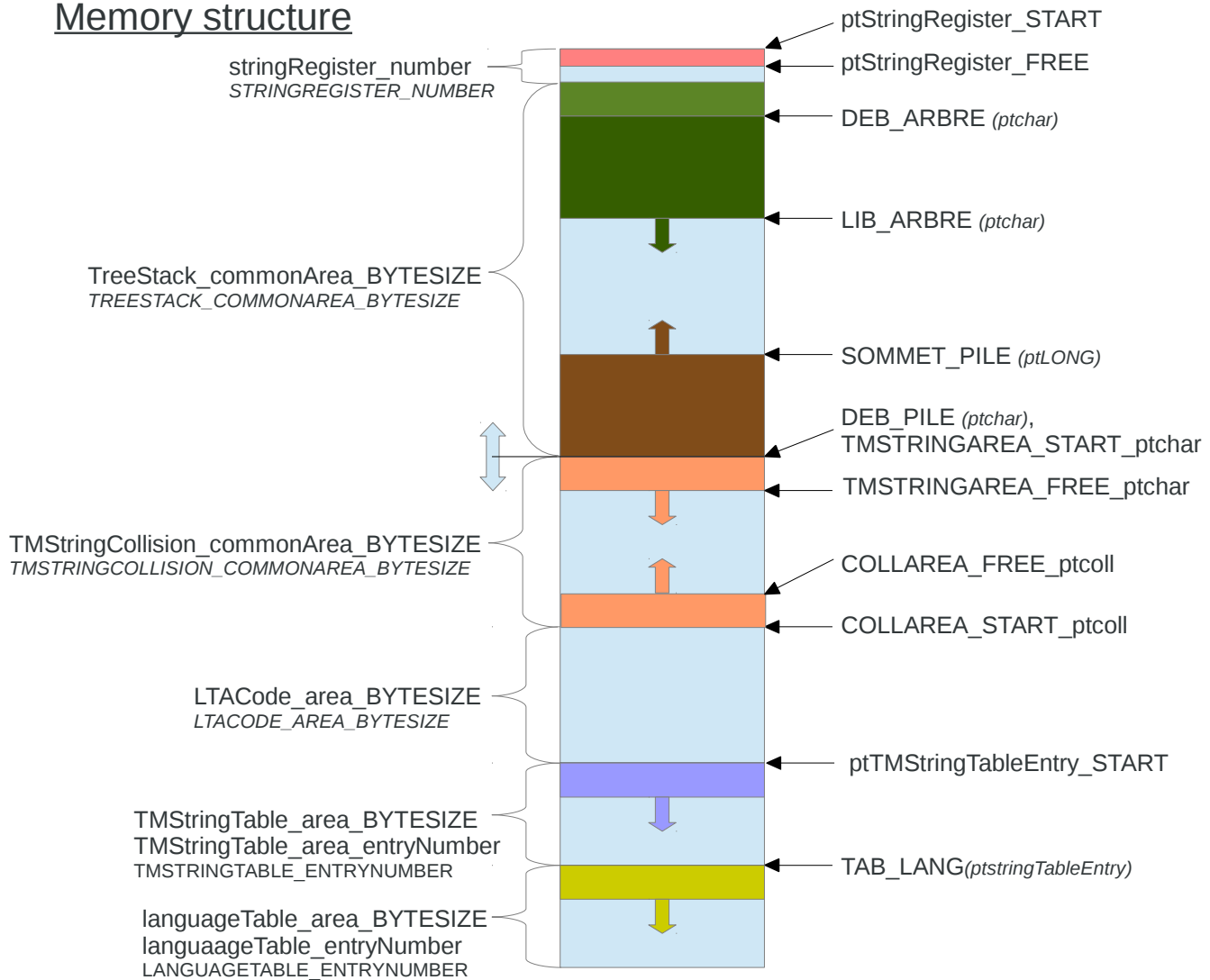


## Memory structure



- **void initTM\_memory(void).**  
required initial action that allocates memory space (set above variables); the implicate parameter size of the memory can be changed using the following environment variables:  
"T\_MEM\_ARB" (total size in byte allocated to trees and stack: TreeStack\_commonArea\_BYTESIZE)  
"T\_MEM\_CHAIN" (total size in byte allocated to strings: )
- **void resetTM\_treeMemoryStart(void).**  
make static the tree memory used until a call occurrence (DEB\_ARBRE get the current position of LIB\_ARBRE).
- **void openTM\_temporaryTreeMemory(void).**  
open a temporary area in tree memory (TEMPORARY\_LIB\_ARBRE get the current position of LIB\_ARBRE).  
openTM\_temporaryTreeMemory and closeTM\_temporaryTreeMemory must alternate (checked);  
openTM\_temporaryTreeMemory is the first allowed action in this alternation(checked).
- **void closeTM\_temporaryTreeMemory(void).**  
