Sintaxe de L1

|  |  |
| --- | --- |
| Termo | Sintaxe |
|  |  |
|  |  |
| op | op |
| if then else | if then else |
|  |  |
|  |  |
| fn | fn |
| fn | fn |
| let in | let |
| let in | let |
| let rec fn in | let rec |
| let rec fn in | let rec |
|  |  |
|  |  |
| isempty | empty? |
| hd | head |
| tl | tail |
| raise | raise |
| try with | try except |

|  |  |
| --- | --- |
| Tipo | Sintaxe |
| int | Int |
| bool | bool |
|  | -> |
| list |  |

Na especificação de funções, o tipo é associativo à direita. Dessa forma, é o mesmo que . É possível utilizar parênteses para obter a associatividade desejada.

Identificadores

Identificadores são strings de tamanho arbitrário, sem espaços em brancos, compostos de quaisquer caracteres com exceção dos seguintes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| . | , | ; | : | \ | + |
| - | / | \* | % | > | < |
| = | ! | ( | ) | { | } |
| [ | ] | @ |  |  |  |

* Numerais não são permitidos no começo de um identificador, porém podem ser utilizados depois.
* A interrogação é permitida em identificadores, e o uso mais indicado para a mesma é no final de funções que retornem um booleano.

Existem termos reservados na linguagem L1, e esses termos não podem ser utilizados como identificadores:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| fn | let | rec | for | in | head |
| tail | empty? | nil | try | except | raise |
| true | false | if | then | else |  |

Operadores

|  |  |
| --- | --- |
| Operador | Sintaxe |
|  |  |
|  |  |
|  |  |
| div |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Além dos operadores definidos na gramática de L1, essa implementação possui os seguintes operadores:

|  |  |  |
| --- | --- | --- |
| Operador | Uso | Significado |
| (prefixo) |  | Oposto numérico de \*\* |
|  |  | Resto da divisão inteira de por \*\* |
|  |  | Concatena a lista com a lista \*\* |
|  |  | Aplica em |
|  |  | Aplica em |
|  |  | Operação logica “e” em e \*\* |
|  |  | Operação logica “ou” em e \*\* |

\*\* Requer a *stdlib* para funcionar

Abaixo está a prioridade de cada operador, além de sua associatividade. É importante ressaltar que é possível mudar tanto a prioridade quando a associatividade com o uso de parênteses.

|  |  |  |
| --- | --- | --- |
| Prioridade | Operador | Associatividade |
| 1 | * (prefixo) | Nenhuma |
| 2 | (aplicação de função) | Esquerda |
|  | **head** | Esquerda |
|  | **tail** | Esquerda |
|  | **empty?** | Esquerda |
| 3 |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
| 4 |  | Esquerda |
|  | (infixo) | Esquerda |
| 5 |  | Direita |
| 6 |  | Esquerda |
| 7 |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
|  |  | Esquerda |
| 8 |  | Esquerda |
| 9 |  | Esquerda |

Açúcar Sintático

|  |  |
| --- | --- |
| Construção | Equivalência |
|  |  |
| let | let |
| let | let |
|  | **\*** |
|  | **\*** |
|  | **\*** |
| fn | fn |
| fn | fn |
|  |  |
|  |  |
| let rec | let rec **\*\*** |
| let rec | let rec **\*\*** |
| let | let |
| let | let |

\* Requerem a *stdlib*

\*\* Apenas o nível mais externo é definido como recursivo.

Bibliotecas