BOLLETTINO
UNIONE MATEMATICA ITALIANA

Sezione A – La Matematica nella Società e nella Cultura

FRANCESCO DE GIOVANNI, TOMMASO LANDOLFI

Le dimostrazioni di teoremi fondate sull'uso di calcolatori

Bollettino dell'Unione Matematica Italiana, Serie 8, Vol. **2-A**—*La Matematica nella Società e nella Cultura* (1999), n.1, p. 69–81.

Unione Matematica Italiana

<http://www.bdim.eu/item?id=BUMI_1999_8_2A_1_69_0>

Bollettino dell'Unione Matematica Italiana, Unione Matematica Italiana, 1999.

Bollettino U. M. I.
La matematica nella Società e nella Cultura
(8) 2-A (1999), pag. 69-81

Le dimostrazioni di teoremi fondate sull'uso di calcolatori.

FRANCESCO DE GIOVANNI - TOMMASO LANDOLFI

1. – Introduzione.

La dimostrazione automatica di teoremi, uno dei primi ambiti di ricerca in intelligenza artificiale, è direttamente legata alla comparsa dei calcolatori elettronici. D'altra parte, la logica matematica offre gli strumenti per la costruzione di teorie matematiche formali, indica cioè come l'insieme dei teoremi di un particolare settore della matematica possa essere ottenuto a partire da una «piccola» collezione di assiomi servendosi di alcune regole di inferenza, e già David Hilbert, nell'ambito del «programma formalista» degli inizi del secolo, si proponeva di introdurre in un «sistema formale» una procedura che consentisse di dedurre tutti i teoremi del sistema in modo assolutamente meccanico. Un *sistema formale* consiste di un alfabeto, una grammatica, un insieme di assiomi ed un insieme di regole di inferenza. Anche se il «sogno hilbertiano» è stato definitivamente infranto dai lavori di Kurt Gödel, logici e matematici hanno perseverato nel tentativo di escogitare metodi di dimostrazione automatica. Lo stesso Gödel, durante una conferenza tenuta a Providence (Rhode Island) nel dicembre 1951, disse: «La mente umana è incapace di formulare (o meccanizzare) tutte le sue intuizioni matematiche [...] D'altra parte, sulla base di quanto è stato finora dimostrato, resta concepibile che possa esistere (e addirittura che possa essere scoperto empiricamente) un dimostratore meccanico che in realtà è equivalente all'intuizione matematica, ma di cui non si può *dimostrare* che lo sia, né si può dimostrare che produca solo teoremi *corretti* della teoria dei numeri finitaria» (cfr. [12], p. 204).

In altri termini quel che si può realmente dedurre dai risultati di Gödel è che non è possibile costruire una macchina che dimostri *tut-*

ti i teoremi di una disciplina matematica; ovviamente un analogo limite esiste anche per la mente umana, mentre la *possibilità* di realizzare un «buon dimostratore automatico», ovvero una macchina in grado di dimostrare teoremi non banali, dipende esclusivamente dalle capacità di progettisti e programmatori.

Fino agli anni '80, i dimostratori di teoremi erano sostanzialmente basati sulla *risoluzione* introdotta da Alan Robinson [11], una procedura per la produzione di dimostrazioni mediante refutazione, cioè tentando di provare che la negazione di un certo enunciato genera una contraddizione con enunciati assunti veri. «La tranquillità fu scossa dalle stridenti critiche da parte di alcuni ricercatori in Intelligenza Artificiale, in modo particolare da quelli del MIT di Boston. Questi ultimi sostenevano che i metodi usati non avrebbero conquistato l'intera matematica ma solo un suo banale sottoinsieme. La generazione di dimostrazioni di teoremi interessanti e non banali richiedeva l'uso di conoscenza sofisticata e specifica, che non era stata prevista nella famiglia di sistemi di dimostrazione basati sulla risoluzione» (cfr. [2], p. 13). Dunque, l'aspetto cruciale nella dimostrazione automatica diventa la conoscenza posseduta da un tipico esperto. Ne segue che il cardine di un dimostratore automatico viene ad essere la *rappresentazione esplicita della conoscenza* inerente l'ambito in cui si intende agire. È chiaro che rappresentare la conoscenza costituisce solo una parte, seppure estremamente rilevante, del problema: *manipolare* tale conoscenza in modo efficace risulta altrettanto importante.