close a temporary area in tree memory all subtrees built since previous openTM\_temporaryTreeMemory occurrence are lost (LIB\_ARBRE get back the current value of TEMPORARY\_LIB\_ARBRE).
- **void clearTM\_userDataMemory(void).**  
clear the tree memory non static space (see resetTM\_treeMemoryStart) and init other memory areas.
- **long int getTM\_availableTreeSpace().**  
return the available space in tree memory (SOMMET\_PILE – LIB\_ARBRE +1).
- **long int getTM\_usedTreeSpace().**  
return the non static used space in tree memory (LIB\_ARBRE – DEB\_ARBRE).
- **bool isNotTM\_availableTreeSpace(int minimalSpace).**  
return true if the total amount of available contiguous tree memory space is smaller that the required space (minimalSpace)
- **void recup\_mem(pttrees pt1, pttrees pt2, pttrees pt3, pttrees pt4, pttrees pt5).**  
Memory tree garbage. **TO BE CHECKED.**

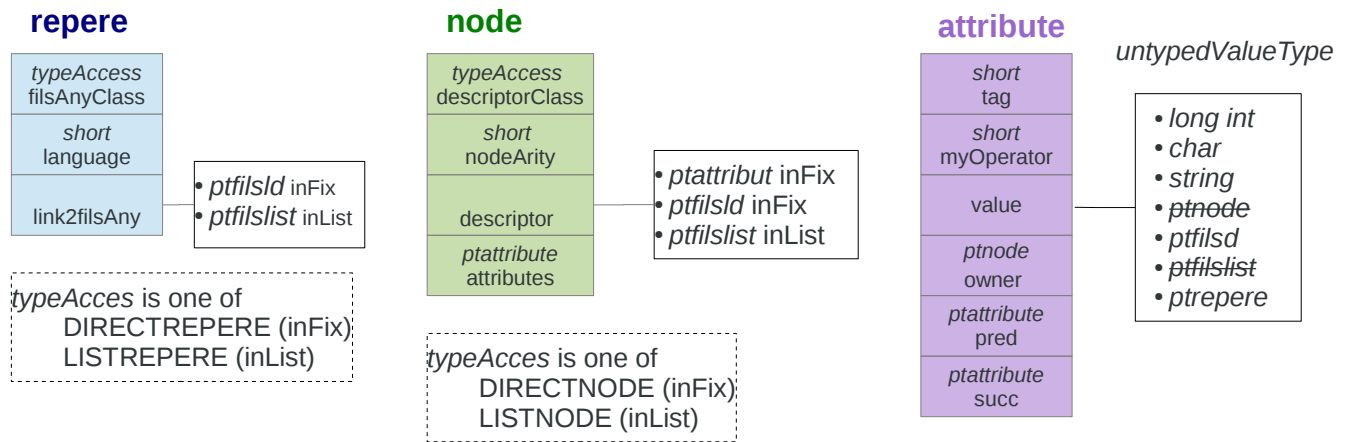
## Basic data structures: external (cf file incl/typesbase.h)

A *ptnode* is a link to a *node* structure that belongs or not to a tree.

