Polinomis i Lògica Matemàtica *

Jörg Flum
Institut für matematische Logik
Albert-Ludwigs-Universität Freiburg
D-79 104 Freiburg i. Br.
flum@sun1.ruf.uni-freiburg.de

Considerem tres problemes relatius a polinomis. Els tres es poden relacionar amb tota una àrea de la lògica matemàtica. Mentre que els dos primers estan resolts i corresponen a parts ja clàssiques de la lògica matemàtica, el tercer és un problema obert que pertany a una àrea relativament jove. Heus aquí —molt breument exposats— els problemes que més endavant veurem amb més deteniment.

- Existeix un algorisme amb el qual poguem decidir, per a qualsevol polinomi $p(x_1, \ldots, x_\ell)$, amb coeficients enters, si té un zero en els enters, és a dir, si existeix $(z_1, \ldots, z_\ell) \in \mathbb{Z}^\ell$ tal que $p(z_1, \ldots, z_\ell) = 0$?

- Suposem donat un polinomi $p(x_1, \ldots, x_\ell)$ sense cap zero a $\mathbb{Z}$. Existeix una demostració del fet que p no té cap zero?

- Existeix un algorisme i un polinomi p tals que, per a cada n ciutats, poguem determinar amb alguna mena d'algorisme i amb menys de $p(n)$ passos un camí de longitud mínima que passi per aquestes n ciutats?

Vegem el primer problema. Suposem que $p(x) = a_n\, x^n + \cdots + a_1\, x^2 + a_0$, amb $a_n \neq 0$, és un polinomi amb coeficients enters en la variable x, $p(x) \in$

*Es tracta de la redacció d'una conferència que vaig tenir l'honor de donar l'11 de maig de 1 993 a la *Societat Catalana de Matemàtiques* de l'*Institut d'Estudis Catalans*. Vull aprofitar l'avinentesa per manifestar el meu agraïment al *Centre de Recerca Matemàtica*, centre del qual en vaig ser investigador convidat des del febrer fins al maig.

$\mathbb{Z}[x]$. Si $x \neq 0$ i $p(x) = 0$, aleshores

$$x^n \cdot \left(1 + \frac{a_{n-1}}{a_n \cdot x} + \cdots + \frac{a_1}{a_n \cdot x^{n-1}} + \frac{a_0}{a_n \cdot x^n}\right) = 0.$$

En resulta que

$$|x| \leq x_0 := \text{màx}\left\{1, \left|\frac{n \cdot a_{n-1}}{a_n}\right|, \ldots, \left|\frac{n \cdot a_0}{a_n}\right|\right\}$$

(atès que, si $|x| > x_0$, aleshores

$$\left|\frac{a_{n-1}}{a_n \cdot x} + \cdots + \frac{a_1}{a_n \cdot x^{n-1}} + \frac{a_0}{a_n \cdot x^n}\right| < \left|\frac{1}{n}\right| + \cdots + \left|\frac{1}{n}\right| = 1$$

i, per tant, $p(x) \neq 0$).

Tenim doncs el mètode següent per tal de determinar si un polinomi $p(x) \in \mathbb{Z}[x]$ té un zero a $\mathbb{Z}$:

- Calculem x_0
- Comprovem si un dels enters de $[-x_0, x_0]$ és un zero de $p(x)$.

En el Congrés Internacional de Matemàtiques, celebrat a París l'any 1 900, Hilbert [5] va presentar 23 problemes de la resolució dels quals n'esperava considerables impulsos en àrees diverses de les matemàtiques. El desè era:

- "10. Entscheidung der Lösbarkeit einer diophantischen Gleichung. Eine dipophantische Gleichung mit irgendwelchen Unbekannten und mit ganzen rationalen Zahlkoeffizienten sei Vorgelegt; man soll ein Verfahren angeben, nach welchem sich mittels einer endlichen Anzahl von Operationen entscheiden lässt, ob die Gliechung im ganzen rationalen Zahlen lösbar ist."

