Association for Literary and Linguistic Computing

BULLETIN

1978
Volume 6
Number 2

ASSOCIATION FOR LITERARY AND LINGUISTIC COMPUTING
BULLETIN
VOLUME 6 **NUMBER 2** **1978**

EDITORS:

Professor Frank E. Knowles
University of Aston in Birmingham
Gosta Green
Birmingham B4 7ET
United Kingdom

Dr John L. Dawson
Literary and Linguistic Computing Centre
Sidgwick Site
Cambridge CB3 9DA
United Kingdom

EDITORIAL ADVISERS:

Professor Roy Wisbey .... King's College, University of London, UK
Mrs Joan M. Smith .... University of Manchester Regional Computer Centre, Oxford Road, Manchester M13 9PL, UK

BOOK REVIEWS AND ABSTRACTS EDITOR:

Professor Frank E. Knowles

EDITOR OF BIBLIOGRAPHY:

Dr W. J. Jones .... Dept. of German, Westfield College, London NW3 7ST, UK

ASSISTANT EDITOR (GERMAN):

Dr Kurt Gärtner .... University of Marburg, Germany

ADVISORY BOARD:

Professor Roberto Busa, S.J. .... Venice, Italy
Professor Joseph Raben .... Queens College, CUNY, USA

ALLC REPRESENTATIVES:

Australia	Mr John R. C. Martyn	University of Melbourne
Austria	Professor Franz V. Spechtler	University of Salzburg
Belgium	Professor Louis Delatte	LASLA, University of Liège
Canada	Professor John R. Allen	University of Manitoba
Czechoslovakia	Dr Eva Hajicová	Charles University, Prague
Denmark	Professor Karl Hyldgaard-Jensen	University of Copenhagen
Finland	Professor Ilpo T. Piirainen	University of Jyväskylä
France	Professor Bernard Quemada	Sorbonne Nouvelle, Paris
Germany (West)	Professor Winfried Lenders	University of Bonn
Hong Kong	Professor Shiu-Chang Loh	Chinese University of Hong Kong
Hungary	Professor Ferenc Papp	The L. Kossuth University, Debrecen
Iceland	Dr Baldur Jónsson	University of Iceland
India	Dr S. K. Havanur	Tata Institute, Bombay
Israel	Professor Uzzi Ornan	Hebrew University of Jerusalem
Italy	Professor Antonio Zampolli	CNUCE, Pisa
Mexico	Professor Luis F. Lara	El Colegio de México
Netherlands	Dr Félicien de Tollenaere	Beatrixlaan 7, Warmond
Norway	Professor Kolbjørn Heggstad	University of Bergen
Poland	Dr Witold Marciszewski	University of Warsaw
Romania	Professor Solomon Marcus	University of Bucharest
Sweden	Professor Sture Allén	University of Gothenburg
Switzerland	Dr Herbert E. Bruderer	ILD, Münsingen/Bern
United Kingdom	Professor Roy Wisbey	King's College, University of London
USA (East)	Professor Richard W. Bailey	University of Michigan, Ann Arbor
USA (West)	Professor James Joyce	University of California, Berkeley

SUBSCRIPTION (1978):

Individual Membership — United Kingdom £ 5.00; others US $12.00 or £ 6.00
Institutional Membership — United Kingdom £15.00; others US $39.00 or £18.00
Bulletin Subscription — United Kingdom £ 8.00; others US $21.00 or £10.00

Individual Membership includes receipt of one complete Volume of the *Bulletin*, and offers reduced rates at most ALLC-sponsored functions.

Institutional Membership includes receipt of two copies of one complete Volume of the *Bulletin*, and entitles three persons to reduced rates at most ALLC-sponsored functions.

Bulletin Subscription includes no Membership privileges.

Back Volumes of the *Bulletin* are available at reduced rates to Members and *Bulletin* Subscribers. Apply to the Secretary: Dr J. L. Dawson, Literary and Linguistic Computing Centre, Sidgwick Site, Cambridge CB3 9DA, England.

Published with the co-operation of the Conseil International de la Philosophie et des Sciences Humaines, and with the aid of a financial grant from UNESCO.

ISSN 0305—9855

ASSOCIATION FOR LITERARY AND LINGUISTIC COMPUTING

CHAIRMAN: PROFESSOR ROY WISBEY

Department of German, King's College, University of London
(Founder and former Director, University of Cambridge
Literary and Linguistic Computing Centre)

The principal purposes of this international association are to further literary and linguistic research by computer and to provide a means of communication for all those concerned with such research.