A *tree* (a subtree) is a link to a *node* structure that belongs to a tree.

An *ptattribute* is a link to an *attribute* structure.

A *ptrepere* is a link to a *repere* structure

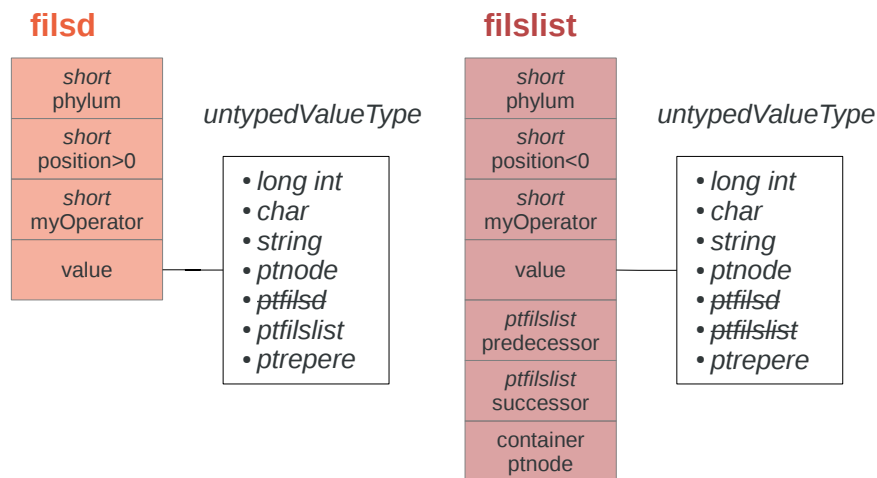


## Basic data structures: internal (cf file incl/typesbase.h)

A *ptfilsd* is a link to a *filsd* structure,

A *ptfilslist* is a link to a *filslist* structure,

A *ptfilsAny* is a link to either a *filslist* structure, or a *filsd* structure,



## Basic macros: internal (cf file incl/typesbase.h)

A macro XXX\_AAA\_SSS apply action AAA to an XXX structure where

XXX can be

REPERE, NODE, ANYFILS, FILSD, FILSLIST, ATTRIBUTE

AAA can be

GET, SET, ISIN, IS, HAS, MAKE

SSS generally refers to field name

For a macro XXX\_GET\_YYY\_TTT, YYY is the category of (NODE, ANYFILS, FILSD, FILSLIST, ATTRIBUTE) the resulting structure

## Basic constructors cf mod/lta\_cbase.cpp

**newTM\_repere()**   **newTM\_fixSizeNode(n)**

direct
current lang
NIL

direct	every		every		every
n	1		i		n
NIL	OPTROU		OPTROU		OPTROU
NIL	NIL		NIL		NIL

