

Expert System Shell 128

di Fabio Villone - Napoli

Expert System Shell, o più brevemente ESS, come il nome stesso indica è uno «shell» in cui inserire varie regole per creare un «sistema esperto». Spieghiamo.

Un sistema esperto è schematicamente un programma che, in base a determinate conoscenze e mediante varie regole, determina nuove conoscenze.

Le regole sono espressioni del tipo: «se (condizione) allora (conseguenza)», da interpretarsi nel seguente modo: se la condizione è vera, allora è vera anche la conseguenza.

La condizione e la conseguenza sono chiamati «nodi» della «rete inferenziale».

I nodi sono affermazioni di qualunque genere, ad esempio «il tempo è bello» o «il tennis è uno bello sport», legate tra loro da rapporti specificati dalle regole di cui sopra, che formano la rete inferenziale.

Il tutto verrà chiarito, spero, da un esempio grafico.

Le regole:

- 1) se A e B allora C;
- 2) se B e D allora E;
- 3) se E allora F;
- 4) se C e F allora G;

dove A, B, C, D, E, F, G sono nodi generici, cioè affermazioni di qualunque tipo, generano la rete inferenziale di figura 1.

Una rete inferenziale può essere percorsa in avanti o all'indietro: nel primo

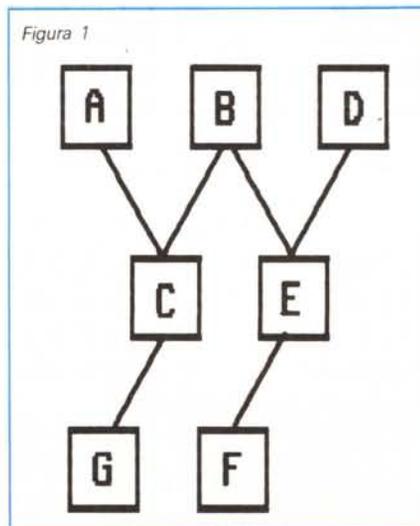
caso il computer, basandosi sugli stati già noti di verità o falsità dei nodi, cerca di procedere il più avanti possibile lungo la rete inferenziale. Rifacendoci all'esempio precedente, supponendo A falso, B e D veri, sarà E vero per la regola 2), F vero per la regola 3), ma C e G rimarranno ignoti, dato che le regole 1) e 4) non resteranno soddisfatte, essendo rispettivamente A falso e C ignoto.

Il secondo metodo, quello all'indietro, differisce dal primo poiché necessita l'indicazione di un «goal», una meta, da raggiungere. Stabilito dunque un goal, il computer cerca di raggiungerlo mediante la via più breve, eventualmente ponendo all'utente domande in merito ai nodi il cui valore (vero o falso) sia ignoto. Ricordando l'esempio, stabilendo G come goal e supponendo A e B veri, avremo che C sarà vero per la regola 1), ma F rimarrà ignoto, essendo D e di conseguenza E ignoti, per cui il computer dovrà chiedere all'utente il valore di F; se esso risulterà vero, allora sarà anche G vero; in caso contrario G resterà ignoto.

Passiamo all'implementazione sull'ESS.

Per inserire una regola basta digitare:

Figura 1



«se (condizione) allora (conseguenza).» (attenzione al punto finale). Il computer, dopo aver chiesto la conferma, controlla se i nodi (condizione) e (conseguenza) siano noti (in caso contrario li inserisce nell'apposito array), controlla che la regola sia compatibile con le altre già presenti in memoria, la inserisce nell'array, ed infine controlla se essa può fargli conoscere lo stato di nodi ignoti.

Particolare attenzione merita il criterio di compatibilità delle regole. In genere una regola non viene accettata quando la conseguenza fa pervenire a conclusioni contraddittorie con la condizione. Cioè, avendo già in memoria le regole «se A allora B.» e «se B allora C e D.» non è possibile inserire la regola «se C e D allora non A.», perché altrimenti affermando A si arriverebbe, scandendo le varie regole, a negare lo stesso nodo A, il che, ovviamente, è una contraddizione.

