

M E M O R A N D U M

23 April 1967

TO: Professors Allen Newell and Herbert Simon
FROM: Steve Coles
SUBJECT: Chess at Carnegie Tech

1. At the SJCC on thursday, April 20, I had a sobering experience. I lost a chess game to a PDP-10! (See Game enclosed) Of course, I can rationalize by saying that I was moving very quickly (10 to 15 seconds in the beginning) and was under pressure from many bystanders (10 to 20 people). Nevertheless, after a more leisurely examination of the game, I can honestly say that except for a foolish knight sacrifice, I made no conspicuous blunders and in retrospect can detect no obvious flaws or idiosyncracies in the machine's play which would have permitted an easy win. My fundamental difficulty appears to have been ~~over-confidence~~ confidence, under estimating the program's ability, and looking for an early mate through aggressive play which in turn led to over-emption without adequate king protection. I always wondered when I would lose my first game to a machine. I never expected it would be in 1967.

2. Here are some salient facts--- The program, written by Greenblatt at MAC, has recently been rated for tournament play, and is the same program which defeated Hubert Dryphus, although it has undergone extensive improvement since that time. The game I played took about an hour with about half that time used by the program which was running on a dedicated PDP-10 (1.5 usec. cycle). The program moves in about thirty seconds with four ply look ahead (although this is a variable parameter)-- and this with fairly high consistency even in complex situations. When the move is obvious however, it replies almost instantly. It consumes about 8K of core on the PDP-10 (36 bit words), and is written in a PDP macro-assembly language. It analyses each position separately without recourse to standard games. It also seems to have fairly standard heuristics with a high predilection for pawn pushes-- all else being equal. Its principal virtue does not seem to be any novel representation, but rather appears to be the fact that

it has played a large number of games with the author feeding back experience on every occasion the program was defeated.

3. What are the implications of this program for Carnegie Tech? First, we might ask ourselves could Greenblatt have written such a program in the Carnegie Tech environment-- ostensibly a hot bed of computer chess activity? I think not. Why not? The most conspicuous variable, as I see it, is the G-21. I do not mean to say that as a computer the G-21 is less sophisticated than the PDP-6-- to the contrary. Rather, the fact of the matter is that the G-21 is a much scarcer resource. To see this we need only examine the record, and ask how many full chess games against a human have been completed on the G-21, and at what rate? Note that Packard once took 6 months to complete a single game. Rubinfeld's program written in Algol 20 can make no more than a few moves a day due to teletype restrictions, while Cunningham could make at most one move per day due to 64K restrictions. I have no reason to suppose that Greenblatt is smarter or more highly motivated than Cunningham, Packard, Baylor, Rubinfeld, or Ramey. And will this problem disappear at Carnegie Tech with the advent of the 67? Although I hope I am wrong, I think not. It is ironic to think that a still more sophisticated computer such as the 67 could be operationally inferior to the PDP-6 as a chess machine, due to inadequate access and system software deficiencies.

Second, we need to reexamine the entire issue of a higher level language for chess. It has always been held more or less axiomatic at Carnegie Tech that the highest level language obtainable would be desirable for such a complex and subtle game. Although this postulate may still be correct, it is ironic to consider that the most successful program running today was deliberately written not in the highest level language available, but in one of the lowest.

Thirdly, we need to rethink the entire issue of memory requirements.. For similar reasons, we have always assumed that large core was essential for such a complex game. Of course I fail to see how large core could hurt, but Greenblatt has provided a persuasive counter example for the "necessary" argument. I, for one, have been dramatically persuaded.

LLC