COMMITTEE

OFFICERS

Chairman - Professor R. Wisbey
Honorary Secretary - Dr J.L. Dawson
Honorary Treasurer - Professor D.A. Wells

MEMBERS

Professor A. Zampolli - Italy
Dr F. de Tollenaere - Netherlands
Professor W. Lenders - Germany
Professor W. Martin - Belgium
Professor F.V. Spechtler - Austria
Mr S. Bootle - United States
Mrs S.M. Hockey - United Kingdom
Professor S. Allén - Sweden
Professor J.R. Allen - Canada

SPECIALIST GROUPS - REGIONAL BRANCHES - SYMPOSIA - SUMMER SCHOOLS

REPRESENTATIVES in Australia, Austria, Belgium, Canada, Czechoslovakia, Denmark, Finland, France, Germany (West), Hong Kong, Hungary, Iceland, India, Israel, Italy, Mexico, Netherlands, Norway, Poland, Romania, Sweden, Switzerland, United Kingdom, USA (East), USA (West).

Enquiries should be addressed to the Honorary Secretary:

Dr John L. Dawson
Literary and Linguistic Computing Centre
Sidgwick Site
Cambridge CB3 9DA
United Kingdom.

CONTENTS

MACHINE TRANSLATION OF CHINESE MATHEMATICAL ARTICLES .... S.-C. Loh, L. Kong, and H.-S. Hung 111

THE CHINESE UNIVERSITY LANGUAGE TRANSLATOR (CULT) - A REPORT .... P.H. Nancarrow 121

AUTOMATIC SCANSION OF SANSKRIT POETRY FOR AUTHORSHIP CRITERIA .... D. Wujastyk 122

THE MIZAR-QC/6000 LOGIC INFORMATION LANGUAGE .... A. Trybulec 136

THE DISCOVERY OF SYNTAGMATIC AND PARADIGMATIC CLASSES .... J.G. Wolff 141

REPORT ON A COURSE ON THE USE OF COMPUTERS IN TEXTUAL ANALYSIS AND BIBLIOGRAPHY HELD IN THE COMPUTER UNIT OF THE UNIVERSITY COLLEGE OF WALES, ABERYSTWYTH, 10-14 APRIL 1978 .... P. Sims-Williams 159

COMPARABLE COMPUTER LANGUAGES FOR LINGUISTIC AND LITERARY DATA PROCESSING: PERFORMANCE .... M. Boot 161

A PARTIAL-PARSING ALGORITHM FOR NATURAL LANGUAGE TEXT USING A SIMPLE GRAMMAR FOR ARGUMENTS .... P.J. Sallis 170

WANTED - COMPUTER READABLE DICTIONARIES .... T.D. Crawford 177

COLLOQUIUM ON THE USE OF COMPUTERS IN TEXTUAL CRITICISM: A REPORT .... S.M. Hockey 180

Secretary's Notes 182

Diary 183

Advertisement (Data Bases Conference) 186

Advertisement (ICCH/4) 187

Advertisement (ALLC International Meeting, 1978) 188

Notes on Contributors 189

Editors' Notes 190

News and Notes 192

Book Reviews 196

Documents Received 199

Advertisement 201

Letter to the Editor 202

Bibliography 204

THE MIZAR-QC/6000 LOGIC INFORMATION LANGUAGE

A. Trybulec *(University of Warsaw, Poland)*

* * * *

Abstract

A logic information language is a language for the recording of mathematical texts, including the logical relationships between elements in it such as inferences between theorems and so on. A practical logic information language has to be on the one hand sufficiently simple for automatic processing and, on the other, sufficiently rich for human use.

This article gives a brief structural description of the MIZAR language which is intended to be a practical logic information language. A subset called MIZAR-QC is presented which is restricted to first order calculus with a fixed set of predicate symbols which, however, can be varied by simply changing the appropriate part of the syntactic analyser. We are, in effect, therefore dealing really with a family of languages with different but fixed sets of predicate symbols.

* *

The MIZAR-QC/6000 language is an implementation on a CDC 6000 which provides the following facilities: a) translating a text coded according to MIZAR rules into a semantic 'form'; b) activating a system for checking proofs in logical arguments. A logic information language of the type represented by MIZAR can be used for the following purposes: a) for checking proofs submitted by students in mathematics teaching; b) in computer-aided proof checking, particularly checking program facilities; c) as an input language to a mathematics fact retrieval system; d) for the purposes of semi-automatic machine translation. It must be noted that the main difficulties in machine translation are caused by the analysis of the natural language text. The synthesis of the target text, on the other hand, does not represent a serious problem. It appears, therefore, that a proper solution would be to allow a human to translate from a natural language into a logic information language and then call upon the computer to translate from the logic information language into a number of natural languages. In the design of a fully-fledged machine translation system the logic information language referred to in this context is used as an intermediate language or an interlingua. Much work in the field of logic information languages has been devoted to this topic.

