

A análise léxica é a primeira parte do processo, o objetivo dessa etapa é ler o código fonte caractere por caractere e a partir disso gerar os tokens, que são sequências de unidades. Como isso é feito ? Primeiramente um código fonte de entrada, no caso da aula usamos na linguagem C. Então identificamos as palavras-chaves, como nome de variável, operadores, números e outros símbolos. No final teremos a saída, que consiste na sequência de tokens descritos em uma tabela de símbolos, esse passo é essencial para futuras etapas como a análise semântica.

A segunda parte é a análise sintática, ela é responsável por receber a sequência de tokens gerados pela análise léxica e verificar se eles formam uma estrutura válida de acordo com as regras.

Assim como anteriormente, podemos separar por algumas etapas. Primeiramente nossa entrada será a sequência de tokens, para então o analisador agrupar os tokens em trechos gramaticais, para então verificar se a ordem e combinação faz sentido. Por exemplo, garante que um comando `if` seja seguido por uma condição entre parênteses ou que uma variável tenha uma atribuição do lado direito. E por último a saída, que será uma estrutura em formato de árvore do programa.

A terceira parte é a análise semântica, seu principal foco é analisar o “significado” do código, basicamente se faz sentido lógico. Nesta etapa estaremos usando a árvore da fase anterior como entrada, e vai percorrê-la relacionando os identificadores com seus dependentes de acordo com a estrutura de hierarquia. Isso quer dizer que a verificação de tipo (não permitir a soma de um texto com booleano por exemplo), verificação de escopo (se variável e funções foram declaradas antes de serem usadas), e verificação de unicidade (declaração duplicada por ex) são feitas nessa etapa.