



OTHELLO
CON IL COMPUTER

Tutti sappiamo che i compiti che un elaboratore può svolgere sono innumerevoli, ed i suoi campi di applicazione, diversissimi, sono limitati praticamente solo dalla fantasia e dall'ingegno dei programmatori. Una delle applicazioni più interessanti è senza dubbio la simulazione di uno dei contendenti di un gioco a due persone: i programmi di questo tipo sono purtroppo tra i più difficili da stendere, e mettono a dura prova la perizia di chi ricerca gli algoritmi e ne cura l'applicazione. Una volta presi dal fascino di questa programmazione, però, si può rimanere piacevolmente occupati per mesi nella stesura di versioni sempre più perfette di uno stesso gioco.

Con questo articolo ci proponiamo di avvicinare il lettore verso questo genere di programmi, che rappresenta per molti possessori di elaboratori un campo assai ricco di nuove possibili esperienze.

Giochi a due persone famosissimi come gli scacchi e la dama sono oggetto di programmazione da quando sono nati i computer, e anche da prima: è nota infatti l'esistenza di più o meno funzionanti macchine create per questi giochi in tempi in cui l'unica automazione possibile era quella degli ingranaggi da orologeria. Oggi i discendenti di quei congegni, favoriti dai bassi prezzi dei microprocessori, stanno entrando prepotentemente nelle nostre case sotto l'aspetto di scatolette intelligenti capaci di intrattenerci piacevolmente con discreti livelli di gioco.

È però sconsigliabile accostarsi per la prima volta a questo affascinante settore con giochi complessi come gli scacchi, che per essere ben giocati richiedono un lungo apprendistato anche ad un cervello umano. Per le nostre applicazioni useremo quindi un gioco forse non molto diffuso in Italia ma avvincente, intelligente e nello stesso tempo semplice nelle sue poche regole: l'Othello. Il nostro scopo, come spiegato in altra parte dell'articolo, è quello di spingervi alla realizzazione di vostri programmi di gioco: per fare ciò ci limiteremo a darvi dei suggerimenti e a presentare un programma di esempio, molto semplice e "poco intelligente"; e tanto per mostrare che non c'è bisogno di un particolare hardware ne riporteremo le versioni per due macchine particolarmente diffuse, e profondamente diverse: l'Apple II e la Texas TI-59. Naturalmente prima di fare ciò converrà introdurre il gioco e le sue regole, a vantaggio di chi ancora non lo conosce.

Il gioco

L'Othello, a differenza dei più diffusi giochi a due persone, non è molto antico: è stato messo in commercio una decina d'anni fa su un brevetto giapponese. In verità, però, è praticamente identico al Reversi, un gioco molto popolare nel secolo scorso

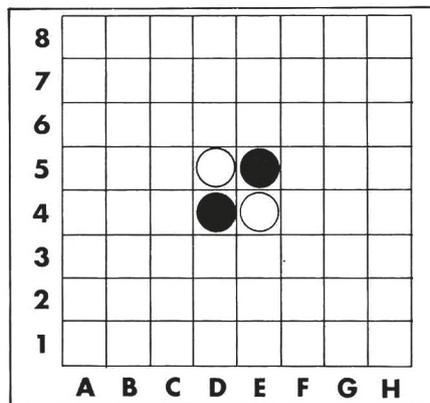


Figura 1 - Il diagramma mostra la posizione iniziale del gioco, e la notazione usata per identificare le caselle. Ricordiamo che il nero gioca per primo.

in Inghilterra, dal quale differisce solo per la rigida posizione di partenza, che è solo una tra le molte possibili del Reversi. Il gioco si svolge su una scacchiera 8 per 8 generalmente verde o comunque di un colore uniforme diverso dal bianco e dal nero, colori destinati alle pedine. Queste, contrariamente a quelle della dama, hanno una faccia bianca ed una nera. Nell'Othello infatti i pezzi appartengono all'uno o all'altro giocatore a seconda del colore della faccia superiore, e nel corso del gioco

possono venire ribaltati in modo da cambiare colore e quindi ruolo; il numero totale di pedine è 64, quanto serve per ricoprire completamente la scacchiera.

Per avere una notazione rigorosa delle mosse possiamo usare una notazione simile a quella degli scacchi: colonne da A ad H (da sinistra a destra) e righe da 1 a 8 (dal basso verso l'alto). All'inizio della partita si dispongono due pedine bianche e due nere nelle caselle centrali del campo, come illustrato in figura 1; la prima mossa è sempre del nero.