Altro criterio è quello secondo cui è inaccettabile una regola che abbia la stessa condizione di un'altra, ma conseguenza contraddittoria con essa, avendo, cioè, in memoria «se A e B allora C e D.» è inaccettabile «se A e B allora non D.». Infine, non è possibile che fra le condizioni o le conseguenze compaia più volte lo stesso nodo: è inaccettabile «se A e B e A allora C.» oppure «se A allora B e C e non C.».

È però possibile «se A allora A» o peggio, «se A allora non A.», per cui sarebbe preferibile non inserire regole del genere. È possibile concatenare varie condizioni o conseguenze, come si è visto, utilizzando la congiunzione «e». Ad esempio: «se A e B e C allora D e E.» oppure «se A allora B e C e non C.».

Per determinare lo stato di verità di un nodo basta digitarlo seguito dal punto, mentre per stabilire quello di falsità bisogna inserire al suo interno la negazione «non». Da notare che questa considerazione vale anche per le regole, per cui, esemplificando:

— «se il tempo non è bello allora io non esco.» significa: se il nodo «il tempo è bello» è falso, allora il nodo «io esco» è falso;

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 243.

```

1 ... Se:
bc e' noto:vero
ac e' noto:vero
ab e' noto:falso
  allora:
ab e' noto:vero
ab=sqr(bc^2+ac^2):vero

2 ... Se:
ah e' noto:vero
hb e' noto:vero
ab e' noto:falso
  allora:
ab e' noto:vero
ab=ah+hb:vero

3 ... Se:
a e' noto:vero
ch e' noto:vero
ab e' noto:falso
  allora:
ab e' noto:vero
ab=2a/ch:vero

4 ... Se:
ac e' noto:vero
ah e' noto:vero
ab e' noto:falso
  allora:
ab e' noto:vero
ab=(ac^2)/ah:vero

5 ... Se:
bc e' noto:vero
hb e' noto:vero
ab e' noto:falso
  allora:
ab e' noto:vero
ab=(bc^2)/hb:vero

6 ... Se:
ab e' noto:vero
ac e' noto:vero
bc e' noto:falso
  allora:
bc e' noto:vero
bc=sqr(ab^2-ac^2):vero

7 ... Se:
ch e' noto:vero
hb e' noto:vero
bc e' noto:falso
  allora:
bc e' noto:vero
bc=sqr(ch^2+hb^2):vero

8 ... Se:
a e' noto:vero
ac e' noto:vero
bc e' noto:falso
  allora:
bc e' noto:vero
bc=2a/ac:vero

9 ... Se:
ab e' noto:vero
hb e' noto:vero
bc e' noto:falso
  allora:
bc e' noto:vero
bc=sqr(ab*hb):vero

10 ... Se:
ab e' noto:vero
bc e' noto:vero
ac e' noto:falso
  allora:
ac e' noto:vero
ac=sqr(ab^2-bc^2):vero

11 ... Se:
ch e' noto:vero
ah e' noto:vero
ac e' noto:falso
  allora:
ac e' noto:vero
ac=sqr(ch^2+ah^2):vero

12 ... Se:
a e' noto:vero
bc e' noto:vero
ac e' noto:falso
  allora:
ac e' noto:vero
ac=2a/bc:vero

13 ... Se:
ab e' noto:vero
ah e' noto:vero
ac e' noto:falso
  allora:
ac e' noto:vero
ac=sqr(ab*ah):vero

14 ... Se:
bc e' noto:vero
hb e' noto:vero
ch e' noto:falso
  allora:
ch e' noto:vero
ch=sqr(bc^2-hb^2):vero

15 ... Se:
ac e' noto:vero
ah e' noto:vero
ch e' noto:falso
  allora:
ch e' noto:vero
ch=sqr(ac^2-ah^2):vero

16 ... Se:
a e' noto:vero
ab e' noto:vero
ch e' noto:falso
  allora:
ch e' noto:vero
ch=2a/ab:vero

17 ... Se:
ah e' noto:vero
hb e' noto:vero
ch e' noto:falso
  allora:
ch e' noto:vero
ch=sqr(ah*hb):vero

18 ... Se:
ab e' noto:vero
hb e' noto:vero
ah e' noto:falso
  allora:
ah e' noto:vero
ah=ab-hb:vero

19 ... Se:
ac e' noto:vero
ch e' noto:vero
ah e' noto:falso
  allora:
ah e' noto:vero
ah=sqr(ac^2-ch^2):vero

20 ... Se:
ac e' noto:vero
ab e' noto:vero
ah e' noto:falso
  allora:
ah e' noto:vero
ah=(ac^2)/ab:vero

21 ... Se:
ch e' noto:vero
hb e' noto:vero
ah e' noto:falso
  allora:
ah e' noto:vero
ah=(ch^2)/hb:vero

22 ... Se:
ab e' noto:vero
ah e' noto:vero
hb e' noto:falso
  allora:
hb e' noto:vero
hb=ab-ah:vero

23 ... Se:
bc e' noto:vero
ch e' noto:vero
hb e' noto:falso
  allora:
hb e' noto:vero
hb=sqr(bc^2-ch^2):vero

24 ... Se:
bc e' noto:vero
ab e' noto:vero
hb e' noto:falso
  allora:
hb e' noto:vero
hb=(bc^2)/ab:vero

25 ... Se:
ch e' noto:vero
ah e' noto:vero
hb e' noto:falso
  allora:
hb e' noto:vero
hb=(ch^2)/ah:vero

26 ... Se:
ab e' noto:vero
ch e' noto:vero
a e' noto:falso
  allora:
a e' noto:vero
a=(ab*ch)/2:vero

27 ... Se:
bc e' noto:vero
ac e' noto:vero
a e' noto:falso
  allora:
a e' noto:vero
a=(bc*ac)/2:vero

28 ... Se:
ab e' noto:vero
bc e' noto:vero
ac e' noto:vero
p e' noto:falso
  allora:
p e' noto:vero
p=ab+bc+ac:vero

