

Documentação Sintática da Linguagem Ruby

1. Elementos Sintáticos.

Um programa em Ruby é composto por uma ou mais funções. Uma função Ruby apresenta sintaxe similar a python, conforme apresentado na seguinte regra:

```
program -> ID | assign | funcao |  
          funcao program
```

```
funcao -> "def" ID "(" "(" sigParams ")" "(" stm "end"
```

Onde os dois primeiros ID se referem, respectivamente, ao token def, utilizado para a definição de uma função e o nome da função. sigParams representa os argumentos da função. Por último temos stms que representam um ou mais comandos finalizados por um end. Na próxima seção será apresentado os comandos da linguagem Ruby.

1.1 Comandos da Linguagem Ruby

Com relação aos comandos aceitos, Ruby lida apenas com comandos de expressões, o comando de repetição while, for e o comando return, conforme apresentado nas seguintes regras:

```
stms → stm | stm stms
```

```
opcional → ELSIF exp "then" stms "end" | ELSE stms "end" | λ
```

```
stm → exp
```

```
| WHILE "(" exp ")" stms "end"  
| WHILE exp stms "end"  
| FOR ID "in" exp "do" stms "end"  
| FOR ID "in" exp stms "end"  
| IF exp "then" stms "end" opcional  
| IF exp stms "end" opcional  
| IF "(" exp ")" "then" stms "end" opcional  
| IF "(" exp ")" stms "end" opcional  
| return exp
```

O comando while inicia com a palavra reservada while, seguido por uma expressão envolvida entre parênteses. Adicionalmente, ele apresenta obrigatoriamente uma sequência de comandos, que tem como delimitador o “end”.

1.2 Expressões em Ruby

Ruby dá suporte a expressões aritméticas de soma, multiplicação, exponenciação, subtração, divisão e resto. Adicionalmente, também dá suporte a chamadas de função (call), atribuição de valores a variáveis (assign). Por fim, expressões também podem ser números (NUM) e variáveis (ID). A sintaxe das expressões Ruby é apresentada pela seguinte regra:

```
exp → exp "+" exp |  
      exp "*" exp |  
      exp "-" exp |  
      exp "**" exp |  
      exp "/" exp |  
      exp "%" exp |  
      exp "==" exp |  
      exp "===" exp |  
      exp "!=" exp |  
      exp ">" exp |  
      exp "<" exp |  
      exp "<=" exp |  
      exp ">=" exp |  
      exp "||" exp |  
      exp "&&" exp |  
      exp "!" exp |  
      call |  
      assign |  
      NUM |  
      ID
```

1.2.1 Chamadas de Função e Atribuição

Ruby dá suporte a chamadas de função com e sem parâmetros. Um parâmetro de função pode ser qualquer expressão Ruby. Adicionalmente, Ruby permite atribuir valores a variáveis, conforme regras apresentadas a seguir:

```
call → ID "(" sigParams ")" |  
      ID "(" " " ")" |  
      ID |  
      ID sigParams
```

```
sigParams → exp "," sigParams |  
            exp
```

assign \rightarrow ID "=" exp

2. Exemplos de Código.

A seguir, alguns exemplos de código na Linguagem Ruby:

Exemplo 1:

```
def sumparabola(a, b, c)
  return a + b + c
end

def some (a, b)
  a = 88 + 44
  b = 70
  sumparabola(1, 2, 3)
  while (true)
    c = 38
    sumparabola(5, 5, 5)
    while (c)
      sumparabola(6, 6, 6)
    end
  end
  soma()
  sumparabolac(2)
  return true
end
```

Exemplo 2 :

```
def dividir (a,b)
  return a/b
end

resultado = dividir (15,29)
while(resultado>1)
  puts resultado
end
```