

Um procedimento de cada vez...

OUTPUT

Suponhamos que queríamos criar um procedimento com um nome português que fizesse exactamente o mesmo que o procedimento **FIRST**. Como se sabe, **FIRST** aplicado a uma lista dá o seu primeiro elemento. Assim, se teclarmos por exemplo

```
PR FIRST [[a < b] [b < c] [a < c]]
```

o computador responde-nos **a < b**, pois o primeiro elemento da lista a que estamos a aplicar **FIRST** é precisamente a lista **a < b**. O que queremos construir é um procedimento, a que chamaremos naturalmente **PRIMEIRO** e que funcione como **FIRST**, ou seja, tal que, se teclarmos

```
PR PRIMEIRO [[a < b] [b < c] [a < c]]
```

também nos responda **a < b**. Uma primeira ideia pode ser a seguinte (como não sabemos se vai dar certo, vamos chamar-lhe por enquanto **PRIMEIRO.TESTE** em vez de **PRIMEIRO**):

```
TO PRIMEIRO.TESTE :LISTA
FIRST :LISTA
END
```

Trata-se de uma tradução «literal» do procedimento **FIRST**.

Se agora teclarmos **PRIMEIRO.TESTE [[a < b] [b < c] [a < c]]** o computador responde-nos com a mensagem:

```
you dont say what to do with [a < b] in PRIMEIRO.TESTE
```

(você não diz o que fazer com **[a < b]** em **PRIMEIRO.TESTE**).

Que quer isto dizer? Em primeiro lugar, que há um problema qualquer, pois recebemos uma mensagem em vez da resposta que esperávamos. Em segundo lugar, que esse problema é *dentro (in)* do procedimento **PRIMEIRO.TESTE**. Em terceiro lugar, que o procedimento calculou bem o primeiro elemento da lista, pois refere-se a **a < b** (isto é natural, pois nós usámos **FIRST** e este funciona bem). Finalmente, que o problema consiste em **nós não termos dito o que fazer com [a < b]** (oh maravilhosas mensagens do LOGO!!). Em algumas versões do LOGO (no IBM LOGO, por exemplo), em vez de «in **PRIMEIRO.TESTE**» estaria escrito «just before leaving **PRIMEIRO.TESTE**» (exactamente antes de terminar **PRIMEIRO.TESTE**), o que queria dizer exactamente o mesmo.

Portanto, o que temos a fazer é alterar o procedimento de maneira a dizer ao computador o que deve fazer com o resultado de **FIRST :LISTA**. Ora nós o que queremos é, **apenas**, que o resultado («output» em inglês) de **FIRST :LISTA** passe a ser o resultado (output) de **PRIMEIRO.TESTE :LISTA**. Há um procedimento em todas as versões de LOGO que tem exactamente esta função, e chama-se — como era de esperar... — **OUTPUT**. Devemos portanto escrever, **OUTPUT FIRST :LISTA** em vez de simplesmente **FIRST :LISTA**. Portanto o novo procedimento, a que chamamos **PRIMEIRO** na esperança de que esteja já correcto, deve ser assim definido

```
TO PRIMEIRO :LISTA
OUTPUT FIRST :LISTA
END
```

Se agora repetirmos a experiência anterior, veremos que **PRIMEIRO** se comporta exactamente do mesmo modo que **FIRST**, como pretendíamos.

Uma tentação habitual quando começamos a programar em LOGO é usar **PRINT** em vez de **OUTPUT**. Dessa forma estaríamos a definir realmente um procedimento válido, mas não seria uma verdadeira tradução de **FIRST**, não se comportaria da mesma forma. Teríamos definido um **comando**, ou seja um procedimento com um **efeito** (que neste caso seria o de escrever no ecrã o primeiro elemento da lista) e não uma **operação com um resultado** utilizável por outros procedimentos. Para verificar isso, faça o seguinte: defina o comando **PRIMEIRO.COM** da seguinte forma

```
TO PRIMEIRO.COM :LISTA
PRINT FIRST :LISTA
END
```

e depois compare o resultado de cada uma das instruções seguintes

```
FIRST FIRST [[a < b] [b < c] [a < c]]
```

```
PRIMEIRO PRIMEIRO [[a < b] [b < c] [a < c]]
```

```
PRIMEIRO.COM PRIMEIRO.COM [[a < b] [b < c] [a < c]]
```

lendo com atenção as mensagens que eventualmente sejam comunicadas pelo computador.

Em conclusão, mesmo o programa tão elementar como a tradução de **FIRST** exige o procedimento **OUTPUT**. Na realidade, compreender a utilização do comando **OUTPUT** é indispensável para quem queira avançar, pouco que seja, na programação em LOGO.

Eduardo Veloso