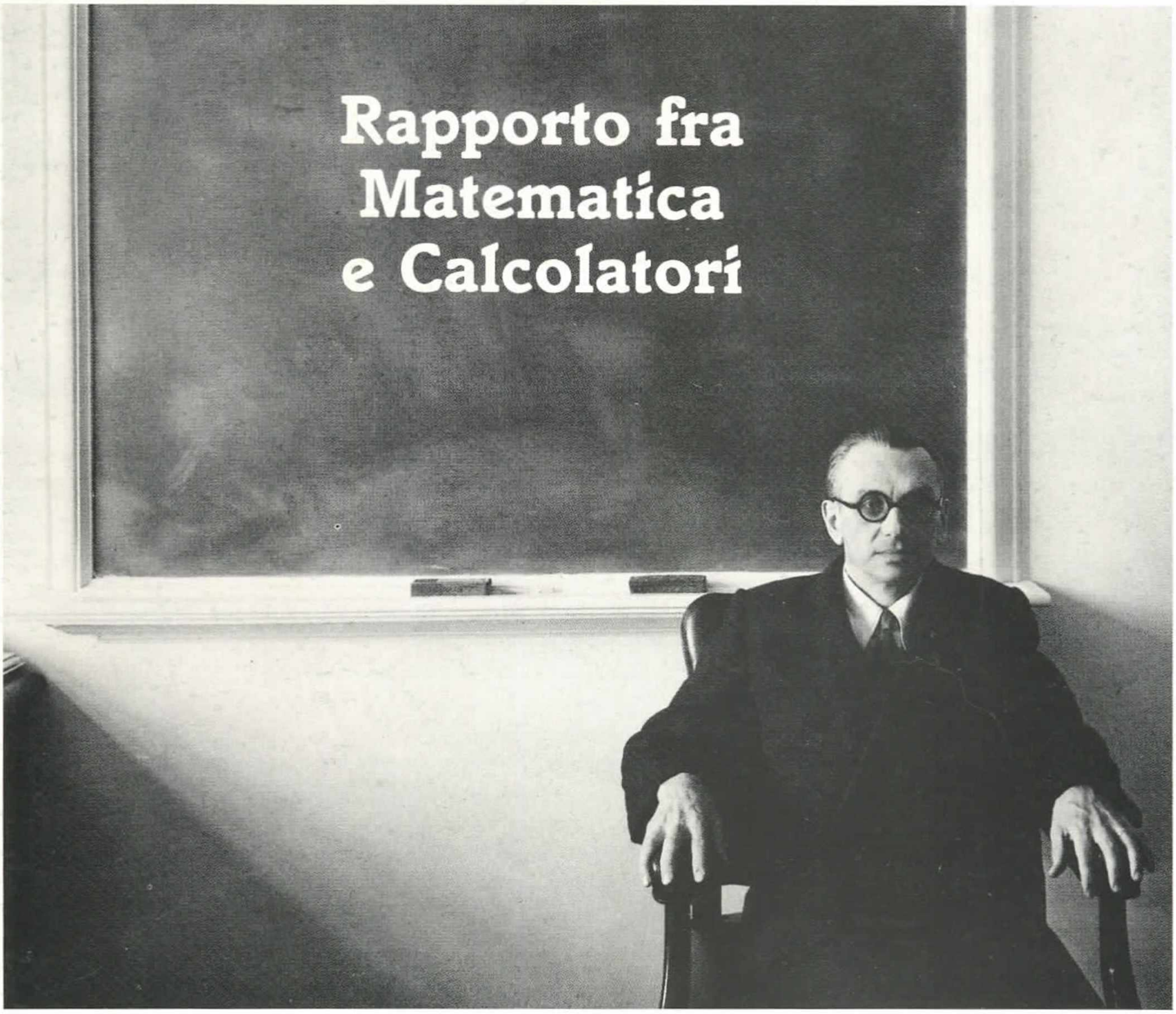
ITC DOSSIER

Periodico trimestrale
Supplemento al n. 3/1988 di ITC Informa
Spedizione in abb. post. Gruppo IV/70%
I trimestre 1988
Reg Trib. TN n. 530 del 28/3/87

Rapporto fra Matematica e Calcolatori



ISIG
ISTITUTO STORICO ITALO GERMANICO IN TRENTO
ITALIENISCH-DEUTSCHES HISTORISCHES INSTITUT IN TRIENT

ISR
ISTITUTO DI SCIENZE RELIGIOSE IN TRENTO

IRST
ISTITUTO PER LA RICERCA SCIENTIFICA E TECNOLOGICA

CIRM
CENTRO INTERNAZIONALE PER LA RICERCA MATEMATICA

I SOCI DELL'ISTITUTO TRENTINO DI CULTURA

I SOCI SONO

a) fondatori
(con quota annua non inferiore a L. 20 milioni)

Provincia Autonoma di Trento
Cassa di Risparmio di Trento e Rovereto
Comune di Trento
Banca di Trento e Bolzano
Associazione Industriali della Provincia di Trento
Comune di Rovereto
Camera di Commercio, Industria, Artigianato e Agricoltura di Trento

b) ordinari
(con quota annua non inferiore a L. 5 milioni):
Istituto di Credito Fondiario
Consorzio dei Comuni della Provincia di Trento
Bacino imbrifero dell'Adige
Istituto Trentino Alto Adige per Assicurazioni

c) aggregati
(con quota non inferiore a L. 250 mila):
Banca Calderari

IL CONSIGLIO D'AMMINISTRAZIONE

Per il triennio 1985/87 il Consiglio di Amministrazione è formato da:

avv. Bruno Kessler, *presidente, senatore della Repubblica*
rag. Fausto Gobbi, *vice presidente*
dott. Tarcisio Andreolli, *assessore alla p.i. e alla cultura, Provincia autonoma di Trento*
dott. Gianni Bazzanella, *presidente della Regione Trentino-Alto Adige*
rag. Mimmo F. Cecconi, *industriale*
dott. Marco Oreste Detassis, *presidente CCIAA, Trento*
prof. Aldo Maurina, *docente*
dott. Renzo Michelini, *sindaco di Rovereto*
p.i. Riccardo Ricci, *assessore all'industria, Provincia autonoma di Trento*
rag. Aimone Sordo, *vice presidente della Cassa di risparmio di Trento e Rovereto*
avv. Dario Vettorazzi, *presidente della Banca di Trento e Bolzano*
prof. Danilo Vettori, *presidente Accademia degli Agiati, Rovereto*
prof. Claudio Visintainer, *assessore all'urbanistica del Comune di Trento*
dott. Franco Zampini, *dirigente ENEA*

Il Collegio dei Revisori dei Conti è formato da:

rag. Ettore Buccella
p.i. Aldo Degaudenz
dott. Paolo Spagni

Responsabile servizi amministrativi ITC:
rag. Mario Tonini, *segretario del consiglio*

Relazioni pubbliche:
dott. Gianni Faustini

ITC DOSSIER

Anno III, numero 1

Direttore: sen. avv. Bruno Kessler
Responsabile: Gianni Faustini
Comitato di Redazione:
Gianni Faustini
Aldo Maurina
Mario Tonini
Franco Zampini
per gli Istituti
Tullio Grazioli
Giovanni Menestrina
Augusto Micheletti
Giuliana Nobili
Progetto grafico: Bruno Zaffoni

Associato all'USPI
Unione Stampa
Periodica Italiana

«ITC Informa» e «ITC Dossier» vengono inviati ad operatori della cultura e dell'economia. Chi desiderasse ricevere copia della presente pubblicazione potrà farne richiesta agli uffici dell'ITC, via Santa Croce, 77 Trento.
Chi intendesse abbonarsi – l'invio è gratuito – potrà segnalare questo desiderio allo stesso indirizzo fornendo i dati utili all'inoltro del periodico.

1° Ciclo di Conferenze del Programma
Matematica è Cultura

Rapporto fra Matematica e Calcolatori

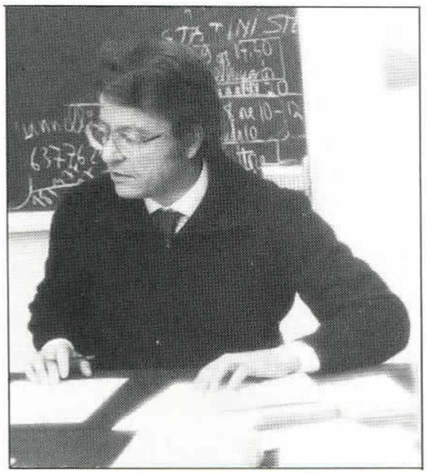
Trento, Aula Grande ITC
9 e 10 dicembre 1987

In copertina: Kurt Gödel, uno dei più grandi logici di questo secolo.

Introduzione

di Mario Miranda

Matematica è cultura è il titolo di un nuovo programma del Centro Internazionale per la Ricerca Matematica. Questo programma rientra nelle finalità dell'Istituto Trentino di Cultura, di cui il CIRM è emanazione, che comprendono oltre alla promozione di ricerche scientifiche di natura fondamentale e applicata, la diffusione delle conoscenze in ogni settore della società.
La matematica è scienza dalla lunga storia, misurata ormai in millenni, ma è solo in questo secolo, dopo essere giunta ad uno stadio avanzatissimo di sviluppo e dopo aver egregiamente servito tutte le altre scienze, che la matematica ha cominciato a guardare a se stessa come ad una creazione a sè stante del genio degli uomini.
La riflessione sui fondamenti della matematica, alla quale hanno contribuito matematici sommi come David Hilbert e Kurt Gödel, e filosofi della scienza come Bertrand Russell, nata per esigenze interne di assestamento ha avuto uno stimolo esterno derivante dal progredire negli ultimi decenni del lavoro attorno alla progettazione e costruzione di macchine intelligenti. Questo lavoro è diventato sempre più frenetico ed economicamente interessante per la scoperta e lo sviluppo di materiali adatti alla bisogna: si pensi alle



vecchie valvole, agli anziani transistor e ai giovani chips.
L'intelligenza che si vuole dare ad una macchina deve essere ovviamente ben conosciuta a priori dai progettisti, i quali hanno così scoperto che è l'intelligenza matematica il soffio divino. Da ciò la spinta esterna a definire cos'è l'intelligenza matematica cioè la matematica tout court.
La diffusione del messaggio culturale portato dalla Matematica è affidata innanzitutto a cicli di conferenze. Il primo ciclo, dal titolo «Matematica e Calcolatori», ha visto come conferenzieri i professori Gabriele Lolli e Annalisa Marcja. Il professor Lolli ha parlato su «I favolosi anni Venti: la logica tra uso e menzione», e «Il dopoguerra: il Calcolatore e le dimostrazioni». La professoressa Marcja ha parlato su «Computabilità e Complessità» e «Programmazione logica e i computer della V generazione».
I testi delle quattro conferenze costituiscono la parte più corposa di questa pubblicazione. Ad essi premettiamo una breve nota biografica relativa ai loro autori.
Gabriele Lolli si è laureato a Torino nel 1965 presentando una tesi di Analisi Matematica scritta sotto la guida del professor Francesco Tricomi. Per la formazione post-laurea Lolli ha visitato la Yale University, dove ha lavorato sotto la guida del professor A. Robinson. Il professor Lolli, che ha insegnato al Politecnico di Torino, alle Università di Salerno e Genova, attualmente lavora nel Dipartimento di Informatica dell'Università di Torino ed è autore, oltre che di numerosi articoli, di alcuni testi di divulgazione come:

1. La Teoria Assiomatica degli Insiemi.
2. Lezioni di Logica Matematica, pubblicati da Boringhieri.
3. Le Ragioni Fisiche e le Dimostrazioni Matematiche.
4. La Macchina e le Dimostrazioni, pubblicati da Il Mulino.

Annalisa Marcja si è laureata a Firenze nel 1966 sotto la guida del professor Roberto Magari. Ha visitato l'Università di Rutgers negli USA ed ha insegnato all'Università di Firenze e a quella di Trento, do-

Kurt Gödel nacque a Brno (Cecoslovacchia) nel 1906 e morì a Princeton (USA) nel 1978.

ve trovasi tuttora. È presidente del Consiglio Scientifico del Centro di Computer Algebra, creato dal Dipartimento di Matematica dell'Università di Trento e dall'IRST. È autrice di numerose pubblicazioni scientifiche, collabora con diversi matematici stranieri ed è particolarmente interessata alla Teoria dei Modelli. Su questo argomento ha organizzato per il CIRM due convegni internazionali di grande successo.

Mi sia permesso ancora un breve commento di carattere generale prima di dare la parola ai nostri illustri esperti.

Le iniziative trentine nel settore Università e Ricerca Scientifica hanno ormai venticinque anni di vita. All'inizio si è trattato di impiegare parte del potenziale offerto dall'autonomia provinciale, come libertà di legiferare e di investire denaro pubblico, per creare strutture e attirare persone capaci di mettere in moto un processo di modernizzazione del trentino.

Oggi trattasi di prendere atto delle trasformazioni avvenute, di conoscere a fondo il nuovo trentino e di operare per il suo inserimento in programmi di respiro nazionale e internazionale. Questo inserimento porterà con sè l'inserimento dei trentini, giovani e meno giovani, nei processi di produzione di cultura e di beni di consumo. Questo inserimento è compito non solo di politici e di professori universitari, ma di tutti coloro che nel Trentino hanno la responsabilità di operare scelte, dagli imprenditori pubblici e privati ai singoli cittadini.

Per svolgere questo compito è necessario, come già più sopra dicevo, conoscere analiticamente la realtà trentina attuale. Poiché in questa realtà la Matematica è viva ed operante, il progetto Matematica è Cultura si inserisce nel grande progetto di modernizzazione della Provincia di Trento.

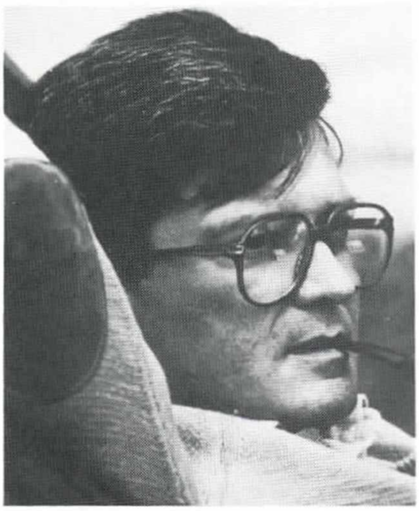
I favolosi anni Venti: la logica tra uso e menzione

di Gabriele Lolli

Vogliamo raccontare alcuni aspetti di un episodio importante della storia della matematica che si è svolto nel ventesimo secolo, quello della progressiva acquisizione da parte della logica di un ruolo centrale nella matematica. La logica è sempre stata accostata alla dimostrazione, ma come generica caratterizzazione di cui non si vedevano però effetti concreti; oggi che la logica si presenta come matrice degli sviluppi rivoluzionari connessi all'informatica è utile vedere di capire cosa sia successo.