És a dir, es demana l'existència d'alguna mena d'algorisme que, per a tot

$$p \in \text{POL} := \bigcup_{\ell \geq 1} \mathbb{Z}[x_1, \ldots, x_\ell],$$

determini si p té un zero a $\mathbb{Z}$. Vegem-ne alguns exemples:

- $x - y$ té un zero: $x = 6, y = 6$
- $x^2 - 17y^2 - 1$ té un zero: $x = 1, y = 0$

- $(x+1)^3+(y+1)^3-(z+1)^3$ no té cap zero
- $x+1$ té un zero: $x=-1$.

Amb el concepte que segueix reformularem el problem 10 de Hilbert.

Definició 1 *Sigui* $p(x_1,\ldots,x_\ell)\in$ POL. *Diem que* p *és* resoluble *ssi* p *té un zero en els nombres naturals, és a dir si existeixen* $n_1,\ldots,n_\ell\in\mathbb{N}$ *tals que* $P(n_1,\ldots,n_\ell)=0$.

Per exemple, $x+1$ té un zero, però no es resoluble.

Un polinomi $p(x,y)$ té un zero en els nombres enters ssi

$$p(x,y)\cdot p(-x,y)\cdot p(x,-y)\cdot p(-x,-y)$$

té un zero en els nombres naturals. D'altra banda sabem que cada nombre natural és la suma de quatre quadrats; per tot això, $p(x,y)$ té un zero en els nombres naturals ssi

$$p(x_1^2+\cdots+x_4^2,y_1^2+\cdots+y_4^2)$$

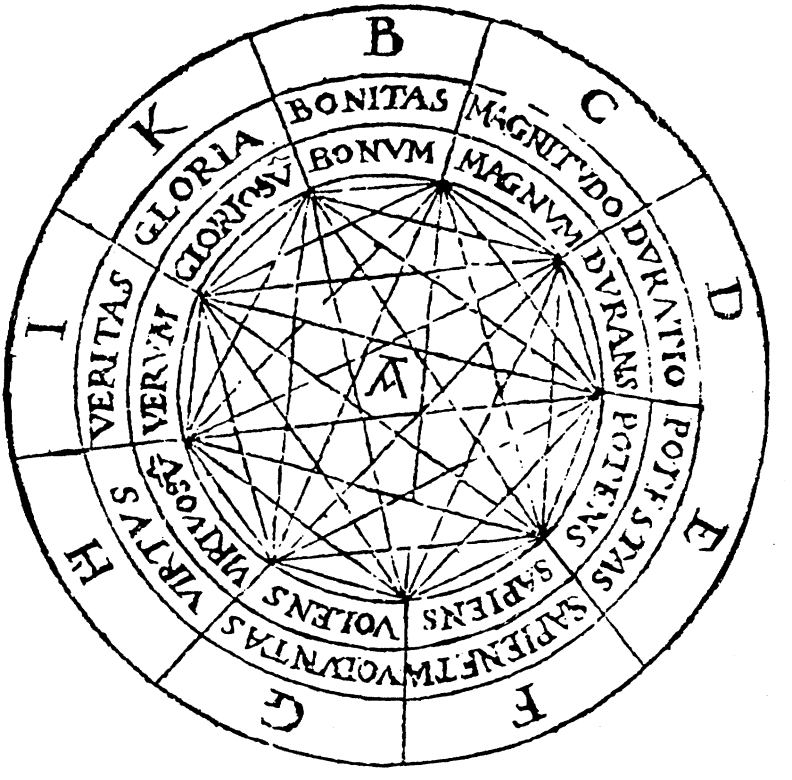
té un zero en els nombres enters. Utilitzant aquestes equivalències podem reformular el problema 10 de Hilbert:

- Existeix un algorisme que, per a tot $p\in$ POL, determini si p és resoluble?

A continuació considerarem el problema 10 de Hilbert d'aquesta forma. Ens demanem per l'existència d'un algorisme —Hilbert ens parlava d'un "Verfahren", d'un mètode, d'un procediment general—, paraules totes elles que usaré sinònimament; avui ens preguntaríem per un programa de computadora.