```

Esempio a - Insieme delle regole che definiscono il triangolo rettangolo.

— «la lezione non è facile.» significherà comunicare al computer che il nodo «la lezione è facile» è falso.

Per stabilire un goal basta digitare il nodo-meta, in forma negativa o affermativa, seguito da un «?»; il computer cercherà di raggiungerlo, se il suo valore è ignoto, mediante la via più favorevole indicando poi il risultato dell'elaborazione. Qualora la meta sia irraggiungibile, comparirà un messaggio di errore.

Questi sono i comandi per manipolare la «conoscenza» del computer; passiamo ora a quelli dedicati all'editing (fra parentesi i parametri opzionali):

— «nodi (start)» fornisce la lista dei nodi da start in poi; essa è interrompibile con la space bar;

— «regole (start)» come sopra, con le regole;

— «save nomefile», «load nomefile» per salvare e caricare regole e nodi; aggiungono a nomefile il suffisso «.ess» a meno che non sia già presente un'altra estensione;

— «dir (file)» funziona come il comando «directory» del Basic;

— «azzerà (numero)» dà il valore «ignoto» al nodo specificato da numero; nel caso esso sia assente azzerà tutti i nodi in memoria;

— «modifica numero» permette di modificare il nodo specificato da numero; esso viene visualizzato con il cursore lampeggiante; basta premere return per non operare alcuna modifica;

— «regola numero» cancella una regola;

— «nodo numero» cancella un nodo; funziona solo per nodi non presenti in nessuna regola;

— «fre» visualizza il numero di nodi e di regole ancora disponibili;

— «fine» fa uscire dal programma; per rientrarvi basta digitare «cont» da Basic.

Note tecniche:

