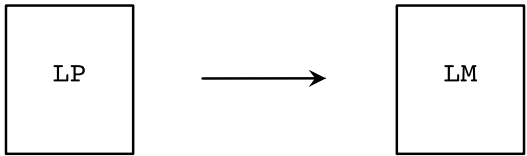


A linguagem LOOP

Por razões de eficiência e simplicidade, os computadores são construídos com a capacidade de reconhecer linguagens muito simples, as chamadas *linguagens de máquina*.

Mas, programar em linguagem de máquina é uma tarefa muito pouco conveniente para um ser humano.

A solução que foi encontrada para lidar com essa situação foi trabalhar com duas linguagens



- uma linguagem de máquina (LM) muito simples que o computador é capaz de entender
- uma linguagem de programação (LP) com uma gramática mais próxima da nossa linguagem natural

E o problema que aparece na prática, então, é que agora é preciso fazer a tradução da linguagem LP para a linguagem LM.

Mas, na medida em que a gramática da linguagem LP é uma gramática livre de contexto conhecida, é possível construir um autômato de pilha que *entende* o código escrito pelo programador e *explica* esse código para o computador.

Em outras palavras, nós podemos construir um autômato que traduz um programa escrito na linguagem LP para código escrito na linguagem LM.

Vejamos um exemplo concreto.

Lembre da linguagem LOOP que foi apresentada na aula 19, e que possui apenas dois tipos de instrução

```
X  <--  <expressão>

      LOOP  X
            <cmd>
            ...
            <cmd>
      ENDLOOP
```

A gramática da linguagem LOOP é a seguinte

```
P  →  C P | C
C  →  LOOP V P ENDLOOP
C  →  V := E
V  →  X | ... | Z
E  →  (...)
```

onde as regras da variável *E* são as regras de **EXPR** e foram omitidas.

E abaixo nós temos o autômato recursivo que reconhece programas escritos na linguagem LOOP