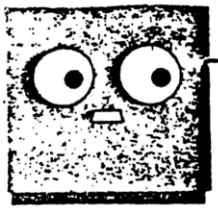

INTERNATIONAL COMPUTER CHESS ASSOCIATION
(I.C.C.A.)

ICCA Newsletter

Vol. 1 No. 2
September 1978

Editor: B. Mittman
Editorial Asst.: J. Cesal
Northwestern University

EDITORIAL

Since we announced the formation of the International Computer Chess Association in May, we have had over 50 computer chess enthusiasts join the Association. (The current membership list is enclosed in the members' mailing.) This is the last copy of the ICCA Newsletter which will be mailed to those who have not yet joined. To continue to receive the Newsletter, fill in the attached application form and send it with $5.00 to ICCA Headquarters as soon as possible.

This issue of the Newsletter coincides with the end of the "Levy decade". David Levy played CHESS 4.7 at the Canadian National Exhibition (Northwestern University: David Slate and Larry Atkin) and MAC HACK in Boston (MIT: Richard Greenblatt) in two challenge matches to finish his famous bet successfully. Levy's article on the challenges begins on page 3.

Computer chess was active in other quarters as well. In early August, six international programs competed in an invitational tournament at the Jerusalem Conference on Information Technology. DUCHESS of Duke University won that tournament. A report begins on page 14.

In the first issue, we indicated that material from our members is essential to sustain a useful and dynamic Newsletter. In this issue, we are publishing some excellent contributions from Dennis Hamilton of Penfield, N.Y. and Don Beal of Queen Mary College, London. We greatly appreciate their interest in the Newsletter. Please send ideas, announcements, comments, and news to:

ICCA Newsletter
Vogelback Computing Center
Northwestern University
Evanston, Illinois 60201
USA

B. Mittman
Editor

USCF REGISTRATION OF PROGRAMS AVAILABLE

The United States Chess Federation, "in order to allow for legitimate scientific testing of chess-playing computer programs", has established procedures for registration of programs and participation in USCF-rated tournaments.

In order to maintain the statistical utility of the player rating system employed in the United States, it is necessary that modifications of existing programs (including operation on a different computer system resulting in a change of program performance) be

(continued on page 2)

USCF REGISTRATION OF PROGRAMS AVAILABLE

(continued from page 1)

re-registered. It is unclear how this proviso is to be accommodated for programs that exhibit learning behavior.

Registration is usually limited to the programmer of the computer algorithm. An annual fee of $15, the same as USCF membership, is charged upon approval of a registration application.

Copies of the single-page procedure are obtained from:

USCF
186 Route 9W
New Windsor, New York 12550
USA

NEW RULEBOOK AVAILABLE - "COMPUTATIONAL MACHINERY" ACCOMMODATED

Morrison, Martin E. (ed.), Official Rules of Chess, 2nd edition, David McKay, New York, 1978. Paperback: $2.95. ISBN 0-679-14043-3.

The second edition of the Official Rules of Chess, effective on January 1, 1978, is now available. Featuring a complete translation of the international (FIDE) rules in English, along with amplifying interpretations of those rules, the rulebook also incorporates a United States Chess Federation (USCF) section. The USCF section applies to official tournament play in the United States and features, among its other provisions, "Suggested Rules for Play Involving Computational Machinery". These rules are consistent with those adopted for conduct of the North American Computer Chess Championship and should be accommodated, along with the overall rules of chess and chess tournaments, by programmers who desire to enter their algorithms in USCF-rated events.

Another noteworthy aspect is the uniform adoption, world-wide, of the abbreviated "algebraic" notation. On January 1, 1981, this will become the sole authorized notation for recording of the moves during tournament play. The notation is usable immediately, as well, so designers of chess-playing programs would do well to accommodate this scheme at once.

Dennis E. Hamilton
Penfield, New York

ARTICLE 12.3 QUESTIONED: WHEN IS A RIGHT NOT A RIGHT?

Article 12.3 of the Official (FIDE) Rules of Chess presents one of the most interesting challenges to the formulation of chess-playing computer programs: correct recognition of a repetition of the position.

Under most circumstances, a repetition of the position is determined by the same player having the move together with an identical arrangement of pieces on the board. But Article 12.3 is more stringent than that; the possible moves of all the pieces must also be the same from each occurrence of the arrangement. The only factors influencing possible continuations and not determinable by inspection of the board alone are, of course, the right to capture a pawn en passant and the rights to castling that may remain to either player. (Of course, it is evident in many positions that no such rights can possibly exist.)

