



Department of  
Computer Science  
Newsletter

no

3  
March  
1975  
?

yes

**PRINTOUT**

Volume 1

Number 3

PRINTOUT, the newsletter of the Department of Computer Science of the University of Maryland at College Park, is published sporadically and distributed to faculty, staff, and students in the Department. Opinions expressed in signed articles may be those of the author, but no opinions represent the policy of the Department, or of the College Park Campus, or of the University.

Contributions may be submitted to the editor, and unless they are obscene or seditious they will probably be used, but minor editing may be done. Complaints directed to the newsletter will be investigated and publicized when possible. It is well to keep in mind however that the Department is subordinate to higher levels of administration, not the other way around; and, the Department does not provide computing service to the campus. Complaints in these areas are best directed to other publications.

## STAFF

EDITOR  
DICK HAMLET

# PRINTOUT

Volume 1  
Number 3

## CONTENTS

DYNAMIC ALGEBRA	R. E. Noonan	3
QUESTIONS	M. Hecht, R. Hamlet	4
MEMORIES	Dick Hamlet	5
CONSULTING	Paul Thompson	8
PUBLICATIONS, ETC.		8
JOHN GANNON		8
(MORE) BUZZ WORDS		9
PROFESSIONAL NEWS		10
COMP. SCI. COMP. REV.		10
THE GOSPEL ACCORDING TO SAINT HEXADECIMAL		11
COLLOQUIUM		11

# DYNAMIC ALGEBRA

Dynamic algebra was originally developed by the late Prof. H. Aiken of Harvard University in order to both simplify programs and allow programs to be mechanically simplified. As such, it emphasizes the commonplace expressions of arithmetic and algebra.

Dynamic algebra programs use only assignments, I/O, loops, and subroutine calls. The use of transfers whether explicit, via the go to or exit, or implicit via the if or case, is discouraged. (Now that all reasonable people have adjusted to the loss of the go to or perhaps losing if's isn't all that bad either.) However, dynamic algebra adds eight new operators (actually redefinitions of old ones), namely:

$=, \neq, >, \geq, <, \leq$

where

$$a \text{ rel } b = \begin{cases} 1 & \text{if } a \text{ rel } b \text{ holds} \\ 0 & \text{otherwise} \end{cases}$$

and G (greater), L (lesser)

where

$$G(a,b) = (a > b) * a + (a \leq b) * b$$

$$L(a,b) = (a < b) * a + (a \geq b) * b$$

(The relational operator treatment will be familiar to SIMPL users.)

At this point you are probably convinced that lots of programs cannot be written without using an if. Trivially, any if can be simulated using a while but that just begs the question. Given dynamic algebra hardware (that is, which implements the eight operators above), if's are needed only for non-total functions (e.g.,  $x/0$ ) and for recursion (otherwise there is no way to break it and start returning). The following argument (I hope) will convince you.

What are if's used for? Clearly, if their only reason for existence was to control other control statements, we would not need them. No, they are used to control assignments and I/O. However, I/O is just a "funny" kind of assignment, and assignments are easily handled as follows:

1) if c then x := a else x := b fi  
becomes x := c \* a + ( $\neg$ c) \* b

2) if c then x := a fi  
becomes x := c \* a + ( $\neg$ c) \* x

Now that we have gotten rid of if's, we promised earlier (actually, Aiken did) to show you how programming becomes simpler (admittedly, it still takes some getting used to). Traditional programming is what might be called case-oriented programming. If's or case's are used to break things up into separate cases and then code is written to handle each case separately. The problem with this method is that the various key variables may not be left in a consistent state under different cases. For example, one case may assume that a stack pointer points to the top of the stack while another assumes that it points to the next empty position. Dynamic algebra programming is characterized by the definition of each variable under all cases.

This contrast in philosophies is illustrated in programs for infix-to-postfix translation. (Both programs lack initialization and a second loop to copy the stack to the output.)