Alla fine del secolo scorso la logica matematica esisteva nella forma della cosiddetta algebra della logica, elaborata tra gli altri da G. Boole, C.S. Peirce e E. Schröder. Nella proliferazione di nuovi formalismi algebrici, ci si era accorti che anche alcune forme classiche di giudizi tramandati dalla tradizione logica potevano essere espressi con l'aiuto di un simbolismo algebrico. Era il simbolismo delle algebre di Boole, sostanzialmente, ma limitato all'uso di identità: ad esempio con $xy = x$ si poteva esprimere il fatto che x è contenuto in y. La restrizione alle identità era favorita dalla tradizione di considerare le leggi logiche alla stregua delle trasformazioni delle equazioni algebriche per mezzo di identità.

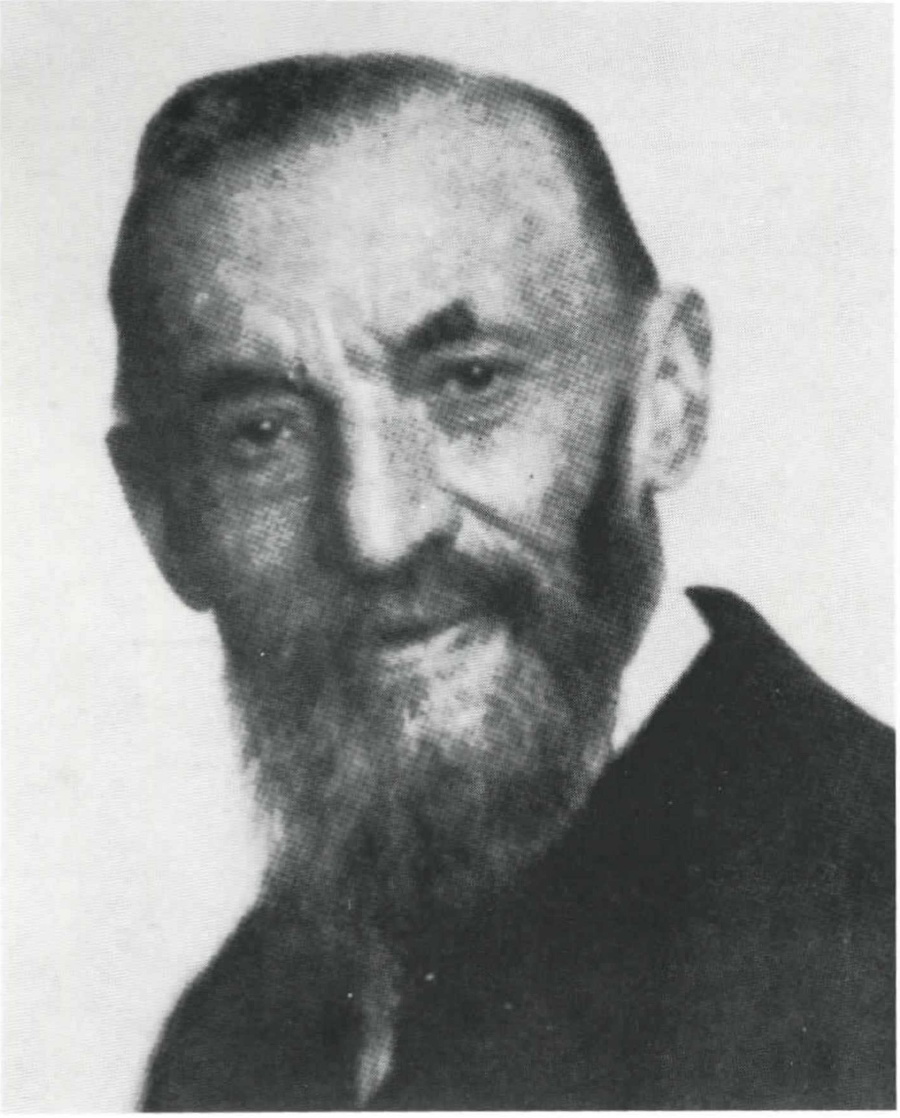
Questo formalismo era molto limitativo; quello che oggi noi esprimiamo con il quantificatore esistenziale si poteva esprimere con la affermazione che una certa classe non è vuota. Ma un calcolo con disuguaglianze offre assai più difficoltà e meno leggi che un calcolo di trasformazione di equazioni. E intanto gli sviluppi della matematica avevano fatto sì che il discorso matematico diventasse assai più complesso e articolato, e non fosse esprimibile in un simbolismo così povero.



È dovuta a G. Frege e a G. Peano la invenzione di un linguaggio simbolico duttile e adeguato alla espressione di tutti i discorsi matematici; dei due, Peano è quello che è conosciuto dal mondo matematico, anche se i sistemi di Frege, soprattutto nella parte deduttiva, erano di molto superiori. Aggiungendo al formalismo algebrico degli operatori per fare affermazioni singolari (in sostanza i nostri quantificatori), Peano era in grado nel 1889 di presentare negli *Arithmetices Principia, novo methodo exposita* per la prima volta una teoria matematica completa (per quel che riguarda naturalmente i teoremi fondamentali iniziali) interamente scritta in linguaggio simbolico. Esaltato da questa possibilità, che realizzava un antico sogno leibniziano, Peano si dedicava a una impresa colossale, raccolta nel *Formulario mathematico*, una opera ingiustamente dimenticata: scrivere tutta la matematica esistente in forma compatta, completa di dimostrazioni, nel suo linguaggio simbolico.

L'uso di un linguaggio artificiale rispondeva per Peano (e in parte, ma con motivazioni diverse, per Frege) alla esigenza di chiarezza, di eliminazione delle ambiguità del linguaggio comune e quindi della intuizione non esplicitata. Valori ormai condivisi universalmente dopo la vicenda delle geometrie non euclidee. Per realizzare la traduzione simbolica di una teoria era necessario individuare le idee primitive, in numero minimo, ed assegnare ad esse dei simboli primitivi, che costituivano come

Giuseppe Peano nacque a Spinetta, vicino a Cuneo nel 1858 e morì a Torino nel 1932. Fu uno dei più grandi matematici italiani, le cui idee hanno precorso la matematica contemporanea.



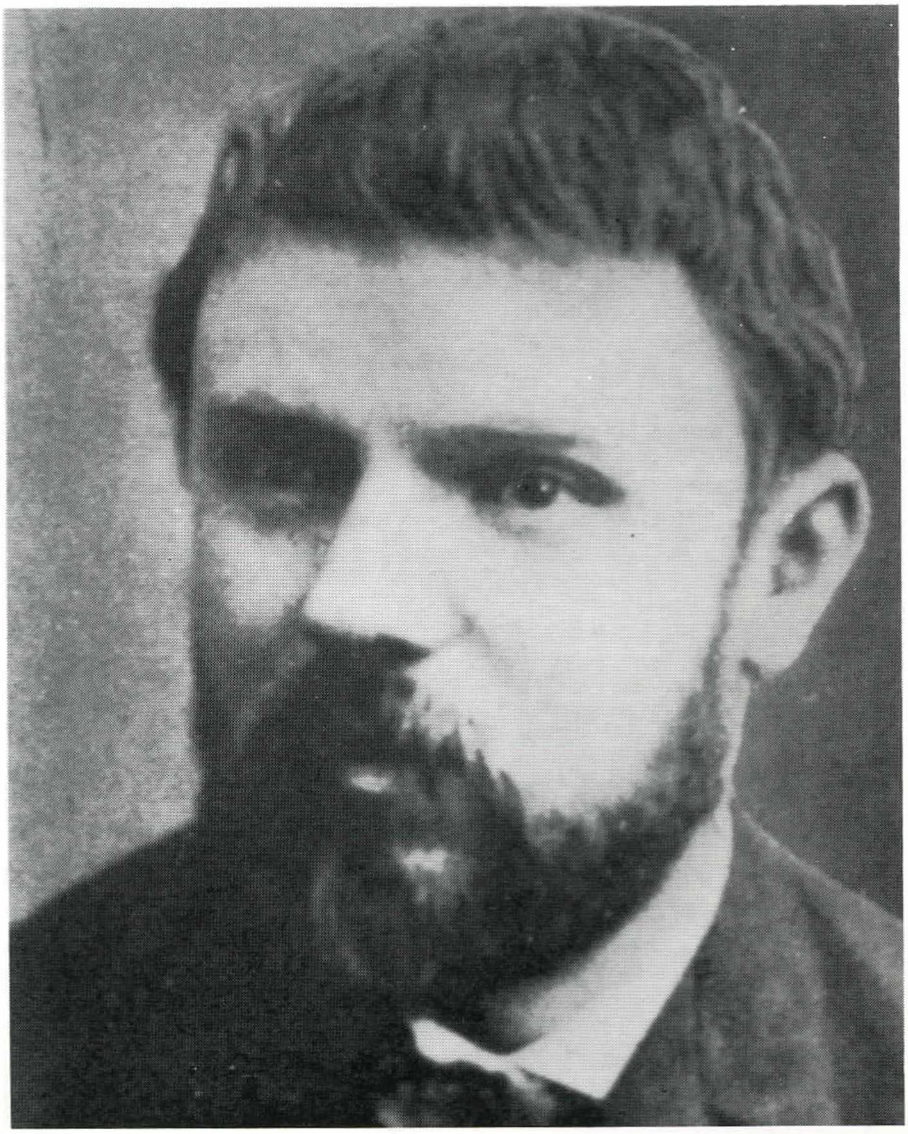
l'alfabeto particolare di quella disciplina. Il lavoro di formalizzazione si accompagnava quindi inevitabilmente a una analisi assiomatica approfondita delle teorie, che non a caso avevano condotto Peano nel 1889 alla presentazione degli assiomi per l'aritmetica che ancora portano il suo nome.

B. Russell colse con perspicacia nel 1900 la utilità del formalismo peaniano, lo fece proprio e ne prese lo spunto per iniziare la sua particolare analisi logica della matematica. La connessione che si viene a stabilire tra linguaggi formali e programma logicista provoca la ribellione di quanti erano contrari alle assunzioni di tale programma, come Poincarè. Con grande intelligenza e anticipazione Poincarè decide di attaccare lo strumento, oltre che il programma, o meglio di rendere lo strumento stesso responsabile di alcune deviazioni. Alla chiarezza e alla concisione sbandierate come meriti del formalismo, egli oppone l'esatto contrario: la mancanza di senso dei simboli artificiali che ne rendono precaria la manipolazione, e soprattutto la lunghezza dei testi formali, che rendendo il controllo difficile e defatigante favoriscono alla lunga proprio gli errori da cui quello strumento ci dovrebbe salvaguardare.

A parte il *Formulario* di Peano e il *Principia Mathematica* di Russell e Whitehead, nessun altro testo corposo è infatti scritto in modo interamente formale; alle difficoltà pratiche si aggiungono le riserve alimentate dalle critiche di Poincarè. Quasi nessuno pensa che la rigidità sintattica dei linguaggi formali ci salvi dalle contraddizioni; queste tra i matematici sono piuttosto superate con un'altra operazione concettuale, quella della assiomatizzazione della teoria degli insiemi, a opera di E. Zermelo nel 1908. Su questa base, la indagine dell'infinito riprende con fiducia e tranquillità.

Agli inizi degli anni venti, il carattere infinitistico della matematica è ormai ben impiantato nella pratica. Non solo l'Analisi, da cui è partito lo stimolo per l'elaborazione della teoria degli insiemi, sta prendendo sempre più il carattere generale e astratto di studio di spazi di funzioni; anche nell'algebra ai numeri reali o a quelli algebrici si aggiungono i campi astratti; la teoria delle strutture è ben avviata, come dimostra il fatto che alla fine del decennio apparirà il libro di van der Waerden; e l'algebra usa liberamente metodi non costruttivi come il contestato, solo pochi anni prima, assioma di scelta. La teoria stessa degli insiemi ad opera della scuola polacca sta organizzandosi in una vera teoria, cioè non serve solo come linguaggio per la formulazione di problemi di altre discipline, ma ha i problemi suoi propri, interessanti, soprattutto relativamente ai grandi cardinali infiniti e alle loro leggi.

Henri Poincaré nacque a Nancy nel 1854 e morì a Parigi nel 1912. Diede importanti contributi in vari campi della matematica.



Questa situazione può essere considerata il paradiso di Cantor di cui parlava Hilbert come di un dono che non si sarebbe più restituito. Un paradiso ancora in formazione, se non è una bestialità teologica, ma ormai ben delineato. In questa situazione le questioni dei fondamenti, intese come preoccupazioni drammatiche sulla capacità di controllare quello che si stava facendo, sarebbero già attenuate e dimenticate se non fosse per colpa di Brouwer, che con la sua costruzione di una matematica alternativa, dell'intuizionismo, tiene vivo il tarlo della cattiva coscienza.

Ma in questa situazione di esaltanti sviluppi, non c'è posto per la logica pedante e vincolante di Peano. Un ulteriore motivo per bandire questa come possibile cornice e vestito della matematica viene dalla elaborazione, che continua non senza alcuni difficili problemi, della assiomatizzazione della teoria degli insiemi. In questo contesto, è da segnalare un intervento profetico di T. Skolem; profetico per i suoi sviluppi, non per lui che si basava su un preciso risultato riguardante i formalismi logici (dell'algebra della logica arricchita con i contributi peaniani). Si tratta del teorema che porta il suo nome insieme a quello di Löwenheim, e che afferma che se una teoria in un linguaggio numerabile ha un modello, allora ne ha anche uno numerabile.

Tra le tante conseguenze, Skolem presentava quello che è noto come paradosso di Skolem: la teoria degli insiemi, nella assiomatizzazione di Zermelo e con le aggiunte opportune (di Skolem stesso e di Fraenkel) era naturalmente esprimibile in questi formalismi; se consistente, avrebbe allora un modello numerabile; ma la teoria afferma la esistenza di infiniti di ordine superiore, che dovrebbero essere presenti in una struttura numerabile.

Skolem pubblicava queste osservazioni nel 1922, anche se le aveva pronte da diversi anni, perché «in tempi recenti ho visto con sorpresa che molti matematici pensano che questi assiomi della teoria degli insiemi forniscono la fondazione ideale per la matematica; perciò mi è sembrato venuto il tempo di pubblicare una critica». La critica di Skolem è alla teoria degli insiemi, ma il suo contributo è preso invece come una ulteriore prova che la logica non deve essere introdotta nella espressione della matematica; i concetti della matematica hanno una consistenza assoluta, mentre l'uso della logica per esprimerli introduce il relativismo.

