Butlletí de la Societat Catalana de Matemàtiques
Vol. 18, núm. 1, 2003. Pàg. 7–17

Complexitat de Kolmogorov i qüestions de fonament*

Albert Atserias

1 Introducció

Considerem les dues seqüències binàries següents:

$$01010101010101010101010101010101$$
$$01110010010101110010100100101 0.$$

Totes dues consten d'uns quants zeros i uns quants uns, fins a un total de trenta símbols. Tanmateix, la primera seqüència és, a simple vista, molt més regular i fàcil de descriure que la segona. En efecte, es tracta d'una seqüència de quinze blocs 01 seguits. D'altra banda, l'afirmació que la segona seqüència no admet una descripció tan senzilla, tot i semblar òbvia, és, si més no, agosarada. En primer lloc, què és el que entenem per *descripció*? En segon lloc, en quin sentit és «quinze blocs zero-un seguits» una descripció més senzilla que cap altra? I en tercer lloc, assumint que hem sabut respondre a les preguntes anteriors, hi ha manera de demostrar la inexistència de descripcions millors?

Per respondre aquestes preguntes podem imaginar-nos que hem fixat un llenguatge de descripció i un intèrpret capaç de generar seqüències binàries a partir de la seva descripció. D'aquesta manera, una *descripció* no és més que una seqüència finita de símbols que pertanyi al llenguatge i podem convenir que una descripció és més senzilla que una altra si, com a seqüència finita, és *més curta* en longitud. Això suggereix les preguntes següents: com escollim el llenguatge de descripció? I l'intèrpret? Com podem garantir que l'elecció no és totalment arbitrària i lliure de contradiccions? També cal preguntar-se: per a què serveix una teoria de la simplicitat de descripcions?

∗ Conferència pronunciada a la sisena Trobada Matemàtica de la SCM, celebrada el 4 d'abril de 2003.

Oblidem-nos per un moment de les qüestions de fonament i imaginem-nos que hem convingut que el citat llenguatge de descripció és un fragment del català. Convé assumir que el llenguatge de descripció és inambigu i complet; és a dir, que cada descripció descriu una única seqüència, i que cada seqüència té almenys una descripció. Imaginem-nos ara que volem transmetre per telèfon una seqüència binària mitjançant un canal de veu que es paga per temps d'ús. Clarament, preferirem transmetre la descripció «mil blocs zero-un seguits» que no pas la descripció literal de la seqüència de dos mil caràcters que li correspon «zero un zero un zero un...». La primera és molt més curta i per tant més barata de transmetre, i a més aporta la informació suficient per recuperar la seqüència en qüestió. No és difícil d'imaginar-se seqüències amb descripcions curtes que no siguin tan òbvies, fins i tot en situacions de la vida quotidiana (pista: zip).

Veiem, doncs, que si busquem alguna utilitat a una teoria de la simplicitat de les descripcions basada en longitud, podem trobar-la en el camp de la compressió de dades. És dubtós que la motivació de Kolmogorov per formular-la, així com la dels seus coetanis Solomonoff i Chaitin, fos únicament una tan tecnològica. La immensa generalitat que es pot assolir en el desenvolupament de la teoria tot mantenint l'elegància, i la diversitat de camps que penetra, des de la teoria de la probabilitat fins a la lògica matemàtica, bé valen l'estudi matemàtic en l'estil més clàssic i general.

Ens proposem l'objectiu d'exposar els fonaments de la complexitat de Kolmogorov deduint-los a partir d'una solució a la paradoxa de Berry. Acabem amb l'observació que la solució dóna lloc a una demostració del primer teorema d'incompletesa de Gödel. Tots els resultats i proves que presentem són clàssics i ben coneguts. Esperem, però, que l'exposició resultarà accessible per a una àmplia majoria dels lectors. Anem a pams, i comencem per les qüestions de fonament.

2 La paradoxa de Berry

Tornem a la qüestió que fins ara hem deixat de banda: què és una descripció? Per veure com és de delicada aquesta qüestió recordarem la coneguda paradoxa de Berry.[1] Aquesta surgeix quan volem considerar

el menor nombre natural que no es pot descriure
en menys de quinze paraules.

