Fabio Sterpetti

Concezione assiomatica, dimostrazioni formali e meccanizzabilità della matematica

1. Alcuni problemi della concezione assiomatica della matematica

Secondo la concezione assiomatica moderna della matematica, che vede tra i suoi massimi rappresentanti Hilbert, il metodo della matematica è il metodo assiomatico e fare matematica significa principalmente dimostrare teoremi, ovvero produrre dimostrazioni in un sistema formale a partire da determinati assiomi. Le dimostrazioni sono una sequenza di deduzioni. In questa prospettiva, è il ragionamento deduttivo a caratterizzare il ragionamento matematico e la produzione di conoscenza matematica.

La concezione assiomatica moderna propugnata da Hilbert si liberava della problematica tesi relativa alla immediata comprensibilità degli assiomi che caratterizzava la concezione assiomatica classica, di ascendenza euclidea, sostituendola con la ricerca della dimostrazione formale della non contraddittorietà degli assiomi di un dato sistema formale[1]. Hilbert riteneva che per fondare la matematica fosse necessario, senza ricorrere a procedimenti infinitari, dati i problemi che il concetto di infinito sollevava, dimostrare che un dato sistema formale F poteva esprimere tutte e sole le verità matematiche e che F era coerente, ovvero dimostrare in F la non contraddittorietà degli assiomi di F.

Un duro colpo a tale concezione fu portato dai teoremi di incompletezza di Gödel (1931). Il primo teorema di incompletezza di Gödel afferma infatti che, per ogni possibile sistema formale F sufficientemente potente da consentire di assiomatizzare la teoria elementare dei numeri naturali, se F è coerente, ci sono proposizioni indecidibili in F. Un enunciato γ è indecidibile in F se non è possibile dimostrare in

[1] Per una breve ricostruzione storica del metodo assiomatico, si veda Rodin 2014, parte I.

Scenari, n. 23, 2025 • Mimesis Edizioni, Milano-Udine • www.mimesisjournals.com/ojs/index.php/scenari • ISSN (online): 2785-3020 • ISBN: 9791222331119 • DOI: 10.7413/24208914224

F né γ né $\neg\gamma$. Un tale sistema formale non può quindi essere completo, ovvero consentire di derivare tutte e sole le verità matematiche, in quanto uno tra γ e $\neg\gamma$ deve essere una verità matematica, ma non può essere un teorema di *F*. Dunque, *F* è incompleto. Il secondo teorema di incompletezza di Gödel afferma che, per ogni possibile sistema formale *F* sufficientemente potente da consentire di assiomatizzare la teoria elementare dei numeri naturali, se *F* è coerente, la coerenza di *F* non è dimostrabile in *F*. Dunque, non possiamo dimostrare con procedimenti finitari che tutta la matematica è coerente derivandola in un dato sistema *F* e dimostrando in *F* che *F* è coerente.

Nonostante ciò, ancora oggi «la concezione assiomatica è la concezione della matematica dominante» (Lavers 2024, p. 54). Molti autori sembrano non dare eccessivo peso ai risultati godeliani e ritengono che la concezione assiomatica e formale della matematica possa essere mantenuta, sebbene ammettano che vada in qualche modo aggiornata (von Plato 2018). Gödel stesso continuò a difendere l'idea che la matematica consista nella dimostrazione di teoremi attraverso il metodo assiomatico e che «tutto ciò che è matematico è formalizzabile» (Gödel 1935, p. 1), sostenendo che i teoremi di incompletezza implicano soltanto che per formalizzare tutta la matematica si debba ricorrere a una sequenza infinita di sistemi formali e non a un solo sistema formale dato come aveva sperato Hilbert. Si consideri, ad esempio, l'aritmetica di Peano, PA. Se PA è coerente, allora per il secondo teorema di Gödel, *Con*(PA), ovvero l'enunciato che esprime la coerenza di PA, non è dimostrabile in PA. Ma se si passa a un sistema formale che contiene gli assiomi dell'aritmetica di Peano del secondo ordine, PA^2, è possibile dimostrare in tale sistema formale *Con*(PA). In PA^2, per il secondo teorema di Gödel, non è possibile dimostrare *Con*(PA^2). Ma se si passa a un sistema formale che contiene gli assiomi dell'aritmetica di Peano del terzo ordine, PA^3, diviene possibile dimostrare in tale sistema formale *Con*(PA^2), e così via. Dello stesso avviso è Curry, secondo cui i teoremi di incompletezza di Gödel mostrano solamente che il concetto di dimostrazione valida non può essere reso esaustivamente per mezzo di un singolo sistema formale (Curry 1977, p. 15).

Tuttavia, se tutta la matematica deve poter essere formalizzata e trattata assiomaticamente, allora anche la transizione da un sistema formale a quello successivo nella sequenza dei sistemi formali necessaria a formalizzare tutta la matematica dovrebbe essere formalizzabile. Infatti, se la transizione tra i successivi sistemi formali non fosse formalizzabile, non potrebbe sostenersi che realmente tutto ciò che è matematico è formalizzabile (Cellucci 2022, p. 32). Ma, se la transizione da un sistema formale a quello successivo nella sequenza di sistemi formali è essa stessa formalizzabile, allora, come scrive McCarthy, per la sequenza dei sistemi formali

sarà possibile dimostrare un teorema che è un «esatto analogo» del primo teorema di incompletezza di Gödel (McCarthy 1994, p. 427).

Più di recente, Calude e Thompson, hanno scritto che, sebbene non sia possibile formalizzare tutta la matematica, «è essenzialmente possibile formalizzare tutta la matematica che "ciascuno usa". La teoria degli insiemi di Zermelo-Fraenkel combinata con la logica del primo ordine fornisce un formalismo soddisfacente e generalmente accettato per essenzialmente tutta la matematica corrente» (Calude, Thompson 2016, p. 139). Per questi autori, se è vero che non si può «dimostrare la completezza di sistemi tanto potenti quanto la aritmetica di Peano», è nondimeno possibile «dimostrare la completezza di molti sistemi, più deboli ma interessanti, come, ad esempio, la logica del primo ordine» (*ibidem*). La prospettiva hilbertiana non andrebbe, quindi, abbandonata, bensì ridimensionata.

Ma questo approccio trascura il fatto che Hilbert non ricercava la formalizzazione come un fine in sé stesso. Il suo scopo principale era fornire una fondazione sicura alla matematica attraverso la formalizzazione e la dimostrazione della coerenza di una sua parte, quella ritenuta meno dubbia, ovvero l'aritmetica, da cui, almeno in linea di principio, sarebbe stato poi possibile derivare tutte le verità matematiche. E questo scopo non può essere raggiunto a causa dei risultati di Gödel (von Plato 2018). Che una qualche porzione della matematica possa essere formalizzata, ma non se ne possa dimostrare la coerenza, o che si possa dimostrare la completezza di una qualche altra limitata porzione della matematica, non significa affatto che lo scopo principale di Hilbert possa essere raggiunto sia pure in una forma più circoscritta. Secondo Weyl, ad esempio, la rilevanza dei risultati di Gödel non può essere sovrastimata, perché a causa di tali risultati «la fondazione ultima e il significato ultimo della matematica rimane un problema aperto», e il mancato successo «dell'audace impresa hilbertiana non può non influenzare l'interpretazione filosofica» della matematica (Weyl 1949, p. 219).

Alcuni autori, poi, ritengono che la concezione assiomatica della matematica non rifletta adeguatamente l'attività dei matematici e che fare matematica significhi principalmente cercare di formulare ipotesi che siano in grado di risolvere un dato problema matematico a partire da ciò che è noto del problema piuttosto che dimostrare teoremi a partire da assiomi (Hersh 1997; Cellucci 2013; 2017; 2022).

2. Concezione assiomatica della matematica e automazione del ragionamento

A fronte delle critiche della concezione assiomatica che provengono da concezioni radicalmente diverse della matematica, e nonostante

alcuni risultati della logica matematica, come quelli di Gödel, rendano irraggiungibile lo scopo che Hilbert assegnava alla formalizzazione della matematica, secondo i sostenitori della concezione assiomatica della matematica vi sono degli altri risultati della logica matematica, come l'isomorfismo di Curry-Howard, che sembrano supportare la bontà della concezione assiomatica e giustificare il perseguimento delle ricerche ispirate da tale concezione, prime fra tutte quelle legate allo sviluppo dell'automazione del ragionamento (*automated reasoning*). In tale ambito di ricerca è particolarmente evidente la consonanza con la concezione assiomatica della matematica. Portoraro, ad esempio, scrive che «il ragionamento automatizzato è simile alla dimostrazione meccanica di teoremi. Costruire un programma di ragionamento automatizzato significa fornire una descrizione algoritmica a un calcolo formale in modo che possa essere implementato su un computer per dimostrare teoremi del calcolo in modo efficiente» (Portoraro 2025).

