

A primeira manipulação é feita na regra **<atribuição>**.

<atribuição> ::= <variável> := <expressão>

Onde substitui-se o não terminal **<variável>**:

<atribuição> ::= <id> <seletor> := <expressão>

Todas as ocorrências da regra **<vazio>** serão substituídas pelo terminal ϵ .

E em seguida, é feita uma manipulação em **<seletor>**, possibilitando escrevê-lo da seguinte forma:

<atribuição> ::= <id> ("[" <expressão> "]")^{*} := <expressão>

Na regra **<comando>**, substitui-se o não terminal **<atribuição>**. Assim temos:

<comando> ::=
 ((<id> <seletor>) := <expressão>)
 | <condicional>
 | <iterativo>
 | <comando-composto>

Na regra **<corpo>**, alterou-se **<comando-composto>**, chegando na seguinte forma:

<corpo> ::= <declarações> begin <lista-de-comandos> end

As regras **<declaração>** e **<declaração-de-variáveis>**, podem ser condensadas na regra **<declarações>**, resultando em:

<declarações> ::= (var <lista-de-ids> : <tipo> ;)^{*}

Na regra **<expressão>** remove-se a recursão à esquerda, assim:

<expressão> ::= (<expressão-simples>) ((<op-rel> <expressão-simples>) | ϵ)

Em **<expressão-simples>** remove-se mais uma recursão à esquerda.

<expressão-simples> ::= <termo> (<op-ad> <termo>)^{*}

Faz-se a substituição dos não terminais **<seletor>** e **<variável>** na regra **<fator>**.

**<fator> ::= <id> ("[" <expressão> "]")^{*} | <literal> |
 "("<expressão>")"**

Para as regras **<lista-de-comandos>**, **<lista-de-ids>**, **<seletor>** e **<termo>** remove-se as recursividades à esquerda.

<lista-de-comandos> ::= (<comando> ;)^{*}

<lista-de-ids> ::= <id>(, <id>)^{*}

<seletor> ::= ("[" <expressão> "]")*

<termo> ::= <fator>(<op-mul> <fator>)*

Regra <comando>:

- ()
 $\text{first}_1(\langle \text{id} \rangle ("[" \langle \text{expressão} \rangle "]")^* := \langle \text{expressão} \rangle)) = \{ \langle \text{id} \rangle \}$
- <condicional>
 $\text{first}_1(\text{if } \langle \text{expressão} \rangle \text{ then } \langle \text{comando} \rangle (\text{ else } \langle \text{comando} \rangle \mid \varepsilon)) = \{ \text{if} \}$
- <iterativo>
 $\text{first}_1(\text{while } \langle \text{expressão} \rangle \text{ do } \langle \text{comando} \rangle) = \{ \text{while} \}$
- <comando-composto>
 $\text{first}_1(\text{begin } \langle \text{lista-de-comandos} \rangle \text{ end}) = \{ \text{begin} \}$

$\{ \langle \text{id} \rangle \} \cap \{ \text{if} \} \cap \{ \text{while} \} \cap \{ \text{begin} \} = \emptyset$

Regra <comando-composto>:

$\text{first}_1(\text{begin } \langle \text{lista-de-comandos} \rangle \text{ end}) = \{ \text{begin} \}$

Regra <corpo>

$\text{first}_1(\langle \text{declarações} \rangle) \cup \text{follow}_1(()) = \{ \text{var} \} \cap \{ \text{begin} \} = \emptyset$

Regra <declarações>:

$\text{first}_1(\langle \text{declarações} \rangle) = \{ \text{var} \}$

Regra <expressão>:

$\text{first}_1(\langle \text{expressão-simples} \rangle (\langle \text{op-rel} \rangle \langle \text{expressão-simples} \rangle \mid \varepsilon)) = \{ \langle \text{expressão-simples} \rangle \}$

Regra <expressão-simples>:

$\text{first}_1(\langle \text{termo} \rangle (\text{op-ad} \langle \text{termo} \rangle)^*) = \{ \langle \text{termo} \rangle \}$

Regra <fator>:

- ()
 $\text{first}_1(\langle \text{id} \rangle ("[" \langle \text{expressão} \rangle "]")^*) = \{ \langle \text{id} \rangle \}$
- <literal>
 $\text{first}_1(\langle \text{bool-lit} \rangle) \cup \text{first}_1(\langle \text{int-lit} \rangle) \cup \text{first}_1(\langle \text{float-lit} \rangle) = \{ \langle \text{bool-lit} \rangle \} \cap \{ \langle \text{int-lit} \rangle \} \cap \{ \langle \text{float-lit} \rangle \} = \emptyset$
 $\text{first}_1(\langle \text{"expressão"} \rangle) = \{ \langle \text{""} \rangle \}$

$$\{\text{<id>}\} \cap \{\text{<bool-lit>}\} \cap \{\text{<int-lit>}\} \cap \{\text{<float-lit>}\} \cap \{\text{"("}\} = \emptyset$$

Regra <iterativo>:

$$\text{first}_1(\text{while } \text{<expressão> do } \text{<comando>}) = \{\text{while}\}$$