La logica formale è dunque bandita per molteplici convergenti motivi, e sarebbe forse dimenticata se Hilbert non la rivitalizzasse in una direzione del tutto nuova. Hilbert tra i grandi matematici era quello più preoccupato del *putsch* che Brouwer voleva fare nella matematica, così lo considerava Hilbert, anche perché vedeva che molti, come Hermann Weyl, lo seguono. Ed egli non era soddisfatto di come si erano assestate le cose. Dal punto di vista di Hilbert non era realizzata, per le teorie matematiche fondamentali, la condizione che egli era arrivato giustamente a porre come base per una impostazione assiomatica corretta, quella della dimostrazione di consistenza, che aveva proposto come problema nella lista di Parigi del 1900.

Tanto più drammatica era la necessità di questa dimostrazione nel caso dell'Analisi perché questa teoria faceva uso di nozioni infinitistiche che egli si rendeva conto non potevano avere una giustificazione normale, un modello intuitivo di partenza come poteva essere lo spazio fisico per la geometria euclidea. L'infinito non è di questo mondo.

La giustificazione dell'infinito per Hilbert si configurava in due momenti: da una parte appunto la dimostrazione di non contraddittorietà della teoria, ma dall'altra anche una giustificazione più generale, cioè la spiegazione di come mai nozioni elaborate dalla mente con una estrapolazione spaventosamente coraggiosa, che portava ben al di là di quello che si incontra nella realtà fisica, come mai appunto queste potessero e dovessero risultare utili e in un certo senso indispensabili per fare matematica.

Per realizzare questa operazione di soffocamento del *putsch* brouweriano e di fondazione della matematica, Hilbert elabora il suo geniale programma. È l'idea di formalizzare i testi matematici, scriverli cioè in un linguaggio artificiale estremamente preciso nella sua sintassi e nel suo ambito, allo scopo di poter svolgere su di essi dei ragionamenti matematici.

Tra i testi matematici si distinguono quelli che riguardano i numeri, le relazioni e le manipolazioni concrete che si affermano o che si fanno su di essi; questi hanno un contenuto intuitivo, come affermazioni di manipolazioni effettivamente eseguibili su stringhe di simboli (i numeri si possono rappresentare in notazione unaria come stringhe di sbarrette). Non appena si scrivono formule con variabili si passa già ad un altro livello, perché una formula con variabili è o rappresenta l'insieme dei suoi casi particolari; ancora peggio le formule con quantificatori, che esprimono pensieri apparentemente infiniti sulla totalità delle possibili combinazioni dei simboli. Ma queste formule non devono necessariamente essere prese con un contenuto, bensì solo come oggetti finiti formali costituiti anche essi da stringhe di simboli, definite da definizioni di tipo induttivo. Questi nuovi oggetti Hilbert pensa di aggiungerli a quelli di base come elementi ideali, allo stesso modo che si fanno estensioni dei campi numerici con elementi ideali, come *i*.

„ È nostra buona fortuna trovare la stessa armonia prestabilita che osserviamo così spesso nella storia della scienza, quella di cui ha beneficiato Einstein quando ha trovato il calcolo generale degli invarianti pienamente sviluppato per la sua teoria della gravitazione; scopriamo che un notevole lavoro di sgrezzatura è stato già fatto: il calcolo logico è stato sviluppato. Vero, originariamente era stato creato in un contesto del tutto differente e, di conseguenza, i suoi segni furono introdotti inizialmente per lo scopo della comunicazione soltanto; ma noi saremo coerenti nella nostra strada se ora spoglieremo anche i segni logici di ogni significato, proprio come abbiamo fatto per quelli matematici, e dichiareremo che le formule del calcolo logico non significano nulla in sè, ma sono delle proposizioni ideali.

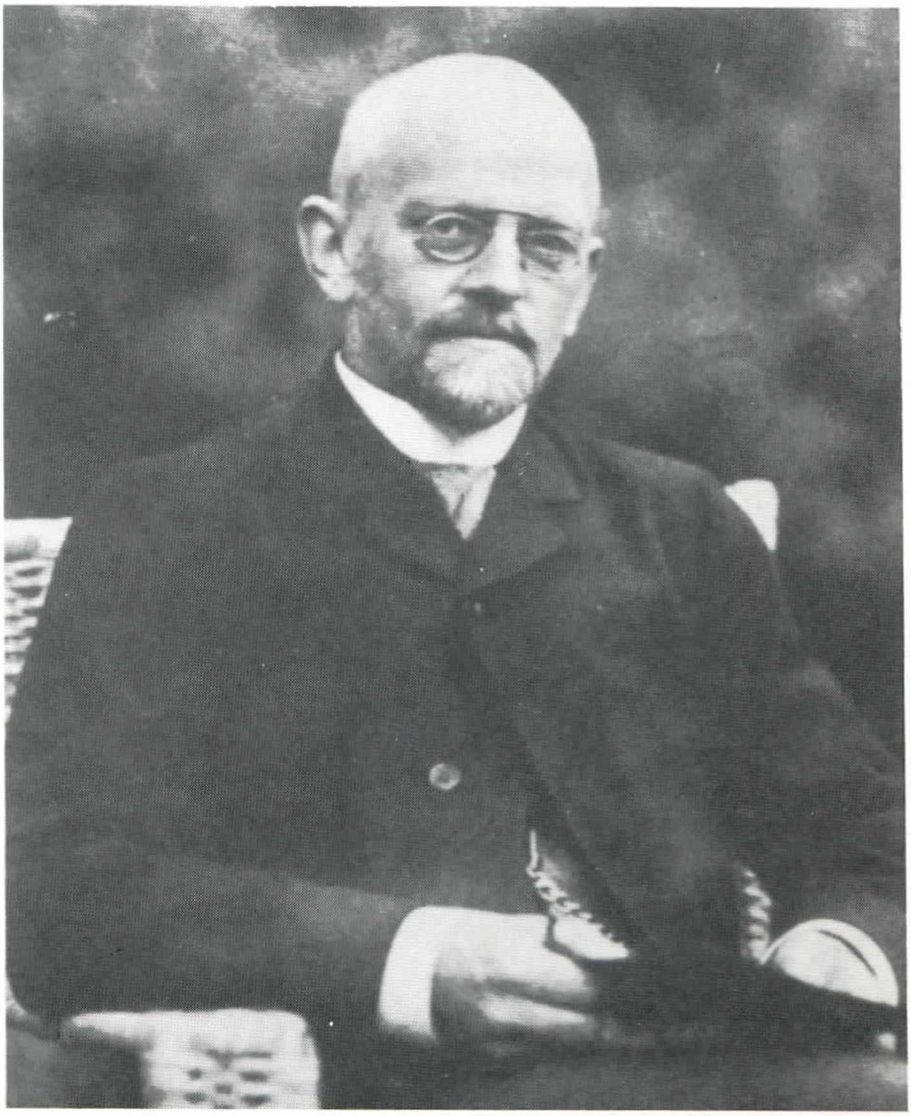
Nel calcolo logico noi possediamo un linguaggio di segni che è capace di rappresentare le proposizioni matematiche in formule e di esprimere le inferenze logiche attraverso preocessi formali... In questo modo noi finalmente otteniamo, invece della scienza matematica contenutistica che è comunicata per mezzo del linguaggio ordinario un arsenale di formule che sono formate di segni matematici e logici e che seguono l'una dall'altra secondo regole definite... Quindi l'inferenza contenutistica è rimpiazzata dalla manipolazione di segni secondo regole, e in questo modo la completa transizione da un trattamento intuitivo a uno formale è ora completa, da una parte per gli assiomi stessi... e dall'altra parte per il calcolo logico, che originariamente doveva essere solo un altro linguaggio. „

Vedete come è rovesciata l'impostazione peaniana; si potrebbe dire che con Hilbert si passa da un uso della logica, per la comunicazione, a una menzione della logica: è una distinzione che la logica stessa ama fare nella sua analisi del linguaggio, quando ad esempio si osserva che la parola «Trento» è usata nella frase che dice che Trento è bella, mentre è menzionata quando si dice che «Trento» ha sei lettere. La logica è usata per la traduzione del discorso in una struttura matematica, costituita da parole di un alfabeto finito, che si può studiare matematicamente. È un oggetto di studio matematico particolare, perché costituito di oggetti finiti, le espressioni del linguaggio intese come sequenze di simboli dell'alfabeto, definiti e manipolati da definizioni e operazioni di carattere induttivo. Le tecniche da usare per tale studio si collocano dunque nell'ambito della aritmetica, o della teoria degli insiemi finiti, e probabilmente a un livello basso di complessità, perché la trattazione dei linguaggi ha un carattere effettivo, concreto (in parte dovuto forse alla necessità della comunicazione intersoggettiva).

L'idea Hilbert la aveva avuta già nel 1904, ma non aveva avuto occasione, o stimoli adeguati per portarla avanti. Questa è la data comunque della nascita, sia pure con un solo esemplare particola-

David Hilbert nacque a Königsberg (ora Kalingrad) nel 1862 e morì a Göttingen nel 1943. Nell'anno 1900 al Congresso internazionale dei Matematici in Parigi presentò un elenco di 23 problemi che assegnava, sono parole sue, come compito ai matematici del XX secolo.

re, della teoria matematica dei linguaggi, che sarà ripresa intensamente a partire dal 1917 (la citazione di sopra è del 1925, dal famoso articolo sull'infinito).
Da questa osservazione nasce la ipotesi coraggiosa che sia possibile studiare con dei metodi matematici ma di tipo elementare, sicuro la struttura delle teorie che nella loro formulazione intuitiva pretendono di parlare anche di oggetti e argomenti di carattere molto più astratto e impegnativo, come l'infinito.
Per la dimostrazione di consistenza appare ora disponibile un nuovo strumento, matematico e semplice: come i linguaggi e i calcoli sono presentati da definizioni induttive, così la dimostrazione che ogni teorema ha una certa proprietà può essere impostata induttivamente, dimostrando prima che gli assiomi ce l'hanno, e poi dimostrando che le regole del calcolo la conservano e la trasmettono. Dovrebbe essere possibile dimostrare che nessuna sequenza costruita secondo le regole termina con una contraddizione.
Formulando esattamente i due problemi che gli stanno a cuore, sulle teorie dell'infinito, Hilbert scopre poi che sono equivalenti: che se si riuscisse a dimostrare, in una aritmetica elementare, la consistenza dell'Analisi o della teoria degli insiemi, allora si dimostrerebbe anche che queste sono estensioni conservative dell'aritmetica per quel che riguarda le affermazioni concrete aritmetiche, quelle che si riferiscono solo a numeri e non usano i quantificatori; questo vuol dire che ognuna di queste affermazioni che fosse dimostrata in una teoria superiore, facendo riferimento alle entità astratte, sarebbe anche già dimostrabile nella teoria elementare senza assunzioni problematiche. Ma questo particolare tecnico non ci interessa, anche se è quello che



ha fatto mettere ancora più fortemente l'accento sul problema della consistenza, e ha forse rafforzato le speranze di Hilbert di essere sulla strada giusta.
Hilbert è troppo modesto nel dire che il calcolo logico è fornito dalla tradizione; in effetti la impostazione rigorosa della sua *teoria della dimostrazione* richiede un perfezionamento del calcolo, che culmina nella presentazione nel 1928, in un volumetto classico di Hilbert e di W. Ackermann, di un calcolo logico per la logica del primo ordine come oggi la conosciamo, insieme a una serie di problemi di base ancora aperti.
Il primo problema è quello della completezza: dimostrare che una proposizione valida in tutte le strutture in cui valgono certi assiomi (che è quello che si intende per teorema nella impostazione assiomatica) è anche dimostrabile a partire da quelli con una dimostrazione logica, sequenza di trasformazioni basate sulle regole pre-

Per ulteriori approfondimenti dei temi trattati si veda:
G. LOLLI, *Le ragioni fisiche e le dimostrazioni matematiche,* Il Mulino, Bologna, 1985;
G. LOLLI, *La Macchina e le dimostrazioni,* Il Mulino, Bologna, 1987.
G. LOLLI *Lezioni di logica matematica*, Boringhieri, 1978;
J. W. LLOYD, *Fondamenti di programmazione logica*, Franco Muzzio Editore, 1986 e A. BUNDY, *L'automazione del ragionamento matematico*, Franco Muzzio Editore, 1986.

scritte dal calcolo logico. Prima di Hilbert questo problema non si era mai posto con tale nettezza, perché mai si era pensato di staccare così radicalmente la forma delle proposizioni, quello che restava quando si scrivevano in un linguaggio artificiale per poterle elaborare matematicamente, dal loro contenuto intuitivo. Il teorema di completezza è dimostrato da Gödel nella sua tesi del 1930.

Ci sono poi altre questioni, che vengono naturali non appena uno ha trovato uno strumento così promettente; perché limitarsi a considerare la consistenza? Ci sono altri problemi riguardanti la attività logica, che Hilbert assegna come compito della nuova disciplina: lo studio del rapporto tra contenuto e forma, la lunghezza delle dimostrazioni, l'esistenza o meno di un metodo di decisione per tutti i problemi matematici. Nel libro del 1928 il problema centrale è dichiarato essere quello della decisione per la validità logica delle formule dei linguaggi del primo ordine, l'*Entscheidungsproblem*. Se esistesse un tale metodo, allora siccome è mostrato praticamente nel libro stesso come molte nozioni matematiche si possano esprimere in tali linguaggi, ne verrebbe un metodo di decisione per la matematica.

