

Um procedimento de cada vez...

REPEAT

Um procedimento popular

O procedimento primitivo REPEAT existe em todos os LOGO's e obedece sempre às mesmas regras de utilização (até ver, está claro). É extremamente útil em muitas situações e de fácil uso, daí ser tão popular... Mas outra razão que o torna popular é o facto de ser empregue para desenhar um quadrado no ecrã e os cursos de iniciação ao LOGO não viverem sem quadrados...

REPEAT é um comando com dois *inputs*, sendo o primeiro um número natural e o segundo uma lista. Dissemos que é um comando, e isso significa que REPEAT produz um efeito, a saber: executa repetidamente — tantas vezes quanto o valor do primeiro *input* — as instruções contidas na lista. Assim, se teclarmos REPEAT 4 [FD 40 RT 90] e depois ENTER ou RETURN, a instrução FD 40 RT 90 é executada quatro vezes, resultando no traçado de um quadrado no ecrã.

O primeiro *input* é um número natural, mas como é habitual no LOGO pode ser obtido como *output* de uma operação. Assim, se criarmos a variável "n e lhe atribuirmos o valor 3 com a instrução make "n 3 e se depois teclarmos REPEAT : n + 1 [FD 40 RT 90], o efeito será idêntico ao anterior. Deve no entanto notar-se que, se o valor do primeiro *input* não é um natural mas um número racional positivo qualquer, o LOGO toma como *input* para o comando REPEAT a sua parte inteira. No caso de um *input* nulo ou negativo, obtemos uma mensagem de desagrado por parte do LOGO.

Quanto ao segundo *input*, ele deverá ser naturalmente uma lista com uma sequência de procedimentos LOGO com os respectivos *inputs*, no caso de existirem. E se assim não for, se introduzirmos na lista uma palavra que não seja um procedimento conhecido do LOGO? Este, muito justamente, responde-nos com uma mensagem dizendo que não conhece tal palavra (o texto exacto da mensagem varia de versão para versão). É o que acontece se tentarmos executar a instrução REPEAT 3 [MARIA] (experimente!; e depois experimente a instrução REPEAT 3 [PR "MARIA]. Se a lista for vazia, como por exemplo em REPEAT 10 [], LOGO repete 10 vezes o procedimento que consiste em não fazer nada, mas nos computadores mesmo não fazer nada leva um certo tempo... e portanto isto é uma maneira de fazer uma pausa, tanto maior quanto o valor do primeiro *input*. Pode fazer umas experiências e servir-se deste processo para fazer pausas com as durações que desejar.

Se o primeiro *input* for 1, então o comando REPEAT torna-se equivalente ao comando RUN (consulte o manual da sua versão de LOGO).

REPEAT é um comando muito útil, mas há tarefas de que não é capaz, está claro. Por exemplo, poderíamos ser tentados, para obter as primeiras duas letras de uma palavra, a usar o seguinte procedimento:

```
to prim.duas.letras :palavra
repeat 2 [pr first :palavra]
end
```

Mas o resultado será diferente daquele que talvez esperemos (experimente!). Se queremos efectuar repetições em que, de cada vez, o *input* sobre que actuamos seja diferente, teremos que usar outro tipo de procedimentos: os procedimentos recursivos. Mas isto fica para outra vez...

Eduardo Vêloso

Desafio aos logoistas!

Uma imagem de uma criança lendo em livro cuja capa é, por sua vez, a imagem da mesma criança lendo o mesmo livro cuja capa é, por sua vez..., é um exemplo de recursividade em expressão gráfica.

O jogo de bonecas russas que se encaixam umas nas outras pode encarar-se como uma concretização artística de recursividade.

Definir avô paterno como «pai do pai» é um exemplo de definição recursiva, aliás muito utilizada em Prolog.

No campo da Matemática as definições recursivas são frequentes. Por exemplo, podemos definir o factorial de um número natural, recursivamente, da seguinte forma:

$$\begin{cases} 1! = 1 \\ n! = n \times (n-1)! \end{cases}$$

A recursividade é, também, uma das características da linguagem LOGO, na medida em que se podem construir procedimentos que se «usam» a eles próprios. Por exemplo, para calcular a potência de expoente inteiro de um número real, podemos construir o seguinte procedimento

```
to potência :b :e
ifelse :e = 0 [op 1] [op :b * potência :b :e-1]
end
```

Vimos, no artigo «Fractais na Escola Secundária» como as curvas fractais são estruturas invariantes por dilatação de escala. O processo geométrico de as obter consiste na repetição sucessiva dos mesmos passos, em diferentes escalas. Parece, assim, ser possível construir, em LOGO, um procedimento recursivo que permita desenhar, por exemplo, a curva do floco de neve e a curva de Peano (ou outras!).

Vamos a isso?

Leonor Moreira