Le prospettive legate alle ricerche relative all'automazione del ragionamento hanno dato nuova linfa alla concezione assiomatica della matematica e al principale prodotto teorico di tale concezione, ovvero alla teoria della dimostrazione (von Plato 2018, sez. 8). Scopo di questo articolo è, mettendo da parte le critiche mosse alla concezione assiomatica a partire da concezioni radicalmente diverse, analizzare se davvero risultati come quello di Curry-Howard e le ricerche sull'automazione del ragionamento forniscano una solida giustificazione per continuare a sostenere l'adeguatezza della concezione assiomatica della matematica.

L'isomorfismo di Curry-Howard mostra come i programmi che vengono studiati in informatica teorica dagli informatici possano essere ritenuti equivalenti alle dimostrazioni che vengono studiate dai logici matematici nella teoria della dimostrazione (Sørensen, Urzyczyn 2006, p. v; Prawitz 2008), ovvero alle dimostrazioni costruttive, tipiche della concezione intuizionista, in cui quando si afferma che p significa che si è in grado di fornire una dimostrazione che p ed estrarre da questa dimostrazione un algoritmo per costruire p ogni volta che si asserisce che p (Moschovakis 2018). Ora, da un lato, è possibile, attraverso le tecniche sviluppate da Gödel e da Turing, "tradurre" lo studio delle proprietà delle dimostrazioni formali nello studio delle proprietà delle funzioni ricorsive, ovvero computabili. Dall'altro lato, il lambda calcolo è stato ideato e sviluppato da Church e da Curry negli anni Trenta del Novecento proprio per studiare le funzioni ricorsive, e dunque la computabilità, e per contribuire alla ricerca sui fondamenti della matematica. In seguito, prima Curry e poi Howard hanno messo in luce la corrispondenza che sussiste tra le dimostrazioni formulate nei sistemi intuizionistici di deduzione naturale e i programmi elaborati nel lambda calcolo tipato semplice, che era ormai

divenuto centrale in informatica teorica[2]. In altre parole, Curry e Howard dimostrano che vi è un isomorfismo tra i sistemi di deduzione naturale, che furono concepiti come modelli delle dimostrazioni utilizzate dai matematici, e il lambda calcolo tipato semplice, che venne concepito come modello della calcolabilità.

L'isomorfismo di Curry-Howard sembra supportare la concezione assiomatica della matematica, perché la corrispondenza tra dimostrazioni e programmi sembra confermare la tesi che ampliare la conoscenza matematica significhi produrre dimostrazioni in un dato sistema formale, e che, quindi, il metodo della matematica sia effettivamente il metodo assiomatico (Weir, 2021, sez. 7). In questa prospettiva, fare matematica non significa altro che dedurre conclusioni da premesse date in un dato sistema formale. Grazie all'isomorfismo di Curry-Howard, dedurre conclusioni in un sistema formale si dimostra essere equivalente all'esecuzione di un programma, e dunque sembra essere un'attività meccanizzabile, dato che i programmi sono algoritmi e sono dunque meccanizzabili. Se la concezione assiomatica è corretta, produrre conoscenza matematica significa produrre dimostrazioni formali, ovvero dimostrazioni all'interno di un dato sistema formale, e se tale attività può essere meccanizzata, significa che la produzione di conoscenza matematica può essere resa del tutto indipendente dal soggetto epistemico umano.

L'idea che il ragionamento logico, inteso esclusivamente come ragionamento deduttivo, possa essere formalizzato efficacemente attraverso la logica matematica ha suggerito a molti autori che vi fosse una sorta di equivalenza tra la deduzione e la computazione. La formalizzazione si ritiene, infatti, ci consente di tralasciare ogni aspetto semantico nell'analisi delle inferenze, di rendere la derivazione di una conclusione da delle premesse per mezzo di una regola di inferenza deduttiva una mera manipolazione simbolica, un processo meramente sintattico del tutto analogo a ciò che avviene quando si seguono delle regole computazionali per effettuare un calcolo in ambito matematico (Dutilh Novaes 2012). Per questo la deduzione è stata considerata strettamente affine alla computazione. Secondo alcuni autori, la computazione può essere considerata una forma particolare di deduzione. Ad esempio, Kripke scrive che «la computazione è un argomento deduttivo da un numero finito di istruzioni», ovvero «una forma speciale di argomento matematico» in cui «viene

[2] Per una presentazione dell'isomorfismo di Curry-Howard e per una breve ricostruzione storica di come Curry e Howard siano giunti a tale risultato, si veda Wadler 2015. Curry scrisse per la prima volta della relazione tra tipi di una funzione calcolabile e proposizioni logiche dimostrabili in Curry 1934. In successivi lavori dimostrò che la relazione si estendeva pure a termini e dimostrazioni. Nel 1969 Howard fece circolare un lavoro, che venne pubblicato solo diversi anni dopo (Howard 1980), nel quale mostrava che la relazione individuata da Curry sussisteva tra i sistemi di deduzione naturale e il lamba calcolo tipato semplice.

fornito un insieme di istruzioni» e in cui si pensa che i diversi «passi nella computazione seguano – seguano deduttivamente – dalle istruzioni» che sono state fornite. Dunque, «una computazione è solo un'altra deduzione matematica, sebbene sia una deduzione matematica di una forma molto specializzata (Kripke 2013, p. 80).

3. L'equiparazione di computazione e deduzione

Ora, anche volendo concedere che l'essere equivalenti alle dimostrazioni in un sistema formale dei programmi informatici, che sono algoritmi, e sono dunque computabili, ovvero sono associati a una funzione computabile, mostri come ogni dimostrazione in un sistema formale possa essere considerata equivalente a una funzione computabile, bisogna però ricordare che non può affermarsi in generale l'inverso, ovvero che ogni funzione computabile corrisponda a una dimostrazione in un sistema formale, dato che, per ogni sistema formale dato, vi sono funzioni computabili che non sono isomorfe ad alcuna dimostrazione in tale sistema formale.

Infatti, se è vero che ogni funzione totale dimostrabilmente computabile f è in effetti computabile, è anche vero che non ogni funzione totale f, sebbene sia computabile, è dimostrabilmente computabile in un dato sistema formale F. Una funzione totale f si dice dimostrabilmente computabile se è possibile dimostrare in un sistema formale F che f è computabile. Dimostrare che una funzione totale f è computabile significa dimostrare che esiste una macchina di Turing, M_f, che computa f e che esiste una dimostrazione nel sistema formale F che f è totale, ovvero che M_f è totale. Una macchina di Turing M è totale se per ogni $n \in \mathbb{N}$ che le viene fornito come input, M si arresta sempre.

Che non ogni funzione totale f, sebbene sia computabile, è dimostrabilmente computabile in un dato sistema formale F si deriva attraverso un argomento diagonale come quello usato da Turing (1937) per dimostrare che il problema della fermata non è decidibile. Il punto è che l'insieme $\boldsymbol{\Phi}$ delle funzioni totali computabili è più che numerabile, mentre ogni sistema formale F che consenta di dimostrare in un numero finito di passi la computabilità di una data funzione totale computabile $f \in \boldsymbol{\Phi}$ non può che essere costituito da un insieme finito di elementi e produrre, quindi, al più un insieme numerabile di dimostrazioni, D_{fi}. Per cui, se enumeriamo tutte le funzioni totali f_i dimostrabilmente computabili nel sistema F, ovvero tutte le funzioni $f_i \in \boldsymbol{\Phi}$ la cui dimostrazione di computabilità $d_{fi} \in D_{fi}$, e otteniamo così l'insieme $\boldsymbol{\Phi}_{DC}$, e se F è coerente, allora è sempre possibile, in un sistema formale diverso da F, diciamo G, identificare, attraverso la tecnica della diagonalizzazione, una macchina di Turing che si riferisce a una

funzione $f_n \in \boldsymbol{\Phi}$ che non è compresa nell'insieme $\boldsymbol{\Phi}_{DC}$, ovvero nell'insieme di tutte le funzioni totali dimostrabilmente computabili nel sistema *F*.