Ma che cosa è un metodo di decisione? Il problema si connette a un altro che viene in evidenza in modo naturale man mano che si cercano di precisare gli strumenti matematici di tipo concreto ed effettivo che si devono usare nella manipolazione dei linguaggi: è il problema di precisare in modo rigoroso cosa si intenda per quei processi che nel corso della storia della matematica sono stati via via chiamati metodi effettivi, algoritmi, procedimenti effettivamente calcolabili.
La necessità di chiarire questa nozione intuitiva emerge per diversi motivi: intanto l'uso sempre più impegnativo di strumenti che soddisfano intuitivamente questa caratterizzazione, ma che non bastano mai, man mano che dai frammenti più facili di logica, come la logica proposizionale, si passa a frammenti più ampi di logica e di aritmetica; il tipo di pensiero ricorsivo, o induttivo, che è implicito nella definizione delle funzioni aritmetiche elementari, e che porta alla delimitazione delle cosiddette funzioni primitive ricorsive, ha bisogno di una migliore caratterizzazione e delimitazione. Ackermann scopre nel 1928 una funzione che è intuitivamente calcolabile, definita da un sistema di equazioni ma che sfugge ai più ristretti metodi definitori delle funzioni primitive ricorsive.
D'altra parte, quando si profila la possibilità di una risposta negativa, la non esistenza di un metodo, la risposta ha bisogno di una delimitazione matematica rigorosa dell'ambito entro cui restringere i metodi in questione, perché la dimostrazione deve in qualche modo passarli in rassegna tutti per escludere che funzionino. Nella storia recente della matematica c'erano esempi analoghi, la non risolubilità per radicali delle equazioni algebriche, la non risolubilità con riga e compasso di certi problemi; quest'ultimo aveva richiesto la formulazione algebrica, e non meccanica, del concetto di risoluzione con riga e compasso. Se si dovesse arrivare ad una risposta negativa quanto alla decidibilità effettiva di questioni logiche, occorrerebbe precisare in modo rigoroso i metodi effettivi ammissibili nella manipolazione degli oggetti formali.
Alla fine degli anni venti ci troviamo allora in questa situazione: l'uso della logica come linguaggio matematico non ha avuto successo, ha prevalso la sensazione della sua impraticabilità, e i dubbi sulla sua adeguatezza per il pensiero matematico; ma la logica come metodo per sottoporre le teorie e la dimostrazione a uno studio matematico, attraverso la traduzione del discorso in un insieme di formule del tutto prive di significato, e collegate solo da regole di trasformazione esplicite, la logica come oggetto di studio matematico insomma, ha preso piede. Attraverso questa deformazione, anche l'uso della logica come linguaggio dovrà tornare in onore, anche se adesso non se ne vede ancora la via. Una prima pietra sarà costituita dal teorema di completezza, che inizialmente passa in secondo piano rispetto al teorema di incompletezza per l'aritmetica che subito lo segue ma di cui non dobbiamo occuparci.
Una esigenza più viva è invece quella della precisazione del pensiero ricorsivo, o della definizione di algoritmo, a cui si dedicheranno i logici negli anni immediatamente seguenti.

Computabilità e complessità

di Annalisa Marcja

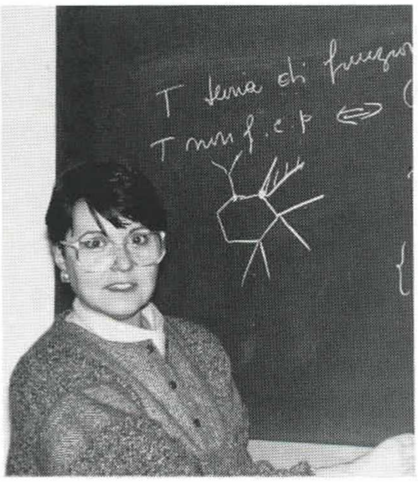
È stato già osservato che molti problemi matematici prendono la forma: trovare un algoritmo (o procedimento meccanico) per mezzo del quale determinare in un numero finito di passi per ogni elemento di un dato insieme se o no l'elemento possiede qualche data proprietà.

Se forniamo un tale procedimento (anche senza aver dato rigorosamente la definizione del concetto di algoritmo) abbiamo trovato una soluzione al dato problema e diciamo che il problema è *risolubile* (decidibile).

In teoria dei numeri abbiamo molti esempi di soluzioni a questo genere di problemi. È questo un fatto generale della matematica? Consideriamo per esempio il seguente problema: (X problema di Hilbert) «C'è un algoritmo tale che data una qualunque equazione polinomiale $P(x_1,...,x_n) = 0$ a coefficienti interi, decida se questa ha o no soluzioni intere?». Ovviamente l'algoritmo è noto in casi particolari. L'indiano Brahmagupta (VII secolo) fornì un algoritmo per decidere se l'equazione $ax + by + c = 0$ ha soluzioni intere o no e la procedura consiste nel verificare se il massimo comun divisore (che si calcola usando per esempio l'algoritmo euclideo) di a e b divide c.

Notiamo che il X problema di Hilbert è equivalente a

«Esiste un algoritmo tale che data



una qualunque equazione polinominale $P(x_1,...,x_n) = 0$ a coefficienti interi decida se $S_p = \{(x_1,...,x_n) : x_1,...,x_n$ naturali e $P(x_1,...,x_n) = 0\}$ è vuoto oppure no?».

Una prima risposta parziale può essere: esiste un algoritmo che elenca tutti gli elementi di S_p (per esempio ordinando le n-ple di numeri naturali e verificando per ciascuna se è soluzione), cioè S_p è semidecidibile. Se S_p è non vuoto, non resta che aspettare una soluzione. In molti casi la più piccola soluzione può essere molto grande. Per l'equazione $x^2 - 991y^2 + 1$ questa è

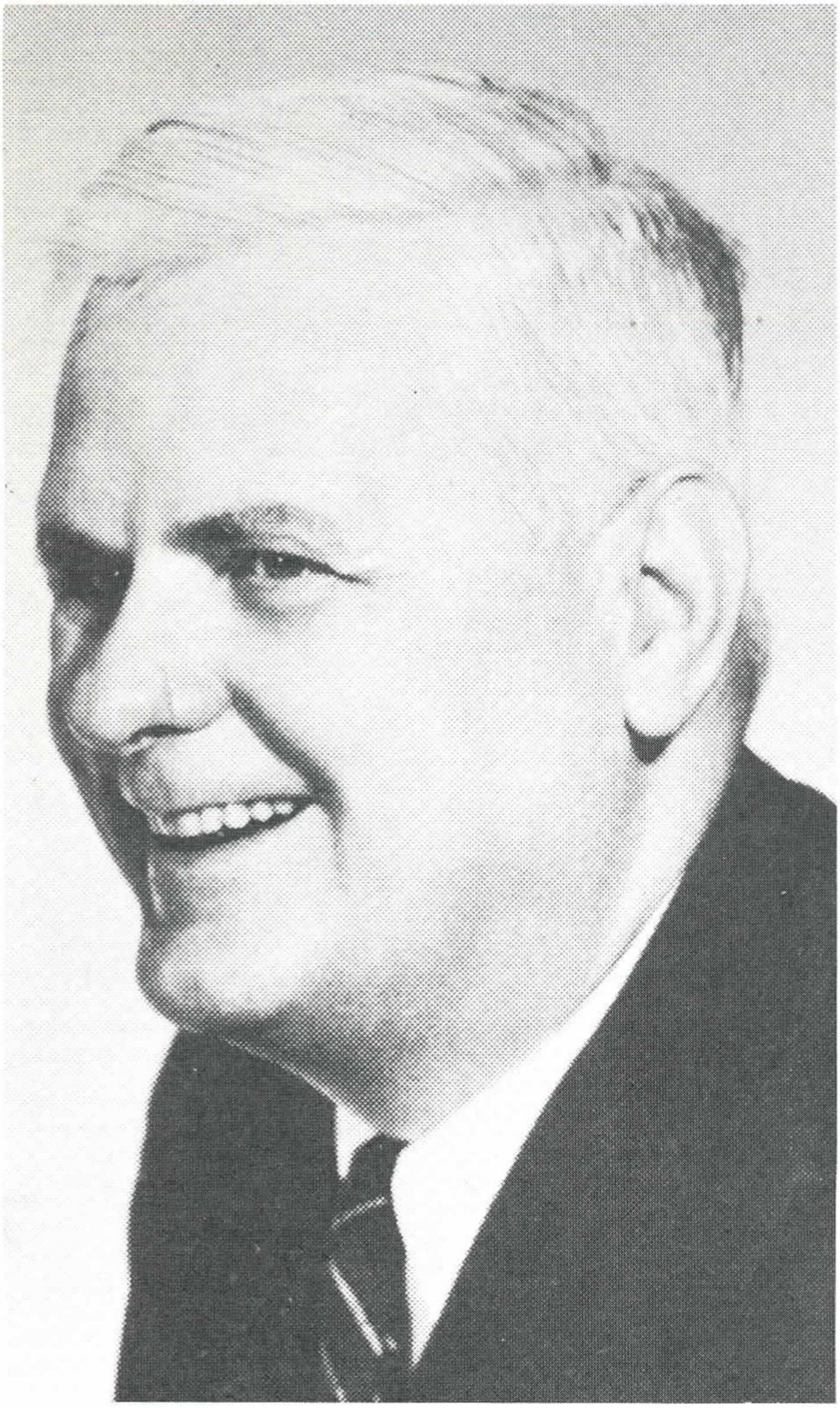
$x = 379516400906811930638014896080$

$y = 12055735790331359447442538767$.

Più in generale Hilbert si poneva la domanda dell'esistenza di un procedimento meccanico per risolvere ogni problema matematico. Assumendo che tutta la matematica classica sia esprimibile nella logica del I ordine, questo diventa il problema di determinare un algoritmo tale che dato un qualunque insieme finito di enunciati T ed un enunciato ϕ della logica del I ordine decida se o no ϕ è dimostrabile da T (Entscheidungsproblem). Questo per Hilbert era il problema fondamentale perché l'esistenza di un tale procedimento meccanico sarebbe potuto essere usato per risolvere algoritmicamente tutti i problemi matematici. Ovviamente una risposta negativa (cioè la dimostrazione di *irrisolubilità*) per un problema particolare (per esempio per il X) risponderebbe negativamente anche a quest'ultimo problema. Per dire però che un problema sia irrisolubile (o indecidibile) dobbiamo dapprima concordare che cosa sia un procedimento meccanico.

Il giovane matematico Alan Turing (1912-1954) del King's College di Cambridge nel 1936 diede una tale definizione rigorosa e fornì un esempio di problema irrisolubile (A. Turing, On computable numbers with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society ser. 2, vol. 42 [1936-37]).

Alonzo Church in una fotografia del 1966.



Nello stesso anno un'altra definizione equivalente era stata data dall'americano A. Church (A. Church, On unsolvable problem for elementary number theory. The American Journal of Mathematics, vol. 58 [1936] – A. Church, A note on the Entscheidungsproblem. Journal of Symbolic Logic 1 [1936]) ma a noi interessa descrivere quella di Turing, per l'influenza avuta sulla moderna Scienza dei Calcolatori.

Turing introdusse un computer ideale (oggi chiamato *macchina di Turing*) per simulare il comportamento umano della computazione. Una macchina di Turing consiste di una testina che scorre su un nastro suddiviso in celle. Questo nastro è potenzialmente infinito, nel senso che può essere sempre aggiunta una nuova cella a destra o a sinistra. In ogni cella c'è un solo simbolo dell'alfabeto di macchina $s_0, s_1, \ldots, s_k$ e la testina legge una cella alla volta. (Per convenzione s_0 rappresenta il simbolo vuoto).

Quello che la macchina farà successivamente, muoversi di una cella a destra o a sinistra o scrivere un nuovo simbolo nella cella dipende dallo stato interno della macchina e dal simbolo letto. I simboli per gli stati siano $\{q_1, q_2, \ldots, q_N\}$.

Il comportamento della macchina nello stato q_i che legge il simbolo s_j può allora essere:

1) rimpiazzare il simbolo s_j con il simbolo s_k e andare nello stato q_e. Questo può essere rappresentato con la quadrupla $q_i s_j s_k q_e$

2) muovere la testina di una cella a sinistra e passare nello stato q_e; in simboli $q_i s_j S q_e$

3) muovere la testina di una cella a destra e passare nello stato q_e; in simboli $q_i s_j D q_e$

Definizione

Una macchina di Turing è un insieme finito di quadruple del tipo descritto sopra. Essa si dice *deterministica* se mai due quadruple iniziano con la stessa coppia $q_i s_j$, *non deterministica* altrimenti.
Assumiamo che una macchina di Turing cominci sempre nello stato q_1 e che si fermi se è nello stato q_i, legge il simbolo s_j e non c'è nessuna quadrupla che cominci con $q_i s_j$. Una macchina di Turing non deterministica T permette l'esecuzione di *uno* dei possibili movimenti. Questo permette più «cammini» possibili. Qualcuno potrà essere terminale, qualche altro no.
Una macchina di Turing può computare una particolare funzione con dominio contenuto in N^n e a valori in N nel seguente modo: Si pone l'input $x_1,\dots,x_n$ codificato nel linguaggio $\{s_0, 1\}$ sul nastro

	1	1	...	1	s_0	1	...	1	s_0	...	s_0	1	...	1

$\underbrace{1\ 1\ \dots\ 1}_{x_1+1}$ s_0 $\underbrace{1\ \dots\ 1}_{x_2+1}$ $s_0 \dots s_0$ $\underbrace{1\ \dots\ 1}_{x_n+1}$

La testina è posizionata sul primo 1 a sinistra e la macchina è nello stato q_1.
Nel caso deterministico se $(x_1,\dots,x_n)$ è nel dominio di f la macchina si ferma e il numero di 1 presenti nel nastro nella configurazione terminale (output della computazione) coincide con $f(x_1 - x_n)$. Se $(x_1,\dots,x_n)$ non è nel dominio di f la macchina non si ferma.
Una funzione f è *Turing computabile* se c'è una macchina di Turing (deterministica) che la computa. Una funzione si dice computabile non deterministicamente se c'è una macchina di Turing non deterministica tale che ogni volta $(x_1,\dots,x_n)$ è nel dominio della funzione c'è almeno un cammino che si ferma e tutti quelli che si fermano, si fermano sullo stesso output e l'output è $f(x_1,\dots,x_n)$. Altrimenti ogni possibile cammino è non terminale.