Is it sufficient, then, to keep track of the "forward" preservation of castling rights along with incidences of pawn advances that satisfy the conditions for capture en passant?

Perhaps not.

(continued on page 3)

ARTICLE 12.3 QUESTIONED: WHEN IS A RIGHT NOT A RIGHT?

(continued from page 2)

What happens in the event that an earlier position is identical in all respects but for the presence of certain rights and there is no way for such rights to be exercised in any legal continuation from the original position?

If you want to ponder this question, try setting up a position in which an en passant capture cannot be exercised because the only pawn in a position to make capture is pinned in front of the player's king. Now consider player's king being in check by other than the just-advanced pawn.

En passant isn't too difficult to verify, of course, because the right is a fleeting one in any case. Unexercisable castling rights are of another breed altogether. What, I wonder, is the least upper bound on a continuation tree before it is determined that a right to castle is unexercisable in every possible continuation?

This problem is not confined to computer chess by any means. Tournament referees may have a devilish time correctly arbitrating a draw claimed under Article 12.3 if the claim hinges on the assertion that a right to castle remaining in an earlier position was, in fact, no right at all!

The United States Chess Federation has been queried on this subtlety, and the FIDE Rules Commission will be asked to consider the question.

In the meantime, it may be prudent to make sure our programs aren't suckered into repetitions because they are a bit too hasty at concluding that a difference in rights is sufficient to make the positions differ.

Dennis E. Hamilton
Penfield, New York

HOW I WON

David Levy

After a decade of speculation I have shown, fairly convincingly, that McCarthy, Michie, Papert, and Kozdrowicki were overly optimistic when they bet me a total of $2,500 that by August 1978 a computer program would win a chess match against me. My two final challenge matches took place during August (and the first few days of September) with a highly satisfactory result. On August 23, at M.I.T., I won the first game of a two-game match against Richard Greenblatt's program MacHack, running on two PDP machines and a piece of special purpose hardware called CHEOPS, which can analyze 150,000 positions per second. The M.I.T. group had prepared their program to play a sharp line in my favourite Dragon Variation of the Sicilian Defense, hoping that the tactics would prove too much for me. In fact, the game quickly reached an ending in which I had the edge, and despite prolonged resistance, MacHack was unable to resist the pressure. This result made the second game of the match moot, since I would only lose the bet if I lost a match. Greenblatt, therefore, suggested that we play a 30-30 game (30 moves in 30 minutes by each player) which was a rather dull struggle in which I eventually won what was quite possibly a drawn ending.

The main interest centred around my match with World Champion CHESS 4.7. After various difficulties, the match was set in Toronto at the Canadian National Exhibition, a kind of gigantic state fair. I was seated in an almost soundproofed glass booth, dressed in a tuxedo, and playing on the special chess board designed by David Cahlander of CDC. This board enables the program to detect my moves without human

(continued on page 7)

HANDICAPPING FOR COMPUTER SPEED

Ever since the first ACM-sponsored computer chess tournament in 1970 in New York, there has been a continuing debate about whether programs should be handicapped as a function of the speed of the computer upon which they are run and the percentage of the capacity of the machine they expect to utilize. Prof. Tony Marsland of the University of Alberta has spoken and written extensively about this problem. The following is part of a letter which Don Beal of Queen Mary College wrote to Barend Swets on the subject. Comments on Mr. Beal's letter are solicited.

16 January 1978

"...Despite the fact that my own program has access to a fairly fast computer, I would like to see computing time allocated to machines in inverse proportion to their basic hardware speed instead of equally as so far. The reasons are as follows:-

1) I believe that the main interest in computer chess lies in the contest between different algorithms for chess play. I ask: would exactly the same program competing on two machines differing only in their speed be particularly interesting or worthwhile?

2) It is beneficial to computer chess if the most efficient algorithms are successful and publicized, rather than algorithms that may or may not be efficient but which happen to have been implemented on very fast machines. It would also give any new algorithms and ideas that exist only in the minds of chess programmers without access to fast machines a better chance to surface.

3) In a sporting sense, it is fairer, as each person has equal opportunity to choose and invent algorithms for chess play, but perhaps not equal access to fast computers.

