

h GCL ANSI Common Lisp Test Suite

Paul F. Dietz*

Abstract

I describe the conformance test suite for ANSI Common Lisp distributed as part of GNU Common Lisp (GCL). The test suite includes more than 20,000 individual tests, as well as random test generators for exercising specific parts of Common Lisp implementations, and has re

indications in the 'Exceptional Situations' sections for operator dictionary entries, as well as tests for calls to functions with too few or too many arguments, keyword parameter errors, and violations of the first paragraph of CLHS section 14.1.2.3. When type errors are specified or when the CLHS requires that some operator have a well-defined meaning on any Lisp value, the tests iterate over a set of precomputed Lisp objects called the `univ`

semantics, it is easy to generate related, but different, forms that should yield the same result (thereby providing a test oracle.)

The Random Tester performs the following steps. For some input parameters t, b and s (each positive integers):

1. Produce a list of t, b symbols that will be the parameters of a lambda e

may be constrained to 64 bits while unboxed values in machine registers may have additional "hidden" bits. The difference

6 Directions For Future Work

The test suite still has a few areas that are not sufficiently tested. Self expanders need more testing, as do logical pathnames and file compilation. Floating point functions are inadequately tested. As mentioned earlier, it isn't clear what precision is expected of these functions, but perhaps tests can be written that check if the error is too large (in some sufficiently useful sense.)

The random compiler tester, as implemented, is constrained to generate forms that remain conforming as they are simplified. This limits the use of certain operators that do not take the entire set of integers as their arguments. For example, `ISQRT`

- [5] Christian Lindig. Random testing the translation of C function calls. At <http://www.st.cs.uni-sb.de/lindig/src/quest/quest.pdf>, Feb. 2005.
- [6] W. M. McKeeman. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, 1998.
- [7] Barton P. Miller, Louis Fredriksen, and Bryan So. An empirical study of the reliability of unix utilities. *Commun. ACM*, 33(12):32–44, 1990.
- [8] Glenford J. Myers. *Software Testing*. John W