SO(i) = ith INPUT TOKEN  
SHR(x) = PRECEDENCE OF TOXEN x  
OPS(j) = OPERATOR STACK TOP  
POL(k) = kth OUTPUT TOKEN

x	SHR(x)
<id>	0
)	1
(	2
+,-	3
*,/	4
↑	5
#	-1

```

while SO(i)\#\#' do /*STILL MORE INPUT*/
case SHR(SO(i)) of
\0\ POL(k) := SO(i) /*OPERAND*/
  k := k+1
  i := i+1
\1\ if SHR(OPS(j)) = 2 /* ( ) */
  then j := j-1
       l := i+1
  else POL(k) := OPS(j)
       j := j-1
       k := k+1
  fi
\2,3,4,5\ /* OPERATOR ,OR, ( */
if SHR(SO(i)) ≤ SHR(OPS(j))
then POL(k) := OPS(j)
  j := j-1
  k := k+1
else j := j-1
     OPS(j) := SO(i)
     i := i+1
  fi
fi
fo
od

```

```

while SO(i)\#\#' do /*STILL MORE INPUT*/
ICP := SHR(SO(i))
STP := SHR(OPS(j))
POL(k) := (ICP=0)*SO(i)+(ICP\0)*
           (ICP≤STP)*OPS(j)
k := k + ((ICP=0) or (ICP≤STP))
j := j + (ICP\0)*((ICP>STP)-(ICP≤STP))
OPS(j) := (ICP≤STP)*OPS(j) +
           (ICP\0)*(ICP>STP)*SO(i)
i := i + ((ICP=0) or ((ICP=1) and
                 (STP=2)))
           or ((ICP>1) and (ICP>STP)))
od
/* COPY OPS TO POL */

```

Note that the structured programming version contains 18 statements while the dynamic algebra version contains only 8 statements. Dynamic algebra programs are typically 25-50% shorter because of the single assignment to each variable within a loop. However, the assignments are considerably longer, (But think of the fun the optimizer pass of a compiler can have with the dynamic algebra version. After all, what can be done with statements like  $K:=K+1?$ ).

--R.E. Noonan

# ? S

The British Computer Society awards a case of wine to the first person to solve computer problems posed in its paper Computing. You can put your cork-screws away.

## ?

ONE. In SIMPL-T one can employ a temporary location to permit the interchange of two variable values:

```
INT A,B,T
```

```
...
```

```
T:=A A:=B B:=T
```

However, it can also be done without the use of a temporary:

```
INT A,B
```

```
...
```

```
INT FUNC XOR
RETURN(A.XOR.B)
```

```
...
```

```
A:=XOR B:=XOR A:=XOR
```

How many other distinctly different ways can you find to do the interchange in SIMPL-T without using a temporary? How obscure can you make the code?

-- M. Hecht

## ?

TWO. Is it possible to write a SIMPL-T program with the property that when two identical copies of the program are submitted, one as program and the other as data, the output is "ME", while any other data deck supplied produces "THEM" instead? (No input/output operations are permitted except card input and printer output.)

--R. Hamlet

# MEMORIES

*(Editor's note. The documents which appear below were found stacked in the trash outside an unnamed office in an unnamed computer center. Some seem to be replies to correspondence which is missing.)*

---

TO : Oliver Twist, Manager  
Computer Users Service  
FROM: R. Bulova, Supervisor  
Systems Programming

I received the complaint from the student whose three-line program wouldn't compile; I suppose there were a lot of them much the same since the program was put on the board in class. But as you know, the systems staff is not responsible for the compilers supplied by RCA, and we are not about to start fixing every problem that some student comes up with. If we did that, since students are the primary users of the system, we would have no time for anything else. Please don't send me any more complaints of this kind. Tell the people with problems anything, promise them anything, but keep it to yourself. If they're students, they will soon be gone anyway, and the instructors know better than to bother us with this sort of thing.

---

TO : John Swazey, Director  
Computer Center  
FROM: R. Bulova, Supervisor  
Systems Programming

Attached is the crash report for this month; as you can see, things are going along very well, and I daresay our record is better than any similar installation. It just shows what a talented set of dedicated professionals can do when properly managed.

*(Attached)*

<u>DATE/TIME</u>	<u>PROBLEM</u>	<u>RESOLUTION</u>
1 Sept 0005	System stopped accepting jobs from one card reader, operator had to restart to continue.	Operations problem--probably had the reader turned off.
1 Sept 0006	System wouldn't come up from restart above. Down until 0940 when someone in systems arrived.	Problem with remote terminal--user tried to dial in when system was not ready. Fixed by turning off terminals for remainder of the day.
1 Sept 0945	System apparently alive, but giving very slow service, and utilizing almost no CPU time.	No action. To learn what was wrong would have required shutting down system, with perhaps serious consequent problems (see 0006 above).

