

I Sistemi Esperti

La volta scorsa, per mancanza di spazio, abbiamo lasciato in sospeso il discorso sulle caratteristiche di ROSIE, il linguaggio di programmazione per Sistemi Esperti che più di tutti ha imposto le direttive per tutto ciò che sarebbe poi venuto in seguito. A differenza dei classici linguaggi, come abbiamo visto in precedenza, le strutture fondamentali di ROSIE sono rappresentate da una serie di elementi di base, che già in parte abbiamo illustrato; in questa puntata concluderemo l'argomento parlando di un blocco di oggetti, appunto, che più di tutti sono caratteristici dell'ambiente di AI

ROSIE usa i suoi database per gli stessi scopi di ogni linguaggio di AI, quello di conservare notizie, «fatti», e risultati intermedi nel processo di calcolo. La componente fondamentale di un database è rappresentato soprattutto da proposizioni, che sono raccolte attraverso l'uso di un sistema logico a tre valori; una proposizione, in un database, può assumere, indipendentemente dal suo contenuto, un valore [vero], o [falso] o non contenere nulla del tutto (vale a dire che resta nel campo dell'indeterminatezza). Questa articolazione ha i suoi vantaggi (essa viene detta in gergo tecnico «open-world assumption»), in quanto implica che Rosie continua a tenere il suo valore di indeterminatezza fino a che non passerà a valore di vero o falso, quando in possesso di una messe sufficiente di dati.

Con queste premesse l'utente può manipolare il database o asserendo proposizioni del tutto nuove, o accedendo a valori particolari di elementi già definiti. Così avremo le seguenti possibili proposizioni (del tutto legali, vista la

semplicissima sintassi utilizzata da ROSIE):

- Affermo che Smith&Wesson è un revolver
- Stabilisco che i revolver sono Smith-&Wesson
- Affermo che Smith&Wesson non è una sciabola

Un difetto di ROSIE è che non contiene grandi tool per manipolare contraddizioni. Se si esegue una asserzione del tipo «Walther non è una automatica» si verifica una contraddizione semplice; bene, solo la seconda (comunque l'ultima) delle affermazioni viene conservata nel database (o comunque è quella significativa). Un leggero vantaggio è dato dal fatto che ROSIE conserva nel database la proposizione, sotto forma di una relazione di base collegata ad un valore di verità (una proposizione è dimostrabile come vera, falsa o semplicemente non esiste nel database).

Per modificare o eliminare un valore dal database occorre porre un minimo di attenzione; inserire il contrario di una proposizione non la esclude o cancella dal database; occorre «negare» la proposizione stessa secondo la formula:

- nego che Colt sia una sciabola
- Ovviamente è possibile inserire proposizioni condizionali del tipo:
- se Colt è un revolver...

Come in tutta l'area della intelligenza artificiale e dei sistemi esperti, il database gioca un ruolo centrale in ROSIE. Ogni affermazione si traduce automaticamente in un inserimento nella base di dati, e, comunque, la maggior parte delle condizioni richiedono accesso e test del contenuto del database. Comunque, anche per ragioni di praticità, la maggior parte dei sistemi esperti scritti con ROSIE contiene più di un singolo database che possono essere anche articolati tra loro. Comunque le nuove tecniche di accesso alle memorie di massa, le nuove tecnologie hardware e la più grande disponibilità di memoria centrale sono state le cause di un continuo aggiornamento e di nuove release di ROSIE. Il file principale di base di dati di Rosie è denominato «global»; comunque il linguaggio possiede una gran messe di comandi e istruzioni, finalizzate proprio alla manipolazione di basi di dati, che permette di collegare, modula-