— La memorizzazione dei nodi avviene nell'array af\$(); il primo byte indica lo stato del nodo (0=falso, 1=vero, 2=ignoto).

```

1 ...ab e' noto:ignoto
2 ...bc e' noto:ignoto
3 ...ac e' noto:ignoto
4 ...ch e' noto:ignoto
5 ...ah e' noto:ignoto
6 ...hb e' noto:ignoto
7 ...p e' noto:ignoto
8 ...a e' noto:ignoto
9 ...ab=sqr(bc^2+ac^2):ignoto
10 ...ab=ah+bh:ignoto
11 ...ab=2a/ch:ignoto
12 ...ab=(ac^2)/ah:ignoto
13 ...ab=(bc^2)/hb:ignoto
14 ...bc=sqr(ab^2-ac^2):ignoto
15 ...bc=sqr(ch^2+hb^2):ignoto
16 ...bc=2a/ac:ignoto
17 ...bc=sqr(ab*hb):ignoto
18 ...ac=sqr(ab^2-bc^2):ignoto
19 ...ac=sqr(ch^2+ah^2):ignoto
20 ...ac=2a/bc:ignoto
21 ...ac=sqr(ab*ah):ignoto
22 ...ch=sqr(bc^2-hb^2):ignoto
23 ...ch=sqr(ac^2-ah^2):ignoto
24 ...ch=2a/ab:ignoto
25 ...ch=sqr(ah*hb):ignoto
26 ...ah=ab-hb:ignoto
27 ...ah=sqr(ac^2-ch^2):ignoto
28 ...ah=(ac^2)/ab:ignoto
29 ...ah=(ch^2)/hb:ignoto
30 ...hb=ab-ah:ignoto
31 ...hb=sqr(bc^2-ch^2):ignoto
32 ...hb=(bc^2)/ab:ignoto
33 ...hb=(ch^2)/ah:ignoto
34 ...p=ab+bc+ac:ignoto
35 ...a=(ab*ch)/2:ignoto
36 ...a=(bc*ac)/2:ignoto

```

Esempio b - Insieme delle condizioni sui nodi.

```

1 ...ab e' noto:vero
2 ...bc e' noto:vero
3 ...ac e' noto:vero
4 ...ch e' noto:vero
5 ...ah e' noto:vero
6 ...hb e' noto:vero
7 ...p e' noto:ignoto
8 ...a e' noto:ignoto
9 ...ab=sqr(bc^2+ac^2):ignoto
10 ...ab=ah+bh:ignoto
11 ...ab=2a/ch:ignoto
12 ...ab=(ac^2)/ah:ignoto
13 ...ab=(bc^2)/hb:vero
14 ...bc=sqr(ab^2-ac^2):ignoto
15 ...bc=sqr(ch^2+hb^2):ignoto
16 ...bc=2a/ac:ignoto
17 ...bc=sqr(ab*hb):ignoto
18 ...ac=sqr(ab^2-bc^2):vero
19 ...ac=sqr(ch^2+ah^2):ignoto
20 ...ac=2a/bc:ignoto
21 ...ac=sqr(ab*ah):ignoto
22 ...ch=sqr(bc^2-hb^2):ignoto
23 ...ch=sqr(ac^2-ah^2):ignoto
24 ...ch=2a/ab:ignoto
25 ...ch=sqr(ah*hb):ignoto
26 ...ah=ab-hb:ignoto
27 ...ah=sqr(ac^2-ch^2):vero
28 ...ah=(ac^2)/ab:ignoto
29 ...ah=(ch^2)/hb:ignoto
30 ...hb=ab-ah:ignoto
31 ...hb=sqr(bc^2-ch^2):vero
32 ...hb=(bc^2)/ab:ignoto
33 ...hb=(ch^2)/ah:ignoto
34 ...p=ab+bc+ac:ignoto
35 ...a=(ab*ch)/2:ignoto
36 ...a=(bc*ac)/2:ignoto

```

Esempio c - Dopo aver introdotto le condizioni indicate in fondo all'articolo, ecco come si modificano i nodi precedenti.

— La memorizzazione delle regole avviene nell'array sr\$(). Il formato è il seguente: primo byte=numero dei nodi condizione; byte seguenti=codici ASCII corrispondenti ai numeri dei nodi condizione (<127:positivi, >127:negativi; ciò spiega il limite di 127 per i nodi); byte seguente=numero dei nodi conseguenza; byte seguenti=come prima, nodi sotto forma di codici ASCII.

— Gli array z() e z\$() e le variabili z e zz sono variabili di sistema utilizzate da tutto il programma, per evitare spreco di memoria.

— La variabile ee indica il numero di errore, fl lo stato di verità o falsità di una espressione.

