**Teoria de Compilador – Tradutor**

(Fonte => Objeto)

- Processador de Linguagens: **Computador**

**Entrada**: Programa Fonte

**Saída**: Linguagem Máquina

Código Fonte

Erros de sintaxe; processador de linguagens; mapeamento de entrada/saída mais rápido; diagnóstico de erros é mais dificultoso.

Máquina

Entrada

Saída

Ações

Programa Fonte

Compilador

- Processador de Linguagens: **Interpretador**

Saída

Entrada

Máquina

Ações

Interpretador

Código Fonte

- Processador de Linguagens: **Abordagem Híbrida (Java)**

Saída

JIT (Just in Time) Compilador (Objeto)

Interpretador do Código JVM

Ações

Máquina

Entrada

Byte Code

Compilador

Código Fonte

Compilador (Linguagens – Forma de Comunicação - Formalismo) - Algoritmos

Aplicações: SQL; HTML; LaTeX

Algoritmo => O que o computador consegue fazer. (Alan Turing – Pai da Computação)

**Teoria do Formalismo** => Algoritmos são linguagens – forma de comunicação – Linguagem Formal)

Linguagem (Conjunto)

Cadeia (Entrada)

{W | W são os caracteres do alfabeto romana/latino}

abbc

Leitura

Fita; Fita Infinita; Máquina de Estado; Código Regulares

Máquina Abstrata

Não Pertence

Pertence

Tipo **Máquina Abstrata**: PDA

C -> As | SbA

A -> cbX (**formalismo**)

Cadeia (Entrada)

{Linguagem Livre de Contexto}

ações

Autômatos Finitos

Leitura

PAD (Pilha)

Não Pertence

Pertence

Linguagem – não livre de contexto;

Sintático – gramática (expressão; palavra reservado)

Em **Compiladores**

S -> Expr;

Expr -> Expr + Expr | Expr – Expr | ...

Cadeia (código fonte)

{Linguagem Livre de Contexto}

Int x = 13

Int y = 17

If ( x > y) {

System.Out.Prinbln

(“X > Y); }

Autômatos Finitos

Leitura

Máquina Abstrata

Compilador

Não Pertence (Rejeita)

Pertence (Aceita)

Mensagem de erro

Mov R1, #47F3

Mov R2, #BB5E

Sub R1, R2, R3

JN2 #47FE

Análise Sintática (forma) – sintaxe

Análise Semântica (significado)

int nome = “Hello World”;

Gramática (dentro – sintático), fora semântico.