

# Descrição Formal da Linguagem *LM*

Leandro Motta Barros

Dezembro de 02001

## 1 Descrição da Linguagem *LM*

*LM* é uma linguagem de programação funcional. *LM* não é ML. *LM* significa “Leandro Motta”, em homenagem a seu criador. Ou “Língua Morta”, já que todas as palavras reservadas são derivadas do Latim [1] [2].

A seguir é apresentada a gramática de *LM*, juntamente com informações sobre a semântica da linguagem.

### 1.1 Elementos Básicos

A definição de dígito não apresenta nenhuma surpresa. Na definição de letra, apenas convém notar que o símbolo *underscore* (`_`) é considerado uma letra.

$\langle \textit{Dígito} \rangle \rightarrow \mathbf{0}$

$\vdots$

$\langle \textit{Dígito} \rangle \rightarrow \mathbf{9}$

$\langle \textit{Letra} \rangle \rightarrow \mathbf{a}$

$\vdots$

$\langle \textit{Letra} \rangle \rightarrow \mathbf{z}$

$\langle \textit{Letra} \rangle \rightarrow \mathbf{A}$

$\vdots$

$\langle \textit{Letra} \rangle \rightarrow \mathbf{Z}$

$\langle \textit{Letra} \rangle \rightarrow \mathbf{-}$

A partir destas definições é possível especificar um inteiro literal e um identificador. Um inteiro literal é simplesmente uma seqüência de um ou mais dígitos:

$\langle \text{Inteiro Literal} \rangle \rightarrow \langle \text{Dígito} \rangle$   
 $\langle \text{Inteiro Literal} \rangle \rightarrow \langle \text{Dígito} \rangle \langle \text{Inteiro Literal} \rangle$

Um identificador é uma seqüência de uma ou mais letras e dígitos que deve começar por uma letra.

$\langle \text{Identificador} \rangle \rightarrow \langle \text{Letra} \rangle$   
 $\langle \text{Identificador} \rangle \rightarrow \langle \text{Identificador} \rangle \langle \text{Letra} \rangle$   
 $\langle \text{Identificador} \rangle \rightarrow \langle \text{Identificador} \rangle \langle \text{Numero} \rangle$

A linguagem *LM* considera que espaços (representados por “ ”), tabulações (**tabs**) e quebras de linha (**eols**) são equivalentes.

$\langle \text{Espaço} \rangle \rightarrow \text{ }$   
 $\langle \text{Espaço} \rangle \rightarrow \text{eol}$   
 $\langle \text{Espaço} \rangle \rightarrow \text{tab}$   
 $\langle \text{Espaços} \rangle \rightarrow \langle \text{Espaço} \rangle \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Espaços Opc.} \rangle \rightarrow \varepsilon$   
 $\langle \text{Espaços Opc.} \rangle \rightarrow \langle \text{Espaço} \rangle \langle \text{Espaços Opc.} \rangle$

Expressões inteiras são expressões *LM* que, quando avaliadas, resultam em um número inteiro. Uma expressão inteira pode ser um inteiro literal, um identificador, ou uma chamada de função.

$\langle \text{Expressão Inteira} \rangle \rightarrow \langle \text{Inteiro Literal} \rangle$   
 $\langle \text{Expressão Inteira} \rangle \rightarrow \langle \text{Identificador} \rangle$   
 $\langle \text{Expressão Inteira} \rangle \rightarrow \langle \text{Chamada de Função} \rangle$

## 1.2 Funções

A seguir são apresentadas regras referentes a funções. Funções em *LM* podem ou não receber parâmetros.

### 1.2.1 Definição de funções

Uma função *LM* é definida através da palavra reservada **definire**. O corpo de uma função é composto de exatamente uma chamada de função. Só seria útil permitir mais de uma chamada se elas pudessem causar efeitos colaterais (como atribuições). Mas este não é o caso, pois *LM* é uma linguagem funcional.