Il gioco consiste nel porre in campo, alternativamente, una pedina col proprio colore rivolto verso l'alto, facendo sì che questa catturi almeno una pedina avversaria (di colore opposto). La presa avviene quando la pedina appena posata chiude, lungo le direzioni uscenti dalla sua casella (verticali orizzontali e diagonali), una o più pedine avversarie fra sé e una delle pedine del proprio colore che già si trovavano sul campo. Ancora più semplicemente: la pedina deve essere posta sulla scacchiera in modo da formare, anche su più direzioni, delle file continue di una o più pedine avversarie terminanti ai due estremi con una o più pedine proprie, di cui una deve essere quella appena giocata. Le pedine avversarie così racchiuse e quindi man-

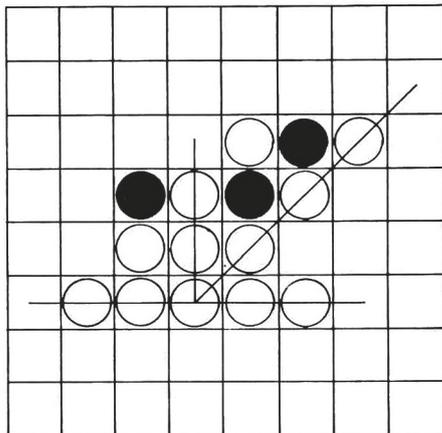
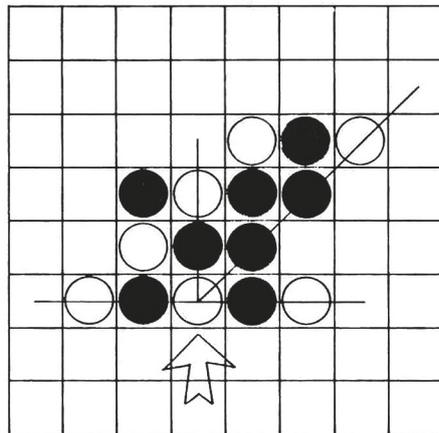
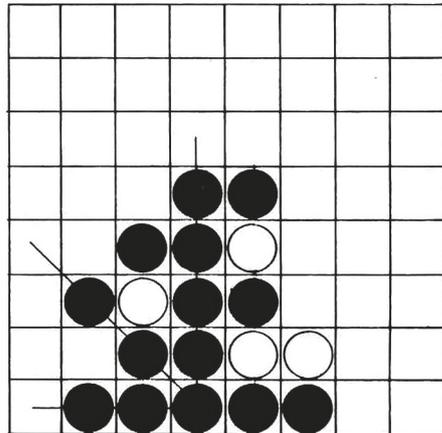
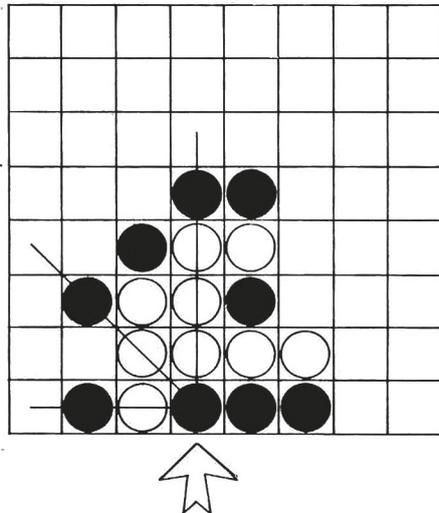


Figura 2 - Due esempi che illustrano come si muove. In alto una mossa del bianco, in basso una del nero: in entrambi i casi la pedina mossa è indicata dalla freccia (nel diagramma di sinistra). A destra è riportata la situazione della scacchiera dopo la cattura dei pezzi.



giate vengono capovolte, assumendo il colore del mangiante. La cattura di almeno una pedina avversaria è una condizione

necessaria per poter eseguire la mossa: se il giocatore che deve muovere non può mangiare nessuna pedina salta il turno, e continua a farlo finché non gli sia nuovamente possibile una mossa con cattura.

Regole tanto semplici possono forse apparire complesse se spiegate a parole senza esempi concreti, ma uno sguardo attento al diagramma di figura 2 sarà certamente sufficiente a fugare ogni dubbio. La vittoria spetterà al concorrente che a fine gioco, quando cioè tutto il campo sarà riempito, avrà più pedine del proprio colore sulla scacchiera. Ripetiamo che la mossa può essere effettuata solo se è possibile mangiare almeno una pedina avversaria, ed in questo caso è obbligatoria; deve invece essere ceduta all'avversario in caso contrario. Se anche lui è impossibilitato a muovere la partita termina prima che tutta la scacchiera sia riempita, e la vittoria va, come sempre, al concorrente rappresentato dal maggior numero di pedine in campo.

