

LISP Pointers



VOLUME 2, NUMBER 3-4
JANUARY-JUNE
1989

TABLE OF CONTENTS

Letter From Editor, Van Deusen	1
Letters to the Editor, Camp.....	2
Puzzling with Current Puzzle, Queinnec and Seniak.....	4
Manipulating Sets in Common Lisp, Eliot	5
A Study of LISP on a Multiprocessor, Preliminary Version, Nuth and Halstead.....	15
Query-io, Dussud, submitted by Yellin	33
Lisp and Symbolic Computation Journal	40
(algorithms), Pavel	41
On Listing List Prefixes, Danvy	42
XI World Computer Congress Registration.....	48
Implementation Summaries, van Roggen	49

EDITORIAL BOARD

EDITOR

Mary S. Van Deusen
IBM Watson Research
P.O. Box 704
Yorktown Heights, NY 10598
(914) 789-7845
(914) 232-7490
(508) 384-2526
maida@ibm.com

LISP IMPLEMENTATIONS

Walter van Roggen
DEC AITG
290 Donald Lynch Blvd.
DLB5-2/B10
Marlboro, MA 01752
(508) 490-8012
vanroggen@hudson.dec.com

X3J13

Robert F. Mathis
9712 Ceralene Drive
Fairfax, VA 22032
(703) 359-0203
mathis@b.isi.edu

TECHNICAL ARTICLES

JonL White
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
(415) 329-8400
edsel!jonl@labrea.stanford.edu

PROGRAMMING ENVIRONMENTS

John Foderaro
Franz Inc.
1995 University Avenue #275
Berkeley, Ca. 94704
(415) 548-3600
jkf%franz.uucp@berkeley.edu

THE SCHEME OF THINGS

Will Clinger
University of Oregon
18 PLC
Eugene, OR 97403
(503) 686-4411
will at cs.uoregon.edu

QUERY-IO

Patrick Dussud
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
(415) 329-8400
dussud%jenner%ti-csl.csnet@csnet-relay

INT'L NEWS

Christian Queinnec
LIX Laboratoire d'Informatique de
l'Ecole Polytechnique
Route de Saclay
91128 Palaiseau Cedex
FRANCE
Eam: QUEINNEERPOLY11.BITNET
UseNet: mcvax!inria!poly!queinnec

ALGORITHMS

Pavel Curtis
Xerox PARC
3333 Coyote Hill Rd.
Palo Alto, CA 94304
(415) 494-4455
Pavel.pa@Xerox.Com

READINGS IN SCHEME

Ozan S. Yigit
York University
Dept. of Computing Services
4700 Keele Street, North York
Ontario, M3J 1P3
CANADA
(416) 736-5257 X3976

MAILING LIST

Marisa Defusco
IBM Research
PO Box 704
Yorktown Heights, New York 10598

This issue of the Lisp Pointers newsletter has been funded by I.N.R.I.A. and Lucid, Inc. Neither group has directed or controlled its publication. The opinions expressed herein are solely those of the authors, editors, publisher and other contributors and do not necessarily reflect the opinions of I.N.R.I.A. or Lucid, Inc. or the opinions of companies affiliated with individuals involved with this effort.

Dear Colleague,

It's official as of July 1st -- we're a technical publication of the ACM SIGPLAN (Special Interest Group in Programming Languages). What this means is that our "current" subscribers will continue to receive Lisp Pointers for one year for free.

All "new" subscribers will be expected to pay the rate set for Lisp Pointers by SIGPLAN. The next issue will contain a new subscription form which will include price information. Because we're a technical publication rather than a separate organization, we have the advantage that the cost of the publication will be the actual publication cost rather than a cost which includes overhead charges.

Once the Volume III, No. 1 issue has hit the street, all subscriptions will be considered "new" subscriptions. In other words, if you were borrowing a copy from your friend and have always intended to order your own, this is the time.

Now, what effects will you see from this change. First, our cover will be modified to fit within the style of SIGPLAN publication covers. Secondly, all issues must be censored by ACM (I once printed this in an editorial in Ada Letters and ACM censored the word "censored"). The only real effect is that ACM conferences which are advertised must have already passed their paperwork. The gentleman who handles SIGs at ACM headquarters, Fred Aronson, is one of their best and I'm looking forward to working with him again. And last, your requests for back issues will go to ACM headquarters rather than to me. I will be talking with Fred in the near future to discuss what to do with all the back issues of current publications. That will be in the next editorial.

My thanks to all the department chairs for their time and energy. They are the ones who have made us successful. Lisp Pointers will always be a changing publication, reflecting the needs we see today for the Lisp and Scheme communities. This means that departments will come and departments will go. The one that is going with this issue is Implementations. Walter van Roggen has worked hard with his section but there are just not enough contributions from implementors to keep the section alive. Unless Walter hears the sound of millions of hands clapping that they believe in Tinkerbell, I'm afraid that we've lost a very important department. My thanks to Walter for the splendid job he's done for us all.

Okay, folks, get your cards and letters coming. This publication is your publication and only works if you contribute. Look through the address list at the front and find a department chair and LET THEM HEAR FROM YOU.

