di Raffaello De Masi

Elementi di Prolog

sesta parte

Lo sviluppo di un programma in Prolog

Prima di completare la nostra discussione sulle variabili è importante dare un'occhiata al modo in cui Prolog maneggia le variabili stesse; in questo linguaggio le variabili stesse assumono importanza e significato ben diverso da quello che accade nel resto dei linguaggi più convenzionali. L'esempio più immediato di differenza tra significato di variabile in Prolog ed in altri linguaggi, come, ad esempio Fortran o C, è rappresentato dal fatto che, nel nostro idioma, una variabile può assumere più di un significato e valore. In altri termini un programma scritto in Prolog può fornire più di una risposta ad una domanda. La cosa diviene sempre più facile man mano che la base di conoscenza su cui si articola lo sviluppo del programma diviene sempre più ampia e strutturata; ovviamente la cosa diviene ancora più complessa quando in un «goal» sono inserite variabili multiple.
La cosa migliore per fugare i sospetti e charire il tutto è vedere come Prolog scava ed investiga una base di conoscenza alla ricerca di valori soddisfacenti ad un «goal»

Il nostro primo esempio è un goal che non coinvolge variabili; vediamo rappresentata la relativa base di dati e la tecnica di «search» nella figura a. Se il goal è:

```
Goal: preferisce(carlo,cioccolata).
```

vediamo effettivamente cosa succede.

Occorre precisare innanzi tutto che Turbo Prolog, al contrario di quanto avviene in altre versioni del linguaggio, richiede che i predicati siano ordinati insieme; tanto per intenderci, se nella base di conoscenza esistono una serie di regole aventi come predicati «preferisce», «rifiuta», e «sceglie», occorre che tutti i predicati «rifiuta» siano inclusi in un gruppo, tutti quelli «preferisce» in un altro, e così via (pare che la versione 2.0, uscita in questi giorni [ottobre '88] superi questo problema). Questo è stato fatto, ad onor del vero, per facilità del compilatore, che è agevolato nella sua ricerca dal fatto che esso sa quando iniziare a cercare e quando smettere nel database, senza perdere ulteriore tem-

GOAL	passo	base di conoscenza	risultati
preferisce(carlo,cioccolata)?	1	preferisce(marco,cioccolata)	errato
preferisce(carlo,cioccolata)?	2	preferisce(corrado,cioccolata	errato
preferisce(carlo,cioccolata)?	3	preferisce(bo,cioccolata	errato
preferisce(carlo,cioccolata)?	4	preferisce(leo,cioccolata)	errato
preferisce(carlo,cioccolata)?	5	preferisce(carlo,cioccolata)	OK 1^ valore
preferisce(carlo,cioccolata)?	6	preferisce(carlo,cioccolata)	OK 2^ valore ; 1^ risultato valido
preferisce(carlo,cioccolata)?	7	preferisce(gabriella,crema)	errato
preferisce(carlo,cioccolata)?	8	preferisce(costantino,gin)	errato
preferisce(carlo,cioccolata)?	9	preferisce(massimo,noci)	errato
preferisce(carlo,cioccolata)?	n	preferisce(andrea,crema)	errato

Figura a - Ricerca di significato di un «goal» da «Dan Shafer - Asking More General Questions With Variables Sams ed. - Indianapolis» con modifiche.

Goal	base di conoscenza	risultati
beve(chi,birra)	beve(franco,aranciata)	errato
beve(chi,birra)	beve(alfredo,coca_cola)	errato
...		
...		
...		
beve(chi,birra)	beve(aldo,limonata)	errato
beve(chi,birra)	beve(marco,birra) » **primo marker**	OK ; "chi" istanziato a "marco"
beve(marco,birra) *(primo risultato intermedio)*		
beve(chi,birra)	beve(nicola,vino)	"chi" inistanziato
beve(chi,birra)	beve(alfonso,birra) » **secondo marker**	"chi" istanziato ad "alfonso"
beve(alfonso,birra) (secondo risultato intermedio)		
beve(chi,birra)	beve(andrea,cedrata)	errato
beve(chi,birra)	beve(corrado,acqua)	errato
		(fine della base di conoscenza)

Figura b - Analisi di un «goal» attraverso una variabile; ibidem.

po. Turbo Prolog guarda al primo predicato «preferisce», e cerca di comparare il primo argomento del predicato con il primo argomento del goal, alla ricerca dei valori che soddisfino al goal. Poiché, nel primo caso, «marco», predicato, non soddisfa la clausola (non coincide con «carlo»), il programma salta al successivo elemento di conoscenza. Il processo continua fino a che (passo 5) il primo predicato viene soddisfatto; a questo punto, contrariamente a quanto si penserebbe Turbo Prolog non passa al confronto col secondo argomento; viene, invece, in corrispondenza del primo predicato della quinta clausola, piazzato un marker e il processo prosegue fino alla fine delle clausole.

