

# Documentação RA2

João Lucas M. N. C., João Luís C. C.

## Tokens e Estados

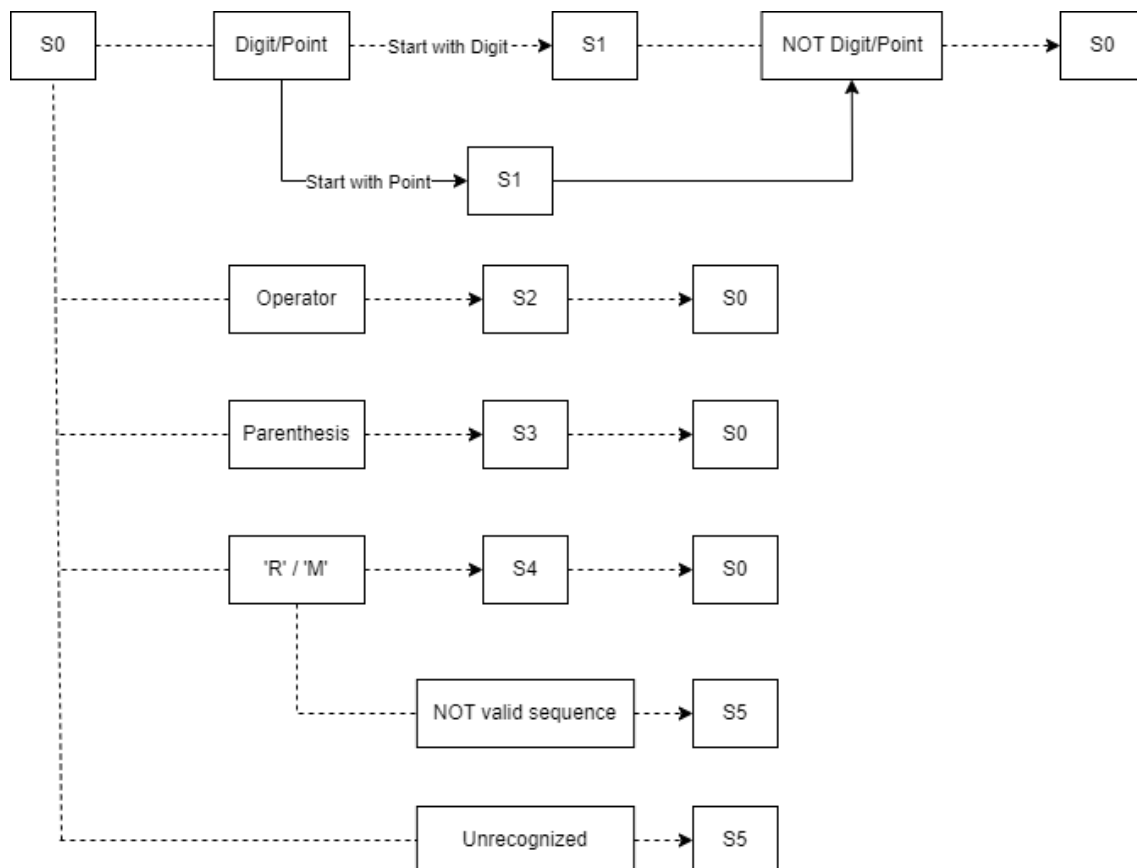
Os tokens incluem operadores matemáticos (+, -, \*, /, %, ^, |), parênteses ((, )), números, e os comandos especiais **RES** e **MEM**.

Estados Básicos:

1. **Estado Inicial (S0)**: O ponto de partida antes de qualquer leitura de caractere.
2. **Estado de Número (S1)**: Um estado que acumula dígitos e pontos decimais para formar números.
3. **Estado de Operador (S2)**: Captura os operadores matemáticos.
4. **Estado de Parêntese (S3)**: Identifica quando um parêntese é encontrado.
5. **Estado de Comando (S4)**: Identifica comandos especiais como **RES** e **MEM**.
6. **Estado de Erro (S5)**: Estado para qualquer entrada não reconhecida.

Transições:

- **De S0:**
  - Se o caractere for dígito ou ponto, transita para S1.
  - Se for um operador (+, -, \*, /, %, ^, |), transita para S2.
  - Se for um parêntese (( ou )), transita para S3.
  - Se começar com 'R' ou 'M' seguido de caracteres específicos, pode transitar para S4 ou S5 se a sequência não formar **RES** ou **MEM**.
  - Qualquer outro caractere leva ao S5.
- **De S1:**
  - Continua em S1 se mais dígitos ou um ponto forem lidos (controlando para não aceitar mais de um ponto).
  - Termina o token de número e retorna para S0 se outro tipo de caractere for encontrado.
- **De S2, S3, e S4:**
  - Após reconhecer um token válido, retorna para S0 para começar a análise do próximo token.



### Exemplos de Aceitação e Não-Aceitação

- Aceitação: 25 + 48, (5 \* 3), RES 1, 10 MEM
- Não-Aceitação: 25 ++ 48, (5 \*\* 3), RESS 1, 10 MEEM

### Regras de Produção

- $E \rightarrow E E \text{ Op} \mid \text{Número} \mid \text{'RES'} \mid \text{'MEM'}$
- $\text{Op} \rightarrow \text{'+'} \mid \text{'-'} \mid \text{'*'} \mid \text{'/'} \mid \text{'\%'} \mid \text{'^'} \mid \text{'\%'} \mid \text{'\^{'}}$
- $\text{Número} \rightarrow [0-9]^+ (\text{'.'} [0-9]^+)?$

**E** é um não-terminal que representa uma expressão, **Op** é um não-terminal para operadores, e **Número** é um não-terminal para números literais.

## Conjuntos FIRST e FOLLOW

- **FIRST(E)** = FIRST(Número)  $\cup$  {'RES', 'MEM'}
- **FIRST(Op)** = {'+', '-', '\*', '/', '%', '^', '|'}
- **FIRST(Número)** = {[0-9]}
- **FOLLOW(E)** = FOLLOW(E)  $\cup$  FIRST(Op)  $\cup$  {EOF}
- **FOLLOW(Op)** = FIRST(Número)  $\cup$  {'RES', 'MEM'}
- **FOLLOW(Número)** = FIRST(Op)  $\cup$  {EOF, 'RES', 'MEM'}

## Tabela de Derivação

Não-Terminal	Input	Produção Utilizada
Expr	(	Expr -> ( Expr Op Expr )
Expr	N	Expr -> Num
Expr	V	Expr -> ( V MEM )
Expr	MEM	Expr -> ( MEM )
Expr	EOF	-
Op	+	Op -> +
Op	-	Op -> -
Op	*	Op -> *
Op	/	Op -> /
Op	%	Op -> %
Op	^	Op -> ^
Op		`Op ->
Num	[0-9]	Num -> [0-9]+(\.[0-9]+)?