Our thanks to Lucid and I.N.R.I.A. for sponsoring this issue.

Sincerely,



Mary S. Van Deusen, Editor



TEXAS INSTRUMENTS INCORPORATED
Computer Science Center

Clyde R. Camp, Senior Member Technical Staff
P.O.Box 655474 • MS 238 • Dallas, Texas 75265
(214)995-0407 • camp@csc.ti.com • COMPMAIL+ C.CAMP

December 20, 1988

Mary S. Van Deusen - Editor *LISP Pointers*
IBM Watson Research
P.O.Box 704
Yorktown Heights, NY 10598

Dear Ms. Van Deusen,

Ever since Texas Instruments first announced *PC Scheme*, I have been distributing, free of charge, a set of utilities and documentation which greatly enhance the development of application programs. Although written initially for the TIPC, everything except the graphics utilities also works on IBMs and IBM clones. They have now been upgraded to support *PC Scheme 3.0* and later releases.

The major packages are:

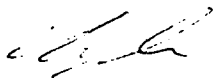
- **UTILITY** - Various text windowing, file printing and keyboard handlers which simplify writing application programs including:
 - A new top-level read-eval-print loop which uses an emacs/unix-like line-editor with the capability to scroll through and edit previous entries
 - Pop-up error window
 - Source File pretty-printer
 - Higher level window manipulators
 - General Purpose application prompts
 - Push/Pop Top Level Loop functions
- **SWI** - A convenient mechanism for invoking 8086 ASSY routines via SWI-INT.
- **HELP** - A user-extendable on-line help facility which includes all of the PCS functions and syntax as well as other information on PCS specific quirks.
- **GRAF** - (TIPC only) A graphics package for creating graphics "windows" somewhat analogous to text windows

- **PLOT** - (TIPC only) A general purpose function plotter
- **GAME1** - Self explanatory - non-graphics for IBM or TIPC
- **GAME2** - (TIPC only)
- **ERR_STAT** - more utilities for controlling the status window, testing for directories and disabling the gc-message
- **MENUSHEL** - two general purpose menu driven command shells handy for application development.

Anyone wishing to receive these utilities and documentation, send one *formatted* 260K 5-1/4" floppy or 720K 3-1/2" micro-floppy and a *self-addressed, stamped* return mailer to the following address.

Clyde R. Camp
Texas Instruments, Inc.
P.O.Box 655474, MS 238
Dallas, TX 75265

Regards,



Clyde R. Camp

Puzzling with Current Puzzle

CHRISTIAN QUEINNEC NITSAN SÉNIAK
Laboratoire de l'École Polytechnique
91128 Palaiseau Cedex
France

March 13, 1989 on 16:00

We wrote a few days ago a validation suite generator and exercised it on lambda and call/cc. We were surprised by some of its output which were meaningful albeit rather weird. They look like quite simple but seem to be unpublished to our knowledge.

Small is Beautiful

What does (call/cc call/cc) ?

Besides being one of the most simple puzzles written yet, it allows separating “jumpy” versus “pushy” call/cc (see [Danvy & Malmkjær] for more informations).

Puzzling with Current Puzzle

What does ((call/cc call/cc) (call/cc call/cc)) ?

Solutions

They are elsewhere in the bulletin.

LISP AND SYMBOLIC COMPUTATION: An International Journal

NOW AVAILABLE

Scope:

- Programming language notations for symbolic computing (e.g., data abstraction, parallelism, lazy evaluation, infinite data objects, self-reference, message-passing, generic functions, inheritance, encapsulation, protection, metaobjects).
- Implementations and techniques (e.g., specialised architectures, compiler design, combinatory models, garbage collection, storage management, performance analysis, smalltalks, flavors, common loops, etc.).
- Programming logics (e.g., semantics and reasoning about programs, types and type inference).
- Programming environments and tools (e.g., knowledge-based programming tools, program transformations, specifications, debugging tools).
- Applications and experience with symbolic computing (e.g., real-time programming, artificial intelligence tools, experience with LISP, object-oriented programming, window systems, user interfaces, operating systems, parallel/distributed computing).

Editors-in-Chief:

Richard P. Gabriel, Lucid, Inc. and Guy L. Steele Jr., Thinking Machines, Inc.

Articles of Volume 1, Issue 1 include:

Expansion-Passing Style: A General Macro Mechanism, R. Kent Dybvig, Daniel P. Friedman, Christopher T. Hayes; *OAKLISP: An Object-Oriented Dialect of Scheme*, Kevin J. Lang, Barak A. Pearlmutter; *The Mystery of the Tower Revealed: A Nonreflective Description of the Reflective Tower*, Mitchell Wand, Daniel P. Friedman; *Technical Issues of Separation in Function Cells and Value Cells*, Richard P. Gabriel, Kent M. Pitman

Submissions and more information contact:

Jan Zubkoff
Associate Editor, LASC
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
jls@lucid.com
415/329-8400

For subscriptions please contact:

Kluwer Academic Publishers
P.O. Box 358
Accord Station
Hingham, MA 02018-9990
617/871-6600