1 Sept 0955	System halted. Was found to be servicing one particular process, and when that process was killed, everything went back to normal.	No action.
1 Sept 1019	System halted at location 10CF. This is a problem in which the memory links become entangled with the scheduling of the resource allocation queue manager, with the result that an error is detected by the routine PUNT, and the system is halted before serious damage can occur. We know exactly what this is, but are still working on the fix. (See the Crash report for June).	Fix in progress.

*(Apparently several pages of this report are missing here...)*

30 Sept 2301	Operator stopped the system because it was too heavily loaded and nothing was getting done.	Operations problem, since system should not have been permitted to degrade in this way.
--------------	---	---

MONTHLY CRASH SUMMARY:	Operations: 80
	Hardware: 18
	Users: 38
	Systems: 3
	TOTAL: 139

TO: John Swazey  
 FROM: R. Bulova

Your budget for the coming year seems essentially sound to me, except that I would like to see the following items included:

1. New operations console. The staff has been complaining that the existing console is out of date. They point out that it gives a poor impression of the system to see a grungy old terminal there. The recommendation that the old console be cleaned up was rejected because this is not a permanent solution, whereas a brand new terminal would be. The cost of \$7000 is particularly reasonable when you note that except for some system down time during installation, the new console will make no change in the operation whatsoever.

2. An additional systems programmer. My staff is currently stretched pretty tightly, particularly since some of the people like to work at home, and are difficult to contact when an emergency arises. (To give just one instance, I haven't seen S. Crumple since last July 17th.) However, we cannot use anything less than a Systems Programmer VI grade, since with less than the \$20K/year salary we couldn't attract anyone good enough to really help.

3. Office suite. I like the proposal for our own offices in the new building across campus from the computer itself. It will remove the distractions to physically get away from the machine. As long as we have our hard-wired priority terminals, we can continue to work there just as well. However, I think you

should investigate the possibility of getting us a separate telephone system which we can use to call out, but which cannot receive calls. That way, we would not be bothered by trivia, particular from old Twist, who sometimes gets the idea that we have nothing better to do than hold user's hands.

---

TO: Oliver Twist  
FROM: R. Bulova

You are hereby formally notified that we put up version 6.21E/Y of the operating system two days ago, and it has been running successfully since then. However, there are certain undocumented changes in this system which may cause some user complaints, and you might want to know about them. For example, no absolute program images created under prior systems will run on 6.21, and the linking loader has a new input format for creating such images. Users can easily make the adjustment, and if you'll tell me what kind of problems they seem to be having, I'll get started writing something up next week. In our tests, we observed that the new system is a little less crash resistant than the old one, but so far its record is excellent. (This may be because for some strange reason the load dropped off dramatically after the first day of running the new system.) I suppose like all new systems, this one will cause a flurry of bitching about the lack of prior notification of the change. You know my position on that very well: we have been working on this system for months, and if we had to announce everything it would slow us down to a crawl, not to mention the impossibility of even finding out what all the changes are so as to announce them.

*(Further scrounging in the trash pile turned up nothing more interesting than used sandwich bags.)*

--Dick Hamlet



Timesharing at the Computer Center

According to the ALGOL 68 Report, a REFETY ROWSETY ROWWSETY NONROW slice may be rewritten as weak REFETY ROWS ROWWSETY NONROW primary, followed by a sub symbol, followed by ROWS leaving ROWSETY indexer, followed by a bus symbol.

For example, X[3].

For details, see the Report, Section 8.6.1.1.

## Consulting

The Computer Science Department's program consulting service is alive and living in Room 2347 of the Computer Center. The service, staffed by the graduate teaching assistants, is intended as an aid in degugging programs written by the students (all levels), faculty and staff of the University. To aid the consultant in locating your bug, bring output from several runs, flowcharts and data sets. Bugs of all types are found, in programs of all languages supported by the Center. The service is staffed continuously from

8:30-5:00 Monday - Thursday  
9:00-5:00 Friday.

Any questions, suggestions, complaints should be directed to Paul Thompson Room 2342, X4247.

## Publications, Etc.

- Basili, V.R. Sets and graphs in GRAAL. *Proceedings of the ACM 27th National Conference, 1974*, 289-296.
- VanderBrug, G. and J. Minker. State-space, problem-reduction and theorem proving: some relationships. *CACM 18* (Feb., 1975), 107-115.
- Zelkowitz, M.V. Structured operating system organization. *Info. Proc. Letters 3* (Nov., 1974), 39-42.
- Zelkowitz, M.V. *Pointer Variables within a Diagnostic Compiler*. TR-343 University of Maryland Computer Science, December 1974.
- Rosenfeld, A. Multidimensional formal languages. In W.R. Spillers, ed., *Basic Questions of Design Theory*, Elsevier, 1974, 417-420
- Rosenfeld, A. Digital straight line segments. *IEEE Trans. on Computers C-23* (Dec., 1974), 1264-1269.
- Weszka, J., R. N. Nagel, and A. Rosenfeld. A threshold selection technique. *IEEE Trans. on Computers C-23* (Dec., 1974), 1322-1326.

