Questa volta restiamo nel vero e proprio campo dell'informatica con due lavori che sono per un certo verso agli antipodi di questa scienza: l'Intelligenza Artificiale e le mappe di Karnaugh.
Le mappe di Karnaugh permettono di semplificare un circuito logico in modo da utilizzare il minor numero di componenti possibile e quindi risparmiando denaro.
L'Intelligenza Artificiale ci permetterà un giorno, forse non troppo lontano, di avere dei computer che, ad altissima velocità, compiono esattamente gli stessi errori di un essere umano...!

È disponibile, presso la redazione, il disco con i programmi presentati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 311.

Sistema Esperto AIES

Artificial Intelligence: the Expert System

di Salvatore Ruggieri - Matino (LE)

AIES nasce come «esercitazione» in ambiente Prolog delle tecniche di IA. Esso è un SE implementato per lavorare nel campo della sistematica informatica.

In particolare AIES è stato progettato come gestore di informazioni (DB) in grado di «dialogare» in modo naturale con l'operatore. Gli applicativi che sono attualmente in commercio offrono una serie di opzioni (organizzate in menu e sottomenu) per cui la «libertà di scelta» dell'utente è limitata alle feature offerte dal programma. In un programma convenzionale egli è «imprigionato» nelle opzioni della procedura: nulla di ciò che non è stato previsto dall'analista potrà mai essere offerto dal sistema. Infine, il modo di accedere alle opzioni di menu, per quanto intuitivo, è molto lontano dall'uso naturale di «chiedere» ciò che si vuole e, nei limiti del possibile, di ottenere una risposta.

AIES è organizzato in modo da avere la conoscenza dinamica (i fatti della base dati) registrata su memoria di massa. Il database è un file con estensione ".DB" cui il programma accede tramite una intuitiva interfaccia accessibile da menu principale o da «interrogazione della banca dati». Letta la base di conoscenza, è possibile analizzarne ed ampliarne la struttura aggiornando la banca dati, l'insieme dei sinonimi delle parole chiave e delle parole che esprimono massimo e minimo. Sono infatti queste le strutture principali contenute su database:

```
persona (symbol, anagrafica, symbol)/
persona (codice, anag, note)
sinonimi (symbol, list) /
sinonimo (chiave, lista)
min (symbol) /
parole che esprimono minimo
max (symbol) /
parole che esprimono massimo
```

I fatti «persona» contengono codice, anagrafica (tipo complesso costituito da nome, cognome, sesso, età e professione) e note riferite ad un elemento (cliente, fornitore, alunno, cittadino) del database. Il file sorgente principale (AIES.pro) gestisce il menu principale, scanner, filtro, interpretazione e valutazione della domanda, motore inferenziale per il calcolo della risposta.

Le procedure specifiche per la gestione della finestra dei menu è devoluta all'include file MENU.pro (ispirato da modulo omonimo fornito assieme al compilatore), mentre le regole di uso generale e le regole per la visione e l'aggiornamento del linguaggio sono registrate, rispettivamente, nei file PROC.pro e LANG.pro. Nella interrogazione della banca dati, l'operatore può inserire al prompt "DOMANDA:" richieste del tipo:

```
Visualizza i codici registrati nel DB
Voglio tutte le informazioni sul nome "Salvatore"
Esiste il cognome "De Jaco"?
Quale è il codice del nome "Anna Lisa"?
Stampa i nomi ed i cognomi del database.
```

Elaborazione del Linguaggio Naturale: gli analizzatori sintattici

AIES è un SE in grado di «interpretare» determinati tipi di domande proposte in lingua italiana. Interpretazione ed esposizione di risposte costituiscono ciò che in IA viene indicato con il nome di Elaborazione del Linguaggio Naturale (ELN).

Nell'ELN il nucleo principale del sistema (che, in teoria, potrebbe essere sviluppato separatamente dal resto del SE) è costituito dall'Analizzatore Sintattico (AS), da quella parte del codice che ha il compito di analizzare le parole (scanner) e le strutture (parser) di una frase. Gli analizzatori sintattici «tradizionali» sono l'analizzatore a stati finiti, l'analizzatore non contestuale a discesa ricorsiva e l'analizzatore ad eliminazione di rumore.