Si aquest nombre existeix, aleshores l'hem descrit en menys de quinze paraules perquè la frase anterior en té catorze. Si aquest nombre no existeix, aleshores tot nombre natural es pot descriure en menys de quinze paraules i per tant n'hi ha un nombre finit. Què ha fallat?

1 Alguns autors, com Li i Vitányi al seu llibre *An Introduction to Kolmogorov Complexity and its Applications*, s'hi refereixen com *la paradoxa de Richard-Berry*. Sovint s'atribueix la seva publicació a Russell.

Comencem la teoria definint el marc de treball. Sigui $\mathcal{O}$ un conjunt numerable i infinit d'objectes a descriure. Per exemple, $\mathcal{O}$ és el conjunt dels nombres naturals en el cas de la paradoxa de Berry. Com a llenguatge de descripció $\mathcal{D}$ utilitzarem el conjunt dels mots sobre l'alfabet binari, és a dir, el conjunt de les seqüències finites de zeros i uns:

$$\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots\}.$$

Com a intèrpret, de moment farem servir qualsevol funció f del conjunt de les descripcions $\mathcal{D}$ en el conjunt dels objectes $\mathcal{O}$. Per cert, si $f(x) = y$, és a dir, si la imatge de x per f és y, diem que x és una descripció de y.

L'elecció de Σ^* com a llenguatge de descripció no és perquè sí. És ben conegut, especialment des que els ordinadors formen part de la nostra vida diària, que l'alfabet binari és capaç de codificar qualsevol esquema sintàctic de comunicació de manera eficient. Dit això, observem que el fet que $\mathcal{O}$ sigui una col·lecció d'*objectes* no té especial rellevància, excepte pel fet que sigui numerable i infinit. Per les mateixes raons que hem escollit Σ^* com a llenguatge de descripció $\mathcal{D}$, podríem haver demanat que $\mathcal{O}$ fos Σ^* ell mateix, i així ho farem per a la resta de l'article. Així doncs, adoptant aquesta disciplina, un intèrpret no és més que un renombrament de cadascu dels elements de Σ^*; una traducció, si es vol. Destaquem també que cada objecte pot tenir més d'una descripció (de fet, infinites) atès que no requerim que la funció f sigui injectiva. També es pot donar el cas que un objecte no tingui cap descripció segons l'intèrpret f.

Arribats a aquest punt, i tenint en compte el destacadíssim paper que donem a Σ^* dins la teoria, convé entrar-hi en més detall.

3 L'alfabet binari i codificacions

Els elements de Σ^* s'anomenen *mots binaris*, o simplement *mots*. El símbol λ denota el mot buit. La longitud d'un mot $x \in \Sigma^*$ es denota per $|x|$ i és el nombre de símbols que el componen. Així doncs, $|0110| = 4$ i $|\lambda| = 0$. Sobre el conjunt dels mots podem definir una operació de concatenació. Donats dos mots $x \in \Sigma^*$ i $y \in \Sigma^*$, denotem per $x \cdot y$ la concatenació de x i y, és a dir, el mot que resulta d'afegir els símbols de y al final dels de x. Per exemple, $0010 \cdot 100 = 0010100$. Clarament, la concatenació és una operació associativa i el mot buit λ és l'únic element neutre. Sovint ometrem el símbol $\cdot$ i escriurem xy en comptes de $x \cdot y$. També farem servir la notació x^n per denotar la concatenació de x amb ell mateix n vegades. Així doncs, $x^3 = xxx$.

Com ja ha quedat dit, una de les propietats més interessants dels mots sobre l'alfabet binari és la seva capacitat de codificar. Potser la més coneguda és la codificació dels nombres naturals en binari. Farem ús extensiu d'aquesta codificació. De fet, identificarem un nombre natural amb la seva codificació en binari sense zeros a l'esquerra. Així doncs, si $n \in \mathbb{N}$ és un nombre natural, tindrà sentit parlar de la seva longitud $|n|$ com a mot. Ja que hi som, fixem-nos que $|n| = \lceil \log_2(n+1) \rceil$, i que $|0| = 0$ la qual cosa és consistent amb $|\lambda| = 0$.