Il panorama che va delineandosi è ora chiaro: accanto alla necessità di costruire una *base di conoscenza* paragonabile a quella di un esperto, troviamo l'esigenza di realizzare un efficiente *motore inferenziale*, al fine di utilizzare le conoscenze possedute per derivarne nuove. Come conseguenza di questi due aspetti, possiamo rilevare un terzo elemento che entra in gioco, ovvero l'*apprendimento*: un programma che dimostri teoremi deve anche «imparare» ad utilizzare ciò che ha già provato per poter svolgere bene il suo compito.

Per quanto riguarda i tentativi di costruire sistemi che effettivamente dimostrino teoremi, si possono distinguere diversi indirizzi di ricerca. Da un lato possiamo citare il «Logic Theorist» [8], programma che dimostra teoremi tratti dal primo capitolo dei *Principia Mathematica* di Russell e Whitehead, la «Geometry Machine» di Gelernter [4] che prova asserti di geometria euclidea e l'«Equational Prover», un sistema rivolto alle algebre di Boole. In questi casi, siamo di fronte a tentativi di costruire procedure il cui scopo è quello di generare teoremi in un particolare ambito, ovvero dimostrazioni formali inerenti uno specifico settore ma non uno specifico problema. Esistono, tuttavia, anche altri modi di dimostrare teoremi con l'uso del calcolatore: progettare ad esempio un programma al fine di risolvere un ben determinato problema di un particolare settore della matematica. È quanto accaduto riguardo il classico e famoso «problema dei quattro colori». Il programma progettato da Kenneth Happel e Wolfgang Haken [1] era finalizzato alla soluzione di questo specifico problema e si basava su un approccio di tipo essenzialmente computazionale.

Una tipologia particolare di dimostrazione automatica è rappresentata dal sistema AM progettato da Douglas Lenat ([5], [6]) all'inizio degli anni '80, il cui ambito di lavoro è la teoria dei numeri. In questo caso, al calcolatore vengono forniti una serie di assunti iniziali e di regole sul loro uso, lasciando al sistema non solo il compito di derivare nuovi teoremi, ma soprattutto la funzione di dedurre nuovi concetti. Dunque AM rappresenta il tipico esempio di programma che «apprende»: il suo scopo infatti è anche quello di generare definizioni appropriate di concetti inerenti la teoria dei numeri.

Pur senza alcuna pretesa di classificazione, e tantomeno di esaustività, si può dire che esistono approssimativamente tre grandi famiglie di macchine per la dimostrazione automatica di teoremi: dimostratori di tipo computazionale, dimostratori formali e sistemi esperti. Nei paragrafi successivi si cercherà di chiarire con opportuni esempi le differenze esistenti tra questi tipi di dimostratori automatici.

Sembra opportuno, a questo punto, porre l'attenzione su un aspetto estremamente delicato quale il controllo delle dimostrazioni prodotte da sistemi automatici. Gli stessi Appel e Haken, nel descrivere la loro soluzione del problema dei quattro colori, evidenziano le perplessità che possono sorgere a tale riguardo. «Molti matematici, in particolare quelli formatisi prima dello sviluppo dei calcolatori ad alta velocità, si oppongono al trattare tali macchine come normali strumenti matematici. Hanno l'impressione che un'argomentazione sia debole quando non può essere controllata, tutta o in parte, direttamente. [...] Le dimostrazioni tradizionali dei teoremi matematici sono ragionevolmente brevi e prettamente teoretiche — quanto più potente è la teoria tanto più elegante è la dimostrazione — e ricontrollarle a mano è, normalmente, il metodo migliore. Ma, anche quando il controllo a mano è possibile, se le dimostrazioni sono lunghe e piene di calcoli, è difficile credere che il controllo con carta e matita escluderà ogni possibilità di errore» (cfr. [1]). Appare difficile assumere una posizione netta al riguardo. Infatti, anche nel caso di una dimostrazione tradizionale, può essere arduo controllare la correttezza di ciascun passo; si pensi, ad esempio, alla *classificazione dei gruppi semplici finiti*, dove alle difficoltà intrinseche si aggiunge l'estrema lunghezza dell'apparato dimostrativo. In casi come questo, può risultare praticamente impossibile un controllo puntuale e definitivo da parte di un singolo ricercatore.