Work on languages of this type has been underway for about ten years: one thinks for instance of de Bruijn's AUTOMATH and Glushkov's KIEV project. Our aim was the elaboration of a system, the use of which would not depart too radically from accepted practices in mathematics.

The MIZAR project is a continuation of work conducted for the Plock Scientific Society in 1975 which comprised the implementation on an ODRA 1204 computer of the MIZAR-PC language which was designed for the purposes of computational logic, but which was limited at that stage to sentential calculus. This language was used experimentally in a teaching situation and can be considered to have brought positive results.

It must be noted that the analyser of a logic information language represents only a part of the system for most of the above-mentioned applications. For teaching and for computer-aided proof checking a supplementary module, known as a proof checker, must be incorporated into the system and it must be parameterizable. For teaching purposes the proof checker does not need extensive automatic proof checking facilities, but it must be 'transparent' and must have a run time for inference checking that is limited linearly or at least multinomially. For the purposes of computer-aided proof checking, on the other hand, a 'rich' system is preferable and run time limitations are not so essential, particularly when the system is being used interactively because the user can always break in and halt the algorithm or go into step mode. In a fact retrieval system the smallest parts of the system are the search and data bank access procedures; data bank access would also be very useful in computer-aided proof checking. The proof checker can be discarded altogether in the case of semi-automatic machine translation. Notwithstanding the above remarks we are of the opinion that the implementation of the MIZAR project represents a certain step forward in the design of more advanced systems. Incidentally, the MIZAR project was implemented using the PASCAL 6000 programming language and the program uses a slightly modified version of the PASCAL compiler's scanner.

The MIZAR-QC/6000 language makes use of the following symbols in its alphabet:

a) upper case letters
b) numerals
c) the following special symbols -

, : ; - ∧ ∨ > ≡ = ( ) [ ]

Any symbols can be used within comments. The sign for the beginning of a comment is (* and a comment ends with *). Spaces and new lines are used to delimit identifiers and may not occur therefore in the middle of an identifier; elsewhere they are ignored. The definition of MIZAR's vocabulary uses a number of terms familiar to those involved with ALGOL 68 - terms such as 'indicant', 'defining occurrence', 'applied occurrence', 'scope', for instance. The basic grammar of MIZAR-QC/6000 is quoted below in BNF notation. Certain identifiers are used as indicants. They are as follows:

BEGIN, PROOF, LET, CONSIDER, SET, ASSUME, THAT, SUCH, X, FOR, HOLDS, AND, THUS, BY, BE, END.

The remaining identifiers are subdivided into the following classes:

a) labels (defining occurrence), designators (applied occurrence)
b) variables (defining occurrence), arguments (applied occurrence)
c) predicates (applied occurrence)
d) specifications (applied occurrence).

In order to test the translator we used the following predicates:

CONTRADICTION, P, Q, R, P1, Q1, R1, P2, Q2, R2, P3, Q3, R3;

the first four of these take zero arguments and the 'arity' of the remainder is indicated by the numeral following the letter. The set of predicates is given by means of a table and can be varied; this, however, does not apply to the contradiction predicate. This particular variant of the MIZAR language allows the occurrence of only one specification - ANYTHING. Indicants, predicates, and the specification are treated as reserved identifiers and may not be used in other constructions. Labels and their corresponding designators are local in proofs; variables and arguments are local in proofs in existential sentences and in universal sentences. The fact that an identifier can belong to labels and designators on the one hand as well as to variables and arguments on the other does not cause problems because a context sensitive recognition system is used. The defining occurrence of an identifier must precede any applied occurrence of the same identifier. Finally, only the first six symbols of identifiers are significant.

In addition to the BNF grammar of MIZAR-QC/6000 we append a short example of MIZAR coding.

* * *

(Translated from Polish by F. Knowles)

