MC39 Homework 5: Keeping Track of List Ends

Max Hailperin

Due: March 18, 1996

Another possible use for class derivation is to change the performance characteristics of a class. (Or, more precisely, to make a derived class that has different performance characteristics than the base class does—the base class's performance remains unchanged.) In this assignment, you will derive a class from Wang's `ArbList` for this reason.

There are a number of applications of lists in which quick access to the last list cell is an important operation. For example, this would be the case if items always need to be added to the end of the list. Wang's `ArbList` class has a `last` member function, which returns a pointer to the last `ArbCell`, but it requires time $\Theta(n)$, where n is the number of items on the list. Thus, building an n element list by repeatedly appending at the end (using Wang's `append` member function) would wind up taking time $\Theta(n^2)$, which would be unacceptable in many applications.

Define a new class, `ArbListFastLast`, derived publicly from Wang's `ArbList`. You should not make any changes in `ArbList` or `ArbCell`. (To save you fussing with the diskettes in the backs of your books, I've put `ArbList.C` and `ArbList.h` in the directory `~max/MC39`.) Your new class should not extend the public interface specification at all, but rather should simply have a new private data member to keep track of the last cell and new implementations for some of the public member functions. These new implementations should result in the `last` and `append` member functions being constant-time, while not changing the asymptotic order of growth of any of the other member functions.