Esistono sostanzialmente due approcci opposti per elaborare il Linguaggio Naturale. Il primo tenta effettivamente di utilizzare tutte le informazioni contenute in una frase, esattamente come farebbe un essere umano, cercando di mettere il calcolatore in condizione di gestire una conversazione (AS a stati finiti). L'altro, invece, tenta di fare in modo che il calcolatore accetti comandi in Linguaggio Naturale, ma estragga soltanto quelle informazioni che sono utili per il comando stesso (AS non contestuale e AS ad eliminazione di rumore). Una delle principali difficoltà che si incontrano nel realizzare dei sistemi basati sull'ELN è rappresentata dal come tener conto della complessità e della flessibilità del Lin-

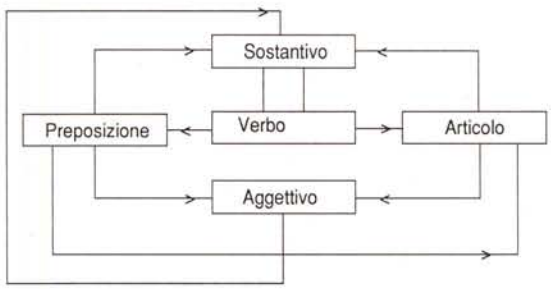
guaggio Naturale. Quando si implementa un programma ELN si è tentati di restringere a un piccolo insieme i tipi di frasi che il programma sarà in grado di comprendere. È evidente, però, che se ad un restringimento della grammatica corrisponde uno snellimento del codice, nel contempo vi corrisponde anche una perdita di «naturalezza» da parte dell'AS.

Un analizzatore sintattico a stati finiti utilizza l'informazione sullo stato corrente per stabilire a quale categoria dovrà appartenere la successiva parola della frase. Per implementare un automa del tipo rappresentato in figura 1 vengono definite due basi di dati: una contenente il dizionario delle parole note al sistema, e l'altra contenente lo stato corrente del sistema. Le regole del motore inferenziale avranno il compito di mettere in relazione lo stato attuale con la categoria della prossima parola. Questo è causa di una immane proliferazione di conoscenze (di parole) e di regole (nell'esempio ne richiede una dozzina), le quali divengono molto complesse quanto più si voglia «naturalizzare» il sistema. Per questo motivo, gli analizzatori sintattici a stati finiti sono utilizzati soltanto in quelle situazioni in cui è richiesto un ristretto sottoinsieme della grammatica (ad esempio vengono usati nella progettazione dei sistemi operativi).

Negli analizzatori non contestuali viene utilizzato un approccio del tutto differente dagli analizzatori appena descritti. Un approccio rivolto a vedere una frase costituita da strutture e non da semplici sequenze di parole. Le regole che stabiliscono come le varie parti possono unirsi sono chiamate regole di produzione della grammatica. Un AS non contestuale utilizza queste regole di produzione per analizzare una frase. Se vengono utilizzate le regole ricorsive per analizzare tutti i nodi dell'albero costruito dalle regole di produzione, allora l'AS viene denominato «a discesa ricorsiva». Il problema maggiore di questi analizzatori (molto usati nell'analisi sintattica dei linguaggi di programmazione) è quello di non poter descrivere tutta la complessa grammatica di una lingua reale.

