在上学期编译原理课中涉及到的语法分析有自顶向下分析法（比如LL（1）），自底向上的语法分析（比如LR）。该文档在解析Kaleidoscope的语法时，运用了递归下降分析法即自顶向下分析法+运算符优先级解析。递归下降分析利用函数之间的递归调用模拟语法树自上而下的构造过程，每个非终结符的处理对应一个递归过程（函数），分析过程就是从根节点出发，自顶向下为输入串寻找一个最左匹配序列，建立一棵语法树。语法分析器就是将词法分析器输出的单词串起来，形成AST。首先要建立一个语法树的基类，之后再根据需要创建用于保存变量名，运算符和函数名以及用作参数的表达式等的子类。之后便是对语法树的解析，需要我们解析的包括数值常量，变量，二元表达式，在涉及到二元表达式时还要考虑优先级的问题。另外还需要扩展例子中的相应报错的辅助函数帮助处理解析过程中产生的错误。在处理复杂的二元表达式时，首先定义运算符的优先级，因此先制定出优先级表，在进行运算符优先级解析时，表达式将按二元运算符分隔，形成一串有序对，而此时需要函数用来解析这些有序对，该函数会优先检查当前语元的优先级，如果优先级过低就直接返回，并且只有在检查通过的时候，可以确定当前运算符为二元运算符，可以被纳入表达式中，在一次处理完二元运算符后，有序对中相应的表达式也解析完成，以上操作也就是先对表达式进行处理，判断是否为二元表达式，在解析完成后，就需要考虑表达式的结合次序，而此时就需要根据优先级表来判断执行的前后顺序了。之后在完成其他结构的解析后，加上引导代码，也就是在最外层的循环中按当前语元的类型选定相应的解析函数，这样语法解析器就可以成功运行了。