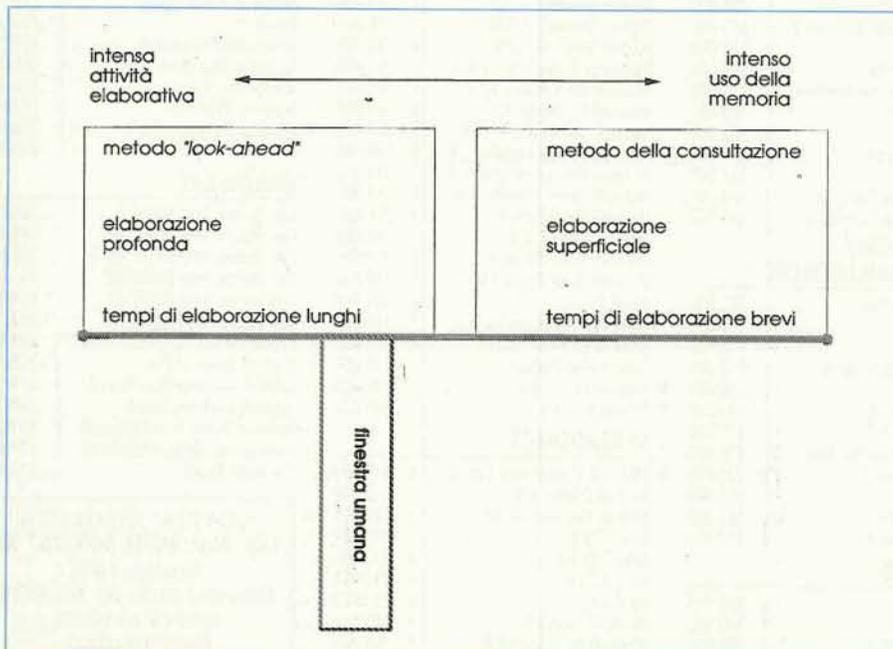


Figura a - Tecnica di elaborazione «look-ahead» contro tecnica delle tabelle da Michie & Johnston, «Intelligenza artificiale e futuro dell'uomo», Milano, 1989.

rizzare, interconnettere, rendere relazionali, parti di diversi database. I più grossi sforzi degli implementatori sono stati comunque rivolti a rendere, per quanto possibile semplici e sovente trasparenti le tecniche di connessione e manipolazioni di queste strutture, che, quando assumono una mole elevata, divengono sempre più lente e complesse da manovrare.

I database condivisi

Pensavate di aver raggiunto risultati complessi? Pura illusione. Il passo successivo nella manipolazione delle basi di dati è rappresentato dai database condivisi.

Premesso che l'utility dei database condivisi esiste solo per la versione di ROSIE dedicata allo XEXOX SIP 1100 (non per la versione Interlisp dedicata alle macchine VAX), occorre precisare che si applica a una speciale classe di sistemi esperti, cosiddetti «distribuiti» o, anche «agenti euristici distribuiti» (a tal riguardo esiste una trattazione notevolmente articolata in Sowizral H. A. «Experiences with Distributed Heuristic Agents in ROSIE», Proceedings of The IEEE Conference on Systems, Man, and Cybernetics, Bombay, 1983), che rappresenta una classe di SE che sono capaci di consultarsi tra loro per risolvere problemi.

Una base di dati condivisa funziona come un normale database; essa, per sua natura conserva informazioni, organizzate però sotto forma di relazioni; tramite questa base di dati è possibile manipolare affermazioni, negazioni, e test condizionali. Il loro uso può essere inoltre disattivato. Ma la vera, fondamentale caratteristica è che essa lega tra loro gli agenti euristici che condivide. Il suo principio di funzionamento è conforme al progetto teorico di Erman e Lesser (Erman L.D. e Lesser V.R. «A multileve Organization for Problem Solving Using Many Diverse Cooperating Sources of Knowledge» Proceedings of the fourth international Joint Conference of artificial intelligence, Tbilisi, URSS, 1975) e, concettualmente, è piuttosto semplice; operatori e agenti di diverso tipo possono concorrere a modificare e alterare un database comune, con la

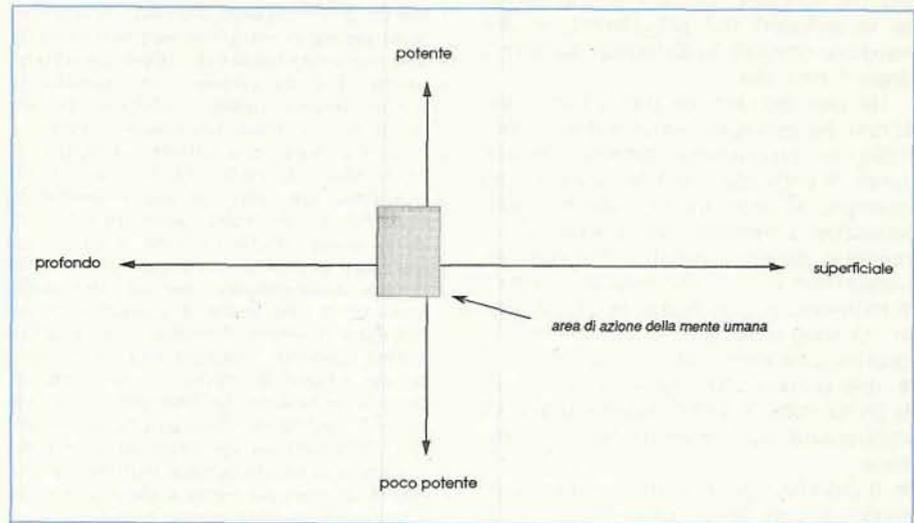


Figura b - Le direzioni di sviluppo delle due diverse tecniche (ibidem).