Una codificació potser menys coneguda és la que permet codificar un parell de mots binaris en un de sol. La idea és que dos mots $x \in \Sigma^*$ i $y \in \Sigma^*$ es poden codificar en un de sol així: $1^{|x|}0xy$. Per exemple, si $x = 01010$ i $y = 11$, aleshores la codificació del parell $(x, y) \in \Sigma^* \times \Sigma^*$ és el mot

$$\langle x, y \rangle = 1^{|x|}0xy = 1^5 0 \cdot 01010 \cdot 11 = 1111100101011.$$

La gràcia d'aquesta codificació es que, donat $\langle x, y \rangle$, és fàcil recuperar x i y: només cal buscar la posició del primer zero i això ens dóna la longitud de x, i per tant la de y. Fixem-nos que si agaféssim xy com a codificació del parell $(x, y) \in \Sigma^* \times \Sigma^*$ no sabríem on hem de tallar per obtenir cadascuna de les parts. Per cert, la longitud de $\langle x, y \rangle$ és $2|x| + 1 + |y|$.

4 Intèrprets computables

Un intèrpret ha de servir per recuperar objectes a partir de les seves descripcions de manera mecànica, sistemàtica i consistent. Idealment, l'intèrpret hauria de ser una màquina que sempre retorni la mateixa sortida amb una mateixa entrada, i que el procés per arribar-hi consisteixi a executar una llista d'instruccions fixada *a priori*. Això ens porta a proposar que els intèrprets siguin funcions computables, és a dir, funcions calculades mitjançant una màquina de Turing, per dir un dels diversos models de càlcul equivalents.

Recordem que una màquina de Turing és un model abstracte de càlcul idealitzant el concepte de màquina d'estats amb una memòria externa de lectura i escriptura potencialment infinita. Vegeu la figura 1. Tot i que no necessitarem la definició formal de màquina de Turing, potser convé recordar breument en què consisteix. Es tracta d'un conjunt finit d'estats Q i una funció de transició $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, on Γ és un alfabet finit, que indica quin és el comportament de la màquina. Intuïtivament, quan la màquina es troba a l'estat $q \in Q$ i llegeix el símbol $a \in \Gamma$ de la memòria externa mitjançant un capçal de lectura/escriptura, la transició $\delta(q, a) = (q', a', M)$ indica que el símbol a de la memòria es substitueix per $a' \in \Gamma$, el capçal es desplaça a dreta o esquerra segons si $M = R$ o $M = L$, i el nou estat intern de la màquina és $q' \in Q$. El còmput de la màquina acaba quan s'assoleix un estat especial de Q, anomenat *estat final*. Vista d'aquesta manera, una màquina de Turing computa una funció (possiblement parcial). L'entrada és el mot sobre l'alfabet Γ contingut a la memòria a l'inici del còmput, i la sortida és el mot sobre l'alfabet Γ contingut a la memòria al final, si és que acaba.

Com ja ha quedat dit, no necessitarem la definició formal i suggerim un text clàssic com [4] per al lector interessat. Tot el que necessitarem saber és que el conjunt de les màquines de Turing es pot enumerar $M_0, M_1, \ldots$ de manera natural, i que el conjunt de les funcions computades per les màquines de Turing coincideix amb el conjunt de les funcions computades per qualsevol altre model de càlcul introduït fins al moment com és, sense anar més lluny, qualsevol llenguatge de programació com ara el PASCAL, el C, o el JAVA. Atès