— Particolare attenzione merita la varia-

bile sk\$; essa ha la funzione di stack, dove vengono memorizzati i valori intermedi delle variabili di sistema quando una subroutine chiama se stessa; in altre parole, essa permette la ricorsività indispensabile ad un programma del genere (non è molto bello, ma funziona).

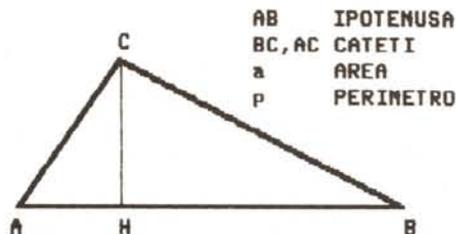
Negli esempi a, b, c, sono proposte le regole che definiscono un triangolo rettangolo.

Il metodo migliore per utilizzarlo è questo: bisogna inserire quali siano gli elementi noti e ignoti mediante «xx (non è noto)», dove xx è un elemento del triangolo riportato nella figura 2, poi listare i nodi per vedere quali siano state le formule impiegate.

Ad esempio, volendo conoscere AB in base a CH e BC, basterà inserire «ch è noto.» e «bc è noto.», oltre a «ab non è noto.», «ac non è noto.», «ah non è noto.», «hb non è noto.».

Chiedendo in seguito «ab è noto?» (in pratica poniamo ab come goal), avremo come risposta «ab è noto:vero», quindi il problema ha soluzione. Per conoscere le formule adoperate bisognerà listare i nodi; le formule utilizzate saranno seguite da «vero». Nel nostro caso avremo «hb=sqr(bc^2-ch^2):vero» e «ab=(bc^2)/hb:vero». Per ragioni di spazio non è possibile pubblicare il listato completo del programma, peraltro disponibile nei modi indicati.

Figura 2





Elenco del software disponibile su cassetta o minifloppy

Per ovviare alle difficoltà incontrate da molti lettori nella digitazione dei listati pubblicati nelle varie rubriche di software sulla rivista, MCmicrocomputer mette a disposizione i programmi più significativi direttamente su supporto magnetico. Riepiloghiamo qui sotto i programmi disponibili per le varie macchine, ricordando che i titoli non sono previsti per computer diversi da quelli indicati. Il numero della rivista su cui viene descritto ciascun programma è riportato nell'apposita colonna; consigliamo gli interessati di procurarsi i relativi numeri arretrati, eventualmente rivolgendosi al nostro Servizio Arretrati utilizzando il tagliando pubblicato in fondo alla rivista.

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Carlo Perrier 9, 00157 Roma.

| Codice | Titolo Programma | MC n. | Prezzo |
|-----------------|---------------------------|-------|--------|
| APPLE II | | | |
| DA2/06 | Miniset + LevaDOS | 37 | 15000 |
| DA2/07 | 27 programmi grafici | 38 | 30000 |
| DA2/08 | Adventure Editor | 38 | 15000 |
| DA2/09 | Animazione Funzioni | 42 | 15000 |
| DA2/12 | Routine Grafiche Estese | 44 | 15000 |
| DA2/13 | Scroll 300 linee | 46 | 15000 |
| DA2/14 | Assembler in Basic | 50 | 15000 |
| DA2/15 | G-Basic II | 53 | 15000 |
| DA2/16 | Disk Editor | 54 | 15000 |
| DA2/17 | Latino | 57 | 15000 |
| DA2/18 | Battaglia | 61 | 15000 |
| DA2/19 | Catalogo | 64 | 15000 |
| DA2/20 | Apple Puzzle II | 65 | 15000 |
| DA2/21 | Precisione Multipla | 66 | 15000 |
| DA2/22 | Sistema 2 + Toto 5.3 IIGS | 68 | 15000 |
| DA2/23 | Operazione Apokalypsis | 71 | 30000 |
| DA2/24 | Classifiche di Formula 1 | 72 | 15000 |
| DA2/25 | Programmabile RPN | 73 | 15000 |
| DA2/26 | Supercircle + Poligonale | 74 | 15000 |
| DA2/27 | Hard Copy OKI 83/A | 76 | 15000 |
| DA2/28 | ProDOS Utility | 77 | 15000 |
| DA2/29 | Modulo Base | 78 | 15000 |
| DA2/30 | List do | 79 | 15000 |
| DA2/31 | Bioritmi | 80 | 15000 |