A questo punto il processo si ferma e Turbo Prolog pensa (? - N.d.R.): «Ho trovato il primo dei predicati che soddisfa alle richieste del goal; avanti col secondo!». Ricomincia daccapo cercando i marker che ha lasciato per strada e raggiunto il primo (e l'unico, in questo esempio) esegue una nuova comparazione e rilascia il valore «True»; la richiesta del goal è soddisfatta in quanto esistono, per la stessa clausola, due True, coincidenti nello stesso dominio, ambito e locazione dei predicati: ciononostante Turbo Prolog non esegue mai accomunamenti; per esso le soluzioni sono due; in altre parole il programma esegue due ricerche separate e pur comparando i risultati per rispondere effettivamente alle richieste del goal, tiene sempre distinte le soluzioni cui arriva.

In questo caso la soluzione era una ed una sola e si è giunti ad una sola risposta; vediamo invece cosa succede se il goal, attraverso l'uso di una variabile, chiede più soluzioni (possibili) ad una base di dati.

Il tutto è esemplificato nella figura b; immaginiamo, in base alla base di conoscenza rappresentata di imporre il goal:

Goal:beve(Chi,birra).

dove «Chi» è una variabile; lo scopo, come si sa, è di soddisfare, stavolta, ad un qualunque valore che, sostituito a «Chi», appunto variabile, riesce a completare la clausola esposta. Per la presenza nel predicato, di una variabile al primo posto non viene, ovviamente, eseguito alcun test sulla prima parte dei predicati della clausola stessa; viceversa tutti i secondi elementi vengono vagliati, uno per uno, e, ove mai venisse accertata la loro equivalenza col secondo predicato, fisso, della clausola, tutte

le occorrenze vengono segnate con un marker. In pratica, dopo un certo numero di tentativi, rappresentati in figura dai punti sospensivi [...], la variabile «Chi» viene istanziata, per la prima volta, a «marco»; il programma prosegue indisturbato marcando imperturbabilmente tutti i predicati soddisfacenti all'assunto, fino alla fine. Ogni volta che si soddisfano le condizioni volute, la variabile «Chi» viene istanziata, per procedere immediatamente dopo alla posizionatura di un marker; dopo di ciò la variabile viene instanziata prima di procedere alla successiva richiesta.

Ovviamente tutto questo lavorio è del tutto trasparente all'utente, che, anche grazie alla velocità del codice compilato, non si accorge per nulla di ciò che succede.

Nel frattempo, comunque, Prolog tiene conto di quante conclusioni positive ha raggiunto. Terminata la ricerca, come abbiamo già precedentemente visto, Turbo Prolog avverte l'utente del numero di conclusioni cui è arrivato e prosegue chiedendo un successivo goal.

Come si vede questo procedimento non è dei più semplici né dei più razionali, ma non esiste ancora oggi, in Prolog, un algoritmo meglio implementabile e compatibile con le esigenze e le possibilità dei compilatori oggi disponibili. È ovvio che la cosa diviene estremamente complessa quando, ad esempio, gli argomenti non sono più due, ma tre o quattro, e su di essi viene chiesta una ricerca eseguita con più variabili.

Proprio questa è una delle situazioni più critiche per il linguaggio ed un vero e proprio tallone d'achille di un idioma per certi versi straordinario; poiché non credo che esistano idiomi perfetti (come non esiste la donna più bella del mondo; ve li immaginate 2 miliardi di uomini tutti innamorati della stessa ragazza?), neppure Prolog è eccezionale e paga certe sue originalità e capacità potenziali in termini di complessità e tortuosità di azione del compilatore.

Cosa succede quando Turbo Prolog maneggia clausole contenenti più variabili, vale a dire che esiste, nel goal, uno statement a variabile multipla; non si fa altro che esaltare alla ennesima potenza la sfacchinata di cui dicevamo precedentemente. Per rendere però un poco più agevole il lavoro, stavolta (e non si capisce bene perché non lo abbia fatto anche prima) il linguaggio esegue una marcatura al primo soddisfacimento della prima variabile, quindi, secondo una struttura ad albero, prosegue sulla seconda variabile eseguendo i relativi controlli. Esaurita la ricerca ritorna al marker della prima variabile e prosegue alla ricerca della seconda, e così via.