2. – Il problema dei quattro colori.

Nel 1976, Appel e Haken [1], dell'Università dell'Illinois, forniscono una soluzione positiva del «problema dei quattro colori». La peculiarità di questa soluzione è data dal fatto che essa è fortemente basata sull'uso di computazioni prodotte da calcolatori: 1200 ore di tempo macchina per colorare complessivamente circa 2000 carte in 200.000 modi differenti. Il problema fu posto nel 1852 da Francis Guthrie, il quale, in una lettera al fratello Frederick (allievo di Augustus De Morgan), osserva che per colorare una qualsiasi carta geografica, in modo tale che due regioni confinanti non abbiano lo stesso colore, sembrano bastare quattro colori soltanto. Guthrie si chiedeva se esistesse una procedura matematicamente corretta per verificare questa affermazione. Da quel momento sono trascorsi oltre 120 anni senza ottenere una risposta, nonostante l'impegno di matematici di primo piano, tra cui lo stesso De Morgan, Arthur Cayley e George Birkhoff.

I primi importanti progressi verso una soluzione della congettura furono compiuti intorno al 1880 da Alfred Kempe, un avvocato membro della London Mathematical Society. Alcune delle idee concepite da Kempe si sono rivelate fondamentali: egli introdusse il concetto di *carta normale*, ovvero una carta in cui nessuna regione ne include altre ed in ciascun punto concorrono al più tre regioni, e dimostrò che l'esistenza di una carta che richiede cinque colori, cioè *pentacromatica*, implica l'esistenza di una carta pentacromatica normale. A questo punto, Kempe prese in considerazione carte pentacromatiche normali minimali, cioè carte pentacromatiche normali contenenti un numero minimo di regioni, e cercò di arrivare ad una contraddizione. Preliminarmente dimostrò che in ogni carta normale esistono solo regioni con due, tre, quattro o cinque confinanti, ovvero che non esistono carte normali in cui sia presente una regione con sei o più confinanti. Successivamente, arguì che se una carta pentacromatica normale minimale contiene una regione con meno di sei regioni confinanti, allora deve esistere una carta pentacromatica normale con un numero inferiore di regioni, contraddicendo l'ipotesi di minimalità. Per dimostrare tale asserto Kempe studiò separatamente i vari casi, cioè esaminò carte normali in cui erano presenti, rispettivamente, regioni con due, tre, quattro o cinque confinanti: la sua argomentazione però funzionava bene nei casi due, tre e quattro, ma falliva purtroppo nel caso cinque. La dimostrazione prodotta da Appel e Haken consiste nel rimediare a questo errore. Nella terminologia attuale, l'asserto di Kempe inerente l'esistenza di carte normali con regioni aventi al più cinque confinanti, si esprime dicendo che ognuna di tali *configurazioni* è *inevitabile*: ogni carta normale contiene necessariamente almeno una di tali configurazioni. L'altro importante risultato di Kempe riguardante l'inesistenza di carte pentacromatiche normali minimali in cui siano presenti regioni con due, tre o quattro confinanti, si traduce affermando che tali configurazioni sono *riducibili*. Volendo riassumere, si può dire che il lavoro di Kempe è stato un «tentativo di trovare un insieme inevitabile di configurazioni riducibili» (cfr. [1]). Ottenere un tale insieme equivale ovviamente a risolvere il problema dei quattro colori.