Ovviamente una funzione è computabile se e solo se è non deterministicamente computabile.
A questo punto è necessario soffermarsi sul concetto intuitivo di computabile (o di algoritmo) e quello di Turing computabile. La *tesi di Church-Turing* afferma che il concetto intuitivo di procedimento meccanico (algoritmo) può essere precisato definendo procedimento meccanico una computazione effettuabile da una macchina di Turing. Ciò non può essere dimostrato (si ha da una parte un concetto intuitivo e vago) tuttavia esistono parecchi elementi a sostegno di questa tesi.
Con una codifica opportuna si può dare una corrispondenza biunivoca (effettiva) fra macchina di Turing e numeri naturali.
Possiamo definire allora la funzione
$\Phi(x,z)$ = la funzione computata della macchina di Turing di numero z sull'input x.
Φ è computabile da una macchina di Turing U, detta universale, perché può simulare il comportamento di una qualunque macchina di Turing.
U provvede un modello suggestivo di «all-purpose computer» in cui i dati e i programmi sono raccolti in una memoria e ne fu un'anticipazione. Di fatto sembra che la costruzione nel 1945 del moderno computer fosse proprio ispirata all'idea di macchina universale di Turing.
Consideriamo adesso il seguente problema: «C'è un algoritmo tale che data una qualunque macchina di Turing T (di numero z) e un input x determina se la macchina T si ferma sull'input x oppure no?». Questo è il cosiddetto *«problema dell'arresto»* per le macchine di Turing.
Per la tesi di Turing-Church questa domanda può essere posta in termini più precisi nella forma:
«È Turing computabile la funzione

$$g(x,z) = \begin{cases} 1 \text{ se } \Phi(x,z) \text{ è definita} \\ 0 \text{ se } \Phi(x,z) \text{ non è definita?»} \end{cases}$$

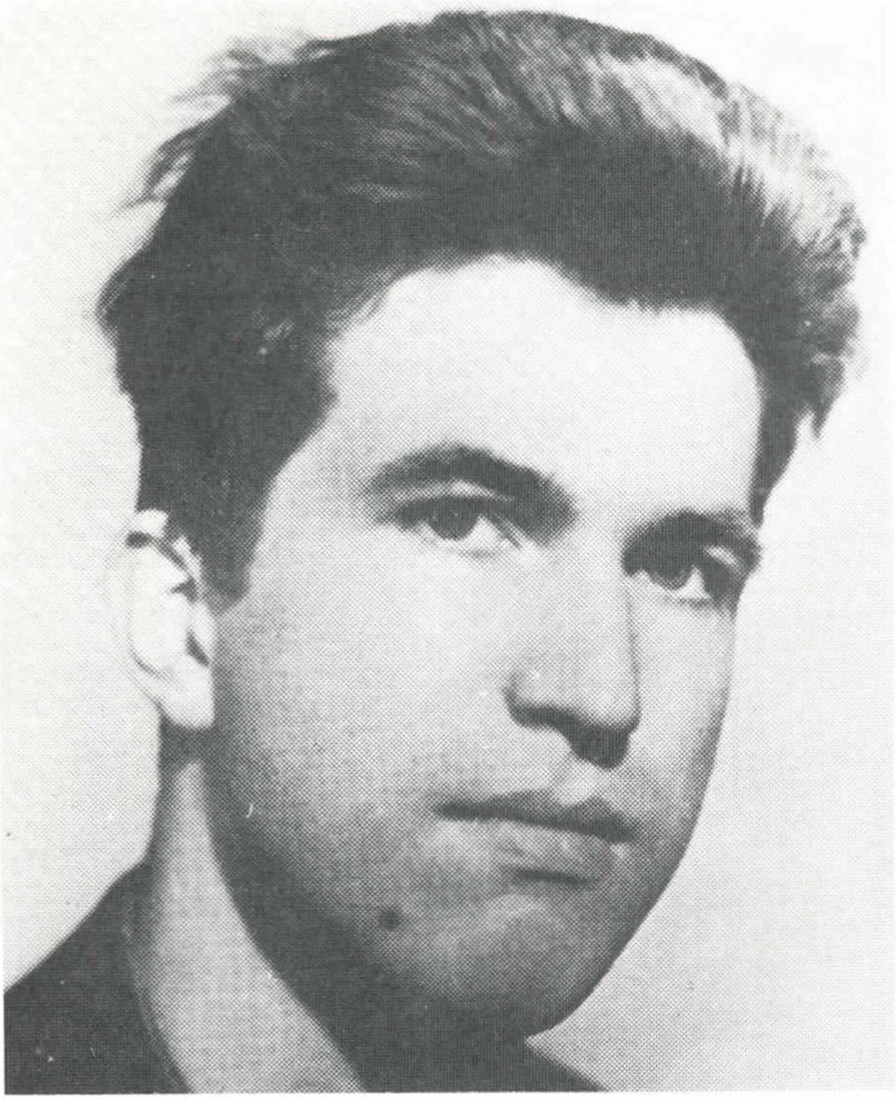
Supponiamo che g sia Turing computabile. Definiamo la funzione

$$\psi(x) = \begin{cases} 1 \text{ se } g(x,x) = 0 \\ \text{non definita se } g(x,x) = 1 \end{cases}$$

Anche ψ è Turing computabile, se g lo è.
Sia y_0 il numero della macchina che computa ψ.
Dalla definizione di Φ segue che: $\Phi(y_0, y_0)$ è definita se e solo se $g(y_0, y_0) = 0$ se e solo se $\Phi(y_0, y_0)$ non è definita.
Abbiamo dimostrato quindi, per assurdo, che la funzione g non è Turing computabile e quindi che il problema dell'arresto è irrisolubile.

Questo primo risultato di irrisolubilità è stata la base di molti altri risultati di questo genere. Le tecniche di dimostrazione sono del tipo «se il dato problema fosse risolubile allora sarebbe risolubile il problema dell'arresto». Con queste tecniche per esempio si è dimostrato l'irrisolubilità del problema della parola per i semigruppi, per i gruppi e l'irrisolubilità del X

Yuri V. Matijasevič, sovietico, in una fotografia del 1970. All'età di 22 anni egli riuscì a dare una soluzione al X problema di Hilbert.



problema di Hilbert già menzionato (Matijasevič 1970).
La macchina di Turing era servita per fornire una risposta negativa al problema di Hilbert e la moderna teoria della computabilità ci rende capaci di distinguere chiaramente e precisamente tra problemi per cui ci sono algoritmi e problemi per cui non ci sono. Però c'è una grande differenza tra la risolubilità «in principio» e la risolubilità «in pratica». Usualmente i problemi risolubili anche in pratica vengono chiamati *trattabili*, quelli risolubili in principio, ma non in pratica *intrattabili*. Il problema della soddisfacibilità della logica proposizionale (*SAT*) è notoriamente risolubile. Un algoritmo potrebbe essere quello delle tavole di verità. Però non sappiamo se un tale problema sulla base di un qualunque algoritmo sia trattabile. Infatti come è noto queste procedure richiedono un numero di passi che è una funzione esponenziale della lunghezza dell'espressione. È proprio per la crescita rapida della funzione esponenziale che queste procedure esauriscono rapidamente le risorse disponibili. Poiché una funzione esponenziale cresce più rapidamente di una qualunque funzione polinomiale, si può pensare a un problema come trattabile se il risolverlo richiede un numero di passi limitato da un polinomio nella lunghezza dell'input.

Definizione.

Una funzione f è detta *computabile (non deterministicamente computabile)* in tempo polinomiale se c'è una macchina di Turing (*non*) deterministica che computa f ed un polinomio $p(n)$ tale che il numero di passi nella (*in almeno una*) computazione di T con input x è $\leqslant p(x)$.
Alla luce della tesi di Turing-Church a un problema risolubile è associata una funzione Turing computabile. Per esempio a SAT è associata la funzione:

$$S(x) = \begin{cases} 1 \text{ se la formula il cui numero è } x \text{ (in una opportuna codifica) è soddisfacibile} \\ \text{non definita altrimenti.} \end{cases}$$

Indicheremo con P la classe dei problemi risolubili per cui la funzione associata è computabile in tempo polinomiale.
Un analogo della tesi di Church è la *Tesi di Cook-Karp*: Un problema è trattabile se e solo se è in P.
Indichiamo invece con NP la classe dei problemi risolubili per cui la funzione associata è computabile non deterministicamente in tempo polinomiale. Ovviamente $P \subseteq NP$ e non è conosciuto alcun problema di NP non in P, quindi potrebbe essere $P = NP$. Si può provare che $SAT \in NP$ e che se $SAT \in$ P allora $P = NP$. Sono i problemi come SAT (cosidetti NP-completi) che sono pensati intrattabili. Ma come già detto è ancora aperto però il problema
$P = NP$?

Il dopoguerra: il calcolatore e le dimostrazioni

di Gabriele Lolli

Abbiamo lasciato la logica nel 1930 rafforzata con il teorema di completezza per i calcoli logici e alle prese con la definizione del pensiero ricorsivo. Negli anni seguenti si accumulano molti risultati, che costituiscono il corpo della moderna disciplina della logica matematica, e molti anche utili per la matematica stessa, ma l'uso della logica come linguaggio della matematica resta tabù: è troppo diversa dal linguaggio naturale che serve per comunicare, troppo rigida, domanda uno sforzo e uno spreco che non è nella direzione della scoperta matematica. Ma adesso ci sono i calcolatori, no?

Tutti coloro che nella storia hanno costruito o anticipato o sognato le macchine calcolatrici (Pascal, Leibniz, quelli che nella prossima lezione saranno detti della prima generazione) hanno sempre pensato di affidare alle macchine delle funzioni eminentemente logiche, anche se poi si dovevano limitare a realizzare le quattro operazioni. L'equiparazione di ragione e calcolo percorre la storia del pensiero occidentale, a partire da Hobbes, e ne è rimasta traccia nella lingua inglese, dove *to reckon* significa calcolare ma anche concludere. L'idea si inseriva e discendeva dalla generale visione meccanicistica dell'universo, si appoggiava ai meccanismi degli orologi e degli automi-bambole. È importante che con il venir meno di tale visione generale l'ipotesi resti in piedi per tutt'altre motivazioni ed esperienze.

Quando Turing aveva concepito l'idea di una macchina universale, per dimostrare la non risolubilità algoritmica del problema della validità logica e di altri problemi riguardanti la attività computazionale stessa, non aveva in mente solo un artificio per ottenere risultati negativi; per Turing la macchina universale era un concetto positivo che forniva le basi teoriche per quello che sarebbe stato il progetto di tutta la sua vita, la costruzione di un *cervello*.

È naturale che ci sia una resistenza, una ripulsa quasi a considerare meccaniche le funzioni umane più alte, e soprattutto da parte dei matematici che sanno bene quanto tortuosa, non garantita, eroica e qualche volta casuale sia la elaborazione di una dimostrazione. Ma ci sono aspetti non secondari che spingono a ritenere fondato il modello di dimostrazione fornito dai calcoli logici.

C'è innanzi tutto un fatto psicologico, sperimentato da tutti, che quando si deve far capire una dimostrazione che non viene capita la si ripete diluendo i suoi passi in una successione di passi più semplici. Con passi più semplici si intende passi che fanno più esplicito appello a conoscenze precedenti o condivise dall'ascoltatore; in linea di principio si dovrebbe poter scendere a passi così semplici che non contengano più alcuna conoscenza matematica...

Tale base psicologica è forse a fondamento della teorizzazione di questo modello compiuta da Descartes, e che è passata nella tradizione:

> Io mi dilettavo soprattutto della matematica, a causa della certezza ed evidenza delle sue ragioni...
> Quelle lunghe catene di ragioni, tutte semplici e facili, di cui i geometri sono abituati a servirsi per pervenire alle loro dimostrazioni più difficili, mi avevano dato occasione di immaginare che tutte le cose che possono cadere sotto la conoscenza dell'uomo si raggiungono nello stesso modo, e che purché ci si astenga dall'accettarne alcuna per vera quando non lo è, e che si rispetti sempre l'ordine che ci vuole per dedurle le une dalle altre, non possono essercene di così lontane che alla fine non le si raggiunga, né di così nascoste che alla fine non le si scopra.

Su questo modello si innestano i calcoli logici, con i loro passi elementari di manipolazione sintattica; ma con la definizione delle macchine di Turing i piccoli passi perdono la loro giustificazione psicologica e assumono quella di fondamento del processo algoritmico. Calcolo logico e processo

Alan Mathison Turing nacque a Londra nel 1912 e morì a Wilmslow nel 1954. Egli introdusse un computer ideale (macchina di Turing) per simulare il comportamento umano della computazione.

algoritmico si identificano, e il primo trova la sua legittimazione nel secondo.

Innanzi tutto, tra le definizioni equivalenti della classe delle funzioni calcolabili ce ne è anche una di carattere logico: le funzioni calcolabili sono quelle definibili da sistemi di equazioni i cui valori si ottengono *derivando* una opportuna equazione numerica dalla definizione, con prescritte e ristrette regole sintattiche di sostituzione. Viceversa una volta che si hanno i concetti della teoria della calcolabilità si vede che la logica si colloca in una posizione cruciale.

A fianco del concetto di decidibilità, e di algoritmo di decisione, un concetto non meno importante è quello di semidecidibilità, associato a quello di algoritmo di decisione parziale. Un algoritmo è tale quando di fronte a ogni caso del problema dà una risposta in un numero finito, per quanto imprevedibile, di passi se la risposta è di un tipo, diciamo positiva, mentre non dà nessuna risposta, lavorando all'infinito e senza che sia mai possibile scoprire che l'attesa è senza fine, nel caso opposto. È facile vedere che in tale caso si dispone anche di algoritmi di enumerazione di tutte le soluzioni positive, una macchina che sputa fuori uno dopo l'altro tutti i casi positivi, supposti infiniti, in un ordine non conosciuto, ad esempio non in ordine di grandezza se si tratta di numeri, altrimenti si ricadrebbe nella decidibilità. Questi problemi si chiamano anche per ovvi motivi ricorsivamente enumerabili o generabili.

Si immagini una macchina che elenca una dopo l'altra tutte le soluzioni di una equazione, mentre non esiste però una macchina che di fronte a una proposta soluzione sia sempre in grado di rispondere in un numero finito di passi se si tratta proprio di una soluzione o no. Un esempio specifico è quello della logica del primo ordine, il problema della validità logica delle formule: non è decidibile per il teorema di Church, ma è semidecidibile. L'algoritmo di semidecidibilità si ottiene descrivendo una procedura che a partire dagli assiomi e usando le regole genera una dopo l'altra tutte le possibili sequenze di formule, accettando solo quelle che nel loro legame interno sono derivazioni ed emettendo la conclusione.

Si vede allora che se la logica non è decidibile, se non è realizzabile il sogno leibniziano del *calculemus*, almeno è importante e consolante che sia semidecidibile: la chiarezza, la intercomunicabilità, la universalità che si è sempre richiesta alla logica pretende che questa abbia un carattere effettivo, e non misterioso, indominabile. Se con il nostro modello riusciamo a collocare la logica a questo livello, abbiamo motivo di credere di aver colto degli aspetti essenziali.

Il livello della semidecidibilità è un livello di non decidibilità che è il meno non effettivo possibile; al di sopra di esso si eleva una gerarchia infinita di classi di problemi caratterizzati da gradi diversi di non effettività. I problemi decidibili sono in un certo senso banali; una volta provata la decidibilità gli unici problemi riguardano eventualmente la loro trattabilità, la esistenza e il ritrovamento di algoritmi efficienti. I problemi matematici e logici più interessanti che non sono decidibili sono semidecidibili; qui, prendendo ad esempio tipico quello della generazione di tutte le possibili derivazioni, i problemi teoricamente interessanti che si pongono sono quelli di cortocircuitare la attesa passiva della uscita di quello che ci interessa, di trovare delle strategie di esplorazione dell'albero infinito di tutte le possibilità che siano efficienti se pure esaustive, di introdurre elementi euristici che facciano imboccare più rapidamente la strada utile. Altrimenti si verifica presto la cosiddetta esplosione combinatoria: ammettendo l'ipotesi semplificatrice che a ogni stadio di una dimostrazione l'ultima formula sia combinabile con due assiomi, per non parlare delle formule intermedie ottenute, si vede che una sequenza di dieci formule, che è invero molto corta, si trova come nodo al livello dieci di un albero binario, e una ricerca esaustiva dell'albero richiede la produzione di *2* elevato a *10* nodi.

