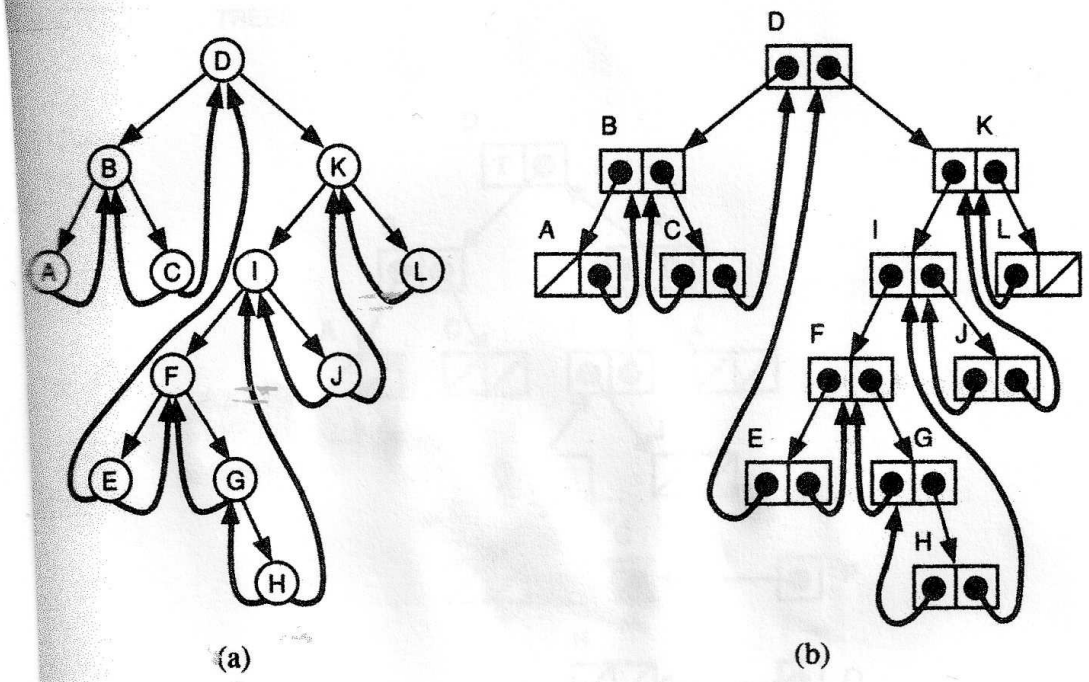

Figure 4.18 The tree of Figure 4.13(b) on page 114 as a threaded binary tree. (a) Logical organization; (b) internal representation.

Threaded Trees

Inorder traversal of part or all of a binary tree is a frequent operation in applications involving expression trees and search trees. The recursive methods of Algorithm 4.4 on page 107 and Algorithm 4.6 on page 113 have two principal disadvantages: they require extra memory to store a stack, and they require always starting at the root—it is impossible in general to start from an arbitrary node in the tree and to pass to the node's **inorder successor**, that is, the node that would be visited next during an inorder traversal. The following method stores additional structural information in the tree that makes it easy to find the inorder successor of any node. Since the tree is not altered during the traversal, the method is quite flexible and the tree can be stored in read-only memory. Moreover the method is entirely symmetric, and makes it equally easy to find the inorder *predecessor* of a node or to traverse the tree in "reverse inorder."

An observation that suggests how such a representation might be achieved is that in the natural representation of a binary tree, there are a great many Λ pointers. In a binary tree with n nodes there are $2n$ pointer fields, but $n + 1$ of these are Λ. (Each node except the root has a unique parent, so the number of pointers from parents to children is $n - 1$; the rest are Λ.) We make better use of these pointers by *storing as the* RC *of a node with no right child a pointer to the node's inorder successor, and storing as the* LC *of a node with no left child a pointer to a node's inorder predecessor.* These pointers must be distinguishable from ordinary pointers that point to left and right children, so one extra bit is required in each pointer field. Pointers of the new type are called **threads**. Figure 4.18 shows the tree of Figure 4.13(b) on page 114 with all the threads added; rather than showing an extra one-bit field with each pointer, we draw the threads using heavier lines.*