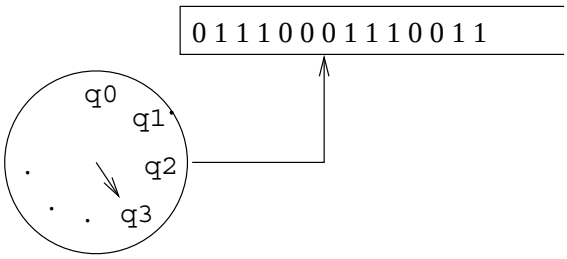

FIGURA 1: Una màquina de Turing.

que les màquines de Turing es poden enumerar, és immediat que el conjunt de les funcions computables mitjançant màquines de Turing també admet una enumeració $\phi_0, \phi_1, \ldots$. A partir d'aquest moment les anomenarem *funcions computables*. Dit sia de passada, que les funcions computables no han de ser totals. En efecte, una màquina de Turing pot no aturar-se per alguna entrada (quedar-s'hi penjada) i això deixaria la funció computada indefinida per aquella entrada. Finalment, assumirem que les màquines de Turing treballen únicament amb l'alfabet binari i per tant computen funcions parcials de Σ^* en Σ^*.

5 Complexitat de Kolmogorov

Ja tenim tots els ingredients per definir el concepte de complexitat de Kolmogorov:

1 DEFINICIÓ (DE COMPLEXITAT DE KOLMOGOROV GENERAL) *Sigui ϕ una funció computable i sigui $x \in \Sigma^*$. La complexitat de Kolmogorov de x segons ϕ es defineix així:*

$$C_\phi(x) = \inf\{|y| : y \in \Sigma^*,\ \phi(y) = x\}.$$

Si $\phi(y) = x$ diem que y és una descripció de x segons ϕ.

Hi ha dos aspectes en aquesta definició que hem de destacar. En primer lloc, es pot donar el cas que no existeixi cap descripció de x segons ϕ. En aquest cas obtenim $C_\phi(x) = \infty$ usant el conveni que $\inf \varnothing = \infty$. Idealment, desitjaríem que el nostre intèrpret ϕ fos exhaustiu sobre Σ^* de manera que cap mot no tingui complexitat infinita. Per les aplicacions, sovint és aquest el cas. En segon lloc, fixem-nos que la definició preveu una complexitat de Kolmogorov per cada funció computable ϕ. Això deixa oberta la possibilitat que un mot $x \in \Sigma^*$ tingui complexitat molt petita segons un intèrpret i molt gran segons un altre. Això suggereix la pregunta següent: existeix una complexitat de Kolmogorov intrínseca a cada mot? És a dir, ens preguntem si podem definir una complexitat de Kolmogorov de manera que no depengui essencialment de l'intèrpret.

Per tal de respondre la pregunta anterior recorrerem al concepte de funció universal de Turing. Possiblement la fita més important de Turing dins la seva teoria de les funcions computables fóra el descobriment de l'existència de funcions universals. La simple existència d'aquestes ens pot semblar, avui en dia, pràcticament òbvia. Al cap i a la fi, els ordinadors actuals no en són més que una realització física.

2 TEOREMA (DE LA FUNCIÓ UNIVERSAL, TURING 1936) *Sigui* $\{\phi_n : n \in \mathbb{N}\}$ *l'enumeració de les funcions computables. Existeix un índex* $U \in \mathbb{N}$ *tal que per a tot* $n \in \mathbb{N}$ *i* $x \in \Sigma^*$ *tenim*

$$\phi_U(\langle n, x\rangle) = \phi_n(x).$$

En particular, si ϕ_n *està indefinida sobre* x*, aleshores* ϕ_U *està indefinida sobre* $\langle n, x\rangle$*.*

Amb l'eina de la funció universal ja podem formular i demostrar el primer resultat de la teoria. Aparentment, Kolmogorov acostumava a atribuir-lo a Turing en les seves ponències sobre el tema.