I have heard some objections to allocating time according to machine speed, but it seems to me that they are all readily refuted:-

1)"It is impossible to compare accurately the speeds of two different computers because of the different instruction sets and other factors."

For non-numerical work I have found that an extremely crude assessment (the average time to add 2 numbers of at least 16 bits, at least one of which is in main store, putting the result elsewhere in main store) is surprisingly effective - within about 50% (as judged by running large lumps of program) in estimating the relative speeds of a wide range of computers. This, or even a slightly wider margin, is sufficiently small in comparison to the difference in performance between chess algorithms (the object of the contest) to be a tolerable degree of possible unfairness, whereas today's 100:1 range of basic hardware speeds is not.

2)"There are differences in programming languages - if you compensate for processor speed you should also compensate for compiler efficiency."

On the whole, I do not support this argument. Program efficiency is very much under the programmer's control. Most computers have some efficient language and, whatever language is used, it is usually possible to incorporate time-critical parts as modules in assembler language, say.

(continued on page 5)

HANDICAPPING FOR COMPUTER SPEED

(continued from page 4)

3) "How could the tournament take place in real time if a very slow machine has to be allowed hours per move to equal the processing of a fast one?"

The tournament can operate in exactly the same time scale as at present. In each game, the same total time is set on the clocks - it is merely divided unequally between the two opponents.

4) "Setting the time allowed on the clock for 'dedicated' machines is straightforward, but what about programs on 'multi-access' systems?"

This is slightly more difficult. It does not seem reasonable to ignore time 'lost' to non-chess users while taking care to equalize computing effort between dedicated machines. I favour the following approach: the programmer to specify before each game a fraction (1/n) of the machine's time which is the minimum he is confident of receiving. His time is then calculated as if he had a machine n times slower, and his time control becomes primarily by processor time - he forfeits the game if the processor time (to be printed by the program and verified by the operating system for the whole run) exceeds 1/n of the clock limit. He also forfeits if the clock limit itself is exceeded - the tournament must run to schedule - although the tournament director could apply this rule leniently, perhaps. Also, 'n' should probably be limited to some reasonably generous maximum, 10 say.

5) "What about parallel machines, and special chess hardware?"

So far, none have been entered in tournaments, so it is in any case a problem for the future, but I think an adequate assessment of the effective speed of such machines could be made. This of course denies such machines their competitive advantage, but I think that the unrestricted use of specialized hardware really belongs in a separate league.

6) "If the clock times are unequal, the player with the lesser time suffers more from the time lost in moving pieces, typing moves etc."

I think the best answer is to adopt firmly the basic principle that processor time is the primary concern. Then the chess clocks become primarily a means of keeping the tournament to schedule. It has already been suggested that programs on 'multi-access' systems print the processor time for each move; let programs on 'dedicated' machines do so as well (easily verifiable since for dedicated machines the time printed should be the elapsed time between the operator typing the move and getting a reply). The time set on each player's clock can then include an adequate allowance for human operating delays without upsetting the balance of allocated processor time.

7) "Won't the quality of the chess decline disastrously if the faster machines are liable to be held down to much less computing time?"

No, it could even improve. A fast machine is only penalized when opposing a slow one. Under the present system of equal time, the chances are that the slow one will play badly because of the lack of computing muscle, even if it is as good a program. Is a game where the loser plays badly a convincing win for the winner or a good quality game of chess?

(continued on page 6)

HANDICAPPING FOR COMPUTER SPEED

(continued from page 5)

Overall, in every game, at least one machine will receive as much or more time as at present, so at least half the moves in the tournament will be as good or better; and, for every move choice that suffers from less computing, another one benefits from more.

8) "Such sponsorship as there is from computer manufacturers would disappear because the winner would no longer be an advertisement for the power of that make of computer."

I think that the general publicity of participating is mainly what motivates sponsors. They cannot all win.

9) "What exactly would the new rules be then?"

1) Each competitor's computer is given a 'rated speed' by the tournament organizers.
2) Programs must print the processor time used for each move.
3) The allocation of time for each game is as follows:
(Let T be the time that would be set on each clock under equal-time rules.)
Processor time, faster machine = (2T-2WT)xRS/(RS+RF)
Processor time, slower machine = (2T-2WT)xRF/(RS+RF)
where: RF = rated speed of faster machine
RS = rated speed of slower machine
WT = allocation for wasted time in human operations
(20 sec/move, say)
The time to be set on the clock is WT + processor time.
(If it should be desired to allocate differing amounts of 'wasted time', the formulae can be easily adjusted.)
4) A game is forfeit if the processor time allowance is exceeded. (And also if the clock limit is exceeded.)