particolarità di poter selezionare gli operatori abilitati.

Rosie assicura che ogni agente che condivide un database abbia esattamente la stessa informazione di ogni altro agente condividente la stessa base; ovviamente un cambio eseguito da un operatore è immediatamente disponibile per gli altri. Poiché ROSIE non costruisce sistemi in tempo reale, non può garantire che ogni agente abbia identiche informazioni in ogni momento, ma assicura che ogni cooperante al database condiviso abbia a disposizione gli stessi cambi nello stesso ordine degli altri agenti. Inoltre, per motivi operativi, tutti gli agenti non hanno a disposizione lo stesso cambio contemporaneamente, né che il tempo passato tra due cambi allo stesso agente è eguale per tutti gli agenti stessi.

La tecnica dei database condivisi è particolarmente utile in certe operazioni di ingegneria della conoscenza e, quando utilizzate insieme ai demoni (di cui parleremo tra poco) permette di costruire strutture di controllo particolarmente complicate con poche difficoltà.

I set di regole

Una delle caratteristiche più efficaci che distinguono ROSIE da altri linguaggi dello stesso tipo è la facilità di costruire set di sottoregole. Un utente ROSIE può controllare l'affidabilità di un blocco di regole logicamente o non raggruppate, inserendole in un subset chiamato appunto «ruleset». Questi blocchi possono essere di tre tipi: procedure, generatori e predicati. Ognuno serve a

differenti funzioni, ognuno è invocato diversamente, ognuno infine, da risultati diversi.

Un esempio di procedura è il seguente:

- Per sparare con una Smith&Wesson
 - aprire il tamburo della pistola
 - inserire le cartucce del calibro appropriato
 - richiudere il tamburo
- END

La procedura ha il compito di aggiornare il database quando viene impartito un ordine del genere:

- sparare con la Smith&Wesson
- Un generatore permette di produrre sia un valore singolo che tutta una serie di valori appartenenti alla stessa classe. Un esempio di generatore può essere:
- sparare con una Smith&Wesson mod 14 calibro 38
 - sparare con una Smith&Wesson mod 29 calibro 44M
- e, nel momento in cui si desse l'ordine:
- mostra le Smith&Wesson che sparano.

Il sistema di funzionamento è ancora basato sul binomio vero-falso, e, ancora, sulla possibilità di manipolare valori sconosciuti.

I Demoni

La parola demoni evoca sensazioni e pensieri terrifici; in gergo di AI essi hanno una speciale funzione; si tratta infatti di programmi o blocchi di regole che restano inattivi (dormienti, ancora in gergo) fintanto che si presentano condizioni particolari, nel qual caso eseguono operazioni particolari e specifiche operazioni. De-

moni possono essere attivati, ad esempio, per eseguire tracing e debug durante lo sviluppo dei programmi, e per eseguire controlli su database durante o dopo il loro uso.

Un uso dei demoni può essere realizzato, ad esempio, per prendere il controllo del programma durante l'esecuzione di particolari routine, quando, ad esempio, si verificano condizioni caratteristiche. I demoni non possono, comunque, essere invocati in maniera generalizzata; essi vanno invocati in precisi momenti, e, con Rosie, le circostanze in cui sono utilizzabili si suddividono in quattro categorie, così riassumibili:

- una proposizione viene asserita per la prima volta, o viene negata, o ancora confrontata col contenuto di un database
- Il database genera un elemento partendo da una descrizione.
- ogni elemento prodotto da un generatore genera a sua volta una descrizione
- viene usato un verbo imperativo.

Casi di invocazione di demoni possono essere ritrovati negli esempi precedenti. Un demone può essere anche usato per eseguire valutazioni e operazioni all'interno di un blocco di programma; in questo caso può agire attraverso un «return» che fa uscire incondizionatamente dal blocco o attraverso un «continue», che invece permette il completamento della operazione in corso di esecuzione.