Possiamo, infatti, immaginare una macchina di Turing M_{fn} che corrisponda alla funzione diagonale relativa all'insieme di funzioni $f_i \in \boldsymbol{\Phi}_{DC}$, ovvero una macchina di Turing che enumeri tutte le dimostrazioni $d_{fi} \in D_{fi}$, ovvero le corrispondenti macchine di Turing M_{fi} che computano tali funzioni secondo le dimostrazioni $d_{fi} \in D_{fi}$, e che per ogni input *n* computi la funzione $f_n(n)$, ovvero applichi a *n* la funzione corrispondente alla ennesima dimostrazione di questa enumerazione, funzione che sappiamo essere computabile proprio grazie alla ennesima dimostrazione di questa enumerazione, ovvero al fatto che si è dimostrato in *F* che la macchina di Turing che la computa è totale, ovvero si arresta per ogni input *n*, e restituisca poi come output un valore opposto a quello restituito dalla ennesima macchina di Turing considerata per calcolare $f_n(n)$. Se, ad esempio, la ennesima macchina di Turing dell'enumerazione di tutte le funzioni totali dimostrabilmente computabili nel sistema *F* restituisce come output "1", perché si arresta per l'input *n*, M_{fn} restituirà come output "0". E così sarà per ogni altra macchina di Turing M_{fi} di tale enumerazione, dato che si dimostra in *F* che tali macchine di Turing sono tutte totali, e dunque tutte si arrestano per ogni *n*. In altre parole, per ogni valore di *n* sappiamo che vi è una macchina di Turing che si arresta quando computa la funzione corrispondente alla ennesima dimostrazione di questa enumerazione. La nostra macchina di Turing M_{fn}, per computare $f_n(n)$, quindi, non farà altro che, per ogni *n*, attivare la macchina di Turing che computa la funzione corrispondente alla ennesima dimostrazione di tale enumerazione. Eppure, non è possibile dimostrare in *F* che la macchina di Turing M_{fn} è totale, ovvero si arresta per ogni *n*, ovvero che la funzione diagonale che corrisponde alla macchina di Turing M_{fn} è computabile in *F*, e quindi possiamo concludere che tale funzione non appartiene a $\boldsymbol{\Phi}_{DC}$. Infatti, per ogni *n*, i valori restituiti come output dalla macchina di Turing M_{fn} sono opposti a quelli della diagonale di una matrice costituita da ogni possibile input *n* e da tutte le dimostrazioni $d_{fi} \in D_{fi}$, ovvero da tutte le corrispondenti macchine di Turing M_{fi} che computano tutte le funzioni $f_i \in \boldsymbol{\Phi}_{DC}$. Non vi è, pertanto, in tale matrice nessuna macchina di Turing che restituisca gli stessi valori di output che restituisce M_{fn}, i valori restituiti da ognuna di queste differiscono da quelli restituiti da M_{fn} per il valore ennesimo, e quindi la funzione che corrisponde alla macchina di Turing M_{fn} non appartiene a $\boldsymbol{\Phi}_{DC}$. Dunque, la funzione corrispondente alla macchina di Turing M_{fn} non ha una dimostrazione d_{fn} della propria computabilità in D_{fi}, ma, nonostante ciò, sappiamo che è computabile, per così dire, per costruzione, dato che per ogni *n* sappiamo che vi è una macchina di Turing che si arresta quando viene computata $f_n(n)$. Dunque, se *F* è coerente, abbiamo la garanzia che, per ogni input *n*, la funzione $f_n(n)$ è

computabile, ma non possiamo dimostrarlo in *F* (Boolos, Burgess, Jeffrey 2002). Tale procedura può ripetersi per ogni sistema formale. Dunque, per ogni sistema formale non tutte le funzioni totali, benché computabili, sono dimostrabilmente computabili in tale sistema. E dunque, non ogni funzione computabile corrisponde a una dimostrazione in senso intuizionista. La sovrapponibilità tra computazione e deduzione non è, quindi, affatto assoluta.

4. Concezione assiomatica della matematica e dimostrazioni formali

Fatta questa precisazione, assumiamo, per amore di discussione, che la concezione assiomatica sia corretta, e che fare matematica significhi produrre dimostrazioni di teoremi in un dato sistema formale a partire da assiomi dati. Conseguenza diretta dell'idea che la conoscenza matematica sia ampliata attraverso la dimostrazione di teoremi in un sistema formale è la tesi secondo cui tale attività può essere meccanizzata, e che quindi la conoscenza matematica può essere ampliata indipendentemente dall'attività epistemica del soggetto umano. E difatti tale tesi è stata difesa da coloro che ritengono che vi siano programmi in grado di produrre autonomamente matematica (Colton, Bundy, Walsh 1999; Colton 2002; Larson 2005; Hales 2008; Portoraro 2025).

Bisogna procedere a un'analisi attenta per mettere in luce come l'associazione tra l'idea che la matematica consista nella dimostrazione di teoremi in un dato sistema formale e l'idea che la matematica possa essere automatizzata sia più problematica di quanto possa apparire a un'ispezione veloce.

Per potere affermare che risultati come l'isomorfismo di Curry-Howard sostengono la bontà della concezione assiomatica della matematica, non basta infatti sostenere genericamente che fare matematica significhi dimostrare teoremi, bisogna impegnarsi a sostenere che fare matematica significhi produrre dimostrazioni in un dato sistema formale, ovvero produrre dimostrazioni formali. Chi intende percorrere questa linea argomentativa a favore della concezione assiomatica della matematica deve, quindi, impegnarsi a difendere la seguente tesi relativa alla formalità delle dimostrazioni matematiche:

(T_f) tutte le dimostrazioni matematiche prodotte dai matematici possono essere, almeno in linea di principio, ridotte a dimostrazioni formali, ovvero a dimostrazioni che possono essere autonomamente verificate da programmi informatici e che sono dello stesso tipo delle dimostrazioni che potrebbero essere autonomamente prodotte da programmi informatici.

Bisogna a questo punto verificare se T_f può essere adeguatamente difesa.

La tesi secondo cui tutte le dimostrazioni matematiche prodotte dai matematici possono essere, almeno in linea di principio, ridotte a dimostrazioni formali è equivalente a quella che viene definita in filosofia della matematica la concezione derivazionista delle dimostrazioni matematiche, secondo la quale le dimostrazioni matematiche sono accettate dai matematici quando tali dimostrazioni sono in grado di mostrare che sarebbe possibile, almeno in linea di principio, tradurle in un linguaggio formale, ovvero che sarebbe possibile derivare il teorema che dimostrano all'interno di un dato sistema formale, che sarebbe possibile, in altre parole, renderle delle dimostrazioni formali (Azzouni 2004; Antonutti Marfori 2010). La stragrande maggioranza delle dimostrazioni matematiche prodotte dai matematici, infatti, sono dette dimostrazioni informali, proprio perché non sono presentate in modo rigorosamente formalizzato.

Nella concezione derivazionista delle dimostrazioni matematiche, ogni dimostrazione matematica può essere ridotta a una dimostrazione formale, e il rigore della matematica e la certezza della conoscenza matematica sono garantiti da tale possibilità. Ad esempio, Mac Lane scrive che una dimostrazione matematica «è rigorosa quando è (o potrebbe essere) scritta nel linguaggio del primo ordine $L(\in)$ come una sequenza di inferenze dagli assiomi di ZFC, in cui ogni inferenza sia fatta in accordo con una delle regole stabilite» (Mac Lane 1986, p. 377).

Dunque, se la matematica è dimostrazione di teoremi in un dato sistema formale, ogni dimostrazione matematica può essere resa una dimostrazione formale, e se ogni dimostrazione matematica è una dimostrazione formale, allora la produzione di matematica può essere meccanizzata. In effetti, affinché «le macchine possano fare matematica ciò che è necessario è prima di tutto un linguaggio che descriva la matematica in termini che siano adeguati alle macchine. Tale linguaggio deve essere completamente *formalizzato*» (Longo 2003, p. 85). Come scrivono Davis e Hersh, la formalizzazione «è il processo attraverso cui la matematica viene resa adatta al processamento automatico» (Davis, Hersh 1981, p. 136). La formalizzazione è il processo attraverso cui un linguaggio viene "desemantizzato" per consentirne una manipolabilità esclusivamente sintattica (Dutilh Novaes 2012).