Questo della esplosione combinatoria è l'argomento più forte che sostiene la rinnovata opposizione all'uso dei calcoli logici. Quando negli anni cinquanta sono disponibili calcolatori sufficientemente potenti, subito vengono fatti esperimenti di dimostrazione automatica. Ma i primi programmi per dimostrare teoremi preparati alla scuola di Intelligenza Artificiale del MIT, la *Logic Theory Machine* di Newell, Shaw e Simon e la *Geometry Theorem Proving Machine* di Gelernter, non sono una implementazione dei calcoli logici esistenti, anche se il primo è usato per dimostrare teoremi dei *Principia Mathematica*. Invece questi programmi sfruttano delle strategie di riduzione di problemi a sottoproblemi, o di appoggio a determinare figure tipiche.

Quello che viene rimproverato paradossalmente ai calcoli logici non è di non essere sufficientemente meccanici, ma al contrario di non essere a misura d'uomo. Da allora si instaura una contrapposizione tra la mera replica meccanica dei risultati e la simulazione della attività umana. I calcoli logici che possono essere meccanizzati solo generando sistematicamente ed esaustivamente tutte le dimostrazioni incorporano gli aspetti più deleteri della stupidità della forza bruta meccanica; quello che si vuole aggiungere sono strategie euristiche di avvicinamento alla dimostrazione che corrispondano a quelle realmente messe in funzione dall'uomo.

La polemica si rivela presto una tempesta in un bicchiere d'acqua. Di calcoli logici ce ne sono molti; quelli in stile hilbertiano derivavano dalla tradizione di considerare alcune leggi logiche più fondamentali di altre, e assumerle come punti di partenza; altri sono stati inventati dopo Hilbert proprio per formalizzare in modo più naturale le dimostrazioni matematiche; ce ne sono ad esempio in cui la dimostrazione si cerca non a partire dagli assiomi ma a partire dalla conclusione voluta risalendo all'indietro verso i possibili assiomi che la implicano, guidati dalla struttura sintattica delle formule. E questo sostanzialmente è il principio, riscoperto indipendentemente, della riduzione ai sottoproblemi.

Infatti i logici tornano presto alla carica; nel 1958 Hao Wang con un calcolo del tipo sopra detto scrive un programma che dimostra tutti i teoremi dei *Principia Mathematica* in circa otto minuti un *exploit* allora sconvolgente. Raccolta la sfida, il risultato più interessante è che un nuovo calcolo viene elaborato nel giro di pochi anni da J. Robinson a partire da uno spunto di M. Davis e di D. Prawitz, il calcolo della risoluzione che è alla base degli attuali dimostratori automatici e degli interpreti dei linguaggi di programmazione logica (di cui si parla nell'ultima lezione). Questo calcolo si basa anch'esso su vecchie idee di Skolem e Herbrand degli anni venti, ma è influenzato dalla preoccupazione della efficienza. In effetti ci sono apparentemente limiti teorici alla efficienza se si vuole considerare un calcolo completo rispetto a tutte le formule del primo ordine; ma restringendosi a classi ridotte, sia pure ancora significative, di formule si può ottenere un raffinamento del calcolo che lavora su di esse in modo soddisfacente; tale classe è stata individuata in quella delle formule di Horn, e la cosiddetta risoluzione lineare applicata ad esse è il fondamento della programmazione logica.

Il metodo della riduzione dei problemi risulta logicamente equivalente, e traducibile in modo diretto, in questo tipo di calcolo; c'era da aspettarselo, una volta provato che il metodo era anche esso completo rispetto alla stessa classe di problemi; il fatto è che quando l'euristica è traducibile in un pro-

Ernst Zermelo nacque a Berlino nel 1871 e morì a Freiburg im Breisgau nel 1953.

René Descartes du Perron nacque a La Haye Touraine (Francia) nel 1596 e morì a Stoccolma nel 1650.

cedimento effettivo allora *ipso facto*, in quanto implementabile, cattura processi ricorsivamente enumerabili, e ricade nella logica del primo ordine. Questo è il destino di qualsiasi euristica che possa essere descritta in modo preciso e che si applichi non a un singolo problema.

Se la ricerca orientata alla efficienza continua, dal punto di vista teorico è ormai ben consolidata e compresa la funzione della logica come rivelatrice della presenza universale delle procedure di semidecidibilità. Ne segue che la macchina universale di Turing si è ormai imposta, se non come modello almeno come metafora. D'altra parte è bene osservare che questa presenza inevitabile non va a scapito delle altre funzioni superiori della intelligenza, di cui ciascuno porta testimonianza. Se vogliamo fare una incursione nella filosofia della Intelligenza Artificiale, possiamo osservare che il vero problema dei modelli computazionali della mente è quello di sapere come si offrono in input alla macchina universale i programmi particolari, ovvero come si stampano nel cervello i nuovi circuiti che incorporano programmi per nuove soluzioni trovate; non come vengono eseguiti una volta che sono stampati. In termini tradizionali di operazioni mentali, come si formano le idee, i concetti, come si arriva a riconoscere una strategia nella soluzione di problemi.

Contributi utili in questa direzione possono venire dalla disciplina nota come sintesi dei programmi: questa studia appunto il modo di codificare per quanto possibile i passaggi che portano da una idea intuitiva di un processo effettivo, da una formula che non è ancora algoritmica a un programma, attraverso raffinamenti e traduzioni di linguaggi. Alcuni di questi passaggi diventano standard su una ampia classe di casi e possono a loro volta essere formalizzati e meccanizzati.

Un settore vicino, sempre dell'informatica, dove l'uso della formalizzazione e della dimostrazione formale è stato rivalutato ed esaltato è quello della correttezza dei programmi. Correttezza non vuol dire soltanto correttezza sintattica, cioè rispetto delle regole formative dei linguaggi di programmazio-

ne, ma correttezza rispetto alla funzione che si vuole calcolare, e che di solito è data in un formalismo diverso da quello del linguaggio di programmazione. Si può dare ad esempio una definizione di somma come cardinalità di due insiemi disgiunti, e poi scrivere un programma iterativo che aggiunge ogni volta *+1*; non si tratta soltanto di dimostrare che sono definizioni equivalenti, ma anche che l'esecuzione di quel programma porta alla fine al risultato corretto rispetto alla prima definizione. La natura dei programmi, sequenze di istruzioni, suggerisce diversi metodi che in sostanza corrispondono ad associare dei commenti a ogni istruzione, commenti relativi agli effetti della esecuzione di quella istruzione. I commenti sono scritti in un linguaggio dichiarativo, che descrive dei fatti, e non imperativo come le istruzioni; la sequenza dei commenti si configura allora come una sorta di dimostrazione del fatto che il risultato voluto segue dalle posizioni iniziali. Con un minimo di rigore, i commenti diventano in senso preciso una dimostrazione, e una dimostrazione formale per di più, della correttezza.
La correttezza del programma si riduce allora alla costruzione di questa dimostrazione, e alla verifica che proprio di una dimostrazione corretta si tratta; la forma in cui si presenta, la sua lunghezza, in generale, il carattere ripetitivo di molti suoi passaggi suggeriscono di fare eseguire questo compito a un dimostratore automatico, un verificatore automatico di programmi. Alcuni successi rilevanti sono stati ottenuti, ad esempio dal verificatore di Boyer e Moore, che incorpora anche applicazioni del principio di induzione matematica.
Naturalmente se da una parte è comodo avere un unico programma che esegue tutte le dimostrazioni, o un unico motore inferenziale, come si dice in programmazione logica, che interpreta tutti i programmi, questo va a scapito della efficienza e duttilità. Anche trascurando i risultati teorici sulla complessità, è evidente che la rigidità e la universalità non si sposano facilmente con la agilità. Quello verso cui ci si orienta quindi, nel campo della Intelligenza Artificiale come in quello della dimostrazione automatica, è costituito dalla realizzazione di procedimenti interattivi, dalla disponibilità di diversi metodi di decisione che si richiamano da parte dell'utente quando ce ne sia bisogno come ausilio. Parti della matematica decidibili attraverso algoritmi efficienti sono affidate alla macchina quando nella ricerca della costruzione di una dimostrazione si perviene a un modulo che ricade in quel dominio. Questo orientamento non è tuttavia teoricamente incompatibile con l'esistenza di dimostratori universali, dati dai calcoli logici.
Ma a fianco dei successi e della propaganda a favore di questa impostazione si è anche sollevata una fiera polemica, in cui sono riprese e sottolineate molte delle ragioni che hanno sempre sostenuto gli oppositori della logica.
La verifica automatica dei programmi insiste sulla opportunità di affidare a una macchina una verifica che, riguardando un oggetto che nasce in modo quasi naturale come formale ha dei caratteri non adeguati a un controllo umano, la lunghezza in primo luogo. Si accetta dunque la critica che il formale allunga, ma si oppone che il calcolatore abbrevia, che l'operatore meccanico a differenza di quello umano riesce a superare questo ostacolo. L'opposizione allora insiste su un altro aspetto che è quello della mancanza di senso e quindi di controllabilità umana di queste sequenze di simboli privi di significato.
L'uso del formalismo introduce una incontrollabilità di principio. Ne risulta che l'accettazione della risposta eventuale non può essere basata su un reale controllo del ragionamento fatto, ma si affida soltanto alla supposta correttezza del programma, e non solo del programma, anche della macchina che dopo una serie di calcoli nascosti e inverificabili sputa fuori un risultato. La fiducia nella esecuzione della macchina, e la necessità di fare controlli solo ripetendo eventualmente i calcoli con un'altra macchina, introduce un carattere empirico, di esperimento fisico nel risultato ottenuto. La esecuzione meccanica delle dimostrazioni allora invece che esaltazione ultima della logica si rovescia nella introduzione di elementi empirici nella matematica. Dopo la dimostrazione con calcolatore del Teorema dei Quattro Colori, dimostrazione che per ora non appare avere alternative se non con altri programmi e altre macchine, la polemica già vivace nel campo informatico ha investito anche su questi temi la dimostrazione matematica.
Siccome però la dimostrazione automatica è il portato ultimo e coerente della logica, il suo rifiuto finisce per investire la caratterizzazione logica stessa della dimostrazione. La storia del metodo assiomatico, culminata con la vicenda delle geometrie non euclidee, ha ben fissato questa caratterizzazione: gli assiomi non hanno un significato intrinseco ma diverse possibili interpretazioni, i teoremi sono le proposizioni che sono valide in tutte le interpretazioni che soddisfano gli assiomi. La dimostrazione stabilisce questo legame, che tecnicamente è quello della conseguenza logica. Ma se si dice che la dimostrazione consiste nello stabilire la relazione di conseguenza logica tra assiomi e conclusione, allora attraverso i passi che abbiamo esaminato, calcoli

Ada Lovelace nacque in Inghilterra nel 1815 e morì nel 1852.

logici formali e completezza, si arriva a dover accettare come perfettamente legittima la dimostrazione al calcolatore.
Chi nega il carattere logico della dimostrazione sottolinea altri aspetti della sua funzione in matematica: in breve si tratta di una funzione retorica di convinzione della convenienza o utilità di fare uso di certi risultati. I criteri di accettazione sono variabili nel tempo e dipendono dalle assunzioni condivise da una comunità scientifica.
L'impressione di chi scrive è che queste posizioni mettono in luce aspetti reali della dimostrazione nella sua funzione di comunicazione e convinzione, ma che trascurino con troppa disinvoltura una tradizione millenaria che in modo non sempre coerente e non sempre tirandone le stesse conclusioni ha tuttavia sempre tenuto presente la relazione logica come riferimento privilegiato della matematica; che siano in difficoltà a dare un quadro complessivo, una definizione globale della matematica al di fuori di una fenomenologia superficiale; e che sia difficile spiegare come all'improvviso tutta la tradizione della matematica si riveli illusoria per essere sostituita dalla rivelazione di una nuova natura.
Nel denunciare la illusoria certezza e sicurezza della macchina, a favore di una dinamica più discorsiva e di certezze più problematiche, gli empiristi finiscono poi di trovarsi alleati di coloro che rifiutano la logica in nome di facoltà superiori e più profonde. Gli sviluppi che hanno reso possibile la scrittura e l'uso della dimostrazione formale sono gli stessi che hanno prodotto i teoremi che stabiliscono l'inevitabile relativismo delle nozioni matematiche. Torniamo un momento al paradosso di Skolem, e al teorema di completezza che dà una spiegazione del suo apparire e lo generalizza.
Il teorema di completezza per la logica del primo ordine afferma che se una proposizione è conseguenza logica di un'altra, in un senso di conseguenza logica che è molto ampio, compatibile con diverse teorie semantiche (ma quella insiemistica, con le sue strutture va benissimo), allora detta proposizione è anche derivabile da quella nel calcolo logico. In modo equivalente si vede subito che il teorema si può anche formulare nel seguente modo: una teoria ha una interpretazione, un modello, se è sintatticamente consistente, cioè se da essa non si possono derivare contraddizioni nel calcolo logico.
Se la teoria è sintatticamente consistente, un suo modello si definisce in modo concreto con il materiale sintattico disponibile, cioè su di un universo costituito dai termini della teoria. Ne segue che se il linguaggio è numerabile allora il modello è numerabile e si ritrova il teorema di Löwenheim - Skolem.
Il teorema di completezza afferma che i calcoli sono forti; ma esso può essere letto come affermazione che qualche cosa d'altro è invece debole, o difettoso: la nostra intuizione, o la nostra pretesa di avere una idea chiara e distinta, assoluta delle nozioni matematiche fondamentali che involvono l'infinito. Prendiamo la più semplice di tutte, quella della totalità dei numeri naturali.
Se si aggiungono agli assiomi dell'aritmetica i seguenti infiniti as-

siomi
$c > 0, c > 1, c > 2, \ldots$
dove c è una nuova costante di cui gli altri assiomi non dicono nulla, allora si vede subito che la nuova teoria è sintatticamente consistente, se l'aritmetica lo è: ogni dimostrazione, essendo finita, può appoggiarsi al più ad un numero finito dei nuovi assiomi, e allora se si sostituisce ovunque in essa c con un numero maggiore di tutti quelli menzionati nella dimostrazione se ne ottiene una nel linguaggio originario.

Un modello della teoria, che è garantito dal teorema di completezza, oltre a contenere gli usuali numeri, detti standard, deve contenere anche un elemento maggiore di tutti gli standard, oltre poi al suo successore, al suo predecessore, al successore del successore e così via. Una struttura abbastanza complessa, in cui comunque c'è un insieme di numeri naturali che possono essere detti infiniti, o non standard, al di là di quelli standard.

Qualcuno potrebbe obiettare che una tale struttura non è un modello dell'aritmetica, perché all'insieme dei numeri non standard non si applica il principio di induzione, nella forma del principio del minimo, il che è vero; ma un tale insieme risulta non definibile nel linguaggio e quindi non viene a contraddire alcun caso esplicito dell'assioma di induzione.