Come già accennato, in questo articolo porremo soltanto le basi generali del gioco, e non entreremo nel dettaglio delle varie strategie possibili, per non limitare la fantasia di chi volesse eventualmente aderire al nostro invito a cimentarsi nella scrittura di un proprio programma. Riteniamo comunque doveroso introdurre alcuni ele-

9	3	5	5	5	5	3	9
3	0	2	1	1	2	0	3
5	2	4	3	3	4	2	5
5	1	3			3	1	5
5	1	3			3	1	5
5	2	4	3	3	4	2	5
3	0	2	1	1	2	0	3
9	3	5	5	5	5	3	9

Figura 3 - Questa tabella mostra i valori potenziali delle varie caselle della scacchiera. Naturalmente le quattro posizioni centrali, essendo occupate fin dall'inizio, non compaiono.

mentari comportamenti di gioco, senza conoscere i quali non è possibile competere neanche con giocatori poco esperti. È sufficiente giocare pochissime partite per capire che durante le fasi iniziali è inutile affannarsi alla ricerca della mossa che faccia conquistare più pedine avversarie, mentre è molto proficuo cercare di guadagnare le caselle strategicamente più favore-

voli: le posizioni più forti, ad esempio, sono i quattro angoli della scacchiera, in quanto una volta conquistati non possono più venire ripresi dall'avversario. Da essi inoltre si può partire per la conquista definitiva dei bordi del campo: altre posizioni molto forti strategicamente. Una pedina sul bordo può venire catturata esclusivamente dal bordo o dall'angolo. Se l'avversario è valido cercherà, a sua volta, di conquistare angoli e bordi; quindi altro compito di un buon programma è di opporsi a che ciò avvenga con facilità. Alla luce di quanto detto si intuisce subito che le caselle sono più o meno buone a seconda che facciano conquistare all'avversario posizioni negative o positive. Esempio tipico di posizione negativa è quello della casella adiacente in diagonale all'angolo, presa la quale diventa molto difficile impedire all'avversario di conquistare l'angolo.

