



Figura 1. (Tipos de análises em compiladores)

1 – Análise Léxica

2 – Análise Sintática

3 – Análise Semântica

Análise Léxica

Na análise léxica, o código é dividido em uma tabela de tokens, onde podemos visualizar de forma separada conteúdos como id, token, tipoVariável, linha, coluna e dispensamos também espaços em branco e comentários do código, nesse processo utilizamos autômatos finitos e expressões regulares para poder identificar as sequencias de caracteres.

	ID	token	tipo	l	c
0	1	int	nome_variavel	1	1
1	2	x	nome_variavel	1	5
2	3	=	atribuicao	1	6
3	4	10	inteiro	1	7
4	5	;	ponto_virgula	1	9

Figura 2. Tabela de tokens gerada na análise léxica

Análise Sintática

Na análise sintática, após análise léxica, verificamos se a sequencia de símbolos está organizada de acordo com as regras gramaticais da linguagem definida, utilizando uma árvore de derivação para realizar a confirmação de que está de fato correto o que foi analisado.

```
int x = 0, y, z = 5, w, t = 10;
```

DECLARAÇÃO

V

Declaração

V

[TIPO] > [NV] > [SA] > [VALOR] > [DIM] [PV]

V

[VIRGULA] > [NV] > [DIM]

V

[VIRGULA] > [NV] > [SA] > [VALOR] > [DIM]

V

[VIRGULA] > [NV] > [DIM]

V

[VIRGULA] > [NV] > [SA] > [VALOR]

Figura 3. Árvore de derivação na análise sintática

Análise Semântica

Na análise semântica é a etapa onde é verificado o sentido do que foi definido anteriormente na análise sintática, é analisado se as variáveis foram corretamente declaradas antes de serem utilizadas, e também analisando atribuições, como a atribuição direta, a atribuição por retorno e a atribuição por passagem de parâmetro, para assim saber se o dado que foi passado como valor está de fato condizente com o que foi definido.

NF	Param	ERRO!
Calcular-fat	"a"	
printf	"abc"	
Soma	10, 20	
main		

Figura 4. Exemplo de comparação dos tipos de variáveis