Chi voglia difendere la concezione assiomatica sulla base di risultati come quello di Curry-Howard si impegna, perciò, non solo a sostenere che la matematica sia dimostrazione di teoremi, ma anche a difendere T_f, ovvero la concezione derivazionista delle dimostrazioni matematiche. Passeremo in rassegna, pertanto, alcune delle principali difficoltà relative alla concezione derivazionista.

5. Dimostrazioni matematiche formali e informali

Il primo problema della concezione derivazionista delle dimostrazioni matematiche deriva dal fatto che contro tale tesi si può obiettare che le dimostrazioni matematiche prodotte effettivamente dai matematici non solo non sono affatto dimostrazioni formali, in quanto sono piuttosto dimostrazioni informali, ma anche che non sia affatto scontato che sia sempre possibile tradurre integralmente tali dimostrazioni informali in un linguaggio formale dato (Rav 2007; Pelc 2009; Goethe, Friend 2010; Tanswell 2015; Larvor 2016; Macbeth 2024). Le dimostrazioni matematiche che vengono effettivamente prodotte dai matematici nella maggior parte degli ambiti della matematica non sono, infatti, delle dimostrazioni formali, ovvero non sono affatto lo stesso tipo di oggetti che sono studiati da quei logici matematici che studiano la teoria della dimostrazione, e difatti vengono chiamate, come detto, dimostrazioni informali, oppure schizzi di dimostrazione (Larvor 2016; Goethe, Friend 2010; Antonutti Marfori 2010).

Le dimostrazioni informali non sono date in alcun linguaggio formale, né rendono esplicito ogni passaggio logico. Piuttosto, le dimostrazioni informali fanno assunzioni circa la conoscenza pregressa «della platea cui si rivolgono e la capacità di questa di seguire certe linee argomentative, saltano i passaggi noiosi o di routine, e fanno riferimento a proprietà semantiche e a proprietà di oggetti matematici senza darne una formulazione completa» (Tanswell 2015, pp. 295-297). Nelle dimostrazioni informali viene usato il simbolismo matematico, così come «il linguaggio naturale, i diagrammi, e spiegazioni dalla modalità mista» (*ibidem*).

Dunque, sebbene, come abbiamo visto, per l'isomorfismo di Curry-Howard, i programmi studiati dagli informatici teorici possono essere considerati equivalenti alle dimostrazioni formali studiate dai logici matematici che studiano la teoria della dimostrazione, ciò non significa che quando un matematico presenta una dimostrazione matematica di un risultato cui è pervenuto stia presentando lo stesso tipo di oggetto che studiano i logici matematici in teoria della dimostrazione o che un programma che sia stato disegnato per derivare deduttivamente teoremi da un insieme di assiomi dati sarebbe in grado di produrre tale risultato.

In altre parole, le dimostrazioni matematiche che i difensori di una prospettiva assiomatica sostengono i programmi informatici siano in grado di produrre autonomamente, ovvero indipendentemente da ogni soggetto epistemico umano, non sono equivalenti alle dimostrazioni matematiche prodotte e ritenute interessanti dai matematici umani (Macbeth 2014; Pantsar 2024). La ragione è che una dimostrazione matematica interessante, anche quando presentata in un contesto rigoroso, come ad esempio in una pubblicazione scientifica, non si risolve mai esclusiva-

mente in una mera sequenza di deduzioni da un insieme di assiomi dati al teorema da dimostrare. Una dimostrazione matematica interessante non convince il pubblico cui è rivolta per il fatto che questo è in grado di seguire diligentemente ogni singola deduzione, ma piuttosto viene ritenuta interessante perché, nel suo svolgersi, oltre a determinate inferenze deduttive, implica anche, da parte del pubblico cui è rivolta, la valutazione di argomenti e ragioni forniti a sostegno o contro determinate affermazioni, implica che il pubblico cui si rivolge comprenda, e giudichi adeguata, la strategia argomentativa complessiva della dimostrazione, che ne colga magari le analogie con dimostrazioni simili precedenti (Larvor 2012; Aberdein 2007), implica che il pubblico cui si rivolge colga i nessi di quanto si viene dimostrando in un dato ambito matematico con ambiti matematici diversi, e possa instaurare così connessioni euristicamente feconde. Una dimostrazione matematica informale viene ritenuta interessante, dunque, perché, sebbene consista nella dimostrazione di un teorema a partire da certi principi matematici e si collochi nel solco della tradizione assiomatica[3], è in grado di innescare anche tutta una serie di inferenze non deduttive e di valutazioni connesse, come i giudizi in merito alla plausibilità di un'ipotesi o di una strategia argomentativa, che non sono meccanizzabili, in quanto sono basati su criteri epistemici non matematici e non formalizzabili, come la plausibilità (Cellucci 2022). Una dimostrazione matematica del genere non è, quindi, formalizzabile nello stesso senso in cui lo è un programma informatico, ovvero lo è una dimostrazione formale, nel senso che non è possibile tradurre integralmente una dimostrazione matematica informale in un linguaggio formale senza che alcuni di quegli aspetti che la rendono una dimostrazione matematica interessante vadano persi (Goethe, Friend 2010).

Molti degli aspetti che rendono interessante e feconda una dimostrazione informale, infatti, non possono essere resi sul piano esclusivamente sintattico. Quando si ha a che fare con una dimostrazione matematica informale non si ha davvero a che fare con la mera manipolazione sintattica di segni privi di significato, come nel caso di una dimostrazione integralmente formalizzata, ma sono piuttosto coinvolti nello sviluppo della dimostrazione in modo essenziale degli aspetti semantici che riguardano gli oggetti matematici coinvolti e le loro proprietà. Quando si lavora con le dimostrazioni formali, come di norma fanno gli studiosi di teoria della

[3] Le dimostrazioni assiomatiche possono essere esplicative o non esplicative (cfr. Cellucci 2022, cap. 14), a seconda che mostrino o meno la ragione della conclusione che viene dimostrata, dato che in una dimostrazione assiomatica una data conclusione può essere dimostrata anche a partire da principi matematici che non sono la vera ragione di quella conclusione. Possiamo approssimativamente considerare, sulla base di tale distinzione, le dimostrazioni matematiche (informali) interessanti cui facciamo qui riferimento, come delle dimostrazioni assiomatiche (informali) esplicative.

dimostrazione, invece, l'elemento cruciale diviene proprio la formalizzazione, e dunque la manipolazione sintattica secondo certe regole di certi segni privati di ogni aspetto semantico. Il punto è che i matematici, di norma, non costruiscono le loro dimostrazioni senza alcuna considerazione del significato matematico e delle implicazioni dei teoremi che intendono dimostrare (Macbeth 2024; Cellucci 2017; Friend 2014; Larvor 2012; Rav 2007).

Che le dimostrazioni matematiche informali e le dimostrazioni formali non possano essere considerate come del tutto equivalenti si evince anche dal fatto che i loro obiettivi sono divergenti. In una dimostrazione formale si è interessati a dimostrare che una certa formula, una certa stringa di segni, formata e manipolata secondo certo regole, e che può essere *eventualmente* interpretata come esprimente una certa asserzione matematica, può essere derivata in un dato sistema formale in un numero finito di passi. Di contro, nei contesti matematici al di fuori della teoria della dimostrazione, l'obiettivo di una dimostrazione informale è mostrare che un certo risultato debba essere accettato perché può essere ricondotto ad altri risultati matematici che sono già stati ritenuti accettabili dalla comunità dei matematici. I matematici che elaborano una dimostrazione informale sviluppano i loro argomenti, attraverso cui veicolano le loro tesi e le loro idee matematiche a sostegno del risultato che intendono dimostrare, facendo riferimento alle caratteristiche matematiche degli oggetti di cui trattano per convincere i loro colleghi e usano il simbolismo matematico e una sorta di formalizzazione incompleta di certi aspetti delle loro dimostrazioni per rendere più preciso, non ambiguo e rigoroso il loro argomentare.

