GEURTS82
Geurts L.; An Overview of the B Programming Language; ACM SIGPLAN Notices 17, 12 (December 1982) pp. 49-58.

This paper presents an introduction to the programming language B.

In the first section, the author explains the goals and briefly introduces B: "B is a powerful, easy to learn and easy to use interactive programming language intended for personal computing ...". B is a language embedded in a dedicated system containing a "syntax-directed" screen editor, file maintenance functions, and facilities for background processes.

In the next section, the author describes the data types and their associated operations: B contains a few carefully designed system and data types and a few associated operations. There are two basic types: "numbers" and "texts", and three structures for creating composite types: "compounds", "lists", and "tables".

B numbers are similar to conventional numbers except there is no limit on integer length and integers are considered special cases of integral fractions (rationals). In B rational numbers are referred to as "exact" numbers and operators are supplied for determining the numerator and denominator of these numbers. Also, to this end, rationals are always reduced to lowest terms. In addition to exact numbers, B also has "approximate" (floating point) numbers.

In B characters strings are referred to as "texts", and the language supplies operations for their concatenation, repetition, generating prefixes and suffixes, length, character counting, character selection (subscripting), and using strings as a generator of a sequence of characters.

Compounds are similar to record constructs except that fields are unnamed. Lists are like "bags" or multi-sets (sets which permit multiple occurrences of elements). All entries in a list must be of the same type and they are kept in sorted order. Lists can have the following operations performed on them: initialization (assignment), addition of an entry, removal of one instance of an entry, number of entries, number of occurrences of an entry, generation of a sequence (the list elements), selection (subscripting), smallest and largest elements (since a list is kept in sorted order). The third structure, tables, are like arrays except that the table indices may be of any type and need not be consecutive (like SNOBOL tables). The operations which may be performed on tables are: assignment, addition of an entry, modification of an entry, length, number of occurrences in the associates, list of indices, deletion of an entry, smallest associate, largest associate, and generation.

In the next section, the author describes the commands YIELD (used to define new functions) and HOW'TO (used to define a new command). Following this, the control commands (IF ... ELSE, SELECT (like a series of if .. elseif ..), and the loop constructs FOR and WHILE with the exit statment QUIT), conditional expressions, B's versions of the mathematical quantifiers (SOME, EACH, and NO) and the TEST statement (used to build conditional functions) are discussed.

Following these, the paper discusses a construct known as a "refinement" used in large hierarchical programs (a refinement is something like a macro) and how I/O is performed using the READ and WRITE commands.

At the end of the paper the author presents two samples programs which demonstrate the use of B data types and control constructs.

This paper serves as a good introduction to a language with a number of interesting features. Both that data types and the control structures in this language are noteworthy. The paper itself is easy to follow and well worth reading.