CHEZ SCHEME VERSION 5 HIGHLIGHTS

Chez Scheme fully conforms to the IEEE/ANSI Standard for Scheme and implements all features described in the revised reports on Scheme. Since 1985, Cadence Research Systems has been dedicated to providing the best and most reliable Scheme implementation available. *Chez Scheme Version 5* will once again set the standard for Scheme implementations, with enhanced performance and many impressive new features.

IMPROVED PERFORMANCE — Most programs will run 15-50% faster in *Chez Scheme Version 5* than in Version 4 due to improvements in program analysis, optimization, register allocation, and code generation. Compile time has improved as well.

LEXICAL MACROS — Support for the **syntax-case** macro system provides pattern-based syntax checking and transformation while respecting lexical scoping for both high- and low-level macros. Local macros are supported and internal syntax definitions may be freely intermixed with variable definitions. The revised report **syntax-rules** form is provided as a simple macro defined in terms of syntax-case.

MULTIPLE VALUES — Complete and efficient support for multiple values has been added. On most architectures, the first several return values are passed in registers, and there is no added overhead for calls that do not involve multiple values.

SHARED INCREMENTAL HEAPS — New shared incremental heaps contain only those portions of the heap that have changed from a previously saved heap, allowing the creation of compact application heaps. Where supported, heaps are memory mapped, maximizing interprocess sharing and minimizing start-up time.

BETTER DEBUGGING SUPPORT — The trace facility now produces noticeably more readable output, and the object inspector has been extended with an **eval** command that permits the evaluation of expressions within the scope of a frame or procedure environment.

GUARDIANS AND WEAK PAIRS — Version 5's new weak pair and guardian mechanisms permit programs to protect objects from garbage collection or detect when objects have been reclaimed. These “finalization” mechanisms are useful for meta-level storage management and deinitialization.

GENERIC PORTS — A new generic port facility allows the creation of efficient, buffered ports with arbitrary input/output semantics, from simple bidirectional ports, string ports, broadcast ports, and transcript ports to ports that represent windows on a bit-mapped display or ports that represent subprocesses connected via pipes or sockets.

FOREIGN PROCEDURE INTERFACE ENHANCEMENTS — New type descriptors increase the flexibility of the interface between *Chez Scheme* and C, and improvements have been made in foreign procedure code size and efficiency.

Numerous other improvements include more pretty printer control parameters, a fast flonum to fixnum conversion operator, efficient block I/O, and more compile-time argument count checks.

Of course, *Chez Scheme Version 5* builds upon the many important features of Version 4 not mentioned above, including a fast incremental optimizing compiler, a sophisticated generation-based garbage collector, an extensible pretty printer with support for printing of cycles and shared structure, an engine-based multitasking facility, subprocess control support, compatibility support for first-class environments, and much more.

Chez Scheme Version 5 supports many new and existing platforms: Sun Sparc SunOS 4.X & Solaris 2.X, DEC Alpha AXP OSF/1 V2.X, Silicon Graphics IRIX 5.X, Motorola Delta MC88000 SVR3 & SVR4, Intel 80x86 NeXT Mach 3.2, Decstation/Decsystem Ultrix 4.3A, Intel 80x86 BSDI BSD/386 1.1, Intel 80x86 Linux.

©1994 Cadence Research Systems ■ 3814 Devonshire South ■ Bloomington, IN 47408 USA ■ 812/333–9269 ■ FAX: 812/332–4688