In seguito, diversi miglioramenti furono apportati alle tecniche di

Kempe da vari matematici, tra i quali George Birkhoff, ma fu necessario aspettare gli anni ’50 per una maggiore presa di coscienza del problema. In questo periodo infatti Heinrich Heesch, un matematico dell’Università di Hannover, occupandosi di questo problema cercò anzitutto di stimarne la «dimensione» reale: secondo i suoi calcoli, il numero di configurazioni riducibili in un insieme inevitabile si aggirava intorno alle diecimila. Questo semplice dato numerico fa capire come fosse impossibile sperare in una dimostrazione della congettura senza l’aiuto di un calcolatore elettronico, almeno seguendo questo tipo di approccio. L’idea centrale di Heesch consisteva nel trasformare la carta iniziale in un grafo piano i cui *vertici* rappresentano una regione della carta e gli *spigoli* i confini tra regioni. In pratica, ogni regione viene «sostituita» con la propria capitale ed i confini tra due regioni con un segmento che congiunge le rispettive capitali, sicchè il *grado* di un vertice corrisponde al numero di regioni confinanti e, per quanto dimostrato da Kempe, ogni vertice ha almeno grado cinque. Costruendo il grafo in tal modo, il piano viene suddiviso in poligoni (*facce*), che sono triangoli in quanto si considerano solo carte normali; pertanto i grafi che si considerano sono *triangolazioni* del piano. Appel e Haken assegnano ad ogni vertice un *numero di carica* pari a $6 - k$, dove k è il grado del vertice: ne segue che tutti i vertici hanno carica negativa (*vertici principali*), tranne quelli di grado cinque o sei. L’importanza del concetto di carica risiede nel fatto che per ogni triangolazione del piano corrispondente ad una carta normale la somma delle cariche risulta costante e positiva. Allora, ad una triangolazione può applicarsi una *procedura di scaricamento*: vertici a carica positiva perdono carica a favore di vertici principali. Lo scopo dello scaricamento è quello di determinare «una procedura che descriva esattamente come trasferire le cariche in modo tale da assicurarsi che ogni vertice a carica positiva, nella configurazione risultante, debba o appartenere ad una configurazione riducibile o essere adiacente ad una di esse. Dal momento che le configurazioni evidenziate da questa procedura debbono formare un insieme inevitabile, se esse sono anche riducibili la congettura dei quattro colori è dimostrata» (cfr. [1]). Per provare la veridicità di quest’ultima affermazione, Appel e Haken hanno analizzato circa

10000 intorni di vertici a carica positiva (senza calcolatore) ed oltre 2000 configurazioni usando 1200 ore di tempo-macchina. Sebbene sia possibile verificare «a mano» la procedura di scaricamento adottata, è ovviamente impossibile un tale controllo per i calcoli delle riduzioni.

3. – La congettura di Robbins.

Di tipo alquanto diverso è la dimostrazione della congettura di Robbins. Il programma che ha generato la prova, «Equational Prover» (EQP), è un esempio di dimostratore formale: sono riportati tutti i passi necessari, tranne quelli ritenuti «facili» dal programma stesso (e questo aspetto avrebbe senz'altro bisogno di una trattazione a parte). Il problema in esame appartiene all'ambito delle algebre di Boole, e per circa 60 anni ha deluso gli sforzi di coloro che hanno tentato di risolverlo. L'algebra booleana può essere vista come un modo di trasformare asserti logici in espressioni algebriche. Ad esempio, asserti del tipo «*P oppure Q*» e «*P oppure non-Q*» possono esprimersi come $p + q$ e $p + N(q)$, rispettivamente. Trasformare un asserto del tipo «*P e Q*» è formalmente più complicato: ad esso corrisponde l'espressione $N(N(p) + N(q))$.

Gli assiomi «standard» per un'algebra di Boole sono:

- $p + q = q + p$
- $(p + q) + r = p + (q + r)$
- $p + N(q + N(q)) = p$
- $p + N(N(q) + N(r)) = N(N(p + q) + N(p + r))$,

e da questi postulati è possibile derivare tutte le altre proprietà canoniche.