$\langle \text{Definição de Função} \rangle \rightarrow \mathbf{define} \langle \text{Espaços} \rangle \langle \text{Identificador} \rangle \langle \text{Espaços Opc.} \rangle$   
 $( \langle \text{Espaços Opc.} \rangle \langle \text{Parâmetros Formais} \rangle$   
 $\langle \text{Espaços Opc.} \rangle ) \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Chamada de Função} \rangle$

$\langle \text{Parâmetros Formais} \rangle \rightarrow \varepsilon$   
 $\langle \text{Parâmetros Formais} \rangle \rightarrow \langle \text{Parâmetros Formais}' \rangle$

$\langle \text{Parâmetros Formais}' \rangle \rightarrow \langle \text{Identificador} \rangle$   
 $\langle \text{Parâmetros Formais}' \rangle \rightarrow \langle \text{Identificador} \rangle \langle \text{Espaços Opc.} \rangle , \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Parâmetros Formais}' \rangle$

### 1.2.2 Chamadas de função

Chamadas de funções definidas pelo usuário são feitas conforme as regras mostradas a seguir. Regras para chamadas de funções pré-definidas são discutidas na seção 1.2.3.

$\langle \text{Chamada de Função} \rangle \rightarrow \langle \text{Identificador} \rangle \langle \text{Espaços Opc.} \rangle ( \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Parâmetros Reais} \rangle \langle \text{Espaços Opc.} \rangle )$

$\langle \text{Parâmetros Reais} \rangle \rightarrow \varepsilon$   
 $\langle \text{Parâmetros Reais} \rangle \rightarrow \langle \text{Parâmetros Reais}' \rangle$

$\langle \text{Parâmetros Reais}' \rangle \rightarrow \langle \text{Expressão Inteira} \rangle$   
 $\langle \text{Parâmetros Reais}' \rangle \rightarrow \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle , \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Parâmetros Reais}' \rangle$

É interessante notar que as funções definidas pelo usuário podem receber outras funções definidas pelo usuário como parâmetro, pois  $\langle \text{Expressão Inteira} \rangle$  pode ser transformada em um identificador. Isto é sintaticamente correto, apesar de que uma função não é realmente uma expressão inteira.

### 1.2.3 Funções pré-definidas

LM possui um conjunto de funções pré-definidas (FPDs) que são discutidas nas próximas seções. Chamadas a funções pré-definidas podem ser divididas em grupos de acordo com o número de parâmetros que recebem.

$\langle \text{Chamada FPD 1} \rangle \rightarrow \langle \text{Nome FPD 1} \rangle \langle \text{Espaços Opc.} \rangle ( \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle )$

$$\langle \text{Chamada FPD 2} \rangle \rightarrow \langle \text{Nome FPD 2} \rangle \langle \text{Espaços Opc.} \rangle ( \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle , \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle )$$

$$\langle \text{Chamada FPD 2+} \rangle \rightarrow \langle \text{Nome FPD 2+} \rangle \langle \text{Espaços Opc.} \rangle ( \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle , \\ \langle \text{Espaços Opc.} \rangle \langle \text{Parâmetros Reais} \rangle \\ \langle \text{Espaços Opc.} \rangle )$$

$$\langle \text{Chamada FPD 3} \rangle \rightarrow \langle \text{Nome FPD 3} \rangle \langle \text{Espaços Opc.} \rangle ( \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle , \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle , \langle \text{Espaços Opc.} \rangle \\ \langle \text{Expressão Inteira} \rangle \langle \text{Espaços Opc.} \rangle )$$

As chamadas de funções pré-definidas *não* são ambíguas: a regra a ser aplicada pode ser facilmente deduzida a partir do nome da função, que é o primeiro símbolo que aparece no lado direito de todas as regras deste grupo.

Evidentemente, chamadas de funções pré-definidas também são chamadas de função:

$$\langle \text{Chamada de Função} \rangle \rightarrow \langle \text{Chamada FPD 1} \rangle \\ \langle \text{Chamada de Função} \rangle \rightarrow \langle \text{Chamada FPD 2} \rangle \\ \langle \text{Chamada de Função} \rangle \rightarrow \langle \text{Chamada FPD 2+} \rangle \\ \langle \text{Chamada de Função} \rangle \rightarrow \langle \text{Chamada FPD 3} \rangle$$

### 1.3 Aritmética de Inteiros

*LM* possui cinco funções pré-definidas que possibilitam a execução das operações aritméticas básicas com números inteiros. Os parâmetros destas funções devem ser expressões inteiras.