La història del concepte d'algorisme està relacionada amb la història dels Països Catalans. La paraula algorisme té la seva rel en el nom del matemàtic àrab Al-Khwārizmī (aproximadament del 800) que, en un llibre, va recopilar els algorismes de l'àlgebra. Aquests mètodes matemàtics, introduïts pels àrabs, van inspirar al mallorquí Ramon Lull l'*Ars Magna* que pretenia ser un procediment general, un algorisme de base combinatòria, que servís per trobar totes les veritats. Vegeu a continuació discos giratoris, concebuts per Ramon Llull. Les diverses configuracions que

s'obtenen en fer-los girar permeten de produir sempre enunciats nous, veritats noves.



Llull va fracassar en el seu intent; els procediments que introdueix tenen un valor escàs. Però el fet del seu fracàs és secundari. Allò que realment importa és que, a la seva obra, tingué lloc la concepció d'una idea certament esplèndida. Aquesta idea va tenir una gran influència d'una forma especial en la posterioritat matemàtica. Per exemple, uns 200 anys més tard, Cardano, l'any 1 545, publicava algorismes algèbrics tot referint-se a Llull, com palesa també el títol de la seva obra "Artis magnæ seu de regula algebraicis liber unus".

Per a Descartes el punt de partida és el seu desig d'aconseguir un tractament algorísmic de les matemàtiques. Creia que l'essencial de la seva geometria analítica residia en el fet que, amb el seu ajut, tots els problemes geomètrics podrien ser traduïts en problemes algèbrics, i aleshores foren susceptibles de ser tractats amb els algorismes desenvolupats per

l'àlgebra.

Leibniz manifesta de forma expressa que estava influït per Llull. Extreu de l'*ars magna* lul·liana dos conceptes: l'*ars inveniendi* i l'*ars iudicandi*, dels quals parlarem més endavant. També va estudiar l'essència de la noció d'algorisme i es va adonar amb claredat que ha de ser possible confiar a una màquina l'ejecució d'algorismes. En relació amb això cal recordar que Leibniz fou el primer que construí una màquina de calcular.

Malgrat tots aquests esforços, l'any 1 900 quan Hilbert va presentar els seus problemes, encara s'utilitzava una noció intuïtiva d'algorisme. També avui, si algú ens presenta un algorisme per resoldre un problema matemàtic, en general, tothom acceptarà que es tracta efectivament d'un algorisme, malgrat que alguns solament coneixen la noció intuïtiva d'algorisme. Però per tal de poder demostrar que un problema no pot ser tractat amb cap algorisme, normalment no serà pas suficient disposar del concepte intuïtiu d'algorisme, sinò que serà necessari tenir-ne una definició exacta.

Aquesta noció no es va introduir fins l'any 1 936. Foren Church i Turing els qui, independentment i per vies diferents, van aconseguir una definició matemàtica del concepte d'algorisme de la qual encara avui hom està convençut que capta de forma correcta la noció intuïtiva d'algorisme.

Vegem algun dels aspectes d'aquesta definició. Els mètodes s'apliquen a paraules sobre un alfabet, per exemple sobre l'alfabet $A = \{a_0, \ldots, a_r\}$, format per les lletres $a_0, \ldots, a_r$. Sigui A^* el conjunt de paraules, on una paraula és qualsevol successió finita d'aquestes lletres.

L'experiència ensenya que qualsevol mena de problema tractable algorísmicament pot ser codificat fent servir les paraules d'un cert alfabet —amb paraules treballen també les actuals computadores.

Per exemple, l'alfabet

$$A_0 := \{0, 1, \ldots, 9\} \cup \{\bar{0}, \bar{1}, \ldots, \bar{9}\} \cup \{\underline{0}, \underline{1}, \ldots, \underline{9}\} \cup \{x, +, -\}$$

serveix per codificar polinomis; per exemple, el polinomi

$$3\, x_1^2\, x_2 - 5\, x_1\, x_3^4$$

podem codificar-lo amb la paraula de A_0^*:

$$3\, x \underline{1} \bar{2}\, x\, \underline{2} - 5\, x\, \underline{1}\, x\, \underline{3}\, \bar{4}.$$

Per aquesta raó podem considerar POL i POL_r subconjunts de A_0^*, on

$$\mathrm{POL}_r := \{p \in \mathrm{POL} \mid p \text{ és resoluble}\}.$$

Implícitament, Leibniz distingia ja, en connexió amb els algorismes, dos conceptes, l'*ars iudicandi* i l'*ars inveniendi*; avui parlaríem d'algorismes de decidibilitat i d'enumerabilitat:

Definició 2 *Sigui A un alfabet i $X \subseteq A^*$. X és* decidible *ssi existeix un algorisme que aplicat a qualsevol $\xi \in A^*$ respon, després d'un nombre finit de passos, a la pregunta "$\xi \in X$?"*

El problema 10 de Hilbert ens pregunta doncs

- És POL_r decidible?