Nel 1933 Edward Huntington dimostrò che tra gli assiomi precedenti gli ultimi due possono essere sostituiti con un'unica espressione:

$$N(N(p) + q) + N(N(p) + N(q)) = p\ .$$

Poco dopo Herbert Robbins, un allievo di Huntington, ipotizzò che in questo contesto fosse possibile rimpiazzare tale assioma con l'espressione

$$N(N(p+q)+N(p+N(q)))=p\ ,$$

che risulta formalmente più semplice, in quanto vi compare una negazione in meno.

Nel corso degli anni, numerosi matematici si sono cimentati senza successo nel tentativo di dimostrare la congettura di Robbins. Tra gli altri vi lavorò Alfred Tarski, il quale era solito proporla ai colleghi in visita all'Università di Berkeley.

È alla fine degli anni '70 che Steve Winker, allievo di Larry Wos (uno dei pionieri nel campo della dimostrazione automatica di teoremi, cfr. [13], [14]), adotta la strategia, suggerita dallo stesso Wos, che si rivelerà vincente: lavorare «all'indietro» sul problema. In altri termini, si cercano quelle equazioni che, combinate con l'assioma di Robbins (e con gli assiomi di commutatività e associatività), risultino sufficienti per generare l'intero sistema assiomatico delle algebre di Boole, con la speranza che una di queste equazioni risulti essa stessa derivabile dall'espressione di Robbins. Sfortunatamente il tentativo non riesce, anche se vengono prodotte molte equazioni del tipo voluto. Poco tempo dopo, William McCune si unisce al gruppo di Wos e, nel corso di un decennio, vengono sviluppati alcuni programmi per la dimostrazione automatica di teoremi, tra cui EQP. All'inizio degli anni '90, una versione preliminare di EQP ottiene dei risultati promettenti: riscopre le equazioni che Winker aveva trovato «a mano». La soluzione definitiva però resta ancora lontana. Infatti è nel 1996 che EQP risolve il problema: una delle equazioni di Winker viene effettivamente derivata dall'assioma di Robbins, concludendo così una ricerca durata oltre sessanta anni. EQP impiega circa otto giorni e coinvolge nella soluzione quasi 50000 equazioni intermedie. Uno degli aspetti sorprendenti è che la dimostrazione vera e propria della congettura consiste in sole tredici righe, a fronte del gran numero di equazioni realmente usate. Il motivo di tale sintesi è da attribuirsi alla «facilità» che EQP ascrive a molti dei passaggi intermedi della dimostrazione.

La tecnica usata da EQP è quella della *paramodulazione* che consiste nel derivare nuove equazioni da altre effettuando delle sostituzioni. Come esempio, consideriamo quello che sembra essere (si ricordi che il sistema non spiega i passaggi intermedi ritenuti facili) il primo passo nella dimostrazione ottenuta da EQP. Se nell'equazione di Robbins

$$N\big(N(N(p)+q)+N(p+q)\big)=q$$

le variabili p e q vengono rimpiazzate con $N(p)+q$ e $N(p+q)$, rispettivamente, si ottiene la nuova equazione

$$N\big(N(N(N(p)+q)+N(p+q))+N((N(p)+q)+N(p+q))\big)=N(p+q)\,.$$

Anche se questo sembra un passo nella direzione sbagliata, EQP riconosce una parte dell'equazione come la componente sinistra dell'espressione di Robbins, e quindi la rimpiazza con la più semplice componente destra della stessa espressione, ottenendo l'equazione

$$N\big(q+N((N(p)+q)+N(p+q))\big)=N(p+q)\,.$$

Usando gli assiomi di associatività e commutatività, si ottiene allora

$$N\big(N((N(p+q)+N(p))+q)+q\big)=N(p+q)\,.$$

È questo il tipico «modus operandi» adottato da EQP.

Stanley Burris, un logico dell'Università di Waterloo (Canada), ha affermato a proposito del successo di EQP: «È una pietra miliare nell'ambito della dimostrazione automatica di teoremi. Ora che sappiamo che un computer può risolvere problemi difficili, si aprono le porte per un gran numero di ricerche interessanti» (cfr. [3]).

4. – AM: un sistema per la scoperta.