As funções **adicere** e **multiplicare** recebem como parâmetros dois ou mais valores e retornam, respectivamente, a soma e o produto dos parâmetros.

$$\langle \text{Nome FPD 2+} \rangle \rightarrow \mathbf{adicere} \\ \langle \text{Nome FPD 2+} \rangle \rightarrow \mathbf{multiplicare}$$

As outras funções para aritmética de inteiros admitem apenas dois parâmetros. **deducere** realiza a operação de subtração; **didere** efetua a divisão inteira; e **relicus** é usada para obter o resto da divisão inteira.

$$\langle \text{Nome FPD 2} \rangle \rightarrow \mathbf{deducere} \\ \langle \text{Nome FPD 2} \rangle \rightarrow \mathbf{didere}$$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{relicus}$

## 1.4 Comparações

Várias funções de *LM* permitem realizar comparações: **superus** (maior que), **intra** (menor que), **aequalis** (igual), **alius** (diferente), **superusaequalis** (igual ou maior que) e **intraaequalis** (igual ou menor que)<sup>1</sup>.

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{superus}$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{intra}$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{aequalis}$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{alius}$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{superusaequalis}$

$\langle \text{Nome FPD } 2 \rangle \rightarrow \text{intraaequalis}$

Todas estas funções admitem dois parâmetros, que devem ser expressões inteiras. A chamada de função “**superus** (*a*, *b*)” retorna 1 se  $a > b$  e 0 caso contrário. As outras funções seguem o mesmo padrão.

## 1.5 Operações Lógicas

Em *LM* o número zero é a única expressão considerada falsa. Qualquer outra expressão é considerada verdadeira.

Três operações lógicas são suportadas por *LM*. A operação lógica E é realizada pela função **et**. A operação OU é feita pela função **vel**. E a operação NÃO é efetuada pela função **nec**.

**et** e **vel** admitem dois ou mais parâmetros e sua semântica é a esperada: **et** retorna 1 caso todos os seus parâmetros sejam verdadeiros (diferentes de zero) e 0 caso contrário; **vel** retorna 0 se todos os seus parâmetros forem falsos (iguais a zero) e 1 caso contrário. A avaliação dos parâmetros é “curto-circuitada”, ou seja, eles são avaliados apenas até o ponto em que seja possível obter o valor de retorno da função.

$\langle \text{Nome FPD } 2+ \rangle \rightarrow \text{et}$

$\langle \text{Nome FPD } 2+ \rangle \rightarrow \text{vel}$

Apenas um parâmetro pode (e deve) ser passado para a função **nec**. Se o

---

<sup>1</sup>Segundo [1], em Latim existe uma palavra, *dumtaxat*, que pode ter dois significados totalmente contrários: “não maior que” ou “não menor que”. Não imagino como alguém possa formar uma frase com um termo desses...

parâmetro for verdadeiro, **nec** retorna 0; se for falso, retorna 1.

$\langle \text{Nome FPD } 1 \rangle \rightarrow \mathbf{nec}$

## 1.6 Testes Condicionais

Testes condicionais podem ser feitos através da função pré-definida **si**, que deve receber três expressões inteiras como parâmetros.

$\langle \text{Nome FPD } 3 \rangle \rightarrow \mathbf{si}$

A semântica da função **si** é a seguinte: se a avaliação do primeiro parâmetro resultar em um valor diferente de zero (verdadeiro), então o segundo parâmetro é avaliado e seu valor retornado. Senão, o terceiro parâmetro é avaliado e seu valor retornado.

## 1.7 Programa

Finalmente, um programa *LM* é uma seqüência de uma ou mais definições de funções possivelmente separadas por espaços. A execução de um programa consiste na avaliação da última função definida.

$\langle \text{Programa} \rangle \rightarrow \langle \text{Espaços Opc.} \rangle \langle \text{Definição de Função} \rangle \langle \text{Espaços Opc.} \rangle$   
 $\langle \text{Programa} \rangle \rightarrow \langle \text{Espaços Opc.} \rangle \langle \text{Definição de Função} \rangle \langle \text{Programa} \rangle$

## 2 Alterações na Gramática

Esta seção apresenta alterações em algumas regras da gramática, que buscam torná-la adequada para a transformação em um *parser*. As regras que não apresentam problemas não são mencionadas.

### 2.1 Inteiros Literais

As regras que definem inteiros literais podem ser fatoradas:

$\langle \text{Inteiro Literal} \rangle \rightarrow \langle \text{Dígito} \rangle \langle \text{Inteiro Literal} \rangle$

$\langle \text{Inteiro Literal}' \rangle \rightarrow \varepsilon$

$\langle \text{Inteiro Literal}' \rangle \rightarrow \langle \text{Inteiro Literal} \rangle$