Infine, presentiamo l'analizzatore sintattico ad eliminazione di rumore implementato in AIES. Alcune applicazioni sono centrate su poche parole chiave contenute in una frase e non si occupano affatto del resto (rumore). Negli AS ad eliminazione di rumore tutte le parole non note o non richieste sono trattate semplicemente come rumore e vengono eliminate. In genere tutte le frasi devono seguire un formato rigido (ad esempio: comando <modificatore> <nome> <operatore> <valore> può interpretare frasi del tipo: stampa tutti i codici maggiori di 5), ma è possibile, così come in AIES, predisporre più formati in modo da rendere più naturale il colloquio. Il principale svantaggio di questo

Automa a stati finiti associato ad un sottoinsieme della grammatica italiana



Un automa a stati finiti è un grafo orientato che evidenzia le transizioni possibili tra stati. Nella figura, ad esempio, un verbo può essere seguito da una Preposizione, da un Articolo o da un Sostantivo, mentre viene sempre preceduto da un Sostantivo.
L'automa presentato descrive i seguenti tipi di frase:

La mamma è andata al mercato.
La "Gioconda" è un bellissimo dipinto realizzato da un grande pittore.
La bella signora fece scacco matto.

Figura 1

Figura 2

Regole di produzione di un sottoinsieme della grammatica italiana

FRASE ---> SN + SV

SN ---> Articolo + Sostantivo
SN ---> Articolo + Aggettivo + Sostantivo
SN ---> Preposizione + SN

SV ---> Verbo + SN
SV ---> Verbo

Nell'esempio una frase è costituita da un Sintagma Nominale e da un Sintagma Verbale. Il sintagma nominale può essere costituito da un articolo e da un sostantivo, oppure da un articolo, un aggettivo ed un sostantivo oppure da una preposizione ed un sintagma nominale (definizione ricorsiva). Il sintagma verbale può essere costituito da un verbo ed un sintagma nominale oppure da un solo verbo.

Si prega il Sig. Marco Arcioni autore del programma Musiclab di mettersi in contatto con la redazione per comunicare i propri dati anagrafici.

tipo di analizzatori è quello di interpretare frasi senza un apparente significato per l'operatore, ma che soddisfano uno stato interno (in AIES, ad esempio, è lecito chiedere: il codice ha ammazzato il piccolo gallo del municipio di "Salvatore" ed ottenere in risposta tutti i codici degli elementi che hanno nome minore di "Salvatore").

Le regole dell'analizzatore sono quelle rivolte all'analisi delle parole (scanner) e al filtro, cioè all'eliminazione del rumore. Le regole «scan» e «scanvalore» implementano lo scanner: esse si occupano di dividere la frase nei suoi componenti essenziali (token), raggruppando tutto ciò che è compreso tra doppi apici in un solo token. Questo è dovuto al fatto che essendo i campi del DB molto più lunghi di una singola parola occorre evitare di confonderne il contenuto con parole chiave o con "rumore". La regola scan esamina la frase separando token da token ed invocando scanvalore quando incontra i doppi apici. All'invocazione, dovuta all'inizio di un token costituito da un campo valore, la regola scanvalore analizza la frase restante fino ai nuovi apici o fino al termine e restituendo tutto ciò che è analizzato come unico token che viene accodato alla lista finale risultante. Molto importante è anche il filtro, implementato dalle regole "filter". Queste si occupano di recepire campi valori (riconosciuti dai doppi apici), parole che esprimono massimo o minimo (codificandole con i simboli ">" e "<") e campi chiave (e sinonimi trasformati in campi chiave), tralasciando tutto il resto.

Il motore inferenziale, ossia le regole di interpretazione della domanda e di generazione della risposta sono implementate principalmente nelle clausole "valuta" (ed in altre che vedremo in seguito) le quali sono invocate con il primo argomento costituito dalla lista di token ottenuta dall'AS ed il secondo argomento libero. A quest'ultimo viene assegnata una lista di symbol contenente le risposte. Il sistema di ELN si preoccupa di inviare allo standard output (regola "num_risposte" contenute in PROC.pro) le eventuali risposte in formato predefinito (nella clausola di uso generale "scrivi_lista" contenuta in PROC.pro. Essa stampa gli elementi della lista su due colonne numerando progressivamente ogni risposta).

Motore Inferenziale: ai limiti della ragione umana?

Il Motore Inferenziale (MI) rappresenta il cuore di un SE. Il modo in cui un MI è organizzato e strutturato dipende sia dalla natura stessa del problema, sia da come si desidera strutturare e costruire il motore stesso.

