Additional Papers

In the December 1991 issue of *The Computer Journal* a new concept of *Additional Papers* was introduced. Such papers will have been typeset and given numbered pages in the same way as other papers, but only the title, authorship and abstract appear in the *Journal*. To obtain copies of the full paper, readers apply to the editor who will mail copies directly to the reader at cost. Libraries will be able to obtain bound volumes at the year end. Reprints will be supplied to authors. For further information, including price lists and order forms, readers and libraries should apply directly to the Editor-in-Chief at the address given on the inside front cover of the *Journal*.

A further list of additional papers is given below.

On the Number of Solutions, S(k,n), to a Class of Crossword Puzzles
G. H. HARRIS AND J. J. H. FORSTER, *Division of CAD, Griffith University, Nathan, 4111, Australia*

Upper and lower limit formulae for the total number of solutions to a standard type of crossword puzzle are proposed. The solutions use words from artificially created lexicons. Formulae are derived for alphabets and square puzzle grids of all sizes and are compared with the numbers of solutions generated by the implementation of a puzzle solution algorithm. (pp. A177–A180)

An Efficient Algorithm for Crosswood Puzzle Solutions
G. H. HARRIS, L. J. SPRING AND J. J. H. FORSTER, *Division of CAD, Griffith University, Nathan, 4111, Australia*

This work investigates the solution of crossword puzzles. A simple pruned tree-search algorithm involving a dynamic dictionary lookup is detailed and the results of its implementation compared against those already reported in the literature. (pp. A181–A183)

An Extension of the Certainty Factor Model in First Order Predicate Calculus
T. PANAYIOTOPOULOS AND G. PAPAKONSTANTINOU, *Electrical Engineering Department, Computer Science Division, National Technical University of Athens, Zographou Campus, 15773 Athens, Greece*

This paper concerns a new approach for the incorporation of the certainty factor model in a full theorem-proving environment. For this reason, an extended-resolution principle is presented, and two refutation methods are described – a general one, and a more systematic one. Both methodologies may be applied to propositional calculus and full first-order predicate calculus.

In the general refutation method it is shown how the certainty factor model may be incorporated in a logic program, with general clauses and an arbitrary selection for unification among these clauses. In the more systematic refutation method the methodology is applied on a logic program which uses depth-first and left-to-right selection of the goals (ordered-input refutation). The ME-reduction operation is incorporated, in order to obtain full theorem-proving capabilities. The methodologies are demonstrated by illustrative examples. (pp. A185–A192)

AGP: A Parallel Processor for Knowledge and Software Engineering
G. PAPAKONSTANTINOU, T. PANAYIOTOPOULOS AND G. DIMITRIOU, *Electrical Engineering Department, Computer Science Division, National Technical University of Athens, Zographou Campus, 15773 Athens, Greece*

In this paper we describe AGP, a new parallel algorithm, which is general enough to be used both for attribute grammar applications and for logic programming applications. AGP can run on a multiprocessing environment and has been implemented using the OCCAM language and the TRANSPUTER architecture. (pp. A193–A199)

Algorithms on Permutation Trees
E. CH. FOUNDAS* and A. A. VAZACOPOULOS†
* *Department of Statistics and Actuarial Science, University of Piraeus, 40 Karaoli and Dimitriou Street, Piraeus, Greece 18532*
† *Department of Decision Sciences and Information Systems, Florida International University, University Park, Miami, Florida 33199, USA*

In this paper we use permutation trees to solve combinatorial problems. We solve the problème des ménages *and problems concerning multipermutations.* (pp. A201–A207)

Experiments with Word-by-Word Compression of English Text using Lexicons
P. D. SMITH, *Department of Computer Science, California State University, Northridge, CA 91330, USA*

Lexicons are added to the MTF technique of Bentley et al. for compressing English text word by word. In the compressed file, certain words and separators are represented by associated bit strings. Source words can be transformed into lexicon entries in a variety of ways. Information about the transformation becomes part of the compressed file. Experiments were carried out to investigate the effects on the size of the compressed file of varying the lexicon and transformations. Generally, adding a lexicon improves the degree of compression. Even a small word lexicon makes the compressed file smaller by about 0.5 bits per character compared with MTF. Using transformation codes and root lexicons of various kinds does not appear to lead to more compression than is achieved when holding many variants of a word in the lexicon. (pp. A209–A213)