Vegem l'exemple d'un conjunt decidible. Sigui $A_1 := \{0, \ldots, 9\}$. Aleshores essencialment podem identificar A_1^* amb el conjunt $\mathbb{N}$ dels nombres naturals. El conjunt PRIM dels nombres primers (en forma decimal) és decidible. Tots hem aprés al col·legi un algorisme per decidir si un nombre natural és primer. Analitzant sistemàticament tots els nombres naturals, podem confeccionar una llista dels nombres primers. PRIM és, doncs, un conjunt enumerable conforme a la definició que segueix:

Definició 3. *Sigui A un alfabet i $X \subseteq A^*$. X és* enumerable *ssi existeix un algorisme que, posat en funcionament, ens dóna com a outputs exactament els elements de X (en qualsevol ordre).*

Com ja hem vist en el cas de PRIM, tenim:

Proposició 1 *Si X és decidible, aleshores X és enumerable.*

Demostració. Analitzem, per a $n = 1, 2, \ldots$, totes les paraules de longitud n amb el mètode de decisió. □

Proposició 2 *POL_r és un conjunt enumerable.*

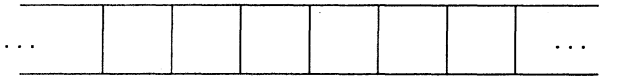
Demostració. Tenim el següent mètode de decisió: per a $n = 1, 2, \ldots$, produïm tots els polinomis amb variables entre $\{x_1, \ldots, x_n\}$, amb coeficients a $\{-n, -n+1, \ldots, n\}$ i amb exponents a $\{0, 1, \ldots, n\}$. I, per a cada un d'ells, comprovem si té un zero a $\{0, 1, \ldots, n\}$. En cas afirmatiu, anotem el polinomi. □

Evidentment, per a qualsevol polinomi $p(x_1, \ldots, x_\ell) \in \mathrm{POL}$, el conjunt $\mathbb{N}-\mathrm{val}(p)$ de valors de p en els nombres naturals és enumerable.

$$\mathbb{N} - \mathrm{val}(p) := \{p(n_1, \ldots, n_\ell) \mid n_1, \ldots, n_\ell \in \mathbb{N}, p(n_1, \ldots, n_\ell) \geq 0\}$$

(produïm sistemàticament totes les ℓ-ples $(n_1, \ldots, n_\ell) \in \mathbb{N}^\ell$ i calculem $p(n_1, \ldots, n_\ell)$).

A les nostres definicions i demostracions hem treballat amb una noció intuïtiva d'algorisme. Per tal de donar la noció exacta d'algorisme, Turing [11] va introduir la noció de màquina i de programa. Vull donar una idea de la seva definició: La memòria d'una màquina de Turing està formada per una cinta, potencialment infinita en ambdós sentits, dividida en cel·les.



Cada cel·la de la cinta és capaç de contenir una lletra i, en tot moment d'una computació, gairebé totes les cel·les estan buides. Per tal de captar de forma fidel la noció d'algorisme, Turing va fer una anàlisi, un estudi dels passos de càlcul més simples possibles en un algorisme. Es va convèncer del fet que, en un pas elemental, la màquina solament pot treballar amb una cel·la, essent les accions possibles:

- observar-ne el contingut
- esborrar-ne el contingut
- reemplaçar el contingut per una altra lletra
- passar a la cel·la de la dreta
- passar a la cel·la de l'esquerra

i l'acció només pot dependre

- de la lletra observada
- de la instrucció a executar, allò que avui en diríem el lloc del programa en què ens trobem.