Un ulteriore e decisivo passo nel tentativo di emulare il processo speculativo umano (e quindi l'apprendimento) consiste nel riprodurre, mediante una macchina, il fenomeno della «scoperta»: di nuovi concetti, di teoremi che li coinvolgano e di nuovi problemi che ne derivino.

AM cerca di fare proprio questo: partendo da poche nozioni di teoria degli insiemi, scoprire nuovi concetti in teoria dei numeri. In un modo che sarà chiarito in seguito, AM ha scoperto una buona parte della teoria standard dei numeri. Per rappresentare la conoscenza, e memorizzare le sue deduzioni, AM usa dei *frames*, cioè delle strutture dati complesse che rappresentano il modello di un contesto o di un oggetto mediante dei *campi* per le entità che entrano in gioco, delle *relazioni* tra i campi e dei *valori di default* (per «default» si intende l'assunzione di un determinato valore per una variabile in caso di mancata specificazione per esso). Un frame può vedersi in analogia con un *record*, ovvero un *tipo di dato* che permette di aggregare più informazioni; in questo senso esiste un'analogia più ampia tra base di conoscenza e base di dati, ove la differenza sostanziale consiste nel potere espressivo addizionale dei campi in un frame.

La base di conoscenza iniziale di AM consiste di circa 80 nozioni elementari di teoria degli insiemi, mentre il suo motore inferenziale utilizza circa 250 *regole euristiche*, ossia suggerimenti di strategie da adottare dipendenti dal contesto. Tra le euristiche più semplici a disposizione di AM, compaiono in particolare le seguenti (per maggiori dettagli cfr. [9], Cap. 17):

◊ *Se alcuni esempi di un concetto X sono esempi anche di un concetto Y, allora costruisci un nuovo concetto che rappresenta l'intersezione di X e Y.*

◊ *Se sono pochi gli esempi di un concetto X, allora trova una generalizzazione di X.*

Infine, le attività in AM vengono gestite da un controllo basato su *agende*, e procedono secondo lo schema seguente:

- l'euristica suggerisce un compito
- il compito viene posto in agenda
- AM sceglie dall'agenda il compito più promettente e lo esegue.

Un esempio illustrativo del modo di operare di AM è dato dalla *scoperta dei numeri primi*. Durante il processo che ha consentito tale «scoperta», AM elabora diverse congetture collaterali, alcune sicuramente banali, altre estremamente interessanti. In queste considerazioni, le affermazioni «AM cerca di ...» o «AM scopre ...», vanno interpretate nel senso che l'azione viene eseguita da una o più euristiche di AM.

Analizzando addizione e moltiplicazione tra numeri naturali (con le loro operazioni inverse), AM scopre il concetto di divisibilità. Seguendo l'euristica che gli indica di esaminare i casi estremi, AM cerca di trovare numeri che possiedano pochi divisori: dapprima tenta con zero divisori (non trovando ovviamente alcun numero), con un solo divisore (trovando il numero 1), poi con due divisori, scoprendo così i numeri primi, come gli «viene suggerito» di denominarli. Successivamente AM cerca numeri con tre divisori, e si accorge, tra l'altro, che i quadrati perfetti godono di tale proprietà. In conseguenza di tale analisi, AM indaga le possibilità di scomporre un numero in fattori primi, e si accorge che per ciascun numero questo si può realizzare in un unico modo: «incidentalmente» AM scopre il teorema di fattorizzazione unica. Per analogia, AM cerca un concetto equivalente alla fattorizzazione in ambito additivo: sviluppa in tal modo alcune congetture banali o irrilevanti, ad esempio che ogni numero può ottenersi come una somma in cui tutti gli addendi sono uguali a 1. Tuttavia, vengono elaborati anche concetti più interessanti. Notando che molti numeri possono essere decomposti nella somma di due primi, AM congettura che tutti i numeri pari maggiori di 2 godano della stessa proprietà; in altri termini AM scopre la famosa *congettura di Goldbach*, che ancora oggi non è stata risolta nella sua generalità.