AIES implementa un Motore Inferenziale in diverse clausole. La principale è «valuta», la quale decide il tipo di domanda fatta dall'utente e richiama le regole costruite per l'elaborazione della risposta. Se la lista di token data in input è vuota (ossia la stringa iniziale è vuota o è costituita solo da rumore) la ricerca di soluzioni fallisce automaticamente senza generare alcuna risposta.

Esistono, altrimenti quattro modi fondamentali di porre una domanda: la richiesta preceduta da un verbo (imperativo), la richiesta di tutti i valori di una lista di campi chiave, dei valori di un campo chiave dell'elemento identificato mediante un diverso campo valore ed, infine, la richiesta di altri tipi di domande. Tutte le clausole del MI controllano, innanzitutto, la correttezza degli argomenti, ossia la coerenza dello stato richiesto con le informazioni contenute nella lista di token.

Quando la lista di token ha come primo argomento un verbo (riconosciuto dall'invocazione "è_un_verbo(X)", come membro della lista di tutti i verbi conosciuti) vengono invocate le regole «esegui». I verbi conosciuti sono DEL (cancellazione di uno o più elementi dal DB), WRITE (redirezione alla stampante dello standard output), HELP (visualizzazione dell'Help), NEW (aggiornamento del DB), LOAD e SAVE (lettura e scrittura su memoria di massa), LOOK (visione e aggiornamento linguaggio), OS (richiamo dell'interprete di comandi DOS) e INFO (richiesta di tutte le informazioni su uno o più record).

Il verbo DEL permette di annullare un elemento (se esiste), o una lista di elementi che hanno un campo in comune (DOMANDA: distruggi tutti i codici che hanno cognome "Gemma" e tutti quelli che si chiamano "Andrea"): esso, una volta individuati i codici, direttamente o per mezzo della regola "più_valori", richiama la regola "cancella" che ritrae (retract) i codici dalla base della conoscenza. La regola "più_valori" ritorna una lista contenente tutti i valori assunti dai campi indicati nella lista di valori passata come primo parametro. Il verbo WRITE redireziona l'output verso la stampante (LPT1). In effetti, la regola si limita a richiamare il MI per la valutazione del resto della frase (DOMANDA: stampa tutti i codici) e la parte dell'ELN destinata alla stampa della risposta.

Il verbo INFO seleziona una lista di elementi (che hanno in comune un determinato campo come in DOMANDA: informazioni sui nomi "Giuliano") ritornando nella lista in output tutti i campi contenuti nei record individuati. Gli altri verbi sono una duplicazione delle opzioni del menu principale che permettono di richiamare qualsiasi feature del programma dall'ambiente di interrogazione del DB. Se la lista di token in input è costituita solo da campi chiave "(solo_campi)" la domanda viene interpretata come la richiesta di tutti i valori assunti dai campi individuati (DOMANDA: visualizza i nomi, i cognomi e le ddn). La lista di liste risultante viene trasformata dalla regola di uso generale "lol_list" (list of list to list) in una lista unica in cui gli elementi sono costituiti dalla successione dei campi chiave comuni ad un elemento e non da tutti i valori assunti da un campo chiave (ciò per ottenere un formato più leggibile).

Ad esempio, alla richiesta DOMANDA: stampa i nomi ed i cognomi, verranno stampati nella prima colonna tutti i nomi e nella seconda tutti i cognomi (e non prima tutti i nomi e poi tutti i cognomi), in modo da avere su una riga le informazioni riguardanti uno stesso elemento.

Un altro tipo di domanda gestita dalle clausole «valuta» consiste nella richiesta di uno o più campi di un elemento individuato da un campo valore (DOMANDA: voglio il nome e il cognome del codice "RS"). Innanzitutto, questa regola si assicura che la lista di token soddisfi lo stato corrente (ossia sia una lista di solo campi chiave fatta eccezione dell'ultimo elemento), poi richiama la regola già vista "più_valori".