Da tutte queste parole si può estrarre subito un algoritmo che faccia dipendere la scelta della mossa da una tabella di valori delle singole caselle, sul tipo di quella in figura 3; e volendo fare le cose più sofisticate la si potrebbe rendere dinamica, ossia far variare i coefficienti in funzione dello svolgimento del gioco e del riempimento della scacchiera. Infatti nelle ultime mosse

```

100 DIM B(71): GOSUB 510: HOME
110 INPUT "CHI INIZIA ? " : Z
120 IF Z# "ID" THEN HOME : GOTO 320
130 HOME
140 GOSUB 580
150 R = - 1 : A = 0 : I = 72 : F = 0
160 I = INT (I - 1)
170 IF I = 0 GOTO 270
180 IF B(I) < 0 GOTO 160
190 RESTORE : FOR S = 1 TO 8
200 READ Q
210 GOSUB 410
220 NEXT
230 IF F = 1 GOTO 300
240 IF I - INT (I) < A - INT (A) GOTO 160
250 A = I
260 GOTO 160
270 IF A = 0 THEN VTAB 22: HTAB 10: CALL - 938: PRINT "PASSO": HOME
280 F = 1 : I = INT (A) : A = - 2
290 GOTO 190
300 IF R = - 2 GOTO 140
310 CALL - 192: VTAB 22: HTAB 10: CALL - 956: PRINT "IL MIO GIOCO E' " : I
320 GOSUB 580
330 VTAB 23: HTAB 10: CALL - 838: INPUT "DOVE GIOCHI ? " : I
340 IF I = 0 GOTO 150
350 R = - 2 : B(I) = - 1 : F = 1
360 GOTO 190
370 D = D - Q
380 B(D) = - 3 - B(D)
390 IF D < > I + Q GOTO 370
400 RETURN
410 D = INT (I + Q)
420 IF D < 0 OR D > 71 THEN RETURN
430 IF B(D) < > R THEN RETURN
440 D = D + Q
450 IF D > 71 OR D < 0 THEN RETURN
460 IF B(D) > 0 THEN RETURN
470 IF B(D) = R GOTO 440
480 IF F = 1 GOTO 370
490 I = I + ( ABS ( ( INT (I) - D) / Q) - 1) / 100
500 RETURN
510 REM - INIZ SCACCHIERA
520 MOSSA = 1
530 FOR J = 0 TO 63 STEP 9
540 B(J) = 1 : NEXT
550 B(32) = - 2 : B(46) = - 2
560 B(31) = - 1 : B(47) = - 1
570 RETURN
580 REM - DISEGNO SCACCHIERA
590 VTAB 2: TAB 1: TUE = 0: MOSSA = MOSSA + 1
600 FOR K = 0 TO 71
610 IF B(K) = 1 THEN PRINT " " + STR$(K);
620 IF B(K) = 0 THEN PRINT RIGHT$( " " + STR$(K); 3);
630 IF B(K) = - 2 THEN PRINT " ( ) "; MIE = MIE + 1
640 IF B(K) = - 1 THEN PRINT " - - "; TUE = TUE + 1
650 NEXT
660 VTAB 12: HTAB 30: PRINT "MOSSA = " : MOSSA
670 VTAB 15: HTAB 30: PRINT "MIE = " : RIGHT$( " " + STR$(MIE); 2)
680 HTAB 30: PRINT "TUE = " : RIGHT$( " " + STR$(TUE); 2)
690 RETURN
700 DATA -10,-9,-8,-1,1,8,9,10
    
```

```

000 01 1 060 06 6 120 73 RC* 180 95 = 241 67 EQ
001 94 +/- 061 42 STD 121 52 52 181 44 SUM 242 02 02
002 42 STD 062 53 53 122 22 INV 182 51 51 243 18 18
003 54 54 063 71 SBR 123 67 EQ 183 92 RTN 244 92 RTN
004 25 CLR 064 01 01 124 01 01 184 43 RCL 245 76 LBL
005 42 STD 065 03 03 125 80 83 185 50 50 246 11 R
006 50 50 066 01 1 126 43 RCL 186 29 CP 247 29 CP
007 04 4 067 94 +/- 127 53 53 187 22 INV 248 22 INV
008 02 2 068 42 STD 128 44 SUM 188 67 EQ 249 67 EQ
009 42 STD 069 53 53 129 52 52 189 01 01 250 02 02
010 51 51 070 71 SBR 130 43 RCL 190 93 93 251 53 53
011 43 RCL 071 01 01 131 52 52 191 25 CLR 252 81 RST
012 51 51 072 03 03 132 29 CP 192 91 R/S 253 42 STD
013 59 INT 073 08 8 133 32 X/T 193 59 INT 254 51 51
014 42 STD 074 94 +/- 134 77 GE 194 42 STD 255 02 2
015 51 51 075 42 STD 135 01 01 195 51 51 256 94 +/-
016 22 INV 076 53 53 136 83 83 196 86 STF 257 42 STD
017 97 DSZ 077 71 SBR 137 04 4 197 01 01 258 54 54
018 51 51 078 01 01 138 01 1 198 02 2 259 01 1
019 01 01 079 03 03 139 22 INV 199 94 +/- 260 94 +/-
020 84 84 080 87 IFF 140 77 GE 200 72 ST* 261 72 ST*
021 29 CP 081 01 01 141 01 01 201 51 51 262 51 51
022 73 RC* 082 02 02 142 83 83 202 61 GTO 263 86 ST*
023 51 51 083 05 05 143 29 CP 203 00 00 264 01 01
024 22 INV 084 43 RCL 144 73 RC* 204 28 28 265 61 GTO
025 67 EQ 085 51 51 145 52 52 205 43 RCL 266 00 00
026 00 00 086 22 INV 146 77 GE 206 54 54 267 28 28
027 11 11 087 59 INT 147 01 01 207 32 X/T 268 76 LBL
028 07 7 088 32 X/T 148 83 83 208 02 2 269 15 E
029 94 +/- 089 43 RCL 149 75 - 209 94 +/- 270 25 CLR
030 42 STD 090 50 50 150 43 RCL 210 22 INV 271 42 STD
031 53 53 091 22 INV 151 54 54 211 67 EQ 272 01 1
032 71 SBR 092 59 INT 152 95 = 212 02 02 273 42 STD
033 01 01 093 77 GE 153 67 EQ 213 15 15 274 00 00
034 03 03 094 00 00 154 01 01 214 81 RST 275 42 STD
035 06 6 095 11 11 155 26 26 215 43 RCL 276 07 07
036 94 +/- 096 43 RCL 156 87 IFF 216 51 51 277 42 STD
037 42 STD 097 51 51 157 01 01 217 91 R/S 278 14 14
038 53 53 098 42 STD 158 02 02 218 43 RCL 279 42 STD
039 71 SBR 099 50 50 159 18 18 219 53 53 280 21 21
040 01 01 100 61 GTO 160 53 ( 220 22 INV 281 42 STD
041 03 03 101 00 00 161 43 RCL 221 44 SUM 282 28 28
042 01 1 102 11 11 162 51 51 222 52 52 283 42 STD
043 42 STD 103 43 RCL 163 59 INT 223 03 3 284 35 35
044 53 53 104 51 51 164 75 - 224 94 +/- 285 94 +/-
045 71 SBR 105 85 + 165 43 RCL 225 75 - 286 42 STD
046 01 01 106 43 RCL 166 52 52 226 73 RC* 287 17 17
047 03 03 107 53 53 167 54 ) 227 52 52 288 42 STD
048 08 8 108 95 = 168 55 + 228 95 = 289 25 25
049 42 STD 109 59 INT 169 43 RCL 229 72 ST* 290 02 2
050 53 53 110 42 STD 170 53 53 230 52 52 291 94 +/-
051 71 SBR 111 52 52 171 95 = 231 43 RCL 292 42 STD
052 01 01 112 29 CP 172 50 50 232 52 52 293 18 18
053 03 03 113 32 X/T 173 75 - 233 32 X/T 294 42 STD
054 07 7 114 77 GE 174 01 1 234 43 RCL 295 34 34
055 42 STD 115 01 01 175 95 = 235 51 51 296 25 CLR
056 53 53 116 83 83 176 55 + 236 85 + 297 91 R/S
057 71 SBR 117 43 RCL 177 01 1 237 43 RCL 298 00 0
058 01 01 118 54 54 178 00 0 238 53 53 299 00 0
059 03 03 119 32 X/T 179 00 0 239 95 = 300 00 0
060 02 22 INV 240 22 INV 301 30 0
    
```

Questo è il programma, nelle versioni per Apple II (a sinistra) e Texas TI-59 (a destra).