10) 'Small print' rules.

1) Users of 'multi-access' systems must obtain a print from the operating system of the CPU time used immediately before the game, and again immediately afterwards.
2) In the event of a discrepancy between alternative timings of a program, the larger will be taken. Programmers should ensure that their program leaves a sufficient margin of unused CPU time to cover possible inaccuracies in system clocks etc.
3) Programmers must supply sufficient data on the hardware they are going to use to enable at least the benchmark addition described to be timed, plus such additional information as may be required by the tournament organizers to rate a particular machine. Entrants must agree to tolerate the inevitable degree of inaccuracy in ratings that will occur. (It will, of course, be vastly less than the uncompensated differences.)

11) "Isn't all this much too complicated?"

No. It just requires a decision to do it this way.

(continued on page 16)

HOW I WON

(continued from page 3)

intervention, and when the program is ready to move, the "from" and "to" squares light up in addition to any intermediate squares, so that a human can move the program's pieces. The next generation of the program, CHESS 5.0, will have its own robot to move the pieces and punch the clock button.

The match started with a most exciting game. Shortly after the opening, I overlooked the fact that a knight sacrifice, which I had deemed useless, was absolutely crushing. The move that I had missed was the program's ...Qg5 which made the sacrifice not only totally sound but also quite devastating. I persuaded the program to trade queens in order to avoid the possibility of a mating attack, and eventually we reached an ending in which I was three pawns down. Here the program began to flounder and after a tenacious defense, I was able to draw.

In games two and three my strategy for the match was seen with good effect. I had resolved before arriving in Toronto that I would employ my usual "Do nothing but do it well" strategy, which is ideal for a tactical genius such as CHESS 4.7. In both cases I won very convincingly.

For game four, by which time I needed only half a point from three games to clinch the bet, I decided to try an experiment and take the program on at its own game - tactics. I played the ultra risky Latvian Gambit but got quite a good opening even though I had sacrificed a pawn. I launched a mating attack on the king side but at each critical position I failed to play the decisive stroke, and finally I succumbed after failing to see a bishop check which left me quite helpless. The program's tactical ability was remarkable - each time that I thought the game was mine, a new resource would appear on the other side of the board. It was almost like playing Petrosian.

Having shown that I am a gentleman, I thought that I ought to dispense with the experiments and get down to the serious business of scoring another point. Game five was a tense struggle in which the program's pieces got tied up on the queen side and I was able to increase the pressure to the point where CHESS 4.7's position was about to crack. Then something unfortunate happened - the machine went down. While the engineers spent 25 minutes putting things right, I came to the conclusion that the program was almost certainly losing. When the machine came up again, it played all the best defensive moves that I could see, but to no avail. I finished it off with a couple of pins and a simple combination, which put me a bishop up. Once again there were technical difficulties at the computer site in Arden Hills, but rather than struggle on hopelessly, David Slate decided to resign the game and the match.

The whole event was great fun. CHESS 4.7 was supported by David Slate (Larry Atkin was on vacation in England), and David Cahlander and John Douglas of CDC. The games were played on the CDC Cyber 176 at Arden Hills, Minnesota.

The games of both matches are on the following pages. Readers interested in my comments on my games with CHESS 4.7 should consult the November 1978 issue of Chess Life & Review.

(continued on page 16)