L'ultimo tipo di domande gestito dalle clausole principali del MI è costituito da un insieme di formati rigidi per la richiesta delle informazioni. Questi formati sono analizzati nelle clausole «risposta». Tali clausole interpretano la lista di token in input a seconda del numero di elementi di cui è costituita. Una domanda in cui siano riconoscibili due soli token viene interpretata come la richiesta di verificare se nel DB esiste un campo che abbia un dato valore, quando il primo elemento è un campo chiave ed il secondo un campo valore (DOMANDA: È vero che esiste il codice "RS"?), oppure se un dato valore è contenuto in un determinato campo (DOMANDA: È vero che "Salvatore" è uno dei nomi contenuti nel DB?).

Se gli elementi della lista sono tre, le possibili combinazioni sono diverse, ma quelle interpretate (o interpretabili, cioè che hanno senso) sono solo quattro. Se gli elementi non contengono confronti max/min, la domanda viene interpretata come la richiesta del contenuto di un campo di un elemento individuato per mezzo del contenuto di un altro campo (DOMANDA: Qual è l'elemento che ha

"Salvatore" per nome. DOMANDA: Qual è il nome dell'elemento "RS").

I confronti sono invece richiesti per individuare campi maggiori o minori di un determinato valore (DOMANDA: Visualizza i nomi maggiori di "Giovanni") o campi di elementi che soddisfano una richiesta di confronto (DOMANDA: stampa il codice dei nomi maggiori di "Giovanni". A questa richiesta, il SE non risponderà con i nomi maggiori di "Giovanni", come nel caso precedente, ma con i codici di tali nomi). I confronti vengono effettuati dalle regole "minmax_val" (campi dei valori maggiori/minori di X) e "valuta_lista" (valori maggiori/minori di X). Per accedere effettivamente alla base della conoscenza, vengono invocate le regole "ent" e "DB".

Interfaccia utente di AIES

L'interfaccia utente di AIES è molto intuitiva e potente grazie alle possibilità offerte dal Turbo Prolog nella gestione delle finestre. Ogni finestra attiva può essere dimensionata a piacere mediante la pressione dei tasti Shift-F10 e, successivamente, dei tasti cursore. Nell'introduzione di un nuovo record nel DB, ad esempio, sono disponibili operazioni di I/O molto sofisticate (anche se non prevedono l'input controllato. Ma, essendo i campi di lunghezza variabile, ciò non è nemmeno necessario): le note sono immesse in una finestra che è del tutto uguale al classico editor like-WordStar dei prodotti Borland e che rende disponibili 64Kbyte per l'editazione di stringhe con avanzate funzioni di cut&paste, lettura, scrittura, ricerca e sostituzione.

Mappe di Karnaugh

CD \ AB	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	0	0	1	0
10	1	0	0	1

Usa i tasti cursore per muoverti
Return per cambiare valore
ESC per calcolare la soluzione
Q per uscire

$Y=\overline{B}\overline{C}\overline{D}+\overline{A}\overline{B}\overline{D}+A\overline{B}\overline{C}\overline{D}+\overline{A}B\overline{C}\overline{D}+A\overline{B}C\overline{D}+ABCD$

$Y=(A+\overline{C}+\overline{D})*(\overline{B}+\overline{C}+D)*(B+\overline{C}+\overline{D})*(A+B+\overline{D})*(A+\overline{B}+D)*(\overline{A}+B+C+D)*(\overline{A}+\overline{B}+C+\overline{D})$

La maschera di inserimento dati; dalla equazione SOP (somme di prodotti) si ricavano (a mano purtroppo) gli «1» da inserire nella matrice. Il programma stampa poi sia la soluzione SOP che quella POS (prodotti di somme) e i corrispondenti schemi elettrici.

Mappk3 e Mappk4

di Enrico Cremonini - Bologna

Mappk3 e Mappk4 sono due programmi da me ideati per risolvere le mappe di Karnaugh a 3 e a 4 variabili tanto usate per la risoluzione di reti logiche.

Queste mappe hanno quell'utile funzione di trovare la funzione matematica di una determinata rete elettronica logica di cui non si conoscono i componenti, ma solo i dati di ingresso e i dati in uscita.

