SNOBOL4 Information Bulletin

Madge T. Griswold and Ralph E. Griswold

SNOBOL4 for Personal Computers

Here is some recent information on the status of the various implementations of SNOBOL4 for PCs:

1. Source code for Minnesota SNOBOL4 is now available. See the flyer attached to this Bulletin.

2. Beginning May 1, 1985, Catspaw SNOBOL4+ will be marketed by Prentice-Hall, Inc. See the attached announcement.

3. The price for Macro SPITBOL is now $195 for the first copy and $50 for additional copies. The current distribution is not copy-protected.

SNOBOL4 for the Motorola 68000?

There are a number of computers that use the Motorola 68000 processor, including the Macintosh, the Apollo Workstation, the Sun Workstation, and a variety of other systems that run UNIX. Consequently, there is considerable interest in SNOBOL4 implementations for this processor. While we have nothing official to announce yet, there are rumors of work under way on both SIL SNOBOL4 and Macro SPITBOL for the 68000. Persons with particular interest should write us for current information. Any completed implementations will be announced in future Bulletins.

A SNOBOL4 Conference

Dakota State College is sponsoring an International Conference on English Language and Literature Applications of SNOBOL and SPITBOL (''ICEBOL'') in Madison, South Dakota from May 30 through June 1, 1985. There will be a number of papers on applications of SNOBOL4, as well as programming clinics for beginning and advanced SNOBOL4 programmers.

Information about the conference is available from

Lynn Ryan
11 Beadle Hall
Dakota State College
Madison, South Dakota   57042

Rebus

Much of the usefulness of SNOBOL4 lies in its pattern-matching facilities. String scanning in Icon is designed to provide similar power, and, unlike SNOBOL4, it allows the use of all language operations during high-level string analysis. Icon's string scanning facility, however, focuses on the computation processes involved in string analysis and does not provide the high-level characterization of string structure that SNOBOL4 patterns does.

On the other hand, SNOBOL4 is an old language. Many of its features seem awkward when compared with those of Icon, and its lack of control structures makes even simple programming tasks painful and often results in poorly structured programs.

A new programming language, called Rebus, has been designed as an experiment in imbedding the pattern matching of SNOBOL4 in a syntax that is similar to Icon's. Without going into the details, the general nature of Rebus can be seen by comparing the following SNOBOL4 and Rebus versions of a program for counting the words in a file.

The SNOBOL4 version is:

```
      letter = "abcdefghijklmnopqrstuvwxyz"
+        "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
      wpat = break(letter) span(letter) . word
      count = table()
readtext = input                        :f(sort)
findw    text wpat =                     :f(read)
      count[word] = count[word] + 1        :(findw)
sortresult = sort(count)                 :f(nowords)
      output = "Word count:"
      output =
      i = 0
print   i = i + 1
      output = rpad(result[i = i + 1,1],15) lpad(result[i,2],4) :s(print)f(end)
nowords output = "There are no words"
      end
```

The Rebus version is:

```
function main()
    letter := &lcase || &ucase
    wpat := break(letter) & span(letter) . word
    count := table()
    while text := input do
       while text ?- wpat do
          count[word] +:= 1
    if result := sort(count) then {
       output := "Word count:"
       output := ""
       i := 0
       repeat output := rpad(result[i +:= 1,1],15) || lpad(result[i,2],4)
       }
    else output := "There are no words"
end
```

Note that the SNOBOL4 and Rebus programs are approximately the same size. There is an obvious trade-off between brevity and program structure in the handling of the flow of control. The Rebus program benefits from some syntactic conveniences, such as &lcase, that would otherwise make it longer.

A description of Rebus appeared in the February 1985 issue of *SIGPLAN Notices*. A technical report describing Rebus also is available; use the form at the end of this Bulletin to request it.

Rebus is implemented by a preprocessor that produces SNOBOL4 code. The Rebus preprocessor is available for UNIX-based systems from The University of Arizona. See the request form at the end of this Bulletin. Note that it is necessary to have SNOBOL4 to run Rebus. SNOBOL4 is not supplied with the preprocessor.

A DOS version of Rebus for IBM PC-compatible computers is available, free, from Mark Emmer of Catspaw, Inc. Send a 5¼″ diskette and a post-paid mailer to:

Catspaw, Inc.
Salida, Colorado 81201

SNOBOL4 Users Group?

A number of persons have expressed interest in a SNOBOL4 User's Group, but there has not been sufficient interest to justify the establishment of a group. However, to facilitate the exchange of information, we are establishing an electronic mail contact at The University of Arizona specifically for exchanging information about SNOBOL4. Use the following addresses for electronic mail:

```
snobol4.arizona@csnet-relay(CSNET and ARPANET)
arizona!snobol4-project          (Usenet and uucpnet)
```

Acknowledgement

Special thanks go to Mark Emmer for his recent contributions to the SNOBOL4 archives.

Request for Rebus Material

Please send the material checked below to:

_____

_____

_____

_____

_____

☐   *Rebus—A SNOBOL4/Icon Hybrid*, Technical report TR 84-9 (*free*)

☐   The Rebus Preprocessor for UNIX Systems ($20)

The Rebus preprocessor is supplied on magnetic tape in UNIX *tar* format, written at 1600 bpi. If the Rebus preprocessor is requested, enclose a check for $20 payable to The University of Arizona.

Return this form to:

     SNOBOL4 Project
     Department of Computer Science
     The University of Arizona
     Tucson, AZ   85721

S4B29