TORONTO, ONTARIO, AUGUST 26, 1978

GAME 1: LEVY X CHESS 4.7

1	G2-G3.		D7-D5.	
2	F1-G2.		E7-E5.	
3	D2-D3.		G8-F6.	
4	G1-F3.		B8-C6.	
5	O-O.		C8-D7.	
6	B2-B3.		F8-C5.	
7	C1-B2.		D8-E7.	
8	A2-A3.		E5-E4.	
9	F3-E1.		O-O.	
10	D3-D4.		C5-D6.	
11	E2-E3.		F6-G4.	
12	H2-H3.		G4*E3.	
13	F2*E3.		E7-G5.	
14	G3-G4.		G5*E3.	CHECK.
15	F1-F2.		D6-G3.	
16	D1-E2.		E3*F2.	CHECK.
17	E2*F2.		G3*F2.	CHECK.
18	G1*F2.		F7-F5.	
19	G4*F5.		C6-E7.	
20	C2-C4.		F8*F5.	CHECK.
21	F2-G1.		C7-C6.	
22	B1-C3.		F5-H5.	
23	G1-H2.		A8-F8.	
24	C3-D1.		E7-G6.	
25	A1-C1.		D7*H3.	
26	G2*H3.		F8-F1.	
27	E1-G2.		F1-F3.	
28	C4*D5.		H5*H3.	CHECK.
29	H2-G1.		C6*D5.	
30	C1-C8.	CHECK.	G6-F8.	
31	B2-C3.		F3-D3.	
32	D1-E3.		H3*E3.	
33	G2*E3.		D3*E3.	
34	C3-B4.		E3-F3.	
35	C8-D8.		H7-H6.	
36	D8*D5.		F3*B3.	
37	D5-D8.		B3-F3.	
38	D8-A8.		G7-G5.	
39	D4-D5.		H6-H5.	
40	D5-D6.		G8-G7.	
41	A8*A7.		F3-F7.	
42	A7-A5.		G7-F6.	
43	B4-C3.	CHECK.	F6-G6.	
44	A5-E5.		F7-F3.	
45	C3-B4.		F3-F4.	
46	E5-E7.		F4-F7.	
47	E7*E4.		F7-D7.	
48	E4-E7.		H5-H4.	
49	G1-G2.		G5-G4.	
50	G2-H2.		B7-B6.	
51	H2-G2.		D7-D8.	
52	A3-A4.		F8-D7.	
53	A4-A5.		D7-F6.	
54	A5*B6.		F6-D5.	
55	B6-B7.		D5*E7.	
56	D6*E7.		D8-H8.	
57	B4-D6.		G6-F6.	
58	B7-B8=Q.		H8*B8.	
59	D6*B8.		F6*E7.	
60	B8-F4.		E7-F6.	
61	F4-D2.		F6-G6.	
62	D2-E1.		G6-G5.	
63	E1-F2.		G5-H5.	
64	F2-E1.			

(.5 - .5)

AUGUST 27, 1978

GAME 2: CHESS 4.7 X LEVY

1	B1-C3.	C7-C5.	
2	E2-E4.	B8-C6.	
3	F2-F4.	A7-A6.	
4	G1-F3.	G7-G6.	
5	D2-D4.	C5*D4.	
6	F3*D4.	F8-G7.	
7	C1-E3.	D7-D6.	
8	D4*C6.	B7*C6.	
9	F1-E2.	A8-B8.	
10	D1-C1.	D8-A5.	
11	E3-D2.	A5-B6.	
12	C3-A4.	B6-A7.	
13	A4-C3.	G7-D4.	
14	C3-D1.	G8-F6.	
15	C2-C3.	D4-B6.	
16	C1-C2.	F6-G4.	
17	C2-A4.	O-O.	
18	E2*G4.	C8*G4.	
19	A4*C6.	G4*D1.	
20	E1*D1.	B6-E3.	
21	B2-B3.	E3*D2.	
22	D1*D2.	B8-C8.	
23	C6-A4.	A7-F2.	CHECK.
24	D2-D3.	F2*G2.	
25	A4-D4.	G2-F3.	CHECK.
26	D3-C2.	F3-E2.	CHECK.
27	C2-C1.	E7-E5.	
28	F4*E5.	D6*E5.	
29	D4*E5.	F8-E8.	
30	E5-G3.	E8*E4.	
31	G3-H3.	C8-D8.	
32	H3-F1.	E2-D2.	CHECK.
33	C1-B1.	E4-E2.	
34	F1*E2.	D2*E2.	
35	H1-E1.	E2*E1.	CHECK.
36	B1-B2.	D8-D2.	CHECK.
37	B2-A3.	E1*A1.	

(0 - 1) 17 MOVES LATER.