Questi miei programmi non solo restituiscono la formula matematica, ma anche il circuito elettronico (la scheda video utilizzata dai programmi per visualizzarlo è la CGA).

Il programma trova sia la formula e il circuito ottenuti raccogliendo i mintermini e sia la formula e il circuito ottenuti raccogliendo i maxtermini, cioè alla fine avremo un circuito che utilizza più AND che conferiscono in un'unica OR e un circuito che usa più OR che vengono collegate ad un'unica AND.

Il programma è stato realizzato con il Quick Basic 4.5 e nel dischetto è presente anche il file in formato EXE autonomi, cioè che non richiedono il BRUN45 fornito con il QB.

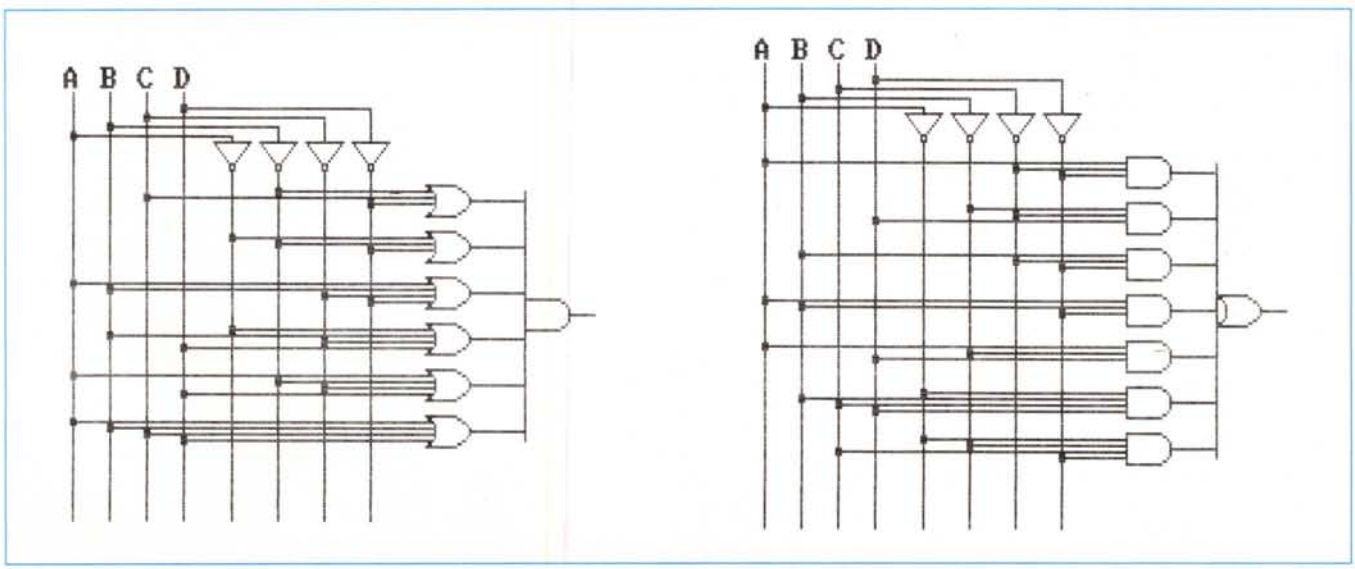
Il programma è molto facile da usare.

Con i tasti cursore ci si può muovere nella tabella. Premendo RETURN si cambia il valore della casella da 0 a 1 e da 1 a 0.

Dopo aver completato la tabella sarà sufficiente premere il tasto ESC e il computer dopo pochi secondi vi fornirà le due formule, dopo di che sarà sufficiente premere un tasto per avere sul video il circuito elettronico inerente alla prima formula e premere un tasto per una seconda volta per avere il circuito inerente alla seconda formula.

Per uscire dal programma e tornare al DOS è sufficiente premere la lettera Q.

MC



Schema SOP (And di Or) e POS (Or di And) della soluzione, i due circuiti hanno la stessa tavola della verità.