Amb aquesta anàlisi, Turing va donar una noció exacta. El primer en proposar la identificació del concepte intuïtiu d'algorisme amb l'exacte de Turing va ser Church. Hi ha diverses raons que parlen a favor d'aquesta identificació. Vegem-ne dues:

- Hi ha hagut molts suggeriments per precisar la noció d'algorisme o, si es vol, dels conjunts decidibles i enumerables. Malgrat que s'originen parcialment a partir de consideracions inicials força diverses, sempre han resultat ser equivalents.

- Fins ara sempre ha estat possible de simular tots els algorismes i tots el programes escrits en qualsevol llenguatge de programació amb algorismes en el sentit de Turing.

Amb aquestes nocions exactes l'any 1 970 Matijasevic [9] (vegeu també [2]) va poder demostrar que coneixem molt bé els conjunts enumerables, si més no els que són subconjunts de $\mathbb{N}$; en efecte:

Lema 1 (Lema de Matijasevic) *Sigui* $X \subseteq \mathbb{N}(\approx \{0,\ldots,9\}^*)$. *Aleshores* X *és enumerable ssi existeix* $p \in POL, p = p(x_1,\ldots,x_\ell)$, *tal que* $X = N - val(p)$ *(recordem que* $\mathbb{N} - val(p) := \{p(n_1,\ldots,n_\ell) \mid n_1,\ldots,n_\ell \in \mathbb{N}, p(n_1,\ldots,n_\ell) \geq 0\}$*).*

I, amb aquest lema, Matijasevic va donar una solució negativa al problema 10 de Hilbert:

Teorema 1 (Teorema de Matijasevic) POL_r *no és decidible.*

Demostració. Suposem que POL_r fos decidible. Enumerem tots el polinomis $p_0, p_1, \ldots$ Sigui

$$X_0 := \{n \in \mathbb{N} \mid n \notin \mathbb{N} - \mathrm{val}(p_n)\}.$$

Atès que

$$n \in X_0 \quad \text{ssi} \quad q := p_n - n \notin P_r,$$

veiem que X_0 és decidible i, per tant, enumerable. Pel lema de Matijasevic existeix un n_0 amb $X_0 = \mathbb{N} - \mathrm{val}(p_{n_0})$. Amb això aconseguim la següent contradicció:

$$n_0 \in \mathbb{N} - \mathrm{val}(p_{n_0}) \quad \text{ssi} \quad n_0 \in X_0 \quad \text{ssi} \quad n_0 \notin \mathbb{N} - \mathrm{val}(p_{n_0}). \qquad \square$$

Amb la solució negativa de Matijasevic al problema 10 de Hilbert obtenim un conjunt clarament definit, POL_r, que no té solució algorísmica. El primer conjunt d'aquesta índole fou aconseguit l'any 1936 i posava de manifest les limitacions dels mètodes algorísmics en matemàtiques.

Fixem-nos que fins ara no hem tingut en compte per a res les limitacions que vénen donades per la manca de temps o de memòria, la qual cosa farem quan estudiem el tercer problema que hem esmentat a la introducció, el problema anomenat el *Traveling Salesman Problem* TSP: Suposem que un representant ha de visitar n clients de n ciutats. El representant busca un algorisme

a) que li proporcioni una ruta òptima, és a dir, una ruta el més curta possible que passi per les n ciutats

o, en una primera aproximació, estaria prou content si disposava d'un algorisme

b) que li digués, per a una longitud k, si existeix una ruta que tingui, com a màxim, una longitud de k kilòmetres.

Abreugem per TSP_b el problema en la seva forma b). Evidentment existeixen solucions algorísmiques a aquests problemes: només cal analitzar de forma sistemàtica totes les rutes possible. Hi ha $n!$ rutes i, per a $n = 100$, una computadora no serà pas capaç de comprovar totes les rutes en un temps acceptable. Existeix un algorisme practicable, és a dir, un algorisme que doni una solució en el temps disponible per al problema?

Direm que un algorisme és *polinòmic*, si existeix un polinomi p tal que, per a cada input ξ, l'algorisme fa, com a màxim, p(longitud de ξ) passos abans de parar-se. Designem amb la lletra P el conjunt de problemes que tenen un algorisme de decidibilitat polinòmica. L'experiència ensenya que, per a problemes que s'esdevenen de forma natural en les matemàtiques o que són d'importància pràctica, l'existència d'un algorisme de decidibilitat practicable és equivalent al fet que pertanyi a P.

