Predicate	Meaning
Ldw *value*	Node level lower than *value*
Lup *value*	Node level greater than *value*
AP *value*	Region with area lower than *value*
FndT	The children are all leaves and at least one leaf is T
FndI	The children are all leaves and at least one leaf is I

Function	Meaning
b+(trees)	The sub-trees listed in `trees` are added as right-brothers to the current node.
b-(trees)	The sub-trees listed in `trees` are added as left-brothers to the current node.
+(trees)	The sub-trees listed in `trees` are added as right-children to the current node.
/(tree)	The tree starting in the current node is substituted with the tree described in `tree`.
rnd(tree)	The sub-tree described in `tree` is added to the children's list of the current node in a random point.

Table 1. Some logical predicates and alteration functions.

non-terminal. Similarly to the XSLT terminology we will refer to α as a *pattern* and to β as a *template*. Generally speaking the pattern defines the trees where the rule can be applied and the template describes how to build the output tree. Each labeled tree can be represented as a string by using a pre-fix notation where the label of a node *precedes*, in the related string, the list of the sub-strings which represent the sub-trees originated by the node's children. Similarly to string grammars we also allow the use of *wildcards* such as star-mark (*) or plus-mark (+) with the meaning of "zero or more repetitions of" and "one or more repetitions of" a tree, respectively. In this way we describe the *structures* of the trees we want to detect or build. In addition, we should pay attention to labels in MXY trees. A label describes a region, like the type of MXY-cut for internal nodes (horizontal or vertical cut along spaces or lines) and the region content for leaves (image, text block, or ruling line). The labels can contain additional information such as the block size and the number of children. We developed a tool that has been designed so as to allow an user to describe the rules in the most flexible way. Several kinds of tree grammars can be described in the tool, such as *regular* or *expansive* ones [14], and the user can define (in a specific JAVA compiled file) *logical predicates* or *alteration functions* to be used into rules.

A *logical predicate* (Table 1, left) is part of the rule pattern, it checks a local property of one tree (related to arrangement, number, and label of nodes) and returns a boolean value. The pattern conditions are satisfied only when all the contained predicates are satisfied. An *alteration function* (Table 1, right) is part of the rule template and modifies the tree structure. The user can change the type of a set of nodes (for instance blocks of text with images), or their arrangement. In addition to the labels introduced in Section 2.1 an additional label (x) is used in the grammar to identify any kind of block. Some examples of rules are shown in Section 4.1.