Una precisazione, infine, in base a quanto avevamo lasciato sospeso in precedenza in una delle scorse puntate. Può essere talvolta necessario utilizzare lettere iniziali maiuscole senza per questo imporle come variabili.

La cosa è molto semplice e si riduce, all'atto pratico, all'uso di virgolette ["] racchiudenti la parola in questione; così se battiamo:

```
Goal:beve("Chi",birra).
```

il programma cercherà davvero la corrispondenza a [Chi] come se fosse un nome di persona.

Abbiamo finalmente terminato con le variabili, che come abbiamo visto, sono di tipo ed uso ben diverso da quelle esistenti in altri linguaggi. La prossima volta affronteremo una delle chiavi di volta di un linguaggio, come questo, dedicato alla AI; le connessioni logiche; a risentirci! MC

Goal	Base di conoscenza	risultati
beve(Chi,Che_cosa)	beve(franco,aranciata)	ricerca 1 valore
beve(franco,che_cosa)	beve(franco,aranciata)	accertato 1 valore
beve(franco,aranciata)	beve(franco,aranciata)	accertato 2 valore
beve(Chi,Che_cosa)	beve(alfredo,coca_cola)	ricerva valore 2.1
beve(alfredo,Che_cosa)	beve(alfredo,coca_cola)	accertato valore 2.1
beve(alfredo,coca_cola)	beve(alfredo,coca_cola)	accertato valore 2.2
...		...
beve(Chi,Che_cosa)	beve(aldo,limonata)	
beve(aldo,Che_cosa)	beve(aldo,limonata)	
beve(aldo,limonata)	beve(aldo,limonata)	
beve(Chi,Che_cosa)	beve(marco,birra)	
beve(marco,Che_cosa)	beve(marco,birra)	
beve(marco,birra)	beve(marco,birra)	*(e cosi via)*
beve(Chi,Che_cosa)	beve(nicola,vino)	
beve(nicola,Che_cosa)	beve(nicola,vino)	
beve(nicola,vino)	beve(nicola,vino)	
beve(Chi,Che_cosa)	beve(alfonso,birra)	
beve(alfonso,Che_cosa)	beve(alfonso,birra)	
beve(alfonso,birra)	beve(alfonso,birra)	
beve(Chi,Che_cosa)	beve(andrea,cedrata)	
beve(andrea,Che_cosa)	beve(andrea,cedrata)	
beve(andrea,cedrata)	beve(andrea,cedrata)	
beve(Chi,Che_cosa)	beve(corrado,acqua)	
beve(corrado,Che_cosa)	beve(corrado,acqua)	
beve(corrado,acqua)	beve(corrado,acqua)	

Figura c - Ricerca di variabile multipla; ibidem.

Quotha32
software & hardware

SOFTWARE
Originale, sigillato con garanzia ufficiale e possibilità di aggiornamento

SPREADSHEET/INTEGRATI	
Microsoft Excel 2.0 (It.)	750.000
Microsoft Works (It.)	295.000
Lotus 1-2-3 Rel. 2.01 (It.)	690.000
Lotus Symphony 2.0 (It.)	890.000
Borland Quattro	350.000
WORD PROCESSING	
Microsoft Word 4.0 (it.)	750.000
Lotus Manuscript (It.)	690.000

Microsoft Excel 2.0 it.
+
Microsoft Mouse 7

L. **850.000**

MicroPro WordStar Professional 4.0 (It.)	595.000
MicroPro WordStar 2000 Rel 3.0 (It.)	890.000
WordPerfect	990.000
Borland Sprint	350.000
DATABASE MANAGEMENT	
Ashton-Tate dBASE III Plus (It.)	890.000

NOVITA'
DA ASHON -TATE

FRAMEWORK III italiano	990.000
dBASE IV	1.090.000

Borland Paradox 2.0 Single (It.)	1.190.000
GRAFICA	
Microsoft Chart 3.0	550.000
Lotus Freelance Plus (It.)	690.000
V.C.N. Concorde 3.0 (It.)	1.290.000
Programmi CAD	Telefonare
DESKTOP PUBLISHING	
Aldus PageMaker 3. 0 (It.)	1.390.000
Rank-Xerox Ventura Publisher 1.2 (It.)	1.390.000
Rank-Xerox Ventura Publisher 2.0	1.490.000
Microsoft Pageview	70.000
Fonts Aggiuntivi per Ventura	Telefonare
LINGUAGGI	
Microsoft QuickBasic 4.0	150.000
Microsoft QuickC	160.000
Borland Turbo Pascal 5.0 (It.)	250.000
Borland Turbo C 2.0 (It.)	250.000

