STANFORD ARTIFICIAL INTELLIGENCE PROJECT June 25, 1964
Memo No. 15

AXIOMATIZATION AND IMPLEMENTATION

by Mark Finkelstein & Fred Safier

Abstract: An example of a typical Advice-Taker axiomatization of a situation is given, and the situation is programmed in LISP as an indication of how the Advice-Taker could be expected to react. The situation chosen is the play of a hand of bridge.

The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183).

AXIOMATIZATION AND IMPLEMENTATION

Mark Finkelstein & Fred Safier

The purpose of this paper is to provide an example of an axiomatization of a situation in the sense of the Advice Taker, and to show how this might then be effected on the computer. The reader should bear in mind that no attempt has been made to construct an elegant situation, nor to produce an efficient program.

The program listed below represents what seems desirable to the minds of the authors, in the way of a program for the Advice Taker to construct, or to operate, given the axiom structure above. Note that the axiomatization differs logically from the program, as the program will "run", i.e. play a hand, while the axiomatization will not.

I. Axiomatization

Following is the axiomatization of the playing of a hand of bridge. We shall assume here that the reader is familiar with the game (if not, see, for example, 1].)

The situation we shall be dealing with is a 10-tuple, of which the first four elements represent the four hands, with the remaining cards in each. The remaining entries are the number of tricks to be played, the next player to play (a direction: North, East, etc.), a history of the trick in progress (a list of the cards already played to the trick), who has won what, the contract (a pair), and the results so far (a complete history of every trick played).

We shall make liberal use below of functions whose definitions will not appear, but whose meaning should be clear, e.g. history[s] is a function which selects the seventh element, that is, the history, from the situation tuple.

1]. Belladonna, Finkelstein, and Safier, "Why you Lose at Bridge" (translated from the Italian by Leo F. Boron), Houghton-Mifflin Co., 1944.

1. Direction of trick

follows [(NORTH WEST), (EAST NORTH), (SOUTH EAST), (WEST SOUTH)]

2. Play of a trick:

(s) Full [history[s]] = >

obtained [Westcards[s], Ncards[s], EC[s], SC[s], number tricks - 1,

newwinner[s], NIL, increment [newwinner[s], whowonwhat[s]],contract[s], add [history[s], result[s]]].

The meaning of this axiom is that when the history is "full", that is, when four hands have played to a trick, then the result is obtained by changing the trick count, determining the winner of the trick and adding to his total, and transferring the history of the trick to the history of the hand (which we are calling "result").

3. Result of a play of a card

mover[p,s] & plays [p,x,s] =>

obtained [remove [x,p,s], --, add [x, history[s]],

follower [p], --]

The meaning here is that when one player plays a card (x), that card is deleted from his hand, and added to the history of the trick. The situation then obtained is one in which the player who follows player p is the current player.

4. Rule on following suit

mover [p,s] & ¬ [null[history[s]]] & ¬[null [insuit[p,

suit[first[history[s]]]]s] =>

(x) [plays[p,x,s] => suit[x] = suit[first[history[s]]]]

i.e. one must follow suit if possible.

5. (∀s) [Number tricks = 0 => obtained [result[s]]]

i.e. when all tricks are played out, the hand is over and the record of the tricks is the result.

II. Implementation

The program below is roughly what might be expected of the Advice Taker when it is presented with the above axioms; there are two main functions, F2, and F3, which correspond to axioms 2 and 3 above. Thus, F2 expresses the play of a trick and F3 the play of a card. The function PLAY is descriptive of the play of the whole hand, consisting of successive applications of F2 and F3, and is expected to give back the whole record of the play with the input of the initial situation.

1. (PLAY (LAMBDA (S)(COND ((ZEROP (TRICKS S))(RESULT S))(T F3 (F2 S))))).

Play of a trick: This corresponds to axiom 5 above.

