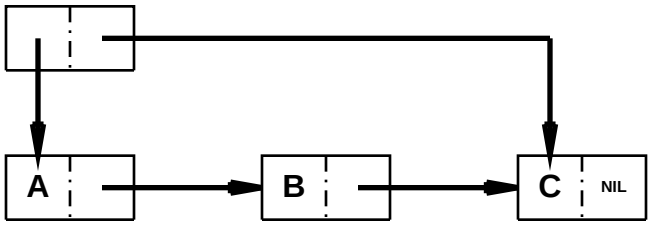
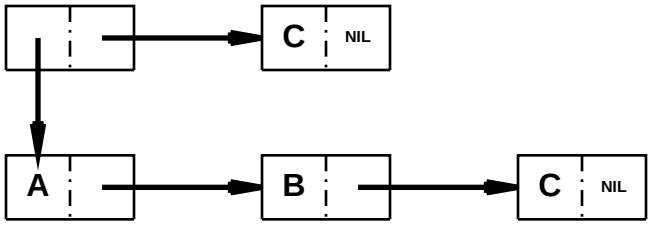
CIRCLPRINT

HPRINT is designed primarily for dumping circular or reentrant list structures (as well as other data structures for which READ is not an inverse of PRINT) so that they can be read back in by Interlisp. The CIRCLPRINT package is designed for printing circular or reentrant structures so that the user can look at them and understand them.

A reentrant list structure is one that contains more than one occurrence of the same EQ structure. For example, TCONC makes use of reentrant list structure so that it does not have to search for the end of the list each time it is called. Thus, if *X* is a list of three elements, (A B C), being constructed by TCONC, the reentrant list structure used by TCONC for this purpose is:

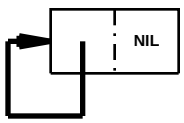


This structure would be printed by PRINT as ((A B C) C). Note that PRINT would produce the same output for the nonreentrant structure:

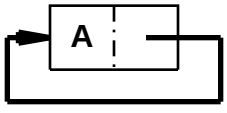


In other words, PRINT does not indicate the fact that portions of the structure in the first figure are identical. Similarly, if PRINT is applied to a circular list structure (a special type of reentrant structure) it will never terminate.

For example, if PRINT is called on the structure:



it will print an endless sequence of left parentheses, and if applied to:



will print a left parenthesis followed by an endless sequence of *A*'s.

The function CIRCLPRINT described below produces output that will exactly describe the structure of any circular or reentrant list structure. This output may be in either single- or double-line format. Below are a few examples of the expressions that CIRCLPRINT would produce to describe the structures discussed above.

First figure, single-line:

((A B *1* C)1)

First figure, double-line:

```
((A B  C)  1)
    1
```

Third figure, single-line:

(*1* 1)

Third figure, double-line:

```
(1)
1
```

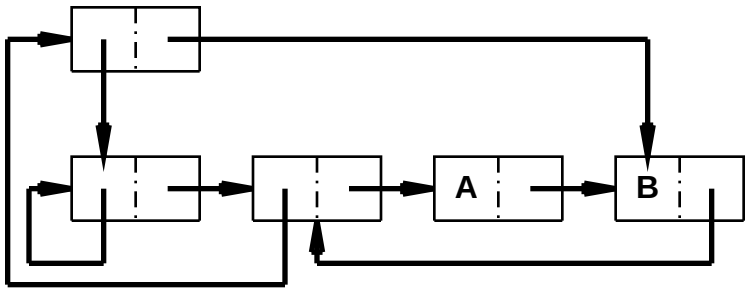
Fourth figure, single-line:

(*1* A . 1)

Fourth figure, double-line:

```
(A . 1)
1
```

The more complex structure:



is printed as follows:

Single-line:

(*2* (*1* 1 *3* 2 A *4* B . 3) . 4)

Double-line:

```
(( 1   2 A  B . 3) . 4)
21    3    4
```

In both formats, the reentrant nodes in the list structure are labeled by numbers. (A reentrant node is one that has two or more pointers coming into it.) In the single-line format, the label is printed between asterisks at the beginning of the node (list or tail) that it identifies. In the double-line format, the label is printed below the beginning of the node it identifies. An occurrence of a reentrant node that has already been identified is indicated by printing its label in brackets.

(CIRCLPRINT *LIST PRINTFLG RLKNT*) [Function]

Prints an expression describing *LIST*. If *PRINTFLG*=NIL, double-line format is used, otherwise single-line format. CIRCLPRINT first calls CIRCLMARK, and then calls either RLPRIN1 (if *PRINTFLG*=T) or RLPRIN2 (if *PRINTFLG*=NIL). Finally, RLRESTORE is called to restore *LIST* to its unmarked state. Returns *LIST.*

(CIRCLMARK *LIST RLKNT*) [Function]

Marks each reentrant node in *LIST* with a unique number, starting at *RLKNT* plus one (or one, if *RLKNT* is NIL). Value is *RLKNT*. Marking *LIST* physically alters it. However, the marking is performed undoably. In addition, *LIST* can always be restored by specifically calling RLRESTORE.

(RLPRIN1 *LIST*) [Function]

Prints an expression describing *LIST* in the single-line format. Does not restore *LIST* to its unCIRCLMARKed state. *LIST* must previously have been CIRCLMARKed, or an error is generated.

(RLPRIN2 *LIST*) [Function]

Same as RLPRIN1, except that the expression describing *LIST* is printed in the double-line format.

(RLRESTORE *LIST*) [Function]

Physically restores list to its original, unmarked state.

Note that the user can mark and print several structures that together share common substructures, e.g., several property lists, by making several calls to CIRCLMARK, followed by calls to RLPRIN1 or RLPRIN2, and finally to RLRESTORE.

(CIRCLMAKER *LIST*) [Function]

LIST may contain labels and references following the convention used by CIRCLPRINT for printing reentrant structures in single-line format, e.g., (*1* . 1). CIRCLMAKER performs the necessary RPLACAs and RPLACDs to make *LIST* correspond to the indicated structure. Value is (altered) *LIST.*

(CIRCLMAKER1 *LIST)* [Function]

Does the work for CIRCLMAKER. Uses free variables LABELST and REFLST. LABELST is a list of dotted pairs of labels and corresponding nodes. REFLST is a list of nodes containing references to labels not yet seen. CIRCLMAKER operates by initializing LABELST and REFLST to NIL, and then calling CIRCLMAKER1. It generates an error if REFLST is not NIL when CIRCLMAKER1 returns. The user can call CIRCLMAKER1 directly to ‘‘connect up’’ several structures that share common substructures, e.g., several property lists.