3 TEOREMA (D'INVARIÀNCIA, KOLMOGOROV 1965 I CHAITIN 1969) *Existeix una funció computable* Φ *tal que per cada funció computable* ϕ *i cada* $x \in \Sigma^*$ *es té*

$$C_\Phi(x) \leq C_\phi(x) + c_\phi,$$

on c_ϕ *és una constant que només depèn de* ϕ*.*

DEMOSTRACIÓ: Sigui $\Phi = \phi_U$, on U és l'índex d'una funció universal de Turing. Sigui $\phi = \phi_n$ una funció computable qualsevol. Per tot $y \in \Sigma^*$ tenim que

$$\Phi(\langle n, y\rangle) = \phi(y).$$

En particular, si $\phi(y) = x$, aleshores $\Phi(\langle n, y\rangle) = x$. Fixem-nos que $|\langle n, y\rangle| = 2|n|+1+|y|$. Per tant, si definim $c_\phi = 2|n|+1$ obtenim que $C_\Phi(x) \leq C_\phi(x)+c_\phi$ per cada $x \in \Sigma^*$. □

El teorema d'invariància enuncia l'existència d'una complexitat de Kolmogorov intrínseca en el sentit que cap intèrpret computable, per més especialitzat que estigui, pot assolir una complexitat significativament més petita en algun mot. Fixem-nos també que una conseqüència immediata del teorema és que la complexitat de Kolmogorov respecte de la funció Φ de l'enunciat mai no és infinita. En efecte, si ϕ computa la funció identitat, aleshores $C_\phi(x) = |x|$ per cada $x \in \Sigma^*$, i per tant $C_\Phi(x) \leq |x| + c_\phi$ per cada $x \in \Sigma^*$.

Vist això, a partir d'aquest punt serà convenient i justificat fixar una Φ, com diu el teorema d'invariància, d'una vegada per totes, i escriure $C(x)$ en comptes de $C_\Phi(x)$.

6 Exemples

Acabem de justificar que per cada $x \in \Sigma^*$ tenim $C(x) \leq |x| + c_{id}$, on c_{id} és la constant del teorema d'invariància corresponent a la funció identitat. Sovint no ens importarà quina és la constant en qüestió i per tant escriurem $C(x) \leq |x| + c$ deixant la constant c sense especificar. Tot seguit veurem que hi ha mots que tenen complexitat de Kolmogorov molt més petita que la seva pròpia longitud.

Sigui $x = (01)^n$ el mot considerat a la primera secció consistent en n blocs 01 seguits. La longitud d'aquest mot és $2n$. Tanmateix, per tal de reconstruir x només ens cal saber n perquè hi ha una funció computable que, donat n, escriu n blocs 01 seguits. Per tant, $C(x) \leq |n| + c$. Sabent que $|n| = \lceil \log_2(n+1) \rceil$, tenim que $C(x)$ és exponencialment més petit que la pròpia longitud de x.

Podem anar una mica més lluny perquè per tal de reconstruir x n'hi ha prou de conèixer n, i per tal de reconstruir n n'hi ha prou de conèixer una descripció de n. Per tant, $C(x) \leq C(n) + c$. En general, $C(n)$ no serà gaire més petit que $|n|$, però en certs casos, com quan n és una potència de dos $n = 2^m$, la complexitat $C(n)$ està afitada per $|m| + c$ que, aquest cop sí, és exponencialment més petit que $|n|$. Òbviament podem iterar aquest raonament. N'hi ha prou de conèixer una descripció de m, i si m és una potència de dos $m = 2^p$, la seva complexitat està afitada per $|p| + c$. Això demostra que $C(x)$ pot arribar a ser molt petit respecte de $|x|$. Tanmateix, tot seguit veiem que en la majoria dels casos, $C(x)$ no és molt lluny de $|x|$.