Si, per a un nombre natural n, un oracle ens proporciona "testimonis" m i k, amb $n = m \cdot k$, podem verificar aquesta igualtat, o sigui que podem verificar que n pertany al conjunt COMP de nombres compostos en temps polinòmic. El mateix succeix amb TSP_b: donades les ciutats i el nombre k, si algú ens proporciona un testimoni, una ruta de longitud inferior a k kilòmetres, podrem verificar aquesta propietat en un temps polinòmic. Denotem per $\mathcal{VP}$ el conjunt de problemes verificables, en aquest sentit, en temps polinòmic. Hem vist que TSP_b, COMP$\in \mathcal{VP}$. Es possible de demostrar que PRIM$\in \mathcal{VP}$. Evidentment tenim que $\mathcal{P} \subseteq \mathcal{VP}$. Hom ha conjecturat que $\mathcal{P} \neq \mathcal{VP}$ (i sabem que $\text{TSP}_b \in \mathcal{P}$ implicaria $\mathcal{P} = \mathcal{VP}$).

L'any 1 991, l'Institut d'Estudis Catalans va organitzar el Symposium on the Current State and Prospects of Mathematic [1]. A la discussió final un dels participants va expressar l'opinió que el problema "$\mathcal{P} = \mathcal{NP}$?" és el problema més important i rellevant de les matemàtiques de la segona meitat del segle XX,

- ... that the greatest problem in mathematics in the last half of the twentieth century comes from the computer. It is the problem "Is $\mathcal{P} = \mathcal{NP}$?"

(aquí es parla de $\mathcal{NP}$, del conjunt de problemes amb algorisme de decisió no determinístic. És fàcil de veure que $\mathcal{NP} = \mathcal{VP}$, en els passos no derterminístics l'algorisme pot endevinar un testimoni).

Totes aquestes teories no li serveixen pas de gaire al viatjant que ha de sortir de viatge. Pregunta: No existeix cap algorisme polinòmic que em proporcioni una ruta de longitud acceptable? Per exemple, existeix un algorisme que, aplicat a un ϵ i a un conjunt C de ciutats, proporcioni una ruta, ruta(C), amb

$$\left|\frac{\text{longitud(ruta}(C)) - \text{longitud(opt}(C))}{\text{longitud(opt}(C))}\right| \leq \epsilon,$$

on opt(C) designa una ruta de longitud mínima? Resultats obtinguts els darrers anys (vegeu [10]) demostren que, a vegades, per a problemes semblants, l'existència d'aquesta mena d'algorismes es pot deduir de l'estructura lògica del problema.

A la darrera part d'aquest article intentarem de tractar el problema que hem mencionat a la introducció, en el segon lloc. Sigui p un polinomi no resoluble, o sigui que no tingui cap zero en el domini dels nombres naturals. Existeix una demostració d'aquest fet? Ens trobem, de bell nou, amb una situació similar a la que teníem amb la noció d'"algorisme": Sempre que, per a un enunciat matemàtic, disposem d'una demostració, els matemàtics acceptaran que es tracta efectivament d'una demostració. Però com podem fer-ho per demostrar que no hi ha cap demostració?

Fixem-nos que estem tractant un problema central de les matemàtiques. En moltes teories matemàtiques partim d'un sistema Φ d'axiomes i volem estudiar les seves conseqüències. Sigui ϕ un enunciat de la teoria. Diem que ϕ és una *conseqüència* de Φ, si ϕ és vàlida en tots els models de Φ. Per exemple, ϕ és una conseqüència del sistema Φ_0 d'axiomes per a grups, si ϕ és vàlida en tots els models de Φ_0, és a dir, en tots els grups. Evidentement, si existeix una demostració de ϕ a partir de Φ, aleshores ϕ és una conseqüència de Φ. Val també el recíproc?

- Sigui ϕ una conseqüència de Φ. Podem aleshores assegurar que ϕ és demostrable a partir de Φ, és a dir, que existeix una demostració de ϕ a partir de Φ?