Questo esempio del *modus operandi* di AM mette in luce le notevoli qualità del programma da un punto di vista strettamente pragmatico. I motivi probabilmente sono da ricondurre a due aspetti fondamentali: da una parte il numero e la qualità delle euristiche a disposizione di AM, dall'altra la rappresentazione ottimale della conoscenza. «AM lavorava modificando sintatticamente vecchie definizioni di concetti, memorizzati essenzialmente sotto forma di brevi programmi LISP. [...] Mentre AM viene interpretato dalle persone come esploratore della teoria dei numeri, esso stava in realtà esplorando lo spazio dei piccoli programmi LISP. Gran parte del successo di AM è dovuto alla stretta relazione tra la teoria dei numeri ed i programmi LISP» (cfr. [9], pag. 509). In [10], gli autori fanno notare come sia importante l'interpretazione da dare a come i programmi di Intelligenza Artificiale realmente operino: AM possedeva una *inclinazione naturale* verso la teoria dei numeri, data appunto dalla relazione tra linguaggio LISP e teoria dei numeri, che si traduce in un'adeguata rappresentazione della conoscenza intorno al dominio di indagine.

Il successo di AM ha spinto Lenat [7] a progettarne un'evoluzione che avrebbe dovuto, nelle intenzioni dell'autore, operare anche ad un *metalivello*, manipolando le stesse euristiche. Questo programma, denominato *Eurisko*, avrebbe dovuto utilizzare le nuove euristiche generate per colmare la «distanza» tra i concetti iniziali ed i nuovi che AM aveva sviluppato. Purtroppo *Eurisko* non ha risposto a queste aspettative, a riprova della necessità di una stretta correlazione tra metodi di rappresentazione e dominio di indagine.

BIBLIOGRAFIA

[1] K. Appel - W. Haken, *Il problema dei quattro colori*, Le Scienze, **113** (1978).

[2] A. Bundy, *L'Automazione del Ragionamento Matematico*, F. Muzzio Editore, Padova (1986).

[3] B. Cipra, *As easy as EQP*, in *What's Happening in the Mathematical Sciences* (American Mathematical Society) (1998/99), 58-72.

[4] H. Gelernter - J. R. Hansen - D. W. Loveland, *Empirical explorations of the Geometry Theorem Proving Machine*, Computers and Thought, ed. E. A. Feigenbaum e J. Feldman, McGraw-Hill, New York (1963).

[5] D. B. Lenat, *Automated theory formation in mathematics*, Proceedings of IJCAI-77 (1977), 833-842.

[6] D. B. Lenat, *AM: An artificial intelligence approach to discovery in mathematics as heuristic search*, Knowledge-Based Systems in Artificial Intelligence, ed. R. Davis e D. B. Lenat, McGraw-Hill, New York (1982).

[7] D. B. Lenat, *Theory formation by euristic search – the nature of euristics II: background and examples*, Artificial Intelligence, **21** (1983), 31-59.

[8] A. Newell - J. C. Shaw - H. A. Simon, *Empirical explorations with the Logic Theory Machine: a case study in heuristics*, Computers and Thought, ed. E. A. Feigenbaum e J. Feldman, McGraw-Hill, New York (1963).

[9] E. RICH - K. KNIGHT, *Intelligenza Artificiale*, 2ª ed., McGraw-Hill Italia, Milano (1992).

[10] G. D. RITCHIE - F. K. HANNA, *AM: A case study in AI methodology*, Artificial Intelligence, **23** (1984), 249-268.

[11] A. J. ROBINSON, *A machine oriented logic based on the resolution principle*, J. Assoc. Comput. Mach., **12** (1965), 23-41.

[12] R. RUCKER, *La mente e l'Infinito*, F. Muzzio Editore, Padova (1991).

[13] L. WOS, *Solving open questions with an automated theorem-proving program, Proceedings of CADE*, ed. D. Loveland, Springer, Berlin (1982), 1-31.

[14] L. WOS - R. OVERBECK - W. LUSK - J. BOYLE, *Automated Reasoning: Introduction and Applications*, Englewood Cliffs Prentice-Hall, New York (1984).

Dipartimento di Matematica e Applicazioni «R. Caccioppoli»
Università di Napoli «Federico II, Via Cintia - 80126 Napoli