_decl param\_decl\_tail | ;
   * func\_declarations: func\_decl func\_declarations | ;
   * stmt\_list: stmt stmt\_list | ;
   * expr\_prefix : expr\_prefix factor addop | ;
   * factor\_prefix : factor\_prefix postfix\_expr mulop | ;
   * expr\_list\_tail : ',' expr expr\_list\_tail | ;
2. To combat not knowing which operator is supposed to be in what order, so ANTLR rewrites the left-recursive rule into a non-recursive rule using precedence of previous operators to judge the precedence of the current operator. A good example of this is for a grammar rule for an expression of basic arithmetic, (e.g. 5+3\*(4-3)), it would come out in the order (4-3), 3\*(1), 5+3 = 8.code
3. ALL(\*) (“all star”), it’s an extension to LL(\*) that parses and analyzes grammar on the fly to have access to input sequences, rather than doing so statically. ALL(\*) parsers can recognize the order of input sequences by reading through the grammar.
4. A screenshot of a computer

   Description automatically generated with medium confidenceGraphical user interface, text

   Description automatically generatedANTLR generates trees that utilizes top down parsing, methods are made from each rule that on methods within them,