

store list, deleted from the triple store, the main store list is manipulated as appropriate, and lastly the altered main store information is returned to the triple store as a set of order information triples.

APPENDIX 2: THE SUBROUTINE INTERFACE TO THE TRIPLE STORE

```
void insert_triple (char s[], char r[],
                  char o[])
```

Takes three strings as parameters; treats them as the subject, relation and object of a triple, and adds them to the triple store.

```
int query_triple (char s[], char r[],
                 char o[])
```

Takes three strings as parameters; treats them as the subject, relation and object of a triple template. Templates have three fields; they can contain any string value or a special wildcard value. `query_triple` compares the template with all stored triples and remembers the location of any triples that matched the template. It returns a 'Boolean' which is TRUE if any triples matched and FALSE otherwise.

```
int query_triple_num (char s[], char r[],
                    char o[])
```

Similar to the above, but it returns the number of triples matched.

```
int retrieve_triple (char s[], char r[],
                   char o[])
```

After a `query_triple` has been issued, `retrieve_triple` is used to sequentially return the matched triples. It returns them in the three string parameters. It also returns the 'Boolean' result TRUE if there are still triples left to be returned, FALSE if the list of hit triples is exhausted. `Query_triple` and `retrieve_triple` are normally used in tandem as follows:

```
query_triple(a,b,c)
while (retrieve_triple(d,e,f) == TRUE)
{
    /* process triple*/
};
```

```
int open_db (char dbname[])
```

This takes as parameter the name of a triple store and attempts to open it. It returns TRUE if successful, FALSE otherwise.

```
void create_db (char dbname[])
```

This takes as parameter the name of a triple store and attempts to create it.

```
void close_db ()
```

This closes the current triple store.

```
void delete_triples (char s[], char r[],
                   char o[])
```

Like `query_triple`, this takes a triple template as parameter, and locates all triples that match the template. Then, however, it goes on to delete all the matching triples.

Book Review

K. DEVLIN
Logic and Information
Cambridge University Press. £17.95
ISBN 0 521 41030 4.

This book starts well, with a clear statement that there is a need for a science of information. Moreover, the author in his acknowledgements makes the valid point that efforts to satisfy this need must initially be driven by scientific curiosity without necessarily an explicit utilitarian objective. These are propositions that need to be published widely to stimulate academic workers in the information engineering field to recognise that the conceptual foundations of systems engineering are conspicuous by their absence, essentially because there is no science of information but there ought to be and there needs to be such a science.

However, the author then assumes that mathematics is the master science so that he devotes the rest of his book to a determined attempt to formulate concepts such as 'Infons', 'Situations' and 'Constraints', and symbols to represent them, as a first step in the foundation of a mathematics of information.

Perhaps in a future book the author will use his symbols to derive propositions whose validity can be checked by experiment, but if there are such propositions in the present book they are well hidden. Curiously there is a clear statement that the concepts proposed are recursively defined, but no mention of observed hyperbolic statistical distribution of symbols in meaningful text that could be interpreted as offering experimental support for the recursive definitions. Indeed, this reviewer could find no reference at all to repeatable observations, which surely should appear in the foundations of a proposed new branch of science.

To force information into a mathematical mould the author had endeavoured to extend the scope of mathematics and logic to include intuition and judgement. Certainly these subjective techniques play an important role in the use of information by all people including mathematicians, but it is generally understood that they are consciously excluded from published mathematical work, indeed it is the objectivity of mathematical techniques derived from the conscious exclusion of intuitive judgements that accounts for the utility of

mathematics in so many branches of established science. It is therefore far from obvious that the author's declared interpretation of a 'science of information' as a 'mathematics of information' is valid with the generally accepted interpretation of the words.

Many readers for whom mathematics is a useful tool but not a way of life will find this book difficult to follow, so that it will probably have less impact than it deserves. This is unfortunate, since the book breaks new ground in a field that is likely to become of increasing importance. The book can be recommended to anyone who has recognised the need for a better understanding of the nature of information and who is prepared to put effort into understanding Devlin's presentation. Perhaps there is gold in this book but it is not clear to this reviewer. Devlin promises more volumes – certainly he should be encouraged to do more work on the problem that he has recognised so clearly, but his objective should be to formulate some novel and checkable conclusions and to distil the essence of his work into a smaller volume.

G. SCARROTT
Welwyn