**newTM\_listNode()**

direct	every
-1	1
NIL	OPTROU
NIL	NIL

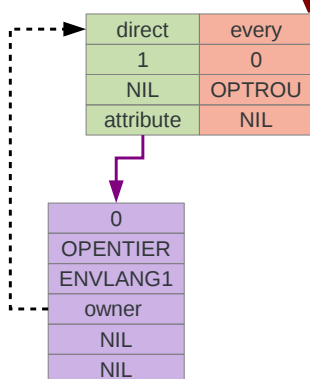
**newTM\_filslst()**

every
-1
OPTROU
NIL
NIL
NIL
NIL

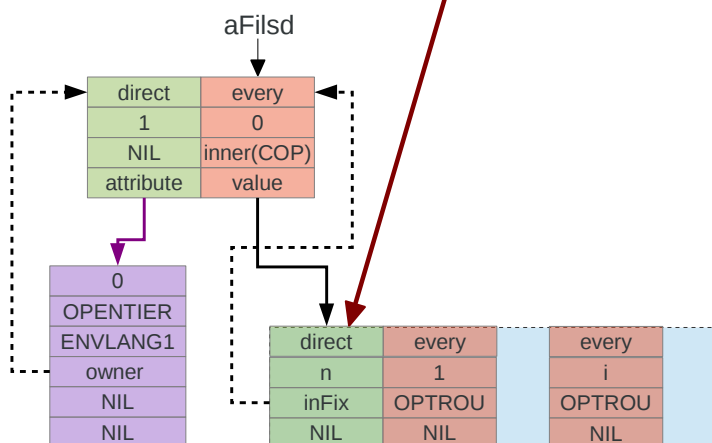
**newTM\_attribute(num)**

num
OPTROU
NIL
NIL
NIL
NIL

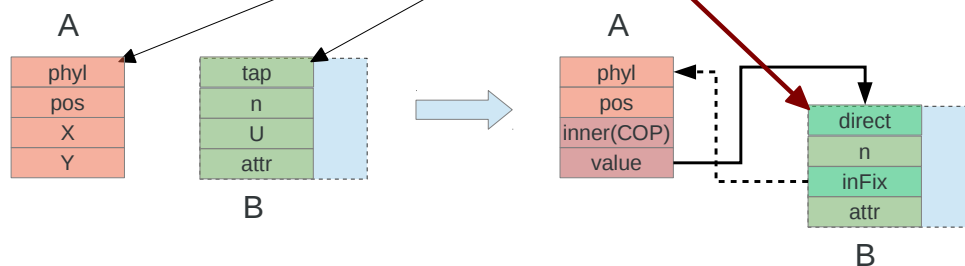
**newTM\_fixSonRoot\_emptyTree()**



**newTM\_userFixeTreeTop(COP, n)**



**linkTM\_filsd\_describedNode(aFilsd, aNode, COP)**

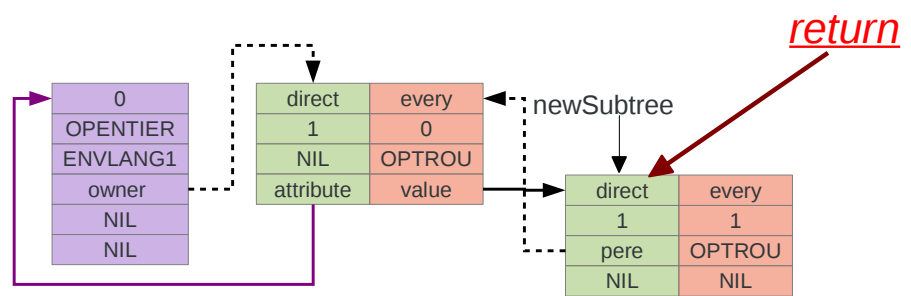


See also

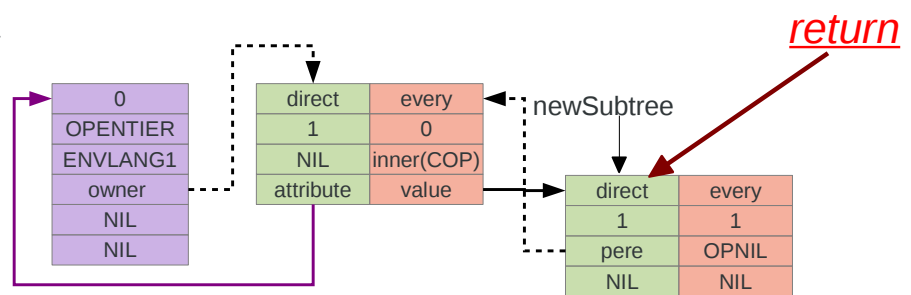
linkTM\_filslst\_describedNode  
linkTM\_filAny\_describedNode  
unlinkTM\_listElement  
cleanTM\_filAny

