**Resumo Sobre As Fases de Análise de um Código**

**Rian Beskow Friedrich**

**Universidade Franciscana - UFN**

A analise de um código de um programa é dividido entre três fases: Analise léxica, Analise sintática e analise semântica.

**Analise léxica:** É a parte de reconhecimento e definição de cada token (sequência de caracteres com significado) do código. Ou seja, identificação do que é cada palavra definida no código, e por isso, por ser algo sequencial e deve de qualquer forma, ser ordenada, é normalmente representada por uma tabela, onde sequencialmente, cada palavra do código, é definida, reconhecida, e explicada, e entre várias outras informações, nesta tabela.

**Análise Sintática:** Após a definição e reconhecimento de cada token, esta parte verifica se há um sentido (formação de identificação) correto em cada linha do código. Em outras palavras, é como se esta analise, reconhecesse frases feitas, na qual é reconhecida, cada função feita, como por exemplo: int x = 10; , na qual, com esta análise, se reconhece que é declarado que há uma palavra reservada inteira chamada que tem o valor de 10 e tem ; para dar continuidade a sequência. Nesta análise, também é comummente usado autômatos para associação de como é feito o processo, pois com um autômato, é possível construir uma forma linear organizada de como é construída cada linha de comando do código e se é feita de maneira que que dê para reconhecer o que é escrito e se faz sentido.

**Análise Semântica:** A análise semântica verifica se o código possui significado e se é consistentemente logico, assim, complementando a análise sintática. Pois enquanto a análise sintática verifica se cada linha consegue sequencialmente ser reconhecida, a analise semântica verifica se a linha feita realmente faz sentido e se é logicamente aceitável como uma parte complementar do código, seguindo sua lógica e raciocínio. Para uma melhor compreensão, a lógica de arvores de derivação podem ser implementadas, na qual, a linha do código é dividida em varias partes para se analisar o que cada parte faz ou se chama outra parte e se as funções são operadas de maneira com que é logicamente aceitável no programa.