I monitor

Qualche parola, infine sull'ultima struttura; Rosie associa ogni blocco di regole con un monitor, che, intuitivamente, controlla lo sviluppo e l'esecuzione del programma e specifica l'ordine di esecuzione nelle regole del blocco. Generalmente Rosie supporta tre tipi diversi di monitor, chiamati sequenziale, ciclico e random. Essi differiscono nelle modalità di esecuzione; il primo, lo standard, esegue le regole in maniera sequenziale, il secondo in maniera ciclica (vale a dire, il set di comandi viene rieseguito daccapo), il terzo infine permette di eseguire regole in maniera casuale. Future implementazioni di ROSIE permetteranno, direttamente all'ingegnere della conoscenza, di costruire monitor a proprio piacimento e secondo le proprie esigenze.

Abbiamo così concluso la trattazione schematica delle parti del più efficiente e potente linguaggio dedicato ai SE. La prossima volta parleremo della struttura linguistica del linguaggio e della tecnica di redazione dei programmi.

Alla ricerca di un po' d'intelligenza

Il programma di Ken Thompson (chi era costui?) si basava, come vedemmo, su un blocco di conoscenza articolato in una maniera piuttosto semplice, rappresentato da una immensa tabella di coppie situazione-azione. C'è da rilevare che, quando gli esperti furono invitati a giocare «più da macchina», le cose funzionarono meglio e ci si rese conto che battere il programma era molto più facile del previsto. Tutto dipendeva dal fatto che era praticamente impossibile ramificare decisioni oltre un certo livello, anche in base al fatto che decisioni piuttosto ovvie per una mente umana abbisognavano, per un calcolatore, sovente di una parte di codice e di un impegno in termini di memoria anche abbastanza rilevante. Tanto per fare un esempio banale, nessun giocatore si sognerebbe di tentare lo scacco del barbiere, visto che esso è facilmente riconoscibile anche dal più smidollato dei giocatori; lasciamo immaginare al lettore la serie piuttosto articolata di «IF» per permettere alla macchina di riconoscere questo banale tentativo!

Abbandoniamo adesso il campo degli scacchi, ma non il problema. Se tutto fosse regolato da una semplice interdipendenza di coppie situazioni-azioni, funzioni particolarmente complesse, che oggi sono gestite da basi di dati e da sistemi esperti molto efficienti, dovrebbero invece essere solo appannaggio di una mente umana. Si intende, noi, in queste chiacchierate informali, stiamo dalla parte dell'uomo e non della macchina, come abbiamo fatto capire più volte in queste pagine. Ma non si può fare a meno di ammettere che la gestione di portafoglio di una grossa banca, o la regolazione del traffico aereo o ferroviario di grandi terminal non è pensabile possano, con le condizioni di movimento di oggi, essere gestite solo da uomini.

La tecnica della situazione-azione non è utilizzabile, o, per meglio dire, non è esclusivamente utilizzabile; cosa succederebbe se un sistema esperto utilizzato in un centro di difesa aerea, stabilisse a un certo punto, di oscurare le piste d'atterraggio e di mobilitare la difesa a terra? Il supervisore umano potrebbe pensare a un malfunzionamento (ce lo auguriamo!) e chiederebbe alla macchina il motivo della sua decisione, sentendosi rispondere, secondo quanto abbiamo visto nelle puntate precedenti, che «così stabilisce la tabella». Non è certo una risposta accettabile; ecco che allora c'è bisogno di quel quid in più, di quella «intelligenza» del sistema di cui stiamo tentando di parlare da quando abbiamo aperto questa rubrica. Cosa succederebbe infatti se alla comunicazione della macchina «Due bombardieri nemici si stanno dirigendo verso il nostro territorio», l'operatore non potesse, prima di premere il bottone (e, fortunatamente, prima di Gorbys) porre la domanda «Cosa te lo fa presumere», senza temere di avere la stupida risposta «Così stabilisce la tabella» o, peggio, «le mie tabelle di combinazione prevedono che questo stia avvenendo?»

