The Ninth North American Computer Chess Championship

Sponsored by the ACM
Dec. 3-5, 1978
Washington, D.C.

Ken Thompson's BELLE brushed off four other chess programs, including the current world champion program CHESS 4.7, to decisively win the Nineth North American Computer Chess Championship held in Washington, D.C. on Dec. 3-5, 1978 as part of the ACM's Annual Conference. BELLE, running on a DEC 11/70 which had a special hardware move generation package attached, was searching trees at rates of about 6000-7000 nodes per second. Its playing strength seemed to be in the 2100-2200 USCF range.

Finishing in second place was CHESS 4.7, the work of David Slate and Larry Atkin of Northwestern University and the current World Champion program. CHESS 4.7 won three games but its loss to BELLE in round 2 cost it the title. That game was a wild affair with plenty of room for interesting analysis. Mike Ciamarra provides the annotations.

Three programs, CHAOS, BLITZ 6.5 and SARGON II finished with 2 1/2 points. DUCHESS, last year's co-champion (with CHESS 4.7) had some tough luck with the draw and was only able to capture 2 points, tying with OSTRICH IV for sixth place. Also participating were MIKE, BLACK KNIGHT, BS6676, AWIT, and BRUTE FORCE.

Carl Diesen, father of World Junior Chess Champion Mark Diesen, along with Monty Newborn, and Ben Mettman of Northwestern University worked out arrangements for the tournament. David Levy served as tournament director. Robert Byrne, Chess Editor for the New York Times and Edward Lasker, International Master and veteran of the U.S. chess world (Lasker celebrated his 93rd birthday on December 3rd!) both attended the tournament. Stories by Byrne appeared in the New York Times on December 11th and 13th.

A Study of Simulated Control of a Robot: Adaptibility, Generation and Execution of Plans

M. Cayrol, B. Fade, H. Farreny
Laboratoire "Langages et Systemes Informatiques"
Universite Paul Sabatier - Toulouse, France

This work constitutes the development of a project previously presented in Sigart Newsletter (August 1976).

A programmed system controlling a robot has to be done. The robot is a utilitary one moving in an unknown (at least partially) environment; it can adapt itself to a large family of problems.

On a first step we made a program constituting the "elementary robot". It has only few but fundamentals operators:

TO LOOK: perception from distance
TO GROPE: perception by contact
TO MARK OUT: conception of moving schemes
TO WALK: checking of execution of the moving schemes.

The environment is represented by a discretized plan (for instance an n x n array); the objects are defined in connection with the fundamental operators: objects which are or not obstacles to moving or to percepting. The elementary robot works out problems of moving itself to a known position or looking for marked objects the position of which is unknown.

On a second step we contrived a system in which the elementary robot constitutes a core able to receive different specializations as a user will. The robot can mainly change its "culture" (its knowledge and capacities) and its environment. The user gives the robot in form of goal-oriented production-rules the operators that seem to him fitting with the selected environment. Some operators will drive terminal acts adapted to the treated application: calling software subroutines, execution of physical processes (manipulators, captors,...). The user teaches the system its initial knowledge. These assertions use objects, relations properties which are not provided for in the elementary robot. So it is added an advancing set of operators objects relations and the robot becomes an "adapted robot". A simple QA4 like language permits to describe the commands which will be stored in the three types of memory: fact instances memory, problem instances memory, production rules memory. These memories are accessible using pattern-matching techniques in an associative mode. The produced plans may involve symbolic identifiers generated by the system itself. These identifiers are necessary to carry on the planning but will be instantiated only when executing the plan. Two updating sequences are associated with each step of the plan they should be used when executing the step. One of them is called in case of success the other is used to manage the failures. The problem-solver is a depth-first one using and-or graphs. A specialized routine controls the execution of the plans in the user's environment (simulated in a conversational mode). The failures of the plan suppose the detection of objects or situations previously unknown. Hence a progress in the state of the knowledge of the robot. This will permit to create new plans, better adapted to the real context.