Difference list

From Wikipedia, the free encyclopedia

In computer science, the term **difference list** may refer to one of two data structures for representing lists. One of these data structures contains two lists, and represents the difference of those two lists. The second data structure is a functional representation of a list with an efficient concatenation operation. In the second approach, difference lists are implemented as single-argument functions, which take a list as argument and prepend to that list. As a consequence, concatenation of difference lists of the second type is implemented essentially as function composition, which is $O(1)$. However, of course the list still has to be constructed eventually (assuming all of its elements are needed), which is plainly at least O(n).

Difference lists as functions [edit]

A difference list of the second sort represents lists as a function *f*, which when given a list *x*, returns the list that *f* represents, prepended to *x*. It is typically used in functional programming languages such as Haskell, although it could be used in imperative languages as well. Whether this kind of difference list is more efficient than another list representations depends on usage patterns. If an algorithm builds a list by concatenating smaller lists, which are themselves built by concatenating still smaller lists, then use of difference lists can improve performance by effectively "flattening" the list building computations.

Examples of use are in the *ShowS* type in the Prelude of Haskell, and in Donald Bruce Stewart's difference list library for Haskell ⧉.

External links [edit]

- Open Lists and Difference Lists ⧉ in Prolog
- Difference Lists ⧉ in Haskell (programming language)

This algorithms or data structures-related article is a stub. You can help Wikipedia by expanding it.

Categories: Algorithms and data structures stubs | Computer science stubs | Linked lists