Othello con il computer

della partita, quando il campo sta per riempirsi, è necessario ricordarsi che il fine del gioco è quello di avere sulla scacchiera più pedine dell'avversario, e quindi bisogna cercare di capovolgere più pezzi possibile tentando di minimizzare le prese dell'avversario nelle mosse successive.

Queste poche notizie non sono che un cenno di tutte le finesse dell'Othello, che come per ogni altro gioco a due persone si possono apprendere solo per esperienza diretta.

Chi volesse programmare un Othello è pertanto vivamente consigliato di giocare qualche partita contro un buon avversario umano, per constatare direttamente quali e quante sottili strategie si possano escogitare e mutare nel corso del gioco, e di quanto la sorte sia alterna fino alle ultimissime fasi della partita. Sorge perciò l'arduo problema, che giriamo ai lettori, di studiare una opportuna funzione di valutazione che metta in grado il programma di selezionare in modo pseudo-intelligente quella, tra le mosse valide, più opportuna in un dato contesto di gioco.

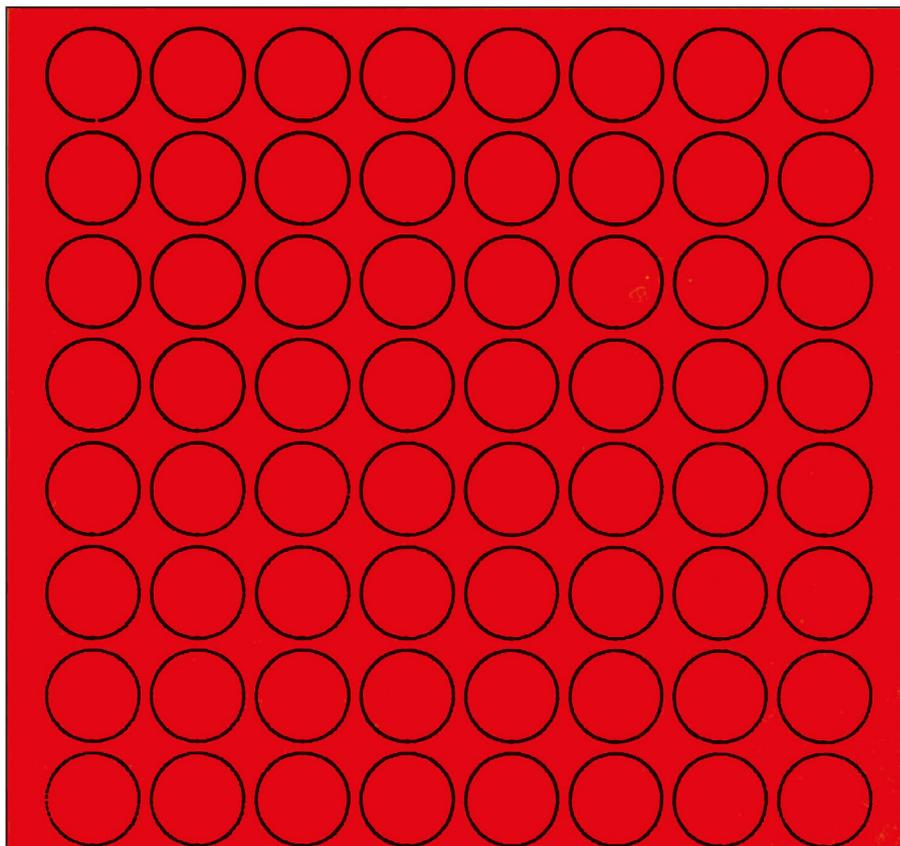
L'Othello sugli elaboratori

Il gioco, anche se a diversi livelli, può essere programmato su tutti gli elaboratori, comprese le calcolatrici programmabili. Per queste ultime però, date le modeste capacità di memoria e soprattutto le basse velocità di elaborazione, proponiamo di sviluppare il gioco su di una scacchiera 6 per 6 anziché 8 per 8; questa limitazione non toglie nulla alla logica della programmazione ed evita il protrarsi delle partite oltre limiti ragionevoli, anche se si perde un po' dell'agonismo della versione originale.