## John Gannon

As the newest member of the Department's faculty, Assistant Professor John Gannon has been placed in our newest office. As a service to the Department (and for Dr. Gannon's personal use) the following algorithm finds that office:

From the 4th floor of the Computer Science Center, go through the double doors into the Space Sciences wing. Continue straight ahead through another set of double doors (not too fast, one of these is usually locked tight), then turn left at the first corridor. If there is no Xerox room on your left just after the corner, you have the wrong building, corridor, double doors, etc. Gannon's office is now just around the first bend.

John comes from Rhode Island, he's 26, and holds an B.A. in mathematics/economics, and an M.S. in computer science from Brown, and a Ph.D. (under J.J. Horning) from Toronto. He has worked on the design and implementation of a systems language, and is interested in programming languages which aid in producing reliable programs. This also involves him in software engineering, project management, programmer professionalism, and software warrantee and liability.

In addition to the walk to his office, John gets his exercise at baseball and basketball (against small, slow opponents), and sailing.

---

Said a rocket man, winking an eye,  
"Into orbits computers must fly.  
Now it might be more sound,  
If they stayed on the ground,  
But the people want  $\pi$  in the sky."

--Hilbert Schenck, Jr.



## Professional News

assistant chairman and in charge of local arrangements, Dr. Ashok Agrawala and Mrs. Jo Ann Thompson were co-treasurers, Dr. Robert Noonan was in charge of the program, and Dr. Victor Basili handled student coordination. The following faculty members chaired technical sessions: Azriel Rosenfeld, James Vandergraft, Laveen Kanal, Jack Minker, Harlan Mills, H.P. Edmundson, Victor Basili, Charles Rieger, and Marvin Zelkowitz. There were about 1100 conference attendees, and the publishers' exhibit and employment register were busy. Dr. Austing is Chairman of the ACM Fifth Technical Special Interest Group on Computer Science Education, whose Symposium immediately followed the Conference at the Statler Hilton.

The topic of the Trends and Applications Symposium of the IEEE Washington Chapter and the National Bureau of Standards is "Computer Networks". Dr. Marvin Zelkowitz is program chairman of the meeting which will be held at NBS in June. Dr. Victor Basili is also on the program committee. Dr. Zelkowitz is also editing the proceedings of the Technical Symposium to be held at NBS in Gaithersburg in June.

## Comp. Sci. Comp. Rev.

The Friday review sessions for the August comps are proceeding along merrily. Each week at 10 a.m. and 12 noon there are one hour review sessions, ~~concerning various~~ systems and programs ~~carefully~~. In-

# (more)      b u z z      w o r d s

The list of definitions originating at Philco in the November PRINTOUT was only part of the original supplied by M. Zelkowitz. The excuse for printing more of them is that it provides an opportunity to include the following joke, somewhat embellished from the one several people recalled because of the definition of *adder*: a variety of snake.

After the Ark had grounded, Noah set the gangplank in place and began releasing the animals in the well-known two-by-two fashion. As each pair passed, he gave them the advice as laid down in the manual for the occasion, "Go forth, be fruitful, and multiply." There were a few cracks about the impossibility of "going forth" when you are far from the second couple, but all went well until the snakes began to slither down the gangway. When Noah gave a pat on the back to one pair they stopped, holding up the entire procession. "We can't do that," said one of the snakes (it was impossible to tell whether it was the male or female), "we're adders." This stopped Noah for a minute, but casting his eyes heavenward, he told them to wait. He quickly fetched materials from the ship, cut down a cherry tree (the ancestor of the one Washington was later to treat in the same fashion), fashioned a crude picnic table and benches, and put on a tablecloth, knives, forks, spoons, and plates for five people (or other creatures). "There," (to the adders), "now you can multiply." "But how does that help?" (Perhaps it was the same snake, perhaps his or her mate.) "I thought you said you were adders? Well, I've given you a five-place log table."

# the gospel according to Saint<sub>16</sub>

In the beginning there was the Lord High Supervisor and nothing else.