Per exemple, si ϕ és vàlida en tots els grups, existeix una demostració de ϕ a partir de Φ_0? Un resultat central de la lògica matemàtica conté una resposta afirmativa d'aquesta pregunta. Evidentment, per tal de donar un tractament precís d'aquest problema cal, en primer lloc, introduir

conceptes precisos. Per a la noció d'"enunciat" s'introdueix un llenguatge formal: el *llenguatge L_1 de primer ordre*, que és un conjunt de paraules sobre un alfabet. En el llenguatge de primer ordre, de forma semblant als llenguatges de programació, està clarament definit quines fileres de signes són enunciats o, com s'anomenen en matemàtiques, *fórmules*. Per exemple, els axiomes de la teoria de grups corresponen a les fórmules següents:

$$\forall x \forall y \forall z (x \circ (y \circ z)) = ((x \circ y) \circ z)$$
$$\forall x (x \circ e = x)$$
$$\forall x \exists y (x \circ y = e).$$

L'experiència ensenya que tot enunciat pot ser expressat d'una manera més o menys directa en L_1, o sigui que L_1 és un llenguatge universal per a les matemàtiques. Per tal de precisar la noció de "demostració", Gödel va fer una anàlisi similar a la que havia fet Turing amb la noció d'algorisme: va estudiar de quina manera podien ser els passos més senzills de què consta una demostració i s'adonà que aquests passos corresponien a regles formals d'aquesta mena:

$$\frac{\phi \wedge \varphi}{\phi} \qquad \frac{(\varphi \to \phi) \quad \varphi}{\phi}$$

on el primer esquema representa la regla que permet passar, en una demostració, d'una conjunció al seu primer membre. Gödel donà un conjunt Γ finit de regles en les quals basà la seva noció exacta de demostració matemàtica.

Definició 4. *La fórmula ϕ és* formalment deduïble *de Φ ssi existeixen $\varphi_0, \ldots, \varphi_n$ tals que, per a $i = 0, \ldots, n$, tenim que*

- $\varphi_i \in \Phi$ *o*
- φ_i *s'obté aplicant una de les regles de Γ als φ_j, amb $j < i$.*

Evidentment, a cada deducció formal li correspon una demostració en el sentit usual de les matemàtiques. Gödel [7] va demostrar:

Teorema 2 (Teorema de completesa) *Si ϕ és una conseqüència de Φ, aleshores ϕ és formalment deduïble a partir de Φ.*

És a dir, per a tota conseqüència existeix una deducció formal.

Aplicant de forma sistemàtica totes les regles de deducció de Γ es demostra

Proposició 3 *Si Φ és decidible, aleshores*

$$\Phi^{cons} := \{\phi \mid \phi \text{ és una conseqüència de } \Phi\}$$

és enumerable.

Si s'esdevé que el conjunt Φ és el conjunt buit, veiem que el conjunt d'enunciats vàlids és enumerable, o sigui que existeix un algorisme que ens confecciona una llista de totes les "veritats matemàtiques", un algorisme que, en un pla diferent i en un altre temps, era el que buscava Llull.

A la matemàtica, en general, fem ús d'algunes propietats dels objectes matemàtics, propietats que no postulem pas explícitament. Per exemple que, per a conjunts X i Y, existeix la unió $X \cup Y$ i el conjunt $\mathrm{Pot}(X)$ de tots els subconjunts de X; o que, per a funcions f i g, existeix la concatenació $f \circ g$. L'experiència ens ensenya que

- podem suposar que tots els objectes matemàtics són conjunts, i que
- en el llenguatge de primer ordre hi ha un sistema ZFC d'axiomes —un sistema d'axiomes per a la teoria de conjunts— que conté precisament les propietats de conjunts que el matemàtic utilitza normalment.

Des d'aquest punt de vista podem justificar la igualtat

$$\text{matemàtiques} \quad = \quad \text{ZFC} \quad + \quad \text{deduccions formals}$$

i per aquesta raó és correcte dir que el mètode matemàtic és un mètode axiomàtico-deductiu.

Quines són les seves limitacions? Gödel [8] va establir que aquest mètode no permet pas demostrar la consistència de les matemàtiques (en el supòsit que siguin consistents).