SPECIALE OS/2

Microsoft C Compiler 5.1	595.000
Microsoft Macro Assembler 4.1	250.000
Microsoft COBOL Compliler 3.0	1.150.000
Microsoft BASIC Compiler 6.0	390.000
Microsoft FORTRAN Compiler4.1	595.000
Microsoft Pascal Compiler 4.0	390.000
Microsoft OS/2 Programmer's Toolkit	495.000
Microsoft Multiplan	telefonare
Microsoft Word	telefonare
Borland Paradox OS/2	1.290.000

Il linguaggi Microsoft operano anche in ambiente MS-DOS

Borland Turbo Prolog 2.0	195.000
UTILITIES	
Microsoft Windows 286 (It.)	195.000
Microsoft Windows 386 (It.)	295.000
Microsoft Windows 2 Toolkit	650.000
Fastback Plus	295.000

HARDWARE
hardware originale con garanzia TOTALE di 1 anno

PERSONAL COMPUTER	
Olivetti M240, 640 KB RAM, 1FDU 5 1/4" 1 HDU 20 MB, monitor monoc., tastiera e cavo stampante	2.750.000
Altri modelli OLIVETTI	Telefonare
Portatili ZENITH	Telefonare

Portatile ZENITH
SupersPORT/20
640 KB RAM, 1 FDU 3,5",
1 HDU 20 MB,
monitor LCD retroilluminato

L. **3.850.000**

STAMPANTI	
PANASONIC KX-P1081	450.000
Altre stampanti PANASONIC a magazzino	telefonare

Stampanti NEC a 24 aghi
P2200, P6 Plus, P7 Plus, P9 XL
Prezzi fantastici e disponibilità a magazzino

Stampanti laser NEC	Telefonare
MONITOR	
NEC MultiSync GS monocromatico	499.000
NEC MultiSync II	1.190.000
NEC Monograph DTP Full Page	2.790.000
Rank-Xerox Full page Display	1.890.000

MONITOR RANK-XEROX FULL PAGE DISPLAY
con scheda grafica
+
VENTURA PUBLISHER 1.2 (It.)
L. 2.990.000

HARD DISK	
Hardcard PLUS 20 MB	1.250.000
Hardcard PLUS 40 MB	1.550.000
Passport PLUS 20 MB	950.000
SCHEDE GRAFICHE, UPGRADE ED ESPANSIONE	
Video Seven VEGA VGA	690.000
Orchid VGA	595.000
IIntel 386/AT 0 KB RAM	1.990.000

Intel Inboard386/PC
trasforma il tuo PC in un 386
con 1 MB di RAM installata
L. **1.790.000**

Microsoft MACH 20	750.000
Schede espansione RAM Above Board Intel	Telefonare
Altre schede	Telefonare
CHIP - MOUSE - VARIE	
Cop. Mat. Intel 80287-10 10 MHz	650.000
Altri Cop. Mat.	Telefonare
Microsoft Mouse 7 (NUOVO!) con Paintbrush	250.000

CONDIZIONI AGEVOLATE
PER ENTI PUBBLICI E SCUOLE
richiedete i nostri preventivi!

Microsoft Mouse 7 (NUOVO!) per PS/2	250.000
Mouse Logitech C7	180.000
Dischetti da 3,5" e 5,25" SONY, 3M, VERBATIM ed altri	Telefonare

Tutti i prezzi sono al netto di I.V.A.

TERMINI E CONDIZIONI DI VENDITA: Tutti i prezzi sono al netto di I.V.A. - Pagamento in contrassegno con assegno circolare NT intestato a Quotha 32 s.r.l. o contante. - Sconto del 3% per pagamento anticipato. - Ci riserviamo di accettare ordini di importo inferiore a L. 300.000. - La merce si intende salvo il venduto. - Ulteriori sconti per quantità. - La presente offerta è valida sino al 15 gennaio 1989 e annulla e sostituisce ogni nostra precedente offerta.

SPEDIZIONI GRATUITE IN TUTTA ITALIA

per ordini o informazioni
telefonare allo

055 - 23.20.240

oppure spedire il tagliando compilato a:

Quotha32 s.r.l.

Via Accursio, 2 - 50125 FIRENZE
Telefax 055-2280674

Ragione Sociale: _____

Nominativo: _____ Qualifica: _____

Indirizzo: _____

C.A.P.: _____ Città: _____ Prov.: _____

Tel.: _____ Telefax: _____

☐ Desidero essere contattato telefonicamente da un vostro funzionario commerciale

☐ Desidero ricevere informazioni su: _____

☐ Desidero ricevere il vostro listino completo

☐ Inseritemi nella vostra mailing list