COMMODORE AMIGA

| | | | |
|--------|----------------------------|----|-------|
| DAM/01 | F-15 | 63 | 15000 |
| DAM/02 | Gest. liste programmi | 64 | 15000 |
| DAM/03 | Studio di Funzioni | 66 | 15000 |
| DAM/04 | Math Pack | 68 | 15000 |
| DAM/05 | Redcode + Mars (Core Wars) | 68 | 15000 |
| DAM/06 | Life | 69 | 15000 |
| DAM/07 | Rubrica Telefonica | 70 | 15000 |
| DAM/08 | Piramidi | 70 | 15000 |
| DAM/09 | Regolazione dei colori | 71 | 15000 |
| DAM/10 | Analitica | 71 | 15000 |
| DAM/11 | Grafici | 72 | 15000 |
| DAM/12 | Traduttore | 73 | 15000 |
| DAM/13 | La Borsa | 74 | 15000 |
| DAM/14 | DMA Music Compiler | 74 | 15000 |
| DAM/15 | Poker | 78 | 15000 |
| DAM/16 | Programmi per il Copper | 79 | 15000 |
| DAM/17 | Mandelbrot mania | 81 | 15000 |

MS-DOS

| | | | |
|--------|--|----|-------|
| DMS/01 | Plotter + Morse | 67 | 15000 |
| DMS/02 | Meloe + Spawn | 68 | 15000 |
| DMS/03 | Pretty + Scritte scorrevoli + Compute | 69 | 15000 |
| DMS/04 | Emulatore CGA per Hercules | 70 | 15000 |
| DMS/05 | Turbo Directory | 71 | 15000 |
| DMS/06 | Math Tool S | 72 | 15000 |
| DMS/07 | Bioritmi + Routine | 72 | 15000 |
| DMS/08 | Salvavideo + Scritte scorrev. + PG151 | 73 | 15000 |
| DMS/09 | Optimizer + Indenter dBase III | 74 | 15000 |
| DMS/10 | Joystick Controller | 75 | 15000 |
| DMS/11 | BootSlow + SlowDown + Turbo Utility | 76 | 15000 |
| DMS/12 | Redcode + Mars (Core Wars) | 76 | 15000 |
| DMS/13 | Gestione Errori Critici Disco + PosCur | 77 | 15000 |
| DMS/14 | Finestre + Desk | 78 | 15000 |
| DMS/15 | General Manager | 78 | 15000 |
| DMS/16 | Tool 05 | 79 | 15000 |
| DMS/17 | Pulldown Menu + Retrace | 80 | 15000 |
| DMS/18 | Righe | 81 | 15000 |

ATARI ST

| | | | |
|--------|--------------------------------------|----|-------|
| DST/01 | Virus Killer | 74 | 15000 |
| DST/02 | Mandelbrot + Proiez. Ori. + Bilancio | 78 | 15000 |
| DST/03 | Diagrammi di Henon | 81 | 15000 |

COMMODORE 128

| | | | |
|--------|--------------|----|-------|
| D28/01 | MMCcalc | 53 | 15000 |
| D28/02 | Hardcopy 128 | 55 | 15000 |

| Codice | Titolo Programma | MC n. | Prezzo |
|--------|------------------------------------|-------|--------|
| D28/03 | Sheet II | 57 | 15000 |
| D28/04 | Star Quest | 58 | 15000 |
| D28/05 | Family Budget | 60 | 15000 |
| D28/06 | La Casa Stregata | 61 | 15000 |
| D28/07 | Strutture 80/33 | 63 | 15000 |
| D28/08 | Bas 80 V.2.0a | 64 | 15000 |
| D28/09 | Paint 80 1.0 | 65 | 15000 |
| D28/10 | Bas 80 V. 2.11 | 66 | 15000 |
| D28/11 | Calendario Perpetuo + Montecarlo | 67 | 15000 |
| D28/12 | Disegna Circuiti | 68 | 15000 |
| D28/13 | Mark's Data Base | 70 | 15000 |
| D28/14 | Label Disk + Disk Editor + Dem DOS | 71 | 15000 |
| D28/15 | Pulldown 128HR + Menu + Drawer | 72 | 15000 |
| D28/16 | Prospettive | 73 | 15000 |
| D28/17 | Char 80 V. 1.0 | 74 | 15000 |
| D28/18 | Italia 128 | 75 | 15000 |
| D28/19 | Super Sprite | 77 | 15000 |
| D28/20 | Ohelco | 80 | 15000 |
| D28/21 | Expert System Shell 128 | 81 | 15000 |

