Operators DEFI_LIST_REEL

1 Drank

To create a strictly increasing list of realities.

The list can be given “in extenso” by the user, or, it can be formed from under lists defined in “constant step”.

Product a data structure of the listr8 `type`.

2 Syntax

```
Lr      [listr8] = DEFI_LIST_REEL

       (         /♦VALE=lr8                        ,
[l_R]

                 /♦DEBUT=debu                  ,                        [R]

                       ♦INTERVALLE=      (_F ( ♦JUSQU_A    =r1       ,    [R]
                                               ♦/NOMBRE       =n1    ,    [I]
                                                  /PAS    =r2       ,    [R]
                                        ),),


                 ◊INFO=/1                    ,
[DEFAULT]
                                    /2     ,

                 ◊TITER=titer            ,
[l_Kn]

       )
```

3 Operands

3.1 Operand VALE

VALE = lr8

List of realities which will form the data structure listr8 result.
This list can be built from a python list.

3.2 Operand debut

♦DEBUT =

It is the first reality of the list of realities which one wants to build.

3.3 Operand INTERVALLE

♦INTERVALLE=

♦JUSQU_A=r1

It is the end of the interval which one will cut out with a constant step.

♦/NOMBRE=n1

It is the number of steps which one wants in the interval which ends in r1.

/PAS =r2

It is the step of division interval.

3.4 Operand INFO

◊INFO=i

Indicates the level of printing of the results of the operator.

1: no printing,
2: printing of the list of realities created

3.5 Operand TITER

◊TITER=titer

Title which the user wants to give to his list of realities.

4 Remarks

- when one uses the key word NOT it may be that the number of calculated step is not rigorously whole. One "will then adapt" the last interval to fall down exactly on the end value (JUSQU_A). So for that, one modifies the step value of more than 1/1000 one emits an alarm,
- attention: this command produces a data structure listr8 which can be used only in the commands expecting such data structures and not in those which expect lists of realities (notation: l_R).

5 Examples

Example 1:

Let us imagine that one wants to create the list:

1. 3. 5. 10. 15. 20. 25. 26. 27. 28.

which is such as the step is: 2.de 1.à5
5.de 5.à25
1.de 25.à28 .

One can write:

```
Lr   = DEFI_LIST_REEL   (DEBUT=1            .,
                        INTERVALLE=    ( _F  (JUSQU_A=5   . , NOMBRE=2 ,   ),
                                         _F  (JUSQU_A=25  ., NOMBRE=4   ,   ),
                                         _F  (JUSQU_A=28  ., PAS=1      .,),),
                        )
```

Example 2:

To create the list: 1.3.12.13 .

One can write:

```
Lr   = DEFI_LIST_REEL   (  VALE =  (1. ,  3. ,   12. ,  13.), )
```

Example 3:

One can build a python list of this way.

```
Lr   = DEFI_LIST_REEL   (  VALE =  [sqrt (I) for I in arranges (5)],   )
```