Dunque, l'obiettivo di un matematico che presenta una certa dimostrazione informale D_{inf} per persuadere i propri colleghi che un certo risultato P debba essere accettato, non è lo stesso obiettivo del matematico che intenda dimostrare che un certo risultato P, che magari è già accettato perché per P esiste una certa dimostrazione informale D_{inf}, è derivabile in modo completamente formalizzato all'interno di un dato sistema formale F, ovvero che intenda produrre una dimostrazione formale D_{form} di P. Non solo gli scopi epistemici di D_{inf} e D_{form} differiscono, quindi, ma sono diverse anche le loro potenzialità euristiche.

Le dimostrazioni formali, per perseguire il proprio obiettivo primario, scompongono e strutturano una dimostrazione matematica in un modo che è tale da ostacolare la comprensione del portato matematico del risultato che viene dimostrato e delle sue implicazioni, e quindi possiedono un basso potere euristico ed esplicativo (Macbeth 2015; Robinson 1997). Ma la comprensione dei risultati matematici ottenuti e delle loro implicazioni è cruciale in vista dell'ampliamento della conoscenza matematica, ovvero della formulazione di nuove ipotesi matematiche. Ciò significa

che le dimostrazioni formali non sono in grado di contribuire in maniera significativa all'ampliamento della conoscenza matematica. Ciò relega la ricerca di dimostrazioni formali di risultati matematici già accettati dalla comunità matematica ad ambiti che, sebbene siano certamente legittimi e degni di essere indagati, sono diversi da quello della produzione di *nuova* conoscenza matematica. Un esempio di tali ambiti è la creazione e l'ampliamento di un vasto archivio di dimostrazioni formali di risultati matematici rilevanti (si veda, ad esempio, Anonymous 1994).

È forse importante sottolineare ancora una volta che una dimostrazione formale D_{form} e una dimostrazione informale D_{inf}, anche di uno stesso risultato P, non sono oggetti dello stesso tipo: una dimostrazione matematica formale è a sua volta un oggetto matematico, che può quindi essere manipolato secondo regole matematiche e valutato secondo criteri matematici; al contrario, una dimostrazione matematica informale non è a sua volta un oggetto matematico, è piuttosto un'argomentazione, che, sebbene presenti un esteso ricorso al simbolismo matematico che non si riscontra nelle argomentazioni non matematiche, consta, come qualsiasi altra argomentazione, di una concatenazione di argomenti volti a fare accettare una data asserzione. Ciò implica che i criteri sulla base dei quali una dimostrazione formale può essere valutata non sono gli stessi sulla base dei quali può essere valutata una dimostrazione informale. Infatti, per valutare una dimostrazione formale si possono utilizzare dei criteri formali. Dato che i criteri formali sono formali, appunto, è possibile ritenere, in prima approssimazione, che la valutazione del soddisfacimento di tali criteri possa essere meccanizzata. Di contro, non tutti i criteri attraverso cui viene valutata una dimostrazione informale sono formali, né possono essere formalizzati, dato che, come detto, non sono indipendenti dagli aspetti semantici e perciò non tollerano quel processo di desemantizzazione propedeutico alla formalizzazione. Dunque, la valutazione del soddisfacimento di tali criteri non può essere meccanizzata. Come scrive Larvor, «la validità o invalidità degli […] argomenti informali non dipende esclusivamente dalla loro forma logica, ma anche dal loro contenuto – sono *dipendenti dal contenuto*» (Larvor 2012, p. 720). È questa loro dipendenza dal contenuto semantico che rende impossibile tradurre in un linguaggio formale una dimostrazione matematica informale senza che qualcosa vada perso (Goethe, Friend 2010). Ed è sempre la dipendenza dal contenuto semantico delle dimostrazioni informali la ragione per cui non è affatto scontato che *ogni* dimostrazione matematica possa essere ridotta a una dimostrazione formale. Il rischio è che nel processo di traduzione vengano persi degli elementi della dimostrazione informale che costituivano le ragioni principali per le quali la conclusione dell'argomentazione avrebbe dovuto essere accettata (Larvor 2012; Lakatos 1976).

6. Concezione derivazionista delle dimostrazioni e regresso infinito

Un'altra rilevante difficoltà che affligge la concezione derivazionista delle dimostrazioni matematiche è la seguente. Se, secondo la concezione derivazionista, il rigore della matematica è garantito dal fatto che ogni dimostrazione matematica può, almeno in linea di principio, essere tradotta in una dimostrazione formale, allora l'informazione relativa a come tale traduzione possa essere realizzata dovrebbe essere veicolata dalla dimostrazione matematica informale stessa, altrimenti vi sarebbe il rischio di un regresso infinito. Vediamo perché. Si consideri la dimostrazione informale D_{inf} del teorema T. Non può certo semplicemente assumersi che D_{inf} sia traducibile in una dimostrazione formale, è necessario fornire delle indicazioni, anche in forma schematica, in merito a come tale traduzione, almeno in linea di principio, possa essere condotta. Ora, se l'informazione relativa a come tradurre D_{inf} nella dimostrazione formale di T, D_{form}, non è veicolata dalla stessa D_{inf}, allora un'altra dimostrazione, D_{inf}*, che dimostri come possa tradursi D_{inf} in D_{form}, dovrebbe essere fornita per poter sostenere che T sia stato davvero dimostrato in modo rigoroso in base alla concezione derivazionista. Dunque, in questo caso sarebbe più corretto dire che la vera dimostrazione di T è data dalla congiunzione di D_{inf} e D_{inf}*, ovvero da [D_{inf} + D_{inf}*], piuttosto che dalla sola D_{inf}. Se consideriamo ora D_{inf}*, abbiamo, di nuovo, due possibilità: o D_{inf}* veicola essa stessa l'informazione relativa a come tradurla nella dimostrazione formale D_{form}*, oppure non veicola essa stessa tale informazione. Se D_{inf}* veicola l'informazione relativa a come possa essere tradotta in D_{form}*, allora si può sostenere che, in effetti, l'informazione relativa a come tradurre una dimostrazione matematica informale di T in una dimostrazione matematica formale di T è veicolata dalla dimostrazione matematica informale di T stessa, dato che, come detto, D_{inf}* può essere considerata una parte della dimostrazione informale [D_{inf} + D_{inf}*] di T, e precisamente quella parte che consente di tradurre la dimostrazione informale di T [D_{inf} + D_{inf}*] nella dimostrazione formale di T [D_{form} + D_{form}*]. Se, invece, D_{inf}* non veicola essa stessa l'informazione relativa a come possa essere tradotta in D_{form}*, allora un'altra dimostrazione, D_{inf}**, di come sia possibile tradurre D_{inf}* in D_{form}* dovrà essere fornita per poter sostenere che T sia stato davvero dimostrato in modo rigoroso in base alla concezione derivazionista. Dovremmo a questo punto considerare che la vera dimostrazione di T sia [D_{inf} + D_{inf}* + D_{inf}**] e non solamente [D_{inf} + D_{inf}*]. E così via. Vi è qui una sorta di dilemma: se si accetta la concezione derivazionista delle dimostrazioni matematiche, o le dimostrazioni informali contengono l'informazione relativa a come possano essere tradotte in delle dimostrazioni formali, oppure ci si trova intrappolati in un regresso infinito.

Questo argomento spiega perché la stragrande maggioranza di coloro

che difendono la concezione derivazionista delle dimostrazioni matematiche ritenga che l'informazione relativa a come una dimostrazione informale possa essere tradotta in una dimostrazione formale debba essere veicolata dalla dimostrazione informale stessa. In altri termini, di norma coloro che difendono la concezione derivazionista delle dimostrazioni matematiche adottano una qualche variante di quella concezione delle dimostrazioni matematiche che considera le dimostrazioni matematiche informali come delle "ricette" per poter produrre delle dimostrazioni formali, ovvero come degli schizzi, degli abbozzi, di dimostrazioni formali.

