Datum-weak Hashtables

Copyright © 2010 Aaron W. Hsu <arcfide@sacrideo.us>

1. Overview

Datum-weak Hashtables are associations whose datum or value association does not create a strong reference to the value. That is, even if there is an association from a given key to some value existing in the hashtable, that existance does not prevent the value from being collected by the garbage collector.

These tables act and behave just like regular hashtables. They have the same set, reference, and update operations, as well as size and deletion. However, they are not recognized at the moment as hashtables using the hashtable? predicate, and you must use the datum-weak-hashtable- prefix instead of using the hashtable- prefix for the operations. Other than this, they work the same way.

2. Implementation Notes

There is current a known memory leak in the registration of collection handlers. This means that you leak memory for every allocation of a hashtable that you make.

The implementation itself uses two hashtables. A main hashtable which holds associations from keys to weak pairs containing the values, and a hashtable whose keys are weak, keyed on the values of the main hashtable with cleanup procedures as the data. A guardian is set for the cleanup procedures, and a collection handler is registered for every hashtable that is created. This collection handler checks for unreachable cleaners, and if found, runs these cleaners. These cleaners remove the entry from the main hashtable, assuming that the main hashtable has that entry. It is possible that the entry could be replaced with a new value, in which case the cleaner should do nothing to change the main hashtable.

3. Future To Dos

Firstly, the memory leak should be cleaned up.

Secondly, I believe it is possible to remove the dependence on checking for the hashtable update explicitly if we store a flag with each cleaner, and allow the cleaners to take an additional parameter which tells the cleaner to set that flag. When this flag is set, the cleaner will do nothing. Since there are a small number of ways in which a hashtable could be updated, this should be same, and we can just set this flag every time that we change the value of the hashtable, thus making sure that the old cleaners will not affect the new entry.