COMMODORE 64

| | | | |
|--------|---------------------------------------|--------|-------|
| D64/11 | Anno Domini | 57 | 15000 |
| D64/12 | The Disk Editor | 54/6/7 | 15000 |
| D64/13 | Boz's Adventure | 57 | 15000 |
| D64/14 | Link-64 | 57 | 15000 |
| D64/15 | New Char 2.2 | 58 | 15000 |
| D64/16 | Music 64 | 59 | 15000 |
| D64/17 | TRX-MEM | 59 | 15000 |
| D64/18 | WDS + WBasic | 60 | 15000 |
| D64/19 | Strange Basic + Dracula | 63 | 15000 |
| D64/20 | File Rescue | 64 | 15000 |
| D64/21 | La Casa | 64 | 15000 |
| D64/22 | Digital Voice | 65 | 15000 |
| D64/23 | Vita 3D | 65 | 15000 |
| D64/24 | Corso di Linguistica | 66 | 15000 |
| D64/25 | Archipus | 66 | 15000 |
| D64/26 | Math Pack Plus | 66 | 15000 |
| D64/27 | Scroll + Multitask + Classifica | 67 | 15000 |
| D64/28 | Calend. Perpetuo + Effetto Telecamera | 68 | 15000 |
| D64/29 | Listing Plus + Utility Data | 69 | 15000 |
| D64/31 | Trucchi e Routine per programmatori | 71 | 15000 |
| D64/32 | Flow-Chart + Flower's Love | 73 | 15000 |
| D64/33 | Sprite Editor | 76 | 15000 |
| D64/34 | Portfolio 64 + Elimina. bordi schermo | 77 | 15000 |
| D64/35 | Alfabeto Morse + Locare + Menu/Driver | 78 | 15000 |
| D64/36 | Schedario Gare | 80 | 15000 |
| D64/37 | Intonatore | 81 | 15000 |

MSX

| | | | |
|--------|-------------------------------------|----|-------|
| DMX/01 | Toto 13 | 60 | 15000 |
| DMX/02 | Painter | 62 | 15000 |
| DMX/03 | MSX Bank | 63 | 15000 |
| DMX/04 | Grafica 3D + Hard Copy | 65 | 15000 |
| DMX/05 | Easy Disk | 66 | 15000 |
| DMX/06 | Classifiche | 67 | 15000 |
| DMX/07 | Magic Paint | 67 | 15000 |
| DMX/08 | Autogest | 68 | 15000 |
| DMX/09 | Compilatore v. 1.01 | 69 | 15000 |
| DMX/10 | Diskmap | 70 | 15000 |
| DMX/11 | Mini dBase MSX | 71 | 15000 |
| DMX/12 | Grafica in Turbo Pascal | 72 | 15000 |
| DMX/13 | Math Pack Plus 3.20 | 73 | 15000 |
| DMX/14 | PCBCAD | 75 | 15000 |
| DMX/15 | Simple Desk | 76 | 15000 |
| DMX/16 | The MSX2 Super Print | 77 | 15000 |
| DMX/17 | Grafica in Turbo Pascal (Graph 1&2) | 77 | 15000 |
| DMX/18 | Hard Copy | 78 | 15000 |
| DMX/19 | HEXDUMP | 79 | 15000 |
| DMX/20 | Utilities in Turbo Pascal | 80 | 15000 |
| DMX/21 | dBase MSX Plus | 81 | 15000 |

Nota:
l'iniziale del codice è C per le cassette, D per i floppy.