Il problema, così si amplia. Se un sistema di tal fatta può essere senz'altro effica-

ce nella gestione di automatismi esclusivamente meccanici (che so, ad esempio, la gestione dei livelli di una diga o il controllo di magazzini di una grossa industria) non è pensabile l'adozione di tali tecniche quando vengono coinvolti risvolti sociali (come ad esempio difesa, nucleare, trasporti, finanze). Mentre nessuno si sognerebbe di chiedere alla macchina per quale motivo ha deciso di acquistare un lotto di bulloni di passo 10, occorre avere la possibilità di comprendere e, eventualmente, di confutare decisioni come quelle degli esempi precedenti.

Occorre allora affrontare il problema in maniera diversa; quella più efficace appare rappresentata da una ricerca completa attraverso gli alberi delle possibilità, che, con gergo tecnico, viene definita come quella del «look-ahead», o metodo delle proiezioni (cosa accadrebbe se...). Questo metodo è esattamente il contrario di quello delle supertabelle; in questo caso la tecnica (detta anche metodo delle consultazioni) richiede una minima potenza elaborativa (si tratta, alla fin fine, di consultare solo una base di dati, ancorché estesa) ma grandi masse di memoria. Nell'altro caso, l'esplorazione di ampi alberi di ricerca in tempi ragionevolmente brevi abbisogna di ampia potenza di calcolo mentre non è necessaria grande quantità di memoria. Molto spesso le due tecniche, con termini peraltro particolarmente azzeccati, vengono definite anche come tecnica di elaborazione «superficiale» e «profonda». La rappresentazione dei rapporti e dei termini di passaggio dall'una all'altra forma è rappresentata in figura a.

Sembrerebbe di aver trovato la soluzione. E invece non è così; come la tecnica delle supertabelle pecca di scarsa disponibilità di dati realtivi alla motivazione della scelta, la tecnica opposta è affetta dal difetto opposto. Se alla domanda «Perché intendi fare ciò» la prima non saprebbe dare una risposta convincente, un sistema basato sulla elaborazione di alberi sarebbe fin troppo eloquente; in scelte di un certo impegno l'operatore sarebbe travolto da una risposta contenente tanti dati e tante notizie da essere praticamente inutile. Avere un'arma troppo potente, diceva Churchill, è lo stesso che non averla! Così quella che pareva la soluzione buona è tanto buona da essere cattiva (tra l'altro, come riferiscono i rapporti finali, il disastro di Three Island fu dovuto sicuramente alla enorme quantità di allarmi e di dati sulla situazione, che impedirono agli operatori di scegliere rapidamente la soluzione giusta).

Nei diagrammi in figura è evidenziata un'area d'azione, la cosiddetta «finestra umana» che rappresenta la regione dove si sovrappongono aree di comprensione e collaborazione reciproca (siamo in tempi di perestrojka, che diamine!) tra macchina e uomo. Il sistema di consultazione delle tabelle fornisce risposte troppo superficiali, quello delle proiezioni troppo complesse, ma soprattutto dispersive. Il meglio sta nel mezzo (d'altro canto l'avevano capito anche i latini); ma di questo parleremo la prossima volta.

SEIKOSHA SL 92AI 1^a DELLA SUA CLASSE IN BELLA SCRITTURA



Una nitidezza mai vista prima, caratteri disegnati con precisione assoluta. Certo, Seikosha SL-92AI vanta il primato della sua classe in bella scrittura, perché riunisce un'elettronica sofisticata e una meccanica di alta precisione in un connubio perfetto. La testina a 24 aghi è dotata di una forza d'impatto superiore e sui moduli multicopia esprime tutta la sua potenza stampando perfettamente dal primo all'ultimo foglio. Naturalmente, offre la più ampia scelta di stili e caratteri, infatti, dispone di due fonts residenti oltre ai sette opzionali. Ma non solo, Seikosha SL-92AI stampa 240 caratteri per secondo e sviluppa questa notevole velocità nel silenzio di soli 55 dBA.



Come se non bastasse, la dotazione comprende un buffer di 44 Kbyte che permette un considerevole risparmio del tempo di impegno della CPU. Anche il trattamento della carta è di prim'ordine. I suoi trattori, del tipo a spinta, consentono il parcheggio del modulo continuo ed è predisposta per il sistema automatico di caricamento ed espulsione dei fogli singoli. Non ultimo e a tutto vantaggio della semplicità operativa, tutte le funzioni principali sono gestite direttamente da una pratica tastiera posizionata sul frontale della macchina. Seikosha SL-92AI, prima della sua classe in bella scrittura, seconda a nessuno in tutto il resto.

SEIKOSHA

COMPANY OF SEIKO GROUP