And He was lonely.

And He created for His pleasure and amusement a 64K environment.

And it came to pass that He again did become lonely in His environment. And He did create for His pleasure and amusement:

JOB MANAGEMENT  
DATA MANAGEMENT and  
TASK MANAGEMENT.

And they all did dwell in the environment in harmony, linking and editing, one with the other.

And it came to pass that the Managers did become restless and dissatisfied. And for their amusement and pleasure, the Lord High Supervisor did create Resource Allocators and Sub-pools.

And soon they all didst quarrel amongst themselves, for they felt too confined, each wanting for himself a greater share of the environment.

And they did beseech the Lord High Supervisor, and He in His infinite capacity and all knowing wisdom did restore harmony by enlarging the environment (to an optional 512K), and allotting to each according to his control block.

And the harmony was short lived, for the Managers had nothing to do and did again grow restless.

And thus it was that he did create System/360 (models 20,30,40,50,65,67,75,85,91,95, and 90).

And He said unto them, make use of yourselves. Go to the land of the Salesman, Engineers, and Programmers. Find which of them is worthy and unto them give them my most precious gift-- a clean compilation.

And they went forth to the land of the Salesmen and they cast them out. And they proceeded to the land of the Engineers. And the Engineers said we wish to know not of base registers and overflows.

And thus it was that they came to the land of the Programmers, a land barren, wasted and full of corruption.

And Lo -- the Programmers cast them not out. And they said unto them, give us of your multiple output. But the

Programmers could not attain the immediate glory of clean compilations, and the system did loop and bomb and they did dump it repeatedly.

And so it was that the managers did create for them: mnemonics, macros, data-sets, access methods, externs, and a multitude of formats.

And the Programmers looked, studied, and learned and they saw that it was good and they were happy.

## Colloquium

January 20: Dr. John Hopcroft,  
"Analysis of Algorithms."

February 3: Dr. Ted Linden,  
"Capability-based Addressing to Support Reliable Software."

February 10: Dr. Richard Hamlet,  
"Systematic Program Testing--Theory and Practice."

February 17: Ms. Sandra Mamrac,  
"Comparative Response Times of Time-sharing Systems on the ARPA Network."

February 21: (11:00 A.M., Room 4107)  
Dr. Robert Daley,  
"Algorithmic Entropy and the Entrance of Optimal Descriptions."

February 24: Dr. William E. Riddle,  
"Proving the 'Correctness' of Large Software Systems."

February 28: (11:00 A.M., Room 4107)  
Dr. Lawrence Yelowitz,  
"Proving Correctness of Pointer-manipulating Algorithms."

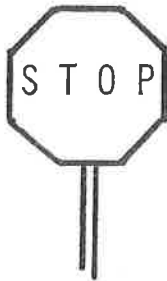
March 3: Ms. Pamela Smith,  
"Applications of Functional Equivalence Relations to the Implementation of Formally-defined Systems."

March 10: Mr. Daniel Hirschberg,  
"Bounds on the Complexity of the Longest Common Subsequence Problem."

Coffee at 3:30 in Room 3316; talks at 4:00 in Room 2324 unless otherwise noted.

# TIRED...

OF RUNS THAT DON'T RUN?  
OF COMPILERS THAT DON'T COMPILE?  
OF SERVICES THAT DON'T SERVE?



DEALING WITH WISHY-WASHY MIDDLEPEOPLE!  
COME DIRECTLY TO A DYNAMIC, DECISIVE  
EXECUTIVE--EXEC 8!

# EXEC 8 NEVER

- ...PASSES THE BUCK, EACH RUN IS TERMINATED WITH A SHORT BUT UPLIFTING MESSAGE!
- ...POSTPONES DECISIONS, INPUT IS REJECTED IMMEDIATELY IF NOT SOONER!
- ...ALLOWS INPUT TO PILE UP, WHAT IS NOT USED IS SIMPLY IGNORED!

COUPON

CLIP AND ATTACH TO RUN

GOOD FOR ONE (ALMOST) BUG-FREE RUN.

\_\_\_1106  
\_\_\_1108  
\_\_\_1110  
\_\_\_B747

Bill my: Mastercharge\_\_\_  
American Express\_\_\_  
Diners' Club\_\_\_  
Signature:

Mr. Operator: This coupon will be redeemed by the Distributer, VacuThatch, Inc., upon presentation of sufficient receipts to cover invoices as set forth in the Retail Agreement. Cash value 1/20¢.