## API constructors cf mod/treefront.cpp

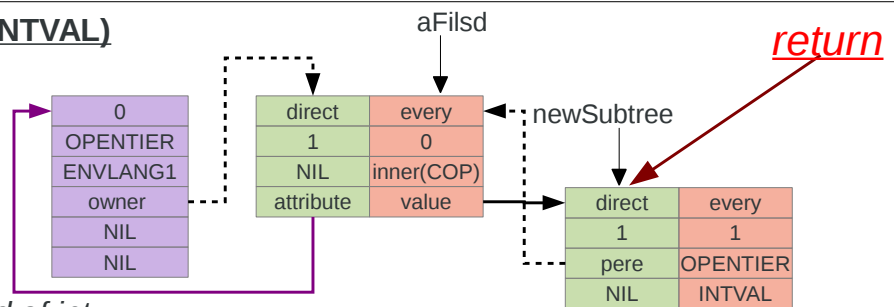
### makeTM\_hole()



### makeTM\_nilAtomTree(COP)

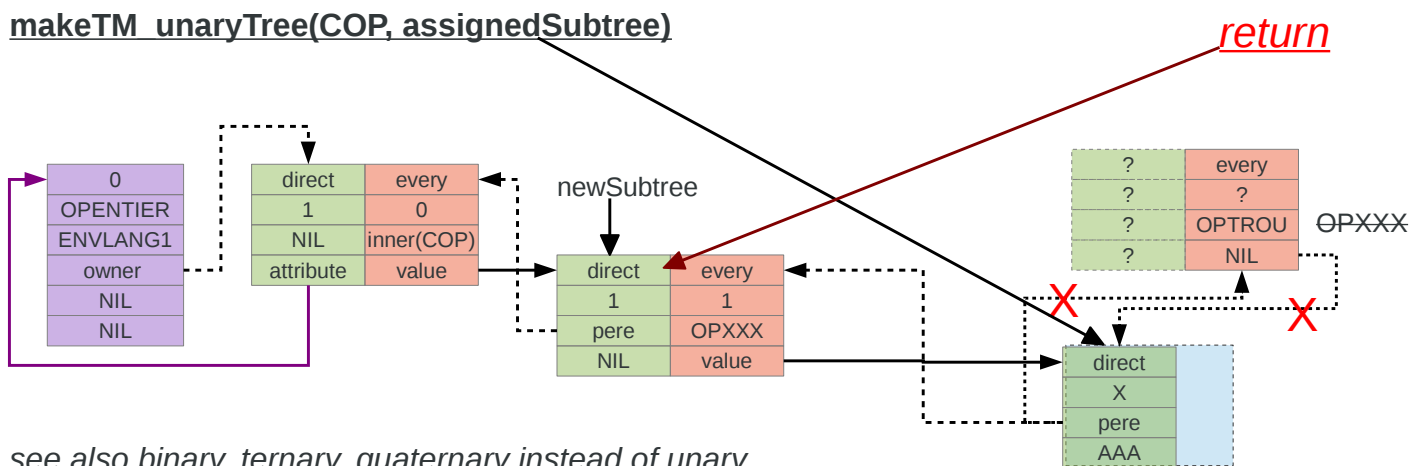


### makeTM\_intAtomTree(COP, INTVAL)



see also *char*, *TMString* instead of *int*

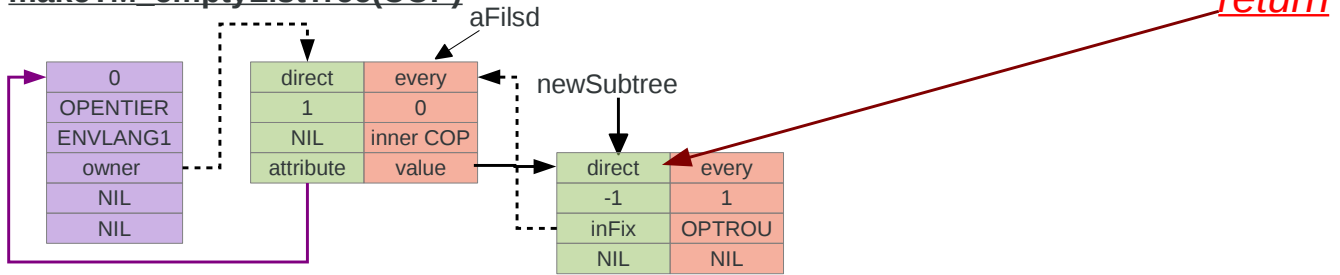
### makeTM\_unaryTree(COP, assignedSubtree)



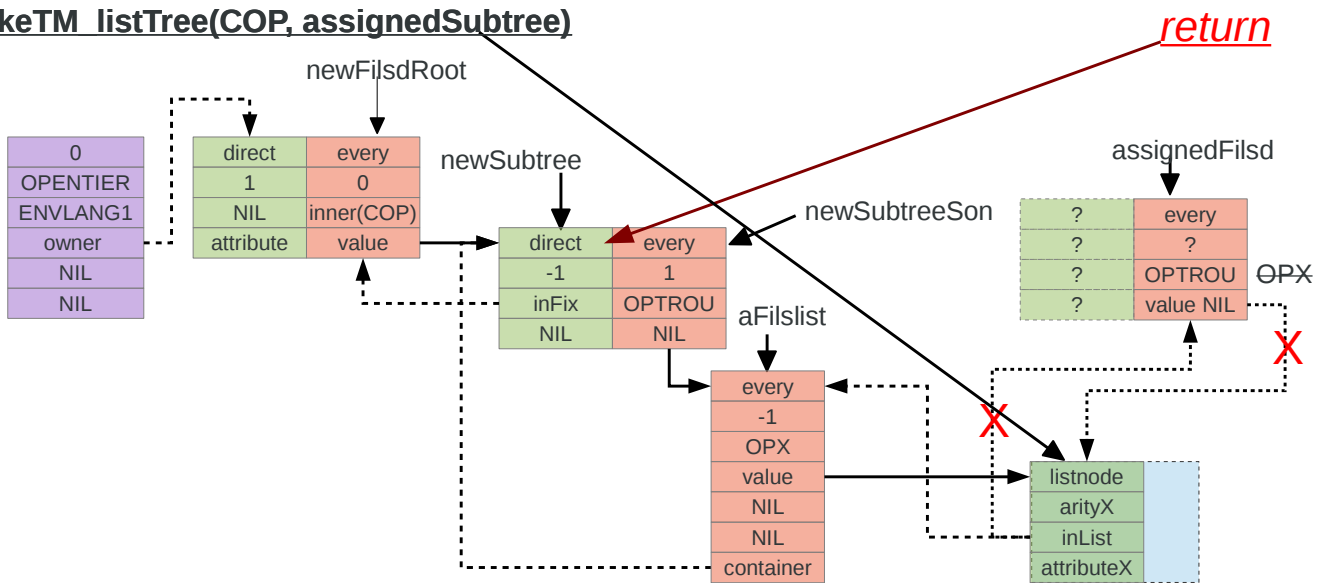
see also *binary*, *ternary*, *quaternary* instead of *unary*