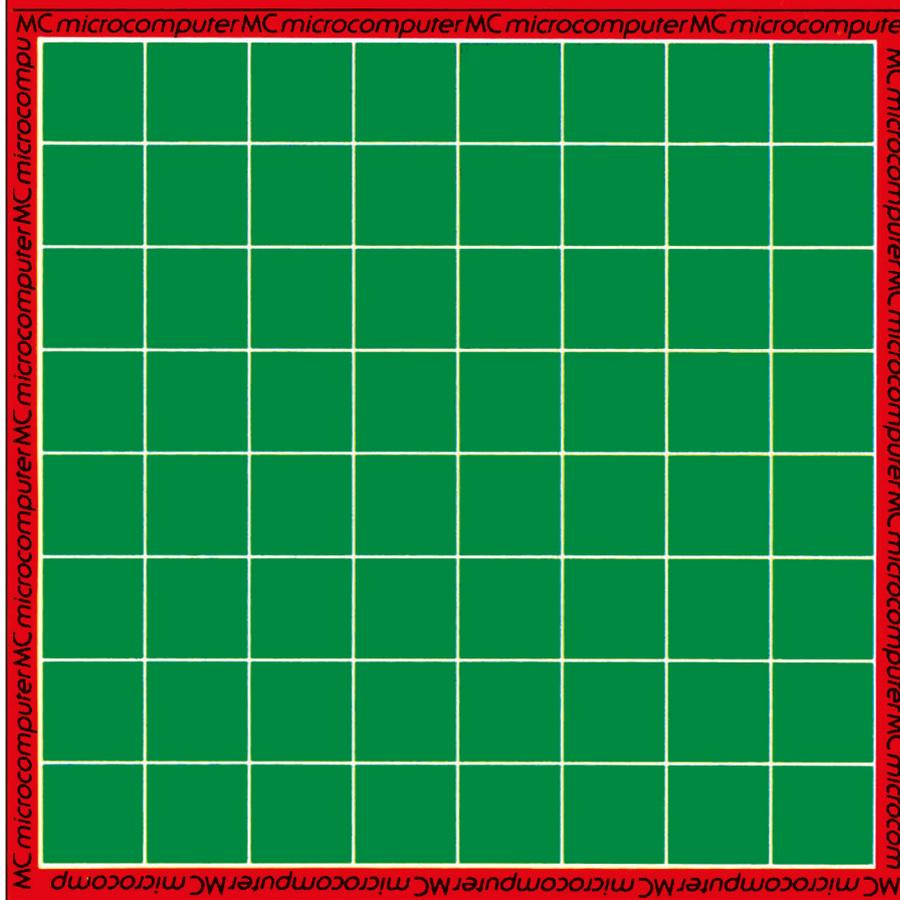
Prima però di occuparsi di far giocare i vostri elaboratori ad Othello, introduciamo il concetto di livello di gioco, altrimenti detto "forecast level" (livello di previsione).

Chi, macchina o uomo che sia, nel giocare ricerca la mossa migliore valutando semplicemente la immediata situazione della scacchiera gioca al primo livello, il più elementare. Giocare al secondo livello vuol dire che per ogni mossa possibile si esaminano tutte le possibili risposte dell'avversario, e si sceglie quindi la propria mossa in modo da non concedere molte chances di buoni colpi-risposta al nemico. Al terzo livello si gioca quella mossa che, supponendo un'intelligente risposta dell'avversario, prepari il campo per il colpo successivo; e così via per gli altri livelli di previsione. La vostra programmabile giocherà quasi certamente al primo livello, dati gli ovvi limiti di memoria e velocità: chi ha invece a disposizione un personal, qualcosina di più la potrà fare.

Detto ciò passiamo a dettagli un po' più operativi. Per far giocare un calcolatore ad Othello è necessario dargli una scacchiera e le pedine, bianche e nere: la scacchiera sarà costituita da un certo insieme di varia-



Costruite la scacchiera di MCmicrocomputer
(le istruzioni sono sul retro del foglio)



Costruite la scacchiera di MCmicrocomputer

1. tagliate questo riquadro
2. incollatelo su un cartoncino (di colore diverso dal rosso)
3. ritagliate la scacchiera
4. ritagliate ciascuna delle 64 pedine, le cui due facce risulteranno una rossa, l'altra del colore del cartoncino.
5. iniziate la vostra partita.
Buon divertimento!

bili, mentre le pedine saranno opportuni valori da inserire nelle stesse. Come già accennato per le programmabili useremo il formato 6 per 6, per i personal l'8 per 8; quale insieme di variabili per simulare la scacchiera consigliamo di usare un vettore (array ad un solo indice): dovendo infatti analizzare tutte le caselle sarà bene non complicarsi troppo la vita con più indici. In figura 4a vediamo la rappresentazione della scacchiera standard; le caselle 0, 9, 18, ..., 63 sono di "bordo", ossia servono per far capire al computer se due caselle sono o non sono sulla stessa riga. Tutto questo perché la scacchiera all'interno del calcolatore non è che un lungo vettore (figura 4b). Lo stesso discorso, con ovvie modifiche, va ripetuto per il formato 6 per 6. In entrambi i casi il nostro consiglio è di porre nelle caselle vuote il valore zero e nei bordi il