2. (F2 (LAMBDA (S) (COND ((EQUAL (LENGTH (HISTORY S)) 4) (LIST (WC S)

(NC S) (EC S) (SC S) (SUB1 (TRICKS S)) (NEWWINNER S) NIL

(INCREMENT (NEWWINNER S) (COUNT S)) (CONTRACT S) (CONS (HISTORY S)

(RESULT S)))) (T S))))

Thus, if all four players have played, the trick is over; give back a new situation with one fewer trick to be played, the winner of this trick, an empty history, with the history of this trick added on to the record of the hand, and the number of tricks taken by the present winner increased by one.

(WC CAR)

(NC CADR)

(EC CADDR)

(SC CADDDR)

(TRICKS (LAMBDA (S) (DDR 4 S)))

(NEWWINNER (LAMBDA (S) (FIND (MAXFRED (HISTORY S) (TRUMP S)) (CURPLAYER S))))

(CURPLAYER (LAMBDA (S) (DDR S)))

(HISTORY (LAMBDA (S) (DDR 6 S)))

(COUNT (LAMBDA (S) (DDR 7 S)))

(CONTRACT (LAMBDA (S) (DDR 8 S)))

(TRUMP (LAMBDA (S) (CAR (CONTRACT S))))

(RESULT (LAMBDA (S) (DDR 9 S)))

(FIND (LAMBDA (N P) (COND (EQUAL N 1) P) (T (FIND (SUB1 N) (PREDECESSOR P))))))

```
(SUCCESSOR (LAMBDA (P) (CADR (ASSOC P (QUOTE ((WEST NORTH (NORTH EAST)
        (EAST SOUTH) (SOUTH WEST)))))))
(PREDECESSOR (LAMBDA (P) (SUCCESSOR (SUCCESSOR (SUCCESSOR P)))))
(DDR (LAMBDA (N X)(COND ((ZEROP N) (CAR X)) (T (DDR (SUB1 N) (CDR X))))))
(INCREMENT (LAMBDA (P Q (COND ((EQ P (QUOTE WEST)) (CONS (ADD1 (CAR Q))
                                                         (CDR Q)))
                              ((EQ P (QUOTE NORTH)) (CONS (CAR Q) (CONS
                                                         (ADD1 (CADR Q))
                                                         (CADDR Q))))
                              ((EQ P (QUOTE EAST)) (CAR Q)(CADR Q) (ADD1
                                                         (CADDR Q)) (CADDDR Q)))
                              ((EQ P (QUOTE SOUTH))(LIST (CAR Q)(CADR Q)
                                                         (CADDR Q) (ADDL
                                                         (CADDDR Q)))))))
```

These should all be self-explanatory, with the following exceptions. DDR is just an abbreviation for finding far-away parts of the tree, e.g. caddddddddddr(s) = ddr(10 s). Newwinner determines which person won the trick, by calling MAXFRED, which determines which card of <u>history</u> won the trick.

The conventions are that a card is a doublet, e.g. (S Q) (D 5), whose first member is the suit and whose second is the value (2--10,J,Q,K,A); a contract is a similar doublet, whose first member is the trump suit, or NT for no trump, and whose second is the bid in the suit, thus (C 3), (NT 6) etc. MAXFRED and auxiliary functions examine a quadruple of cards and find which card is the highest; returning the number of (MAXFRED (LAMBDA (LST TRUMP) (MXX (QUOTE (NIL 0)) LST TRUMP 5 5)))

```
(MXX (LAMBDA (X L TR C D) (COND ((NULL L) D)
                                ((MACHT X (CAR L TR) (MXX (CAR L) (CDR L)
                                                          TR (SUB1 C)(SUBL C)))
                                (T (MXX X (CDR L) TR (SUB1 C) D)))))
(MACHT (LAMBDA (C D TR) (COND ((EQ (CADR C (CADR D)) (LESST (CAR C)(CAR D)))
                              (T (AND (NOT (EQ (CADR C TR)) (EQ (CADR D)
                                                          TR)))))))
(LESST (LAMBDA (X Y)(COND ((NUMBERP X) (AND (NUMBERP Y) (LESSP X Y)))
                              (T (COND ((EQ X (QUOTE A)) NIL)
```

```
                                ((EQ X (QUOTE K)) (EQ Y (QUOTE A)))
                                ((EQ X (QUOTE Q)) (OR (EQ Y (QUOTE A))
                                       (EQ Y (QUOTE K))))
                                (T (OR (EQ Y (QUOTE A))
                                       (EQ Y (QUOTE K))
                                       (EQ Y (QUOTE Q)))))))))
```

