A generic hash table interface specification for Common Lisp

Ingvar Mattsson (ingvar@hexapodia.net)

November 10, 2006

Chapter 1

Generic, extendable hash tables for Common Lisp

1.1 Rationale

The hash table interface specified in the Common Lisp standard is only guaranteed to work with keys that are considered equal with either of EQ, EQL, EQUAL or EQUALP. It is sometimes useful for an application to have hash tables using keys with different equality predicates.

1.2 Interface specification

1.2.1 Package

All symbols of a generic hash implementation should be accessible in a package named so that :genhash is a designator for the package. This has the limitation of restricting a developer to have at most one GENHASH implementation loaded without having to do renaming. It has the advantage that an implementation can probe for the existence of a GENHASH implementation using FIND-PACKAGE.

The functional interface presented here is a minimum interface. The underlying implementation is up to the implementor, though the reference implementation uses CLOS, generic functions and methods of those.

The interface and operation available on generic hash tables are designed to be as close as possible to those available for built-in Common Lisp hash tables.

1.2.2 Functional interface

register-test-designator

(register-test-designator test-designator hash-function equal-function)

This function registers a hash table test designator. It takes a hashing function (that should return an integer) and an equality predicate (so possible hash collisions acan be resolved).

Test designators (with the exception of the four pre-defined functions) must be symbols.

If the hash function ever returns a non-integer, the consequences are intentionally undefined, but implementors are enouraged to make sure an error is signalled.

The only restrictions on the hash function is that returns an integer value and it is required that for any two objects (O1 and O2) so that:

(equal-function O1 O2)

implies that

(= (hash-function O1) (hash-function O2))

This function will raise the HASH-EXISTS condition if an attempt to re-register a test designator if the hash-function and test-function are not sufficiently similar (the reference implementation considers "sufficiently similar" to mean "tests equal with EQL").

There are eight pre-defined test designators

- CL:EQ
- (function CL:EQ)
- CL:EQL
- (function CL:EQL)
- CL:EQUAL
- (function CL:EQUAL)
- CL:EQUALP
- (function CL:EQUALP)

make-generic-hashtable

(make-generic-hashtable &key size (test 'eql))

This function creates a new hashtable, using the test predicate specified. If the test predicate specified is any of 'CL:EQ, 'CL:EQL, 'CL:EQUAL, 'CL:EQUALP a generic hash table with that equality predicate should be constructed (without any need to pre-register them).

If the test predicate is any of (FUNCTION CL:EQ), (FUNCTION CL:EQL), (FUNCTION CL:EQUAL) or (FUNCTION CL:EQUALP), either a system hash table or a generic hash table, using the relevant test predicate, should be returned.

If the :size keyword is used, the value sould be treated as a hint to the initial sizing of the hash table.

If no suitable hash table can be constructed because the specific test doesn't signal a pre-registered generic hash table, the function will signal an UNKNOWN-HASH-TEST error.

hashref

(hashref key table &optional (default nil))

This function is the GENHASH function that corresponds to the Common Lisp GETHASH function. It should accept both system hash tables and GENHASH hash tables.

hashrem

(hashrem key table)

This is the GENHASH function that corresponds to the Common Lisp REMHASH function. It should accept both system hash tables and GENHASH hash tables.

hashclr

(hashclr table)

This is the GENHASH function that corresponds to the Common Lisp CLRHASH function. It should accept both system hash tables and GENHASH hash tables.

hashmap

(hashmap function table)

This is the GENHASH function that corresponds to the Common Lisp MAPHASH function. It should accept both system hash tables and GENHASH hash tables.

generic-hash-table-count

(generic-hash-table-count table)

This is the GENHASH function that corresponds to the Common Lisp HASH-TABLE-COUNT function. It should accept both system hash tables and GENHASH hash tables.

generic-hash-table-size

(generic-hash-table-size table)

This is the GENHASH function that corresponds to the Common Lisp HASH-TABLE-SIZE function. It should accept both system hash tables and GENHASH hash tables.

generic-hash-table-p

(generic-hash-table-p table)

This is the GENHASH function that corresponds to the Common Lisp HASH-TABLE-P function. It should accept both system hash tables and GENHASH hash tables.

1.2.3 Conditions

hash-exists

This condition (warning either a warning or an error, this is implementation-dependent) is signalled when re-registering a nickname with "sufficiently different" hash-function and equal-function.

unknown-hash

This error is signalled when trying to create a generic hash table of an unknown type.

1.2.4 Reference implementation

A reference implementation of GENHASH can be retrieved from http://src.hexapodia.net/genhash.tar.gz

1.2.5 Copying