SEPTEMBER 2, 1978

GAME 3: LEVY X CHESS 4.7

1	C2-C4.		G8-F6.	
2	A2-A3.		B8-C6.	
3	B1-C3.		D7-D5.	
4	C4*D5.		F6*D5.	
5	D2-D3.		D5*C3.	
6	B2*C3.		E7-E5.	
7	G2-G3.		F8-E7.	
8	F1-G2.		D8-D6.	
9	G1-F3.		C8-E6.	
10	O-O.		O-O.	
11	D1-A4.		D6-C5.	
12	C1-D2.		B7-B5.	
13	A4-C2.		F7-F6.	
14	F1-B1.		A8-D8.	
15	C2-B2.		D8-B8.	
16	D2-E3.		C5-D6.	
17	F3-D2.		E6-D5.	
18	G2*D5.	CHECK.	D6*D5.	
19	B2-B3.		D5*B3.	
20	D2*B3.		F6-F5.	
21	E3-C5.		E7-D6.	
22	B1-B2.		G8-H8.	
23	A1-B1.		A7-A6.	
24	C5*D6.		C7*D6.	
25	B3-D2.		F5-F4.	
26	G1-G2.		F4*G3.	
27	H2*G3.		B8-D8.	
28	A3-A4.		C6-A7.	
29	D2-E4.		B5*A4.	
30	B2-B6.		D6-D5.	
31	E4-C5.		A7-B5.	
32	C5*A4.		D8-A8.	
33	C3-C4.		D5*C4.	
34	D3*C4.		B5-D4.	
35	E2-E3.		D4-F3.	
36	C4-C5.		F3-G5.	
37	C5-C6.		G5-E4.	
38	C6-C7.		F8*F2.	CHECK.
39	G2-G1.		F2-F8.	
40	B6-B8.		H7-H5.	
41	B8*A8.		F8*A8.	
42	B1-B8.	CHECK.		

(1 - 0)

SEPTEMBER 3, 1978

GAME 4: CHESS 4.7 X LEVY

1	E2-E4.		E7-E5.	
2	G1-F3.		F7-F5.	
3	E4*F5.		E5-E4.	
4	F3-E5.		G8-F6.	
5	E5-G4.		D7-D5.	
6	G4*F6.	CHECK.	D8*F6.	
7	D1-H5.	CHECK.	F6-F7.	
8	H5*F7.	CHECK.	E8*F7.	
9	B1-C3.		C7-C6.	
10	D2-D3.		E4*D3.	
11	F1*D3.		B8-D7.	
12	C1-F4.		D7-C5.	
13	G2-G4.		C5*D3.	CHECK.
14	C2*D3.		F8-C5.	
15	O-O.		H7-H5.	
16	C3-A4.		C5-D4.	
17	F4-E3.		D4-E5.	
18	D3-D4.		E5-D6.	
19	H2-H3.		B7-B6.	
20	F1-E1.		C8-D7.	
21	A4-C3.		H5*G4.	
22	H3*G4.		H8-H4.	
23	F2-F3.		A8-H8.	
24	G1-F1.		D6-G3.	
25	E1-E2.		D7-C8.	
26	F1-G2.		G3-D6.	
27	E3-G1.		H4-H3.	
28	A1-E1.		H3-G3.	CHECK.
29	G2-F2.		H8-H3.	
30	E2-E3.		C8-A6.	
31	C3-E2.		A6*E2.	
32	E1*E2.		C6-C5.	
33	F3-F4.		G3*E3.	
34	E2*E3.		H3-H4.	
35	F2-G3.		H4-H1.	
36	G1-F2.		H1-D1.	
37	E3-A3.		C5*D4.	
38	A3*A7.	CHECK.	F7-F8.	
39	A7-D7.		D1-D3.	CHECK.
40	G3-G2.		D6-C5.	
41	D7*D5.		D3-D2.	
42	B2-B4.		C5*B4.	
43	D5-D8.	CHECK.	F8-F7.	
44	D8-D7.	CHECK.	F7-F8.	
45	D7*D4.		D2-B2.	
46	G2-F3.		B4-C5.	
47	D4-D8.	CHECK.	F8-E7.	
48	F2-H4.	CHECK.	E7-F7.	
49	G4-G5.		G7-G6.	
50	D8-D7.	CHECK.	F7-F8.	
51	F5*G6.		B2*A2.	
52	F4-F5.		A2-A3.	CHECK.
53	F3-G4.		A3-A4.	CHECK.
54	G4-H5.		A4-D4.	
55	D7-C7.		C5-E7.	

