EXPERIMENTS IN CHESS CAPTURE SEARCH

Prakash Bettadapur
M.Sc. Thesis, Spring 1986
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada

(Xerox copies at US $ 5.00 are available from the author on request.)

We quote the abstract:

Capture search in chess is an example of a tree searching application. The capture analysis, an expensive part of any chess program, checks for quiescence at every leaf node of a depth limited game tree. A simple capture search is implemented with limited hardware support, to study which hardware and software aspects can improve search efficiency. Three experiments are discussed, specifically, the performance comparisons, order experiments, and the depth experiments study the effect of applying algorithms and heuristics to capture search.

Performance measurements suggest that the expensive part of capture search, move generation, can be made faster using bitmaps, but this improved performance may be lost if the moves cannot be provided to the search mechanism efficiently. Move ordering is important in search efficiency. In the order experiments, the capture search is done with different capture orders and their efficiencies are compared. The experiments indicate that ordering the moves using the heuristic of *first* capturing the *biggest pieces* by the *smallest pieces* leads to the most efficient search. In the depth experiments, the capture trees are searched to varying depths. The resulting error and the reduction in tree size for depth limited capture trees are compared to the correct values obtained with no depth limit. The experiments show that selecting a search depth leads to a *trade off* between error and savings. There is no optimum search depth with maximum savings and minimum error, so it is better to do capture search to an unlimited depth.

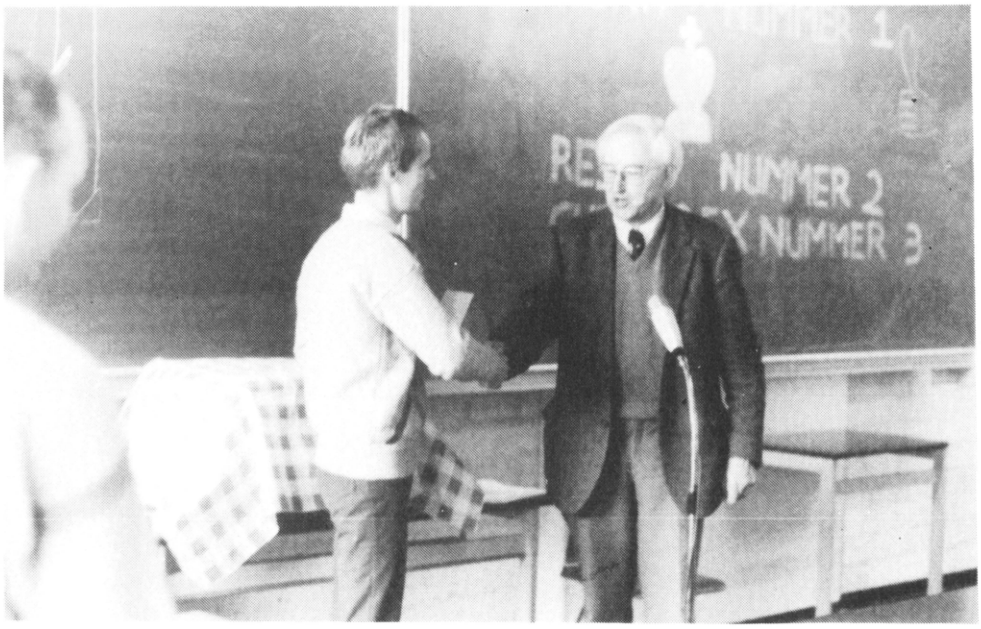

Photo by K.E. Börjars

NONE LIKE NONA
Van Oosterwijk Bruyn presenting the prize when Nona retained her superiority among Dutch programs.