Play of a card:

```
3.(F3 (LAMBDA (S) ((LAMBDA (Z) (FIXUP Z (CURPLAYER S) (LIST (WC S)
                                  (NC S) (EC S) (SC S)
                            (TRICKS S)(SUCCESSOR (CURPLAYER S))(CONS Z
                            (HISTORY S)) (COUNT S)(CONTRACT S)(RESULT S))))
                     (CARDPLAY S))))
(FIXUP (LAMBDA (C P S)(COND ((EQ P (QUOTE WEST)) (CONS (DELETE C(WC S))
                                       (CDR S)
                                ((EQ P (QUOTE NORTH)) (CONS (WC S) (CONS
                                       (DELETE C (NC S)) CDDR S))))
                                ((EQ P (QUOTE EAST)) (CONS (WC S)(CONS (NC S)
                                       (CONS (DELETE C(EC S)) (CDDDR S)))))
                                ((EQ P (QUOTE SOUTH)) (CONS (WC S) (CONS (NC S)
                                       (CONS (EC S) (CONS (DELETE C (SC S))
                                       (CDDDDR S)))))))))
```

Thus if P plays a given card from his hand, this card is deleted from the list of cards in his hand and added to the history of this trick, and the next player in order becomes the current player. Finally, the card to be played is determined by CARDPLAY from the bestcard routine:

```
(CARDPLAY (LAMBDA (S) (BESTCARD S (CURPLAYER S))))
(BESTCARD (LAMBDA (S P) (COND    ((EQ P (QUOTE WEST)) (B1 (WC S) S))
                                 ((EQ P (QUOTE NORTH)) (B1 (NC S) S))
                                 ((EQ P (QUOTE EAST)) (B1 (EC S) S))
                                 ((EQ P (QUOTE SOUTH)) (B1 (SC S) S))  )))
(B1 (LAMBDA (CARDS S)  (COND     ((NULL (HISTORY S)) (B2 CARDS))
                                 (T (B3 CARDS (SUIT (LAST (HISTORY S))))))))
```

```
(B2 (LAMBDA (CARDS)   (COND      ((NULL (CAR CARDS)) (B2 (CDR CARDS)))
                                 (T (CAAR CARDS)))))

(B3 (LAMBDA (CARDS ST)(COND      ((NULL (LOOK CARDS ST)) (B2 CARDS))
                                 (T (CAR (LOOK CARDS ST))))))

(LOOK (LAMBDA (CARDS ST)(COND    ((EQ ST (QUOTE S)) (CAR CARDS))
                                 ((EQ ST (QUOTE H)) (CADR CARDS))
                                 ((EQ ST (QUOTE D)) (CADDR CARDS))
                                 (EQ ST (QUOTE C)) (CADDDR CARDS))   )))

(LAST (LAMBDA (X)     (COND      ((NULL (CDR X)) (CAR X)) (T (LAST (CDR X))))))
(SUIT CAR)
```

Note: The lisp function BESTCARD which appears here is the simplest form of algorithm for making the program choose a card to play when it is its turn to play. The algorithm calls for the play of the first card in the hand which conforms with the rules, i.e. which follows suit, if necessary. This algorithm is inserted here only that the program should operate. Obviously, it is this part of the program where all the interesting work can be done toward a high-quality bridge-playing routine.

Note: The program given above has not been debugged as yet.