

Análise lexical Valida a entrada, agrupa caracteres em tokens, especificada por ER, m^* (ou mais vezes), m^+ (1 ou mais vezes), $m?$ (0 ou 1 vez), $a.b = ab$, +

Análise sintática Valida a estrutura, organiza token em árvore, especificada por BNF, $P::(\text{inicial} | NT \rightarrow NT \text{ e/ou terminal})$

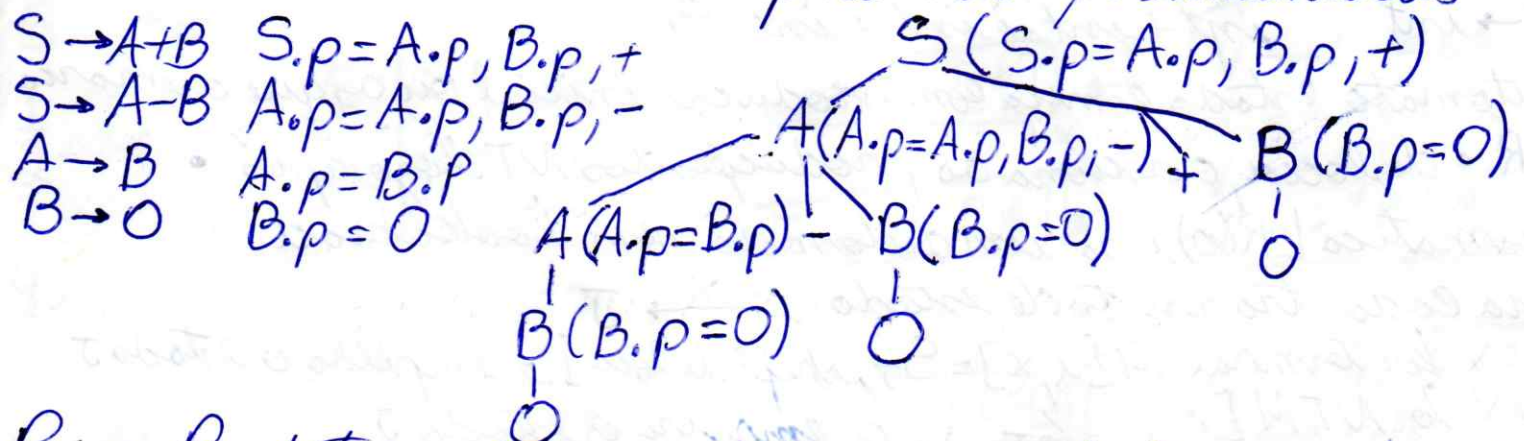
Análise semântica Valida o sentido com a definição da linguagem especificada pela tabela de símbolos, Confere declaração, herança e parâmetros

Gramática ambígua tem mais de uma derivação/redução para mesma palavra

Identificador Variável não inicializada em tempo de compilação

R.: usando a tabela de símbolos fazer uma travessia na AST marcando variáveis utilizadas, fazer outro marcando o uso

Travessia árvore atributos aplicando ações semânticas



Parser Predictivo é um parser recursivo topDown, sem backtrack a gramática deve ser fatorada removendo recursão a esquerda e prefixos comuns em produções de mesmo NT

$\text{First}(\alpha)$ terminais que podem iniciar derivação de α

$\text{Follow}(NT)$ terminais que podem vir logo após NT , $\epsilon \notin \text{Follow}$, $\$ \in \text{Follow}$

1º faz o First de produções iniciados em terminal

No follow substitua o NT logo após por cada terminal do First dele, $\text{Follow}(NT)$ conterá o follow da produção, se NT estiver no final ou antes de outro NT que pode ser ϵ .

Gramática LL(1) produz um Parser Predictivo, $A \rightarrow \alpha | B$

$\text{First}(\alpha)$ disjunto $\text{First}(B)$

se x deriva ϵ , $\text{First}(y)$ disjunto $\text{Follow}(x)$