(1 - 0)

SEPTEMBER 4, 1978

GAME 5: LEVY X CHESS 4.7

1	C2-C4.		G8-F6.
2	A2-A3.		C7-C6.
3	D2-D3.		D7-D5.
4	D1-C2.		D5*C4.
5	C2*C4.		E7-E5.
6	G1-F3.		F8-D6.
7	G2-G3.		C8-E6.
8	C4-C2.		B8-D7.
9	F1-G2.		O-O.
10	O-O.		D8-B6.
11	B1-D2.		B6-C5.
12	C2-B1.		H7-H6.
13	B2-B4.		C5-B5.
14	B1-C2.		D7-B6.
15	C1-B2.		A7-A5.
16	A3-A4.		B5-A6.
17	B4*A5.		A6*A5.
18	B2-C3.		A5-C5.
19	F1-C1.		B6-D7.
20	A4-A5.		C5-A7.
21	C2-B2.		F6-G4.
22	D2-E4.		D6-C7.
23	H2-H3.		F7-F5.
24	H3*G4.		F5*E4.
25	D3*E4.		E6*G4.
26	C3-E1.		D7-C5.
27	C1-B1.		A8-E8.
28	E1-D2.		F8-F7.
29	D2-E3.		C7-D6.
30	B2-C2.		G4*F3.
31	G2*F3.		E8-A8.
32	B1-C1.		B7-B6.
33	G1-G2.		A7-B7.
34	A5*B6.		A8*A1.
35	C1*A1.		C5-E6.
36	A1-A7.		B7-C8.
37	C2-A2.		F7-F6.
38	A7-A8.		D6-B8.
39	F3-G4.		G8-F7.
40	A2-A7.	CHECK.	B8*A7.
41	A8*C8.		A7*B6.
42	G4*E6.	CHECK.	F6*E6.
43	E3*B6.		

(1 - 0)

M.I.T., CAMBRIDGE, MASSACHUSETTS
AUGUST 23, 1978

	Greenblatt WHITE	Levy BLACK
1	e4	c5
2	Nf3	d6
3	d4	cd
4	Nd4	Nf6
5	Nc3	g6
6	f4	Bg7!
7	e5	Nh5
8	Bb5+	Bd7
9	e6	fe
10	Ne6	Bc3+
11	bc	Qc8!
12	Qd4	Nf6
13	Qc4	Nc6
14	Nd4	Nd4
15	cd	Qc4
16	Bc4	Bf5!
17	Bb5+	Kf7
18	Bc4+	d5
19	Bd3!	Rhc8
20	0-0	Rc7
21	Rb1	Rac8
22	Be3?!	Ne4
23	Rf3	Nd6
24	Rb2	b6
25	a4	Bd3
26	cd	Rc3
27	Rh3	h5
28	Bd2	Rc2
29	Rc2	Rc2
30	Be1	Nf5
31	a5	ba
32	Ba5	Nd4
33	Re3	Ra2
34	Bc7	a5
35	Re1	a4
36	Be5	Nc6
37	Bh8	a3
38	Rd1	Rc2
39	Ba1	a2
40	h3	Na5
41	d4	Nb3
42	f5	Rc1
43	Rc1	
	0-1	

COMPUTER CHESS GOES TO ISRAEL

An invitational computer chess tournament, sponsored in part by the Israel Chess Association, was held during the Jerusalem Conference on Information Technology on August 6-9. Organized by Prof. B. Mittman of the United States and Mr. Yoram Alster of Israel, the three-round Swiss tournament featured one Canadian, one Dutch, one Swiss, and three U. S. programs, all running on computers in Jerusalem.

The tournament was won by DUCHESS of Duke University; CHAOS of the University of Michigan and CHESS 4.6 of Northwestern University tied for second place; OSTRICH of McGill University and TELL of ETH in Zurich tied for fourth place; and BS'66'76 of the Netherlands came in sixth. Summarized below are some details about the participants:

Program	Author(s)	Country	Computer Used	Site in Jerusalem
DUCHESS	T. Truscott B. Wright E. Jensen	USA	IBM 370/158	Bank of Israel
CHESS 4.6	D. Slate L. Atkin	USA	CDC CYBER 74	Hebrew University
CHAOS	M. Alexander T. McBride F. Swartz W. Toikka V. Berman J. Winograd	USA	IBM 370/168*	Office Mechanization Center
OSTRICH	M. Newborn G. Arnold	Canada	Data General Supernova	At the tournament site
TELL	J. Joss	Switzerland	HP 2100	At the tournament site
BS'66'76	B. Swets	Netherlands	IBM 370/168*	Office Mechanization Center

One game, which was awaited with considerable interest, was between CHESS 4.6 and DUCHESS. These programs had tied for first place in the most recent North American Championship in Seattle last October. But now, instead of running on a powerful Control Data CYBER 176, CHESS 4.6 was on a CYBER 74, a machine more comparable in speed to the IBM 370/158. DUCHESS won that game, which is shown on page 15, and went on to win the tournament.