4 LEMA (DELS INCOMPRESSIBLES, KOLMOGOROV 1965) *Per a tot $m \in \mathbb{N}$ i $n \in \mathbb{N}$ tals que $n \geq m$, existeixen com a mínim $2^n - 2^m + 1$ mots $x \in \Sigma^*$ de longitud n tals que $C(x) \geq m$.*

DEMOSTRACIÓ: Hi ha 2^n mots de longitud n i només $2^m - 1$ possibles descripcions de longitud menor que m. Per tant, com a mínim $2^n - 2^m + 1$ mots de longitud n tenen complexitat m o més. □

En particular, si posem $m = n - 1$, el lema diu que més de la meitat dels mots de longitud n tenen complexitat $n - 1$ o més. Un cas particularment interessant és quan $m = n$. En aquest cas, el lema diu que existeix un mot $x \in \Sigma^*$ de longitud n per al qual totes les seves descripcions tenen longitud n o més. Són les seqüències per a les quals no tenim una manera més barata de transmetre-les per telèfon que cantar-les literalment: «zero un un un zero zero un...». Aquest tipus de mots s'anomenen *incompressibles* per raons òbvies.

7 Solució a la paradoxa

Tenim totes les eines per formalitzar la paradoxa de Berry i veure què passa. Per a cada $m \in \mathbb{N}$, considerem el mínim nombre natural que no es pot des-

criure en menys de m símbols, si existeix. Formalment, consideram la funció

$$f(m) = \min\{m' \in \mathbb{N} : C(m') \geq m\}.$$

Les preguntes següents són immediates: està definit $f(m)$ per a tot $m \in \mathbb{N}$? Hem definit $f(m)$ mitjançant una descripció en el sentit de les seccions anteriors? La resposta a la primera pregunta és afirmativa i per demostrar-ho farem ús del lema dels incompressibles.

Considerem el conjunt dels mots de longitud $m+1$. D'entre aquests, n'hi ha com a mínim $2^m + 1$ que tenen complexitat m o més, i d'entre aquests, algun ha de començar per 1 perquè només hi ha 2^m mots de longitud $m + 1$ que comencen per 0. Aquest mot, interpretat com a nombre natural, és el nostre m' per al qual $C(m') \geq m$. Per tant, $f(m)$ està definit per a qualsevol $m \in \mathbb{N}$.

Quant a la segona pregunta que ens hem formulat, això és el que podem dir. Fixem-nos que per a què $\min\{m' \in \mathbb{N} : C(m') \geq m\}$ sigui una descripció n'hi hauria prou que $C(x)$ fos una funció computable. Si fos així, aleshores f també seria computable i per tant de la forma ϕ_n per algun $n \in \mathbb{N}$. Això garantiria que $f(m)$ estigués descrit per $\langle n, m\rangle$ ja que $\phi_U(\langle n, m\rangle) = \phi_n(m) = f(m)$. Curiosament, el que ens diu la paradoxa de Berry, com veurem tot seguit, és precisament que això no és possible i, per tant, que $C(x)$ no és una funció computable.

5 TEOREMA (DE LA INCOMPUTABILITAT DE C, KOLMOGOROV 1965) *La funció $C(x)$ no és computable.*

DEMOSTRACIÓ: Suposem que ho fos. Aleshores f també seria computable i per tant $f = \phi_n$ per algun $n \in \mathbb{N}$. Sigui $m \in \mathbb{N}$ suficientment gran, de manera que $|\langle n, m\rangle| < m$. Recordem que

$$|\langle n, m\rangle| = 2|n| + 1 + |m| = 2\lceil \log_2(n+1)\rceil + 1 + \lceil \log_2(m+1)\rceil$$

i per tant un m com el que necessitem existeix. Sigui $m' = f(m)$. Es dóna el cas que $\langle n, m\rangle$ és una descripció de m' perquè

$$\phi_U(\langle n, m\rangle) = \phi_n(m) = f(m) = m'.$$

Alhora tenim que $C(m') \geq m$ per la pròpia definició de f. Això contradiu el fet que $|\langle n, m\rangle| < m$. Per tant $C(x)$ no és computable. □

8 Incompletesa

Hem vist que la paradoxa de Berry ens ha dut a concloure l'existència de funcions no computables amb una prova realment elegant. Però si ens hi endinsem una mica més ens adonem que podem demostrar més coses.