Estudiem aquest resultat amb una mica més de deteniment i vegem-ne la seva relació amb el problema sobre polinomis que ens interessa. Per tot el que hem dit, sabem que la consistència de la matemàtica equival a la consistència de ZFC que, al seu torn, equival al fet que la fórmula $\neg(x = x)$ no sigui deduïble formalmente del conjunt d'axiomes ZFC. Si, per a un alfabet $A = \{a_1, \ldots, a_r\}$, identifiquem per exemple la paraula $a_3\, a_6\, a_3$ amb el nombre natural $2^3 \cdot 3^6 \cdot 5^3$, ens adonem que, a cada paraula,

correspon un nombre natural i que, a cada conjunt (enumerable) $\mathcal{B} \subseteq \mathcal{A}^*$, correspon un conjunt (enumerable) $X_{\mathcal{B}} \subseteq \mathbb{N}$. Per la Proposició 3 sabem que ZFC^{cons} és enumerable i, per tant, $X_{\mathrm{ZFC}^{cons}}$ és un subconjunt enumerable de $\mathbb{N}$. Existeix doncs (vegeu el Lema 1) un polinomi p amb $\mathbb{N} - \mathrm{val}(p) = X_{\mathrm{ZFC}^{cons}}$. Sigui n_0 el número natural que correspon a la paraula $\neg(x = x)$ i sigui q el polinomi $p - n_0$. És clar que tenim la següent cadena d'equivalències:

Les matemàtiques són consistents

ssi ZFC és consistent

ssi $\neg(x = x)$ no és deduïble formalment dels axiomes de ZFC

ssi $n_0 \notin X_{\mathrm{ZFC}^{cons}}$

ssi $n_0 \notin \mathbb{N} - \mathrm{val}(p)$

ssi q no és resoluble.

Així doncs, el resultat de Gödel —esmentat més amunt— sobres les limitacions del mètode axiomàtic, el podem formular de la forma següent:

Teorema 3 (Teorema d'incompletesa de Gödel) *Si el polinomi q no és resoluble, aleshores no podem pas demostrar que p no és resoluble.*

A la introducció hem esmentat que el tres problemes —dels que hem partit— pertanyen a tres àrees diferents de la lògica matemàtica. En efecte, el problema 10 de Hilbert pertany a la teoria de la computabilitat, el Traveling Salesman Problem a la teoria de la complexitat i el darrer problema que hem tractat a la teoria de la demostració. Els llibres [3], [4] i [6] constitueixen, respectivament, introduccions a la lògica matemàtica, a la teoria de la computabilitat i a la teoria de la demostració.

Referències bibliogràfiques

[1] Casacuberta, C., Castellet, M. *Mathematical Research Today and Tomorrow*. Lectures Notes in Mathematics 1 525, (1 992).

[2] Davis, M. Matijasevic, Y.V., Robinson, J. *Hilbert's tenth problem. Diphantine equations: positive aspects of a negative solution*, Proceedings of Symp. in Pure Mathematics 28, AMS, (1 976), 323-378.

[3] Ebbinghaus, H.-D., Flum, J., Thomas, W. *Mathematical Logic*. Springer-Verlag, (1 984).

[4] Hermes, H. *Enumerability, Decidibility, Computability.* Springer-Verlag, (1 969).

[5] Hilbert, D. *Gesammelte Werke*, Bd. 3, Springer-Verlag, (1 970).

[6] Garey, M.R., Johnson, D.S. *Computers and Intractability.* W.H. Freeman and Company (1 979).

[7] Gödel, K. *Die Vollständigkeit der Axiome des logischen Funktionenkalküls*, Monatshefte für Mathematik und Physic, 37 (1 931), 349-360.

[8] Gödel, K. *Über formal unentscheidbare sätze der Principia Mathematica und verwandter Systeme I*, Monatshefte für Mathematik und Physic, 38 (1 931), 173-198.

[9] Matijasevic, J.V. *Enumerable sets are diophantic*, Soviet Math. Dokl. 11 (1 970), 345-357.

[10] Papadimitriou, C. Yannakakis, M. *Optimization, approximation and complexity*, Jour. Comp. Syst. Sciences 43 (1 991), 425-440.

[11] Turing, A.M. *On Computable Numbers with an Application to the Entscheidungsproblem*, Proc. Lond. Math. Soc. (2) 42 (1 937), 230-265.