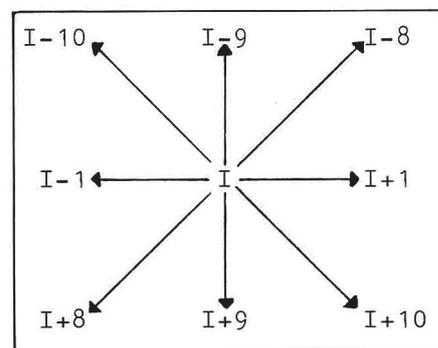


Figura 5 - Lo schema delle costanti necessarie per esplorare, partendo da una generica casella, quelle adiacenti nelle varie direzioni.

valore 1, contrassegnando poi le caselle proprie col valore -2 e quelle dell'avversario col valore -1. Questa scelta per le pedine non è del tutto a caso: infatti questi valori risultano abbastanza comodi per capovolgere le pedine conquistate. La semplice istruzione di assegnazione: contenuto casella = (-3) - contenuto casella restituisce per l'appunto la pedina capovolta, come è facile verificare. Basta applicare la formula a tutte le pedine da voltare ed il gioco è fatto.

Data così una semplice idea della rappresentazione della scacchiera passiamo alla ricerca delle possibili mosse. Non ha molta importanza da quale casella incominciare, tanto bene o male bisogna analizzarle tutte. Cominciamo ad esempio da quella in basso a destra, la numero 71 (o 41 nella scacchiera 6 per 6). Potendo giocare la nostra pedina solo dove non ve ne siano altre, e tantomeno sui "bordi", analizzeremo solo le caselle vuote, ossia quelle che contengono zero. Di ogni posizione utilizzabile dobbiamo vedere se nelle otto direzioni principali (Nord, Est, Sud, Ovest, NE, SE, SO, NO) sia possibile conquistare pedine; a tal fine, lo ricordiamo, è necessario che una o più pedine avversarie siano comprese tra due o più pedine proprie, di cui una già sulla scacchiera e una appena posata. Per effettuare questa ricerca basta cominciare in una qualsiasi direzione, con-



0	1	2	3	4	5	6	7	8	9	63	64	65	66	67	68	69	70	71
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

trollando se nella casella c'è una pedina avversaria: se non c'è si deve cambiare direzione (e al termine delle direzioni cambiare casella), se invece c'è si prosegue nella stessa direzione, controllando il contenuto di ogni casella col seguente algoritmo: si presentano tre casi:

- a) Nella casella c'è un'altra pedina avversaria: allora si prosegue nella stessa direzione.
- b) Siamo usciti fuori dalla scacchiera o siamo capitati sul bordo, o la casella esaminata è vuota: in questo caso non è possibile mangiare, e allora si cambia direzione (o casella, se tutte e otto le direzioni sono state esaminate).
- c) È presente una pedina propria: è pertanto possibile catturare tutte le pedine avversarie comprese fra la casella analizzata per ultima e quella da cui si è partiti; si prosegue poi la ricerca nelle restanti direzioni per controllare se sia possibile conquistare qualche altra pedina con la stessa mossa.

Resta il problema di come effettuare la ricerca nelle otto direzioni. Dando un'occhiata alla scacchiera 8 per 8 (figura 4a), noterete che, considerata una qualunque casella, ad es. la 40, quelle adiacenti ad essa nelle otto direzioni sono contraddistinte dai numeri: 30, 31, 32, 41, 50, 49, 48, 39; ossia 40-10, 40-9, 40-8, 40+1, 40+10, 40+9, 40+8, 40-1. In altre parole, aggiungendo ripetutamente al valore 40 il valore +8, ottenendo via via 48, 56; 64, si percorre la direzione sud-est; un discorso analo-

0	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53
54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71

Figura 4 - La scacchiera così come viene usata dal calcolatore. Benché sia più comodo rappresentarla in forma di matrice (sotto) in realtà in memoria è realizzata come un vettore (sopra).

go, con diverse costanti, vale per le restanti direzioni. Lo schema è riportato in figura 5; per la scacchiera 6 per 6 cambiano le costanti ma non il principio. In entrambi i casi bisognerà stare attenti a non uscire dalla scacchiera durante la ricerca.

Il programma di esempio che vi presentiamo, nelle versioni per Apple II e TI-59, esegue questo test per tutte le caselle della scacchiera, e gioca infine la mossa che conquista più pedine. Abbiamo già detto che questa non è sempre la strategia preferibile, ed infatti questo programma non è certo il migliore; anzi, batterlo non è difficile. Non vogliamo però dirvi altro: il nostro scopo non è quello di darvi un programma

ma di farvelo fare, dandovi il maggior numero di consigli.