The grammar of MIZAR-QC/6000

```
TEXT ::= BEGIN TEXT-BODY END

TEXT-BODY ::= STATEMENT | TEXT-BODY ; STATEMENT

STATEMENT ::= PROPOSITION JUSTIFICATION

PROPOSITION ::= LABEL : SENTENCE

SENTENCE ::= ATOMIC-SENTENCE | (COMPOUND-SENTENCE)

ATOMIC-SENTENCE ::= PREDICATE | PREDICATE [VARIABLE-LIST]

VARIABLE-LIST ::= VARIABLE | VARIABLE-LIST , VARIABLE

COMPOUND-SENTENCE ::= - SENTENCE |
                      SENTENCE BINARY-FUNCTOR SENTENCE |
                      FOR VARIABLE-LIST HOLDS SENTENCE |
                      EX VARIABLE-LIST ST SENTENCE

JUSTIFICATION ::= SIMPLE-JUSTIFICATION | DEMONSTRATION

SIMPLE-JUSTIFICATION ::= EMPTY | BY LABEL-LIST

LABEL-LIST ::= LABEL | LABEL-LIST , LABEL

DEMONSTRATION ::= PROOF DEMONSTRATION-BODY END

DEMONSTRATION-BODY ::= DEMONSTRATION-ELEMENT ; DEMONSTRATION-BODY |
                       CONCLUSION END

DEMONSTRATION-ELEMENT ::= ASSUMPTION | STATEMENT | CONCLUSION | DECLARATION |
                          CONSIDER-CLAUSE | SET-CLAUSE

CONCLUSION ::= THUS STATEMENT

ASSUMPTION ::= ASSUME CONDITIONS

CONDITIONS ::= THAT PROPOSITION | CONDITIONS AND PROPOSITION

DECLARATION ::= LET VARIABLE-LIST BE ANYTHING

CONSIDER-CLAUSE ::= CONSIDER VARIABLE-LIST |
                    CONSIDER VARIABLE-LIST CONDITIONS JUSTIFICATION

SET-CLAUSE ::= SET VARIABLE-LIST = VARIABLE

LABELs, VARIABLEs, and PREDICATEs are identifiers.

BINARY-FUNCTOR ::= ∧ | ∨ | > | ≡
```

An example of MIZAR text

P1 is a unary predicate symbol, P2 binary, and P3 ternary.
EX means 'there exist(s)' and ST means 'such that'.

```
BEGIN
   T1:  ((FOR X, Y, Z HOLDS P3 [X,Y,Z] >
                  (FOR X HOLDS P3 [X,X,X]))
   PROOF
      ASSUME THAT Z:  (FOR X, Y, Z HOLDS P3 [X,Y,Z]) ;
      LET X BE ANYTHING ;
      THUS C:  P3 [X,X,X] BY Z END ;
   T2:  ((EX X ST P3 [X,X,X]) > (EX X1, X2, X3 ST P3 [X1,X2,X3]))
   PROOF
      ASSUME THAT Z:  (EX X ST P3 [X,X,X]) ;
      CONSIDER X SUCH THAT ZZ:  P3 [X,X,X] BY Z ;
      SET X1, X2, X3 = X ;
      THUS C:  P3 [X1,X2,X3] BY ZZ END ;
   T3:  ((FOR X HOLDS P1 [X]) > (EX X ST P1 [X]))
   PROOF
      ASSUME THAT Z:  (FOR X HOLDS P1 [X]) ;
      CONSIDER X ;
      THUS C:  P1 [X] BY Z END ;
   T4:  ((EX X ST (FOR Y HOLDS P2 [X,Y])) > (FOR Y HOLDS (EX X ST P2 [X,Y])))
   PROOF
      ASSUME THAT Z:  (EX X ST (FOR Y HOLDS P2 [X,Y])) ;
      CONSIDER X SUCH THAT ZZ:  (FOR Y HOLDS P2 [X,Y]) BY Z ;
      LET Y BE ANYTHING ;
      SET XX = X ;
      THUS C:  P2 [XX,Y] BY ZZ END ;
   T5:  (FOR X HOLDS (P1 [X] ≡ P1 [X]))
   PROOF
      LET X BE ANYTHING ;
      THUS T:  (P1 [X] ≡ P1 [X]) END ;
   T6:  (FOR X HOLDS ((FOR Y HOLDS P1 [Y]) > P1 [X]))
   PROOF
      LET X BE ANYTHING ;
      ASSUME THAT Z:  (FOR Y HOLDS P1 [Y]) ;
      THUS W:  P1 [X] BY Z END ;
```

```
  T7:  (FOR X HOLDS (P1 [X] > (EX Y ST P1 [Y])))
  PROOF
    LET X BE ANYTHING ;
    ASSUME THAT Z:  P1 [X] ;
    SET Y = X ;
    THUS C:  P1 [Y] BY Z END ;
  T8:  (((FOR X HOLDS P1 [X]) ∧ (FOR X HOLDS (P1 [X] > Q1 [X])))
                                   > (FOR X HOLDS Q1 [X]))
  PROOF
    ASSUME THAT Z1:  (FOR X HOLDS P1 [X])
            AND Z2:  (FOR X HOLDS (P1 [X] > Q1 [X])) ;
    LET X BE ANYTHING ;
    L1:  P1 [X] BY Z1 ;
    L2:  (P1 [X] > Q1 [X]) BY Z2 ;
    THUS C:  Q1 [X] BY L1, L2 END ;
END
```