Perché non usiamo allora un linguaggio in cui si possa scrivere esplicitamente che certe proprietà devono valere per *tutti* gli insiemi, e non solo per quelli definibili? Sono i cosiddetti linguaggi del secondo ordine, con variabili e quantificatori su insiemi; ma nel 1948 L. Henkin ha esteso il teorema di completezza anche a questi linguaggi: non è il linguaggio che conta ma il tipo di logica che si usa. Se la relazione di conseguenza logica deve essere ricorsivamente enumerabile, allora si cade automaticamente nei fenomeni usuali del primo ordine, inclusa la completezza. Un teorema di Lindström afferma che ogni logica (definita da una relazione di conseguenza semantica) che abbia il fenomeno di Löwenheim - Skolem, in modo da potersi ridurre a strutture numerabili, e quello della compattezza, per cui una conseguenza di una teoria è sempre già conseguenza di un numero finito di assunzioni, è di fatto la logica del primo ordine.

Come si fa allora a sfuggire al relativismo, come si fa ad escludere di dover accettare modelli come quelli di sopra come realizzazioni della nostra idea dei numeri? Non si può, almeno non si può in una maniera comunicabile, scrivibile in un testo finito. Si può fare un atto di volontà, dopo aver individuato queste strutture si può dire che non le si vuole considerare, ma da allora in poi la nozione di conseguenza che si usa non è più neanche semidecidibile, è a un livello di non effettività in cui i matematici saranno anche capaci di muoversi, ma che è un po' ardito prendere come base fondazionale.

Escludere le strutture non standard richiede come minimo appunto prima il riconoscerle; e non solo la loro esistenza è dettata dalla nostra logica, ma sono anche utili. Le strutture non standard per i numeri reali hanno permesso di rivalutare il metodo degli infinitesimi, e di dimostrarne l'efficacia in una accezione del tutto inaspettata. Non è solo una curiosità storica, di fronte a cui il matematico possa dire che è solo marginalmente interessato; si tratta di una tecnica nuova che si sta rivelando utile nella modellizzazione di diversi fenomeni fisici, e una tecnica che è il portato inevitabile della presa sul serio del carattere logico delle dimostrazioni. È vero che l'analisi non standard si svolge apparentemente come un discorso insiemistico su strutture piuttosto complicate, ma la sua ragion d'essere sta nel prendere sul serio il carattere logico delle dimostrazioni, la loro coincidenza con le dimostrazioni del calcolo logico, e l'arricchimento, invece dell'impoverimento lamentato, che ne segue dal punto di vista delle possibilità dimostrative.

Non è un caso che gli osservatori più attenti dello sviluppo attuale della matematica accostino tra loro due fenomeni in apparenza superficiale così diversi come l'informatica, con tutto quello che comporta, e la analisi non standard.

Tra l'altro essa mostra che gli infinitesimi non erano una invadenza metafisica, ma erano il portato naturale delle dimostrazioni, secondo la concezione (intravista da Leibniz) del loro carattere fittizio.

Alla fine, la relatività delle nozioni matematiche apparentemente assolute che cosa dimostra? Tutti sappiamo che i tentativi fondazionali dell'inizio del secolo, che volevano dare una versione definitiva di questa credenza sono falliti.

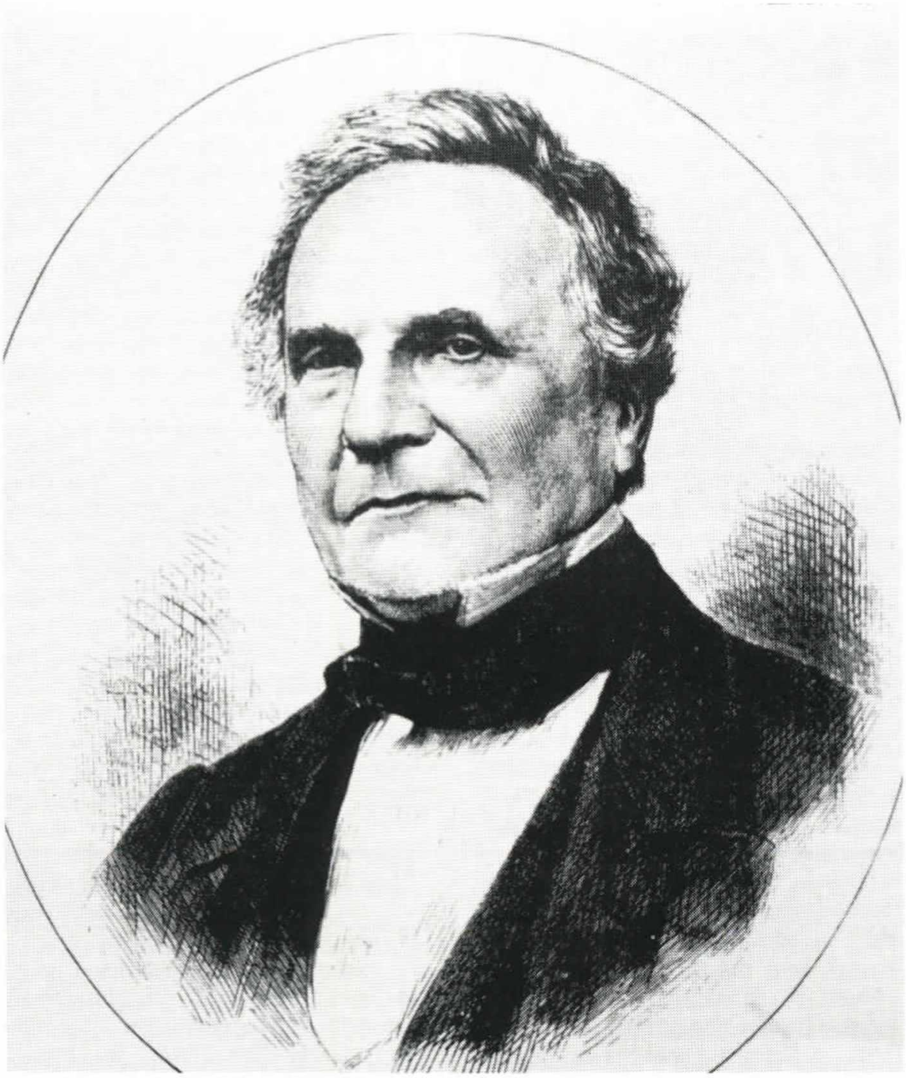
Il logicismo ad esempio voleva riuscirvi stabilendo il loro carattere logico, in un senso da definire, ma che aveva qualcosa di assoluto e preliminare in sè. Quello che è fallito è il tentativo di concepire, e provare che la matematica è una costruzione che si appoggia a qualche nozione assoluta di carattere prematematico. Il fallimento di tali progetti non è un fallimento della logica, come spesso ancora si sente dire, ma è stato una conseguenza dello sviluppo della logica e delle illuminazioni che ne sono venute sulla attività dimostrativa. Demolendo queste aspirazioni la logica ha permesso di riportare in primo piano una concezione della matematica che la identifica, che sorpresa sarà questa per i matematici, con la attività di dimostrazione di teoremi.

Programmazione logica e i computer della V generazione

di Annalisa Marcja

La storia del moderno computer è suddivisa in cinque generazioni. Possiamo prendere come date di inizio di queste rispettivamente il 1642, il 1955, il 1964, il 1972 e forse il 1990.
Nel 1642 Blaise Pascal costruì la prima macchina per fare le somme. Dopo 180 anni da questa invenzione l'inglese Charles Babbage (1791-1871) costruì una macchina che sostituisse le tavole numeriche del suo tempo, calcolate a mano e piene di errori. Dopo la costruzione di questa, pensò a un «general-purpose» computer. Egli pensava ad una macchina potente che avrebbe cambiato il futuro dell'analisi matematica, e che quindi chiamò *macchina analitica*. Questa macchina non fu mai costruita da Babbage (non fu capita dai suoi contemporanei e Babbage non riuscì ad avere un finanziamento governativo per la costruzione), ma il suo progetto è molto importante perché incorporava molti degli elementi del moderno computer.
Le schede perforate furono usate per la prima volta da Jacquard, che nel 1801 costruì un telaio in cui il movimento dei fili era controllato dai buchi nella scheda. Babbage adattò questa idea alla sua macchina analitica, usando schede perforate per le istruzioni e i controlli. La macchina era pensata con una memoria ed una unità aritmetica, dove introdurre operazioni e variabili; i numeri potevano essere posti nella memoria manualmente oppure per mezzo delle schede perforate. Babbage pensava anche a un metodo per stampare le risposte. La macchina analitica di Babbage è l'anello di transizione tra la macchina calcolatrice e il computer. Lady Lovelace, amica di Babbage, viene considerata come il primo programmatore, e i metodi da essa usati erano molto vicini ai linguaggi macchina dei primi calcolatori.
Verso la fine del diciannovesimo secolo, Hollerith pensò di modificare le idee di Babbage per costruire una macchina per rendere veloce il censimento negli Stati Uniti. Questa macchina, che controllava elettricamente la presenza o l'assenza di un buco nella scheda, in corrispondenza alla risposta si o no, rese effettivamente molto veloce il censimento del 1890.
Nel 1936 Aitken, in collaborazione con la IBM, iniziò la costruzione dell'Harvard Mark I, che fu terminato nel 1944. Questa era una macchina elettromeccanica e la memoria (formata da ruote dentate come contatori) poteva contenere 72 numeri in forma decimale di 23 cifre.
L'ENIAC (Electronic Numerical Integrator and Calculator) costruito nel 1946, fu il primo calcolatore completamente elettronico. Questa macchina conteneva 18.000 valvole e poteva in un'ora fare calcoli che avrebbero richiesto una settimana con l'Harvard Mark I. Poteva memorizzare solo 20 numeri, ma poteva sommare, moltiplicare e dividere due numeri in 200, 3000 e 6000 microsecondi rispettivamente.
La svolta che avrebbe portato al moderno computer è dovuta a John von Neumann. Von Neumann era nato a Budapest e nel 1930 aveva raggiunto Einstein all'Institute for Advanced Study dell'Università di Princeton. Nel 1944 von Neumann era un consulente del gruppo che lavorava alla bomba atomica a Los Alamos, e che aveva necessità di computazioni numeriche estremamente lunghe e complicate. Nel 1946, von Neumann pubblicò un articolo dal titolo «Preliminary discussion of the Logical Design of an Electronic Computing Instrument» che descriveva molte delle caratteristiche fondamentali del moderno computer digitale.
Alla fine del 1940, gruppi di ingegneri e matematici dell'Università di Cambridge in Pennsylvania, lavoravano all'implementazione dei piani di von Neumann e Turing per una macchina in cui potesse essere usato un programma, memorizzato nella macchina stessa. Il primo computer di questo tipo fu l'EDSAC (Electronic Delay Storage

Charles Babbage nacque a Walworth Road, vicino a Londra, nel 1791, e morì a Londra nel 1871. Progettò la «macchina analitica».



Automatic Calculator) costruito a Cambridge da Wilkes e l'EDVAC (Electronic Discrete Variable Automatic Computer), costruito da von Neumann. Entrambi i computer divennero operativi nel 1949. Nel 1951 la Remington produsse un grande computer col nome di UNIVAC (Universal Automatic Computer) che può essere pensato come il primo computer commerciale.

I transistor furono scoperti nel 1948 da Shokley ed essi immediatamente sostituirono le valvole nei computer. Comincia così la II generazione. È di questo periodo anche l'introduzione dei linguaggi FORTRAN-COBOL-ALGOL.

La terza generazione ha inizio invece con i computer con circuiti integrati. Di questi fanno parte computer come l'IBM 360 (1964), che impiegava 20 microsecondi per l'addizione di due numeri e 32 microsecondi per la moltiplicazione.

La quarta generazione (l'attuale) è la generazione dei minicomputer ed è caratterizzata dal piccolo ingombro e dalla sofisticata capacità computazionale di questi.

Come abbiamo già osservato i primi computer venivano programmati in linguaggi macchina. Essi permettono solo di far eseguire al calcolatore le singole operazioni elettroniche necessarie al conseguimento del risultato voluto. È facile convincersi che programmare in linguaggio macchina è molto difficile e suscettibile di numerosi errori.

Per rendere più facile la comunicazione uomo-macchina sono stati inventati i linguaggi di programmazione di alto livello, ciascuno pensato per tipi specifici di problemi.

Si può pensare che mentre per comunicare con i linguaggi macchina il programmatore deve conoscere la psicologia della macchina per esprimere i suoi problemi, nei linguaggi di alto livello i problemi vengono espressi in termini più vicini alla loro concettualizzazione originale. Ma ancora un programma redatto in linguaggi come il Fortran, il Pascal, l'Ada è la trascrizione di un algoritmo. Dobbiamo cioè conoscere già il modo di risolvere un problema, e descriverlo alla macchina.

Kowalski e Colmerauer nel 1972 arrivarono alla fondamentale idea che *la logica poteva essere usata come linguaggio di programmazione*, introducendo così la *programmazione logica*.

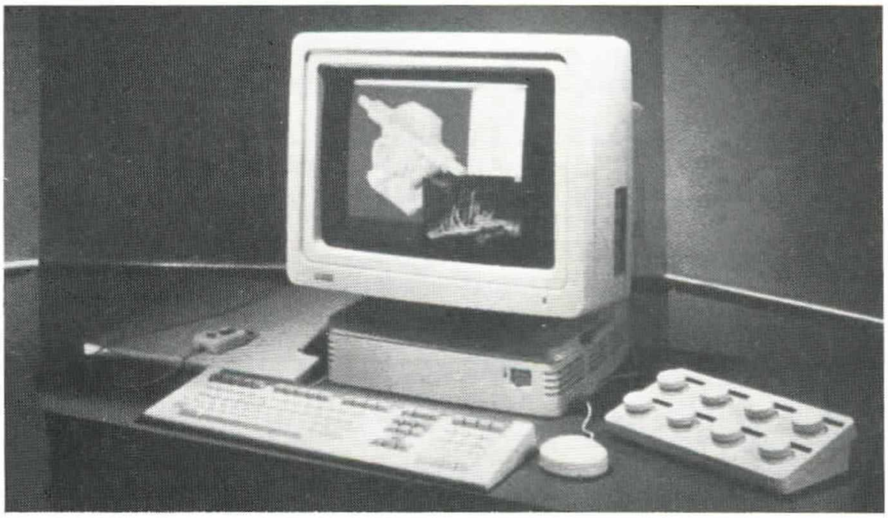
L'idea che la logica del I ordine, o almeno sottoinsiemi di essa, potesse essere usata come linguaggio di programmazione era rivoluzionaria, perché fino ad allora la logica veniva usata in informatica solo come linguaggio di specificazione. La logica invece ha una interpretazione procedurale. Brevemente una clausola di programma $A \leftarrow B_1,...,B_n$ viene vista come una definizione di procedura. Se $\leftarrow C_1,...,C_k$ è una clausola goal, ogni C_j viene visto come una chiamata di procedura. Un programma viene eseguito dandogli un goal iniziale. Se questo è $\leftarrow C_1,...,C_n$, un passo del calcolo implica l'unificazione di qualche C_i con la testa A di una clausola di programma $A \leftarrow B_1,\ldots,B_n$ e quindi la riduzione del goal in esame al goal $\leftarrow (C_1,\ldots,C_{j-1},B_1,\ldots,B_n,C_{j+1},\ldots,C_k)\ \phi$ dove ϕ è la sostituzione unificante. L'unificazione diventa dunque un meccanismo uniforme per il passaggio dei parametri, la selezione e la costruzione dei dati. Il calcolo termina quando viene prodotto il goal vuoto.