Il problema ora è il seguente: come si deve interpretare l'affermazione secondo cui l'informazione relativa a come sia possibile tradurre una dimostrazione matematica informale in una dimostrazione matematica formale è veicolata dalla dimostrazione matematica informale stessa? Secondo Larvor, la concezione che considera le dimostrazioni informali come delle ricette per produrre dimostrazioni formali sostiene che una data dimostrazione informale D_{inf} di un dato teorema T, che è un'asserzione matematica, non sia altro che un argomento volto a sostenere che esiste una dimostrazione formale adeguata di T, ovvero D_{form}. Dato che l'asserzione "D_{form} esiste e può essere derivata da D_{inf}" è un'asserzione matematica a sua volta, indichiamola con T^*, «per evitare un regresso è necessario un qualche altro resoconto di come la dimostrazione [informale D_{inf} che dimostra T] possa dimostrare [T^*]» (Larvor 2016, p. 403). In altre parole, per evitare un regresso infinito, se difendiamo la tesi che D_{inf} sia in grado di dimostrare T sulla base della concezione secondo cui le dimostrazioni informali sono delle ricette per formulare dimostrazioni formali, non possiamo rendere conto di come D_{inf} sia in grado di dimostrare T^* nello stesso modo, altrimenti ci troveremmo in una situazione come quella illustrata sopra, in cui è necessaria sempre un'ulteriore dimostrazione per potere affermare che un dato risultato sia stato davvero dimostrato. Ma questo conduce a una situazione in un certo senso paradossale: affinché sia possibile evitare un regresso infinito, la concezione che vede le dimostrazioni matematiche informali come delle ricette per produrre dimostrazioni matematiche formali sembrerebbe richiedere che una stessa dimostrazione matematica informale (ad esempio, D_{inf}) possa essere considerata al tempo stesso 1) una ricetta per poter produrre una dimostrazione matematica formale (ad esempio, D_{form}) di un dato risultato (ad esempio, T) e 2) una dimostrazione matematica formale (indichiamola con D_{formT^*}) di un altro risultato (ad esempio, T^*, che afferma che "D_{form} esiste e può essere derivata da D_{inf}"). Avremmo, quindi, che la nostra dimostrazione è al tempo stesso D_{inf} e D_{formT^*}. Ora, è effettivamente difficile anche solo riuscire a immaginare come possa una dimostrazione informale di T essere al contempo una dimostrazione formale di T^*. Come può una dimostrazione che non è formulata in alcun linguaggio formale

essere al contempo una dimostrazione formale adeguata? La concezione delle dimostrazioni informali come delle ricette per produrre dimostrazioni formali sembra dunque inadeguata a mettere al riparo coloro che sostengono la concezione derivazionista delle dimostrazioni matematiche dal rischio che la loro concezione conduca a un regresso infinito.

7. Dimostrazioni formali e meccanizzabilità

Abbiamo messo in luce, nel par. 6, il problema di elaborare una formulazione rigorosa e adeguata della concezione derivazionista delle dimostrazioni matematiche. Abbiamo, inoltre, nel par. 5, messo in luce la difficoltà di sostenere la concezione derivazionista, fornendo ragioni a sostegno della tesi che non sia possibile tradurre davvero ogni dimostrazione matematica informale in una dimostrazione formale del tutto equivalente, perché gli elementi delle dimostrazioni informali connessi agli aspetti semantici andrebbero persi in tale processo di traduzione. Ci siamo concentrati sulla concezione derivazionista perché, per difendere la concezione assiomatica della matematica, i sostenitori di tale concezione fanno riferimento ad alcuni risultati della logica matematica, come l'isomorfismo di Curry-Howard. Come già sottolineato, se adottano tale linea, i sostenitori della concezione assiomatica della matematica non possono limitarsi a concepire genericamente la matematica come dimostrazione di teoremi, ma devono impegnarsi a difendere la concezione derivazionista delle dimostrazioni matematiche.

Ora, il sostenitore della concezione assiomatica della matematica potrebbe sostenere che, nonostante la concezione delle dimostrazioni informali di norma sostenuta dai derivazionisti sia inadeguata, e nonostante le dimostrazioni formali non siano in grado di far progredire la conoscenza matematica perché non facilitano la comprensione matematica, la concezione derivazionista è comunque utile alla difesa della concezione assiomatica, perché per suo tramite si può sostenere l'idea che il rigore e la specificità della conoscenza matematica risposano sul fatto che ogni dimostrazione matematica può essere tradotta, almeno in linea di principio, in una dimostrazione formale e che una dimostrazione formale è epistemicamente superiore a qualunque dimostrazione informale e a qualunque altra forma di giustificazione epistemica possibile, perché proprio per il fatto che una dimostrazione matematica è una dimostrazione formale, e proprio grazie a risultati come l'isomorfismo di Curry-Howard, è possibile, almeno in linea di principio, automatizzare la verifica della correttezza di tale dimostrazione, così da renderla indipendente da qualsiasi soggetto epistemico, cosa che non è possibile fare in nessun ambito della conoscenza umana al di fuori della matematica. Ciò garantirebbe

la peculiare certezza della conoscenza matematica rispetto al resto della conoscenza scientifica e fornirebbe un notevole sostegno alla concezione assiomatica della matematica.

Il problema è che non è davvero possibile difendere in questo modo la concezione assiomatica a causa di un altro risultato della logica matematica, il teorema di Rice (1953). Come abbiamo detto, formalizzare una dimostrazione matematica significa renderla adeguata a essere processata meccanicamente. Ciò si ritiene assicuri il massimo grado di certezza possibile alla conoscenza matematica. L'idea, di ascendenza leibniziana, è infatti che un controllo completamente automatizzato di una determinata dimostrazione possa conferire un grado di giustificazione a tale dimostrazione superiore a quello che potrebbe conferirle un controllo effettuato da un soggetto epistemico umano, o da più soggetti epistemici umani. La ragione è che si ritiene che le macchine siano superiori ai soggetti umani nello svolgere compiti di routine, dato che sono meno soggette degli umani a commettere errori in compiti del genere, e che il controllo della correttezza di una dimostrazione, riguardando solo il rispetto delle regole di manipolazione sintattica delle formule proprie del sistema formale nel quale la dimostrazione è stata elaborata, sia proprio uno di quei compiti di routine in cui le macchine sopravanzano le capacità cognitive umane. Il controllo automatizzato consente di escludere dal processo di verifica ogni soggetto umano ed è proprio ciò che conferisce un alto grado di accuratezza a tale processo rispetto a un processo di verifica svolto da matematici umani. La formalizzazione, quindi, non garantisce tanto un elevato grado di giustificazione epistemica a una dimostrazione perché, esplicitando ogni singolo passo della dimostrazione, la rende più facilmente ispezionabile e giudicabile dai matematici umani, quanto perché rende la verifica di tale dimostrazione almeno potenzialmente indipendente da ogni matematico umano.

Il punto è che anche considerando il solo ambito della verifica delle dimostrazioni formali, tralasciando, quindi, ambiti più creativi del lavoro matematico in cui è più dibattuto se le macchine riescano o meno a emulare le capacità umane, non è comunque possibile affermare che sia possibile meccanizzare del tutto il processo di verifica della correttezza delle dimostrazioni formali, e quindi renderlo indipendente da ogni contributo umano in senso assoluto. Kreisel, ad esempio, scrive che il ragionamento matematico «non ci si presenta come l'esecuzione di regole meccaniche […]. La connessione tra la affidabilità e la possibilità del controllo meccanico è di solito, e un po' acriticamente, data per scontata» (Kreisel 1970, p. 21). Ma la connessione tra l'affidabilità del processo di verifica della correttezza di una dimostrazione e la meccanizzazione di tale processo di verifica non può essere affatto data per scontata.

Per comprendere meglio perché tale connessione non possa darsi per scontata, si consideri che per verificare meccanicamente la correttezza

di una dimostrazione formale data FP_1 si deve sviluppare un programma informatico, CP_1. Supponiamo che CP_1 verifichi la correttezza di FP_1. Possiamo ora considerare FP_1 verificata in modo certo e assoluto? Non proprio. Come abbiamo visto, in base all'isomorfismo di Curry-Howard un programma informatico può essere considerato equivalente a una dimostrazione formale. Dunque, prima di potere affermare con certezza che la correttezza di FP_1 è stata verificata si dovrebbe potere dimostrare la correttezza di CP_1, che possiamo rinominare FP_2. Se per verificare la correttezza di FP_2 sviluppassimo il programma CP_2 ci staremmo ovviamente imbarcando verso un regresso infinito. E non c'è modo di evitare un tale regresso infinito perché non c'è modo di verificare la correttezza di un programma in generale.