## 2.2 Identificadores

Originalmente as regras que definiam identificadores possuíam recursão à esquerda. Ela foi eliminada, dando origem às seguintes regras:

$$\langle \textit{Identificador} \rangle \rightarrow \langle \textit{Letra} \rangle \langle \textit{Identificador}' \rangle$$

$$\langle \textit{Identificador}' \rangle \rightarrow \varepsilon$$

$$\langle \textit{Identificador}' \rangle \rightarrow \langle \textit{Letra} \rangle \langle \textit{Identificador}' \rangle$$

$$\langle \textit{Identificador}' \rangle \rightarrow \langle \textit{Numero} \rangle \langle \textit{Identificador}' \rangle$$

## 2.3 Parâmetros

As regras que definem listas de parâmetros (formais e reais) podem ser fatoradas. Para parâmetros formais temos:

$$\langle \textit{Parâmetros Formais} \rangle \rightarrow \varepsilon$$

$$\langle \textit{Parâmetros Formais} \rangle \rightarrow \langle \textit{Parâmetros Formais}' \rangle$$

$$\langle \textit{Parâmetros Formais}' \rangle \rightarrow \langle \textit{Identificador} \rangle \langle \textit{Parâmetros Formais}'' \rangle$$

$$\langle \textit{Parâmetros Formais}'' \rangle \rightarrow \varepsilon$$

$$\langle \textit{Parâmetros Formais}'' \rangle \rightarrow \langle \textit{Espaços Opc.} \rangle, \langle \textit{Espaços Opc.} \rangle \\ \langle \textit{Parâmetros Formais}' \rangle$$

E, similarmente, para parâmetros reais:

$$\langle \textit{Parâmetros Reais} \rangle \rightarrow \varepsilon$$

$$\langle \textit{Parâmetros Reais} \rangle \rightarrow \langle \textit{Parâmetros Reais}' \rangle$$

$$\langle \textit{Parâmetros Reais}' \rangle \rightarrow \langle \textit{Expressão Inteira} \rangle \langle \textit{Parâmetros Reais}'' \rangle$$

$$\langle \textit{Parâmetros Reais}'' \rangle \rightarrow \varepsilon$$

$$\langle \textit{Parâmetros Reais}'' \rangle \rightarrow \langle \textit{Espaços Opc.} \rangle, \langle \textit{Espaços Opc.} \rangle \\ \langle \textit{Parâmetros Reais}' \rangle$$

## 2.4 Expressões Inteiras

Há um problema com as regras apresentadas para expressões inteiras: se o primeiro símbolo da expressão inteira for uma letra, será impossível descobrir se o que está sendo lido é um identificador ou uma chamada de função definida pelo usuário. Para corrigir este problema as regras que definem

uma expressão inteira foram substituídas.

$$\begin{aligned}\langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Inteiro Literal} \rangle \\ \langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Chamada FPD 1} \rangle \\ \langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Chamada FPD 2} \rangle \\ \langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Chamada FPD 2+} \rangle \\ \langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Chamada FPD 3} \rangle\end{aligned}$$

Estas regras são suficientes para o caso de o primeiro símbolo lido ser um dígito ou o nome de uma função pré-definida. Mas, como o primeiro símbolo lido pode ser uma letra (ou seja, o início de um identificador), novas regras são necessárias:

$$\begin{aligned}\langle \textit{Expressão Inteira} \rangle &\rightarrow \langle \textit{Identificador} \rangle \langle \textit{Expressão Inteira}' \rangle \\ \langle \textit{Expressão Inteira}' \rangle &\rightarrow \varepsilon \\ \langle \textit{Expressão Inteira}' \rangle &\rightarrow \langle \textit{Espaços Opc.} \rangle ( \langle \textit{Espaços Opc.} \rangle \\ &\quad \langle \textit{Parâmetros Reais} \rangle \langle \textit{Espaços Opc.} \rangle )\end{aligned}$$

## 2.5 Programa

É possível fatorar as regras que definem um programa:

$$\begin{aligned}\langle \textit{Programa} \rangle &\rightarrow \langle \textit{Espaços Opc.} \rangle \langle \textit{Definição de Função} \rangle \langle \textit{Programa}' \rangle \\ \langle \textit{Programa}' \rangle &\rightarrow \langle \textit{Espaços Opc.} \rangle \\ \langle \textit{Programa}' \rangle &\rightarrow \langle \textit{Programa} \rangle\end{aligned}$$

## Referências

- [1] English to Latin. <http://cawley.archives.nd.edu/cgi-bin/lookdown.pl>.
- [2] Latin Word List. <http://kufacts.cc.ukans.edu/ftp/pub/history/Europe/Medieval/aids/latwords.html>.