In addition to computer vs. computer competition, the large and enthusiastic audiences were also treated to speed chess, pitting rated Israeli players against the programs. CHESS 4.6 was particularly successful in speed chess, winning against several players rated about 2000 ILO points, and drawing against two players rated over 2400 (Chess Masters Kagan and Peretz).

*Shared access

COMPUTER CHESS GOES TO ISRAEL

Round 2 - August 7, 1978

	DUCHESS White	CHESS 4.6 Black
1.	E2-E4	B8-C6
2.	D2-D4	D7-D5
3.	E4-E5	F7-F6
4.	G1-F3	C8-F5
5.	F1-B5	D8-D7
6.	0-0	A7-A6
7.	B5-D3	F5-G4
8.	E5*F6	G8*F6
9.	C2-C3	E7-E5
10.	F3*E5	G4*D1
11.	E5*D7	F6*D7
12.	F1*D1	F8-E7
13.	D3-F5	A8-D8
14.	F5-E6	D7-F8
15.	E6-G4	F8-G6
16.	D1-E1	0-0
17.	B2-B3	B7-B5
18.	A2-A4	B5-B4
19.	E1-E6	C6-A5
20.	G4-D1	D8-D6
21.	E6*D6	E7*D6
22.	C3*B4	A5-C6
23.	D1-F3	F8*F3
24.	G2*F3	C6*D4
25.	B1-D2	G6-H4
26.	C1-B2	D4*F3 check
27.	D2*F3	H4*F3 check
28.	G1-G2	F3-D2
29.	B2-C3	D2-E4
30.	C3-E1	E4-F6
31.	A1-C1	F6-H5
32.	C1-C6	H5-F4 check
33.	G2-F1	F4-D3
34.	E1-D2	D6*H2
35.	C6*A6	H7-H5
36.	F1-E2	D3-F4 check
37.	E2-E3	F4-H3
38.	E3-D4	H3*F2
39.	A6-A8 check	G8-F7
40.	D4*D5	F2-G4
41.	D2-G5	F7-G6
42.	G5-E7	H2-G3
43.	A8-F8	H5-H4
44.	D5-C6	H4-H3
45.	F8-F1	G4-E3
46.	F1-H1	H3-H2
47.	E7-D8	E3-C2
48.	D8*C7	C2-D4 check
49.	C6-B7	G3*C7
50.	B7*C7	D4*B3
51.	H1*H2	G6-F5
52.	C7-D6	G7-G5
53.	D6-D5	G5-G4
54.	D5-C4	G4-G3
55.	H2-A2	F5-E4
56.	C4*B3	E4-D5
57.	A2-G2	D5-E5
58.	G2*G3	E5-D5
59.	A4-A5	D5-E5
		Resign

HANDICAPPING FOR COMPUTER SPEED

(continued from page 6)

Perhaps you could circulate the arguments in this letter within the I.C.C.A. with a request for an agree/disagree/comments response from everyone? If there are enough people who agree with it, at least in principle, I will take it a step further by drawing up a table of relative speeds for all the computers used in the last tournament (as far as I am able)."

Don Beal
Queen Mary College
University of London

HOW I WON

(continued from page 7)

I have decided to renew my bet, but for a shorter period than 10 years. Details will be announced shortly. I am pleased to be able to announce here that a prize of $5,000 is to be awarded to the author(s) of the first program that does win a match against me. $1,000 of the money will be my own; the remaining $4,000 has been offered by OMNI, the new science magazine which began publication in September. Full details will be published in an early issue of OMNI.

APPLICATION FORM

This is to apply for membership in the International Computer Chess Association.

Enclosed is a ☐ check (U.S. only)

or ☐ international money order

for the amount of $5.00 made out to ICCA for the first year's dues.

Name:_____

Address:_____

City:_____

State or Province:_____ Zip Code:_____

Country:_____

Please mail to:

ICCA
Vogelback Computing Center
Northwestern University
Evanston, Illinois 60201 USA