La ragione per cui non è possibile in generale verificare la correttezza di un programma informatico è il teorema di Rice (1953), un risultato della teoria della computabilità che è strettamente connesso al risultato di Turing (1937). Il teorema di Rice, infatti, data la relazione che, come abbiamo visto, sussiste tra programmi, dimostrazioni formali e funzioni calcolabili, dimostra che non vi è alcun algoritmo per decidere in generale questioni non banali riguardanti il comportamento dei programmi informatici, dove una questione è banale se riguarda tutti i programmi informatici o nessuno di essi. La correttezza dei programmi è proprio una delle proprietà non banali che non sono decidibili in generale in base al teorema di Rice (Longo 2003, p. 99; Cellucci 2017, p. 217). Il teorema di Rice può essere formulato come segue:

> *Teorema di Rice*: se *S* è una proprietà non banale dei linguaggi accettabili dalle macchine di Turing, allora il problema di determinare se un dato linguaggio *L* accettabile da una data macchina di Turing abbia la proprietà *S* è un problema indecidibile.

È opportuno ricordare che un linguaggio formale è detto accettabile da una macchina di Turing se è un sottoinsieme ricorsivamente numerabile dell'insieme di tutte le possibili sequenze finite costituite dai simboli che appartengono all'alfabeto Σ di tale linguaggio, ovvero se esiste una macchina di Turing che è in grado di enumerare tutte le formule valide di tale linguaggio. Un linguaggio formale è detto decidibile da una macchina di Turing se esiste una macchina di Turing che quando riceve come input una data sequenza finita di simboli che appartengono all'alfabeto del linguaggio si arresta e la accetta se questa è una formula valida di tale linguaggio, oppure si arresta e la rigetta in caso contrario. Vi sono linguaggi formali che sono accettabili da una macchina di Turing, ma che non sono decidibili da una macchina di Turing. Come abbiamo già ricordato, Turing (1937) ha dimostrato che un esempio importante di tali linguaggi è il

linguaggio in cui può essere formulato il problema della fermata, ovvero il problema di decidere se, dato un certo input, un dato programma si arresterà o meno. Turing ha dimostrato che non vi è alcun algoritmo in grado di decidere in generale il problema della fermata, ovvero che non vi è alcuna macchina di Turing in grado di deciderlo. Il teorema di Rice generalizza tale risultato, dato che dimostra che ogni proprietà non banale dei linguaggi accettabili dalle macchine di Turing è indecidibile, ovvero che non vi è alcun algoritmo in grado di decidere in generale se un dato linguaggio L possieda la proprietà non banale S.

Un insieme arbitrario di linguaggi S è detto essere una proprietà dei linguaggi accettabili dalle macchine di Turing se tutti i membri di S sono linguaggi accettabili da una macchina di Turing, ovvero se per ogni linguaggio che appartiene a S vi è una macchina di Turing che lo accetta, ovvero che è in grado di enumerare tutte le formule valide di tale linguaggio. Diciamo, quindi, che un dato linguaggio L ha la proprietà S se $L \in S$. La proprietà S, come detto, è detta banale se l'insieme S è vuoto o se S è l'insieme di tutti i linguaggi accettabili dalle macchine di Turing. Se M è una macchina di Turing, indichiamo con $L(M)$ il linguaggio che è accettato da M. Possiamo definire il linguaggio della proprietà S come segue:

$$L_S = \{\langle M\rangle \mid M(L) \in S\},$$

ovvero come l'insieme dei numeri naturali che codificano tutte quelle macchine di Turing che riconoscono un linguaggio che ha la proprietà S. Ora, per il teorema di Rice, se S è una proprietà non banale, allora L_S non è decidibile. La dimostrazione del teorema di Rice non è molto dissimile dalla dimostrazione del teorema di Turing e da quanto visto sopra in relazione al fatto che non tutte le funzioni computabili sono dimostrabilmente computabili, ovvero assume per assurdo che L_S sia decidibile e che S sia una proprietà non banale, ovvero che enumerati tutti i linguaggi che appartengono a S ed enumerati tutti i linguaggi che non vi appartengono, sia i l'indice del primo linguaggio dell'enumerazione che appartiene a S e sia j l'indice del primo linguaggio dell'enumerazione che non appartiene a S, per cui $i \in S$ e $j \notin S$, e costruisce poi una funzione diagonale totale $C(x)$ che è tale da restituire i se $x \notin S$ e j se $x \in S$, conducendo così a una contraddizione, dato che, per ipotesi, $i \in S$ e $j \notin S$, e conclude che L_S non è decidibile. Dato che, come detto, la correttezza è una proprietà non banale dei programmi informatici, per il teorema di Rice non vi è alcuna procedura meccanica, ovvero alcun algoritmo, per decidere in generale se un dato programma sia corretto o meno.

Come scrive Thurston, «l'affidabilità non deriva dal fatto che i matematici verificano formalmente argomenti formali; deriva dal fatto che i matematici riflettono attentamente e criticamente sulle idee matemati-

che» (Thurston 1994, p. 170). Analogamente, Hersh scrive che, in ultima analisi, è la comprensione umana «che verifica la correttezza della computazione formale» (Hersh 1979, p. 43). Infatti, la formalizzazione non elimina affatto la possibilità di errore, né assicura la correttezza delle dimostrazioni, per il fatto di rendere il processo di verifica della correttezza delle dimostrazioni indipendente dal soggetto epistemico umano, dato che «semplicemente non si dà il caso che una dimostrazione incerta diventi certa per il fatto di venire formalizzata. Al contrario, l'incertezza della dimostrazione sarebbe a quel punto rimpiazzata dall'incertezza della codifica e della programmazione» (*ibidem*).

Analizzata anche questa possibile linea difensiva della concezione assiomatica della conoscenza matematica e mostratane l'inadeguatezza, sembra lecito sostenere che non siano state ancora fornite sufficienti ragioni per ritenere che il mantenimento della concezione assiomatica della matematica sia adeguatamente supportato da alcuni risultati della logica matematica, come l'isomorfismo di Curry-Howard, e dalle prospettive delle ricerche legate all'automazione del ragionamento.

Bibliografia

Aberdein, A. (2007). *The Informal Logic of Mathematical Proof*, in B. Van Kerkhove, J.P. Van Bendegem (eds), *Perspectives on Mathematical Practices*, Dordrecht: Springer, pp. 135-151.

Anonymous (1994). *The QED Manifesto*, in A. Bundy (ed.), *Automated Deduction – CADE-12*, Springer-Verlag, Berlin, pp. 238-251.

Antonutti Marfori, M. (2010). "Informal Proofs and Mathematical Rigour", *Studia Logica*, 96(2): 261-272.

Azzouni, J. (2004). "The Derivation-Indicator View of Mathematical Practice", *Philosophia Mathematica*, 12(2): 81-105.

Boolos, G., Burgess, J., Jeffrey, R. (2002). *Computability and Logic*, 4th ed., Cambridge: Cambridge University Press.

Calude, C.S., Thompson, D. (2016). *Incompleteness, Undecidability and Automated Proofs*, in V.P. Gerdt, W. Koepf, W.M. Seiler, E.V. Vorozhtsov (eds.), *Computer Algebra in Scientific Computing. CASC 2016*, Cham: Springer, pp. 134-155.

Cellucci, C. (2013). *Rethinking Logic: Logic in Relation to Mathematics, Evolution, and Method*, Dordrecht: Springer.

Cellucci, C. (2017). *Rethinking Knowledge: The Heuristic View*, Cham: Springer.

Cellucci, C. (2022). *The Making of Mathematics: Heuristic Philosophy of Mathematics*, Cham: Springer.

Colton, S. (2002). *Automated Theory Formation in Pure Mathematics*, London: Springer-Verlag.