Il programma

Per concludere diamo un breve sguardo al nostro programmino. Bisogna subito dire che sulla 59, nonostante la riduzione del numero di caselle, il gioco non è molto veloce (dai 3 ai 5 minuti per mossa), e ciò non solo a causa della bassa velocità di elaborazione ma anche per la complessità del programma, caratterizzato da numerosissimi salti e confronti.

I due programmi fanno né più né meno ciò che è stato finora descritto a parole: il diagramma di flusso è in figura 6. Si entra dal punto K se gioca per prima la macchina, dal punto A altrimenti. Sulla 59 si deve innanzitutto premere E per inizializzare la scacchiera, poi si preme RST R/S per far giocare prima lei, altrimenti si imposta la propria mossa e si preme A. Ricordiamo che la numerazione delle caselle è differente nel caso 6 per 6 da quello 8 per 8, e che per passare si gioca nella casella 0.

Entrambi i programmi sfruttano la stessa subroutine (W nel flow-chart) per ricercare le mosse e capovolgere le pedine: lo stato del flag 1 (la variabile logica F nella versione BASIC) indica quale delle due funzioni viene svolta.

Ripetiamo che questo programma è solo di esempio, e manca di molte funzioni accessorie: ad esempio non controlla la validità delle mosse dell'avversario, e non si accorge della fine della partita. L'importante però è che chiarisca il meccanismo di gioco (anzi, uno dei tanti meccanismi possibili), e non è difficile abbellirlo e completarlo un tantino.

Conclusione

Con questo articolo speriamo di aver suscitato il vostro interesse verso l'Othello e l'intelligenza artificiale. Crediamo, di aver mostrato chiaramente come, con un po' di pazienza, sia possibile insegnare ad un computer a fare qualcosa di tipicamente umano: giocare. Va anche detto che il programma-giochetto, ben noto specialmente ai piccoli informatici in virtù della sua onnipresenza sui manuali delle programmabili, ha un alto valore didattico sia perché non deve necessariamente essere una banalità, sia perché psicologicamente dà molta soddisfazione giocare contro un proprio programma: se vince la macchina si ha il merito di aver sviluppato un buon algoritmo di gioco, se vince il programmatore si ha la soddisfazione di non essersi lasciati fregare da una scatola ignorante e presuntuosa...

A questo punto vi lasciamo ai vostri computer: buon lavoro, e non mancate di farci conoscere i risultati.

Andrea De Prisco
Silvio Cavalcanti

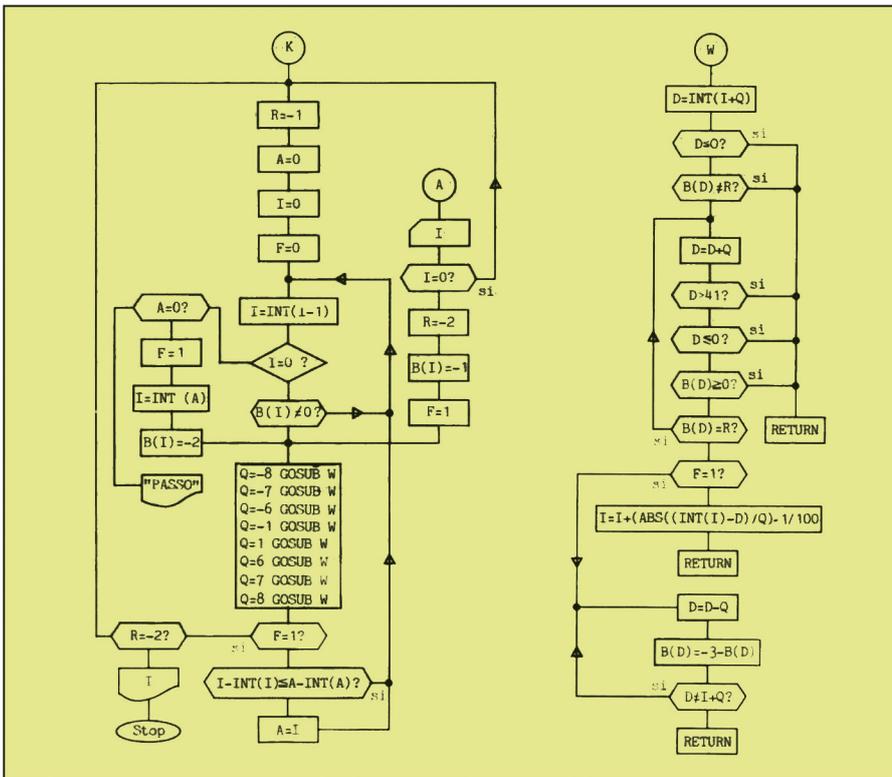


Figura 6 - I diagrammi di flusso del programma: il corpo principale è la A, mentre la routine W è quella che esamina le caselle e capovolge le pedine.