## API constructors (cont'd) cf mod/treefront.cpp

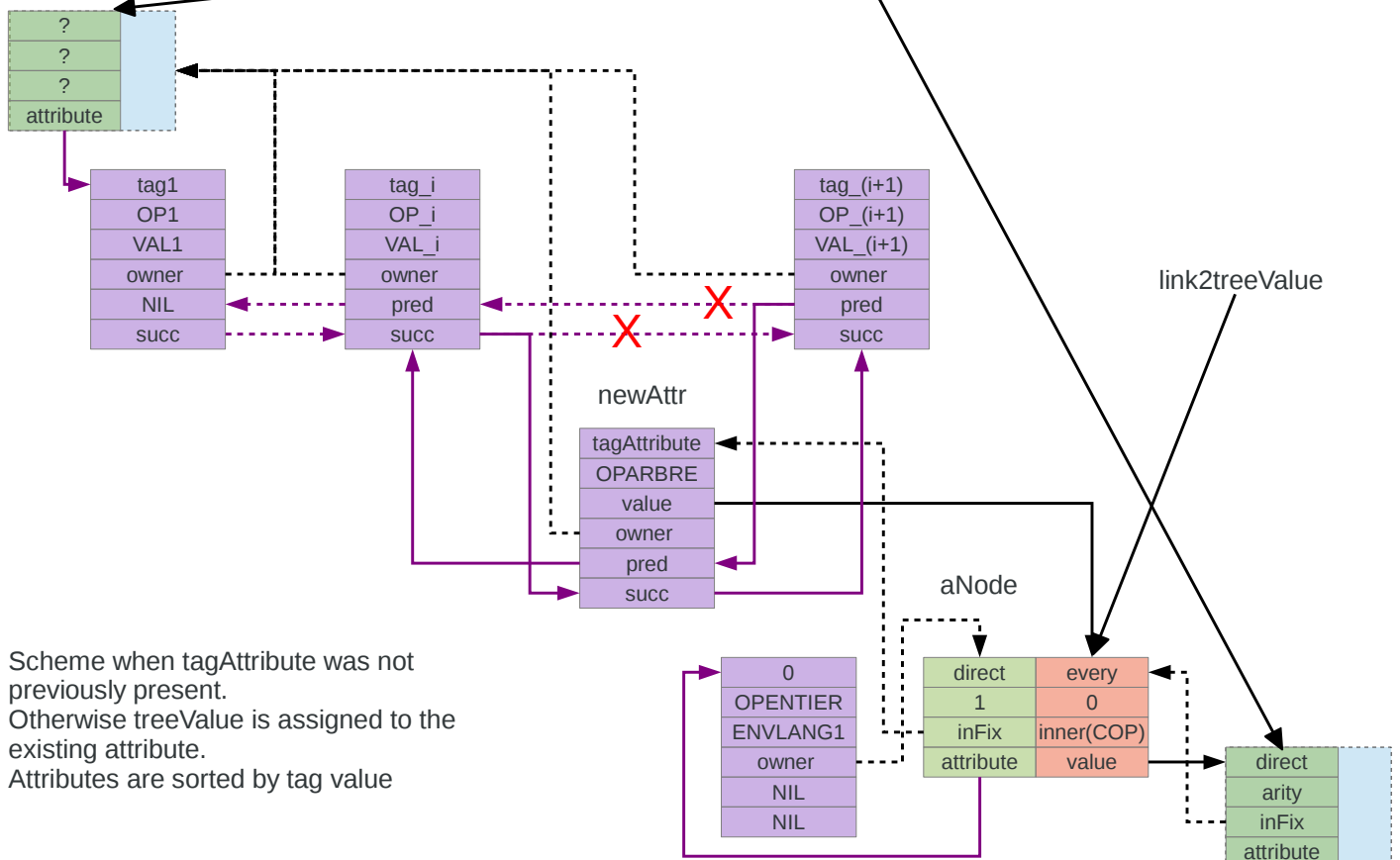
### makeTM\_emptyListTree(COP)



### makeTM\_listTree(COP, assignedSubtree)



### attachTM\_newTreeAttribute(aSubtree, tagAttribute, treeValue)



Scheme when tagAttribute was not previously present.  
Otherwise treeValue is assigned to the existing attribute.  
Attributes are sorted by tag value

see also *char*, *attachTM\_newIntegerAttribute*