Colton, S., Bundy, A., Walsh, T. (1999). *Automatic Concept Formation in Pure Mathematics*, in D. Thomas (ed.), *Proceedings of the Sixteenth International*

Joint Conference on Artificial Intelligence. Volume 2, San Francisco: Morgan Kaufmann, pp. 786-791.
Curry, H.B. (1934). "Functionality in Combinatory Logic", *Proceedings of the National Academy of Science,* 20(11): 584-590.
Curry, H.B. (1977). *Foundations of Mathematical Logic*, Dover: New York.
Davis, P.J., Hersh, R. (1981). *The Mathematical Experience*, Boston: Birkhäuser.
Dutilh Novaes, C. (2012). *Formal Languages in Logic: A Philosophical and Cognitive Analysis*, Cambridge: Cambridge University Press.
Friend, M. (2014). *Pluralism in Mathematics: A New Position in Philosophy of Mathematics*, Dordrecht: Springer.
Gödel, K. (1931). "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte für Mathematik und Physik*, 38: 173-198.
Gödel, K. (1935). Recensione di: Carnap, R. "Die Antinomien und die Unvollständigkeit der Mathematik", *Monatshefte für Mathematik und Physik*, 41: 263-284 (1934), *Zentralblatt für Mathematik und ihre Grenzgebiete*, 11: 1.
Goethe, N.B., Friend, M. (2010). "Confronting Ideals of Proof with the Ways of Proving of the Research Mathematician", *Studia Logica*, 96(2): 273-288.
Hales, T. (2008) "Formal Proof", *Notices of the AMS*, 55(11): 1370-1380.
Hersh, R. (1979) "Some Proposals for Reviving the Philosophy of Mathematics", *Advances in Mathematics*, 31(1): 31-50.
Hersh, R. (1997). *What Is Mathematic, Really?*, Oxford: Oxford University Press.
Howard, W.A. (1980). *The Formulae-as-Types Notion of Construction*, in J.R. Hindley, J.P. Seldin (eds.), *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, New York: Academic Press, pp. 479-490.
Kreisel, G. (1970). "The Formalist-Positivist Doctrine of Mathematical Precision in the Light of Experience", *L'Âge de la Science*, 3: 17-46.
Kripke, S.A. (2013). *The Church-Turing 'Thesis' as a Special Corollary of Gödel's Completeness Theorem*, in B.J. Copeland, C.J Posy, O. Shagrir (eds.), *Computability*, Cambridge (MA): MIT Press, pp. 77-104.
Lakatos, I. (1976). *Proofs and Refutations*, Cambridge: Cambridge University Press.
Larson, C.E. (2005). *A Survey of Research in Automated Mathematical Conjecture-Making*, in S. Fajtlowicz, P.W. Fowler, P. Hansen, M.F. Janowitz, F.S. Roberts (eds.), *Graphs and Discovery*, Providence (RI): American Mathematics Society, pp. 297-318.
Larvor, B. (2012). "How to Think about Informal Proofs", *Synthese*, 187(2): 715-730.
Larvor, B. (2016). "Why the Naïve Derivation Recipe Model Cannot Explain How Mathematicians' Proofs Secure Mathematical Knowledge", *Philosophia Mathematica*, 24(3): 401-404.
Lavers, G. (2024). *Mathematics is (mostly) Analytic*, Cambridge: Cambridge University Press.
Longo, G. (2003). "Proofs and Programs", *Synthese*, 134(1-2): 85-117.
Macbeth, D. (2015). *Reasoning in Mathematics and Machines: The Place of Mathematical Logic in Mathematical Understanding*, in G. Dodig-Crnkovic,

Y.J. Erden, R. Giovagnoli (eds.), *Proceedings of the AISB 2015 Symposium on Social Aspects of Cognition and Computing*, Canterbury: University of Kent, pp. 3-12.
Macbeth, D. (2024). *Formal Proofs in Mathematical Practice*, in B. Sriraman (ed.), *Handbook of the History and Philosophy of Mathematical Practice*, Cham: Springer, pp. 2113-2135.
Mac Lane, S. (1986). *Mathematics: Form and Function*, New York: Springer-Verlag.
McCarthy, T.G. (1994). "Self-Reference and Incompleteness in a Non-Monotonic Setting", *Journal of Philosophical Logic*, 23(4): 423-449.
Moschovakis, J. (2018). *Intuitionistic Logic*, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, URL = <https://plato.stanford.edu/archives/win2018/entries/logic-intuitionistic/>.
Pantsar, M. (2024). "Theorem Proving in Artificial Neural Networks: New Frontiers in Mathematical AI", *European Journal for Philosophy of Science*, DOI: 10.1007/s13194-024-00569-6.
Pelc, A. (2009). "Why Do We Believe Theorems?", *Philosophia Mathematica*, 17(1): 84-94.
Portoraro, F. (2025). *Automated Reasoning*, in E.N. Zalta, U. Nodelman (eds.), *The Stanford Encyclopedia of Philosophy*, URL = <https://plato.stanford.edu/archives/spr2025/entries/reasoning-automated/>.
Prawitz, D. (2008). *Proofs Verifying Programs and Programs Producing Proofs: A Conceptual Analysis*, in R. Lupacchini, G. Corsi (eds.), *Deduction, Computation, Experiment: Exploring the Effectiveness of Proof*, Berlin: Springer, pp. 81-94.
Rav, Y. (2007). "A Critique of a Formalist-Mechanist Version of the Justification of Arguments in Mathematicians' Proof Practices", *Philosophia Mathematica*, 15(3): 291-320.
Rice, H.G. (1953). "Classes of Recursively Enumerable Sets and Their Decision Problems", *Transactions of the American Mathematical Society*, 74(2): 358-366.
Robinson, J.A. (1997). *Informal Rigor and Mathematical Understanding*, in G. Gottlob, A. Leitsch, D. Mundici (eds.), *Computational Logic and Proof Theory*, Berlin: Springer-Verlag, pp. 54-64.
Rodin, A. (2014). *Axiomatic Method and Category Theory*, Cham: Springer.
Sørensen, M.H., Urzyczyn, P. (2006). *Lectures on the Curry-Howard Isomorphism*, Amsterdam: Elsevier.
Tanswell, F. (2015). "A Problem with the Dependence of Informal Proofs on Formal Proofs", *Philosophia Mathematica*, 23(3): 295-310.
Thurston, W.P. (1994). "On Proof and Progress in Mathematics", *Bulletin of the American Mathematical Society*, 30(2): 161-177.
Turing, A.M. (1937). "On Computable Numbers, with an Application to the *Entscheidungsproblem*", *Proceedings of the London Mathematical Society* 42(1), pp. 230-265.
von Plato, J. (2018). *The Development of Proof Theory*, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, URL = <https://plato.stanford.edu/archives/win2018/entries/proof-theory-development/>.
Wadler, P. (2015). "Propositions as Types", *Communications of the ACM*, 58(12): 75-84.

Weir, A. (2021). *Formalism in the Philosophy of Mathematics*, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, URL = <https://plato.stanford.edu/archives/sum2021/entries/formalism-mathematics/>.
Weyl H. (1949). *Philosophy of Mathematics and Natural Science*, Princeton: Princeton University Press.

Concezione assiomatica, dimostrazioni formali e meccanizzabilità della matematica

Secondo la concezione assiomatica della matematica, fare matematica significa principalmente dimostrare teoremi, ovvero produrre dimostrazioni in un sistema formale a partire da determinati assiomi. Per Hilbert, la formalizzazione della matematica aveva come scopo principale fornire una fondazione sicura alla matematica. La concezione assiomatica è stata messa in crisi dai teoremi di Gödel, che mostrano come lo scopo che Hilbert assegnava alla formalizzazione della matematica non possa essere raggiunto. Molti sostenitori della concezione assiomatica ritengono però che ciò non debba significare l'abbandono di tale concezione e affermano che altri risultati della logica matematica, come l'isomorfismo di Curry-Howard, forniscono ragioni sufficienti al mantenimento di tale concezione della matematica. Scopo di questo articolo è valutare tale affermazione.

PAROLE CHIAVE: metodo assiomatico, ragionamento automatizzato, dimostrazioni informali, isomorfismo di Curry-Howard, teoremi di incompletezza di Gödel.

The Axiomatic View, Formal Proofs, and Mechanizability of Mathematics

According to the axiomatic view of mathematics, doing mathematics means primarily proving theorems, that is, producing proofs in a formal system starting from certain axioms. For Hilbert, the main purpose of formalizing mathematics was to provide a secure foundation for mathematics. The axiomatic view was called into question by Gödel's theorems, which showed that the purpose that Hilbert assigned to the formalization of mathematics could not be achieved. Many supporters of the axiomatic view believe, however, that this should not mean abandoning this view and claim that other results of mathematical logic, such as the Curry-Howard isomorphism, provide sufficient reasons to maintain this view of mathematics. The purpose of this article is to evaluate that claim.

KEYWORDS: axiomatic method, automated reasoning, informal proofs, Curry-Howard isomorphism, Gödel's incompleteness theorems.