Nel 1972 a Marsiglia fu implementato il primo interprete Prolog (*Pro*grammation en *Log*ique). Il Prolog rivoluziona completamente il punto di vista della programmazione classica.

Si abbandona il principio della programmazione imperativa che obbliga il programmatore a indicare al computer l'algoritmo, sostituendolo con la programmazione dichiarativa, che consiste nel rappresentare le conoscenze su un soggetto dato, a partire dal quale saranno dedotte le risposte alle domande dell'utente. Un programma scritto in Prolog non fa che enunciare un insieme di fatti e di regole, che sono più vicini ad assiomi matematici, che ad ordini dati ad una macchina. Queste caratteristiche fanno del Prolog un linguaggio particolarmente adatto ai problemi dell'intelligenza artificiale, specialmente a quelli che si cerca di risolvere non applicando metodi di calcolo predeterminati.

Senza definire completamente la programmazione logica ed il Prolog, mostreremo il modo di utilizzo del Prolog in un classico esempio di «Problem Solving» (Mc Carthy).

C'è una scimmia sulla porta di una stanza. Nel mezzo della stanza c'è una banana che penzola dal soffitto. La scimmia è affamata e vorrebbe prendere la banana, ma non ci arriva. Alla finestra c'è una cassetta che la scimmia potrebbe usare.



La scimmia può compiere le seguenti azioni:
- camminare sul pavimento
- spingere la cassetta
- afferrare la banana, salendo sulla cassetta, posta immediatamente sotto la banana.

Il problema è: La scimmia può prendere la banana?

Prima di tutto dobbiamo rappresentare il problema nel linguaggio usato.

Supponiamo per esempio che lo stato iniziale sia:
1. La scimmia è sulla porta.
2. La scimmia è sul pavimento.
3. La cassetta è alla finestra.
4. La scimmia non ha la banana.

Formalizziamo ora le regole del gioco.

Il «goal» è una situazione in cui la scimmia ha la banana cioè, se usiamo il funtore stato con 4 posti, uno stato in cui all'ultimo posto c'è ha.

Stato (-,-,-,ha).

Inoltre ci sono 4 tipi di mosse che possono cambiare uno stato in un altro:
1. afferrare la banana
2. salire sulla cassetta
3. spingere la cassetta
4. camminare.

Non tutte le mosse sono possibili in ogni stato. Per esempio la mossa «afferrare» è possibile se la scimmia è sulla cassetta sotto la banana. Tali regole sono rappresentabili in Prolog da
Mossa(stato1,M,stato2).

Le mosse possibili sono:

mossa(stato(nelmezzo,sullacassa,
nelmezzo,nonha),
afferra,stato(nelmezzo,sullacassa,
nelmezzo,ha)).

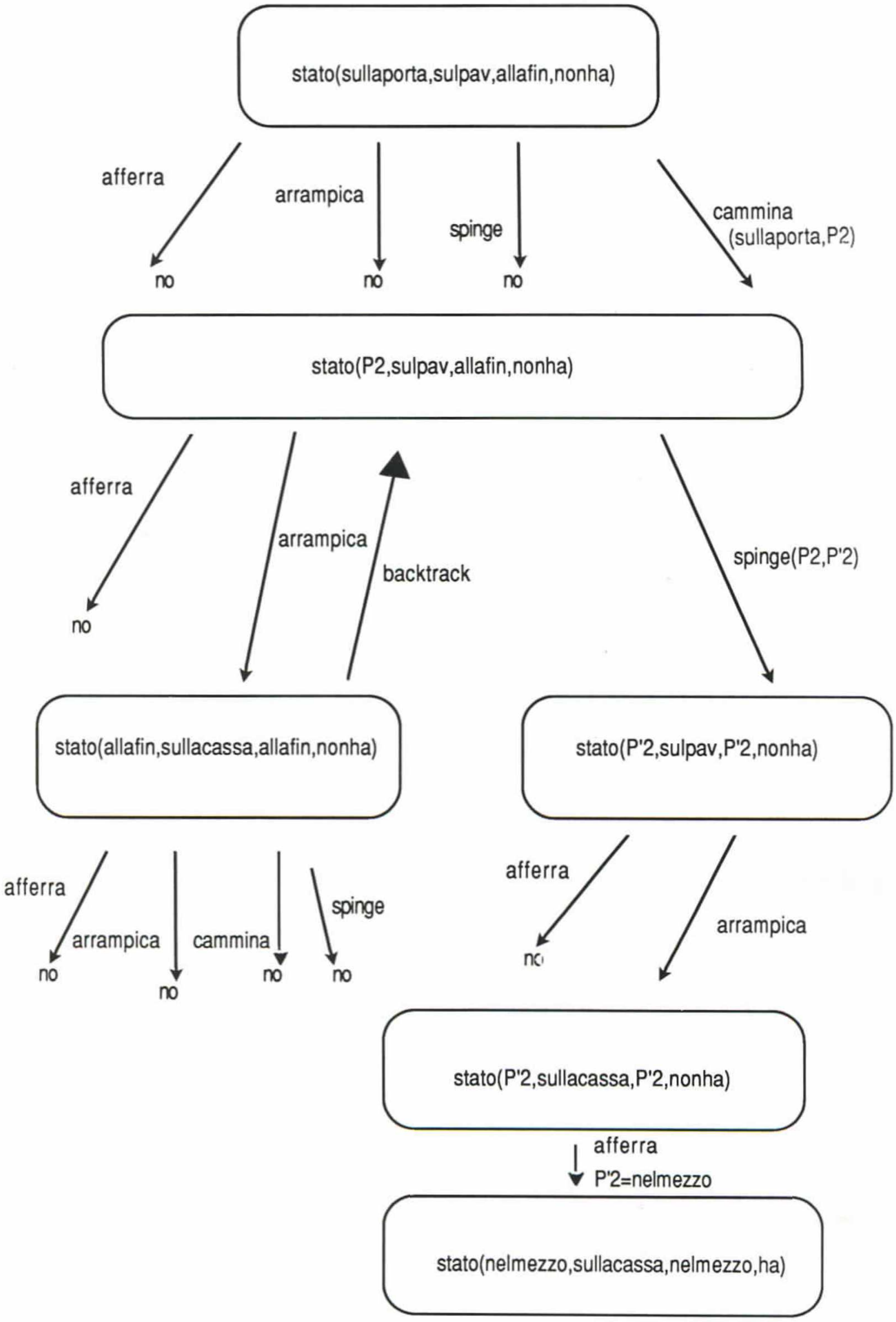
mossa(stato(P,sulpavimento,*P*,
H),arrampica,
stato(*P*,sullacassa,*P*,*H*)).

mossa(stato(P_1,sulpavimento,P_1,*H*),
spinge(P_1,P_2),stato(P_2,sulpavimen
to,P_2,*H*)).

mossa(stato(P_1,sulpavimento,*B*,*H*),
cammina(P_1,P_2),
stato(P_2,sulpavimento,*B*,*H*)).

Il fatto che la scimmia in uno stato iniziale *S* può afferrare la banana, può essere formulato con un predicato ricorsivo.
può(stato(-,-,-,ha)).
può(S_1):-mossa(S_1,*M*,S_2),può(S_2)

Abbiamo quindi descritto le nostre conoscenze in Prolog. Consideriamo ora il goal:
?-può(stato(sullaporta,sulpavimento,allafinestra,nonha)).
La risposta è sì.



Il comportamento procedurale è il seguente:

Se dopo aver scritto il programma ci accorgiamo che la conoscenza del problema era parziale, possiamo correggerlo abbastanza celermente, senza dovere, come nel caso imperativo, modificare completamente l'algoritmo.

È questo tipo di flessibilità che ha reso popolare il Prolog fra gli studiosi di intelligenza artificiale.

Nell'autunno del 1981 è stato pubblicato in Giappone un rapporto sui «calcolatori della quinta generazione», preludio al lancio di un programma di ricerca con lo stesso nome. L'obiettivo era di mettere a punto, per il decennio successivo, macchine specializzate per la rappresentazione della conoscenza ed adatte all'intelligenza artificiale.

In questo programma si tenta di risolvere contestualmente due problemi: da una parte avere linguaggi ad altissimo livello e dall'altro macchine parallele per aumentare la velocità. Si cerca quindi da una parte, e sembra che un prototipo sia pronto, di costruire «Prolog machine» e dall'altra trovare linguaggi per macchine parallele.
La speranza è che la programmazione parallela in linguaggi logici risulti più facile della programmazione sequenziale nei linguaggi tradizionali. Il Prolog ha in sè due possibili tipi di parallelismo. Consideriamo il programma

a:-b,c,d
a:-e,c
?-a

Per provare il goal *a* si può parallelamente verificare *b,c,d* (parallelismo and) oppure si può parallelamente tentare la I e la II clausola (parallelismo or) o entrambe.
Clarc e Gregory dell'Imperial College di Londra hanno introdotto il PARLOG: programmazione logica parallela.

I COMITATI SCIENTIFICI

L'attività di ricerca dell'ITC si sviluppa attraverso l'Istituto Storico Italo Germanico, l'Istituto di Scienze Religiose, l'Istituto per la Ricerca Scientifica e Tecnologica e il Centro Internazionale per la Ricerca Matematica.

ISIG

fondato il 3-11-73

38100 TRENTO - VIA S. CROCE 77
TEL. 0461/981617-981640

prof. Adam Wandruszka, *prof. emerito di Storia austriaca, Università di Vienna* - presidente
prof. Paolo Prodi, *prof. ord. di Storia moderna, Università di Trento* - segretario
prof. Umberto Corsini, *prof. ord. di Storia del Risorgimento, Università di Venezia*
prof. Reinhard Elze, *Direttore Istituto Storico Germanico di Roma*
prof. Rudolf Lill, *prof. ord. di Storia contemporanea, Università di Karlsruhe*
prof. Carlo Guido Mor, *prof. emerito di Storia del diritto, Università di Padova*
prof. Josef Riedmann, *prof. ord. di Storia medievale, Università di Innsbruck*
prof. Konrad Repgen, *prof. ord, di Storia moderna e contemporanea, Università di Bonn*
prof. Iginio Rogger, *prof. di Storia della Chiesa e Liturgia, Seminario Teologico di Trento*
prof. Pierangelo Schiera, *prof. ord. di Storia delle dottrine politiche, Università di Trento*
prof. Heinrich Schmidinger, *prof. ord. di Storia medievale, Università di Salzburg*
prof. Franco Valsecchi, *prof. emerito di Storia moderna, Università di Roma*

IRST

fondato il 21-9-76

38100 TRENTO - LOC. PANTÈ DI POVO
TEL. 0461/810105-810481

prof. Corrado Mencuccini - *prof. ord. di Fisica, Università La Sapienza di Roma* - presidente
dott. Luigi Stringa - *Direttore dell'Istituto per la Ricerca Scientifica e Tecnologica*
prof. Antonio Borsellino - *prof. ord. di Biofisica, Istituto Sup. di Studi Avanzati di Trieste*
prof. Vincenzo Lorenzelli - *prof. ord. di Chimica, Università di Genova*
dott. Angelo Marino - *Direttore del Dipartimento di Tecnologie Intersettoriali di Base ENEA*
prof. Giorgio Musso - *Responsabile Servizio Ricerca Centralizzata ELSAG, Genova*
prof. Salvatore Nicosia - *prof. ord. di Automazione degli Impianti, Università di Roma II*
prof. Emilio Picasso - *Direttore del progetto LEP CERN, Ginevra*
prof. Carlo Rubbia - *Premio Nobel per la Fisica; Senior Research Scientist CERN, Ginevra; professore di fisica Harward University, Cambridge, MA*
dott. Franco Zampini - *Responsabile Unità di Coordinamento Ricerche di Sicurezza ENEA*

ISR

fondato il 29-12-75

38100 TRENTO - VIA S. CROCE 77
TEL. 0461/981617-981640

prof. Iginio Rogger, *prof. di Storia della Chiesa e Liturgia, Seminario teologico di Trento* - presidente
prof. Giovanni Menestrina, *prof. di Italiano e Latino, Liceo Scientifico «Leonardo da Vinci»* - segretario
prof. Wilhelm Egger, *prof. di Nuovo Testamento, Università di Innsbruck e Seminario teologico di Bressanone* - attualmente vescovo di Bolzano-Bressanone
prof. Walter Kern, *prof. di Teologia fondamentale, Università di Innsbruck*
prof. Josef Krejči, *prof. di Antico Testamento, Seminario teologico di Trento*
prof. Claudio Leonardi, *prof. ord. di Storia della letteratura latina medievale, Università di Firenze*
prof. Germano Pellegrini, *teologo, Ministro provinciale dei PP. Francescani di Trento*
prof. Luigi Sartori, *prof. di Teologia dogmatica, Facoltà teologica di Milano-Padova, Presidente dell'Associazione Teologica Italiana*
dott. Sitia Sassudelli, *pubblicista*
prof. Lorenzo Zani, *prof. di Nuovo Testamento, Seminario Teologico di Trento*

CIRM

fondato il 11-7-78

38100 TRENTO - LOC. POVO
TEL. 0461/810629-931136

prof. Mario Miranda - *prof. ord. Analisi Matem., Università di Trento* - presidente
prof. Antonio Bove - *prof. ord. Analisi Matem., Università di Bologna*
prof. Giuseppe Da Prato - *prof. ord. Equazioni stocastiche, Scuola Normale Superiore Pisa*
prof. Dionigi Galletto - *prof. ord. Fisica Matem., Università di Torino*
prof. Francesco Gherardelli - *prof. ord. geom. algebrica, Università di Firenze*
prof. Enrico Giusti - *prof. ord. Analisi Matem., Università di Firenze*
prof. Mimmo Jannelli - *prof. ord. Equazioni stocastiche, Università di Trento*
prof. Carlo Marchioro - *prof. ord. Fisica Matematica, Università di Roma*
prof. Alessandro Silva - *prof. ord. Geometria, Università di Trento*
prof. Giovanni Zacher - *prof. ord. Algebra, Università di Padova*