L'argument per demostrar que $C(x)$ no és computable es basa en la idea següent: si ho fos, aleshores podríem verificar si $C(m') \geq m$ per a qualsevol

$m' \in \mathbb{N}$, i per tant computar el mínim m' tal que $C(m') \geq m$. Formulat d'aquesta manera, ens adonem que no només $C(x)$ és no computable, sinó que ni tan sols podem *verificar* si $C(x) \geq m$. Això suggereix la idea que puguem tenir $C(x) \geq m$, però això no és *demostrable*.

Surt de l'objectiu d'aquesta exposició discutir el terme «demostrable» des del punt de vista de la lògica matemàtica. Tanmateix, sí que estem en disposició de formalitzar el concepte de demostració des d'un punt de vista purament abstracte. De fet, fent-ho així obtenim resultats sorprenentment generals.

Recordem que si f és una funció de Σ^* en Σ^*, aleshores $f(\Sigma^*)$ denota el conjunt de les imatges $\{f(x) : x \in \Sigma^*\}$.

6 DEFINICIÓ (DE SISTEMA DE DEMOSTRACIONS ABSTRACTE) *Sigui $A \subseteq \Sigma^*$ un conjunt no buit de mots. Un sistema de demostracions per A és qualsevol funció computable i total f tal que $f(\Sigma^*) \subseteq A$. Si $f(x) = y$, diem que x és una demostració que $y \in A$. Si la inclusió $f(\Sigma^*) \subseteq A$ és una igualtat diem que f és complet. Altrament diem que és incomplet.*

En paraules planeres, un sistema de demostracions abstracte com el de la definició anterior no és més que una manera mecànica de verificar demostracions. Aquesta definició inclou les de la lògica matemàtica com a casos particulars. N'hi ha prou a fer que A sigui el conjunt de les codificacions de les fórmules vàlides en el model estàndard dels axiomes sota consideració, i fer que f verifiqui demostracions en el sistema formal en qüestió. Com ja ha quedat dit, però, no és el nostre objectiu entrar en aquesta mena de detalls.[2] Finalment, podem enunciar el resultat d'incompletesa. Recordem que el conjunt dels incompressibles és $\{x \in \Sigma^* : C(x) \geq |x|\}$.

7 TEOREMA (D'INCOMPLETESA, SEGONS CHAITIN 1974) *Qualsevol sistema de demostracions per als incompressibles és incomplet.*

DEMOSTRACIÓ: Sigui f un sistema de demostracions qualsevol per als incompressibles i suposem per contradicció que f és complet. Sigui

$$g(m) = \min\{x \in \Sigma^* : |f(x)| \geq m\},$$

on mínim significa dins l'ordre lexicogràfic per longituds. Sigui $h(m) = f(g(m))$. Si assumim que f és complet i ja que existeix un incompressible de cada longitud (lema dels incompressibles), les funcions g i h són totals. També són computables perquè f ho és i per tant $h = \phi_n$ per algun n. Sigui m suficientment gran tal que $|\langle n, m\rangle| < m$. Un m com el que necessitem existeix per la mateixa raó que a la prova del teorema de la incomputabilitat de C. Sigui $x = h(m)$. Òbviament, x és incompressible perquè $x = h(m) = f(g(m))$ i f només retorna incompressibles. D'altra banda,

2 Potser convé mencionar que no és difícil veure que un conjunt no buit té un sistema de demostracions complet si i només si és enumerable recursivament.

$|x| \geq m$ perquè $x = h(m) = f(g(m))$ i $g(m)$ és tal que $|f(g(m))| \geq m$ per definició. Finalment,

$$\phi_U(\langle n, m\rangle) = \phi_n(m) = h(m) = x$$

i per tant

$$C(x) \leq |\langle n, m\rangle| < m \leq |x|.$$

Això contradiu la incompressibilitat de x. □

En particular, qualsevol teoria per a l'aritmètica suficientment forta per formular sentències del tipus $C(x) \geq |x|$ és incompleta. Val a dir que la hipòtesi sobre la qual reposa aquesta demostració és que la teoria en qüestió només demostra fórmules vàlides de l'aritmètica. Aquesta hipòtesi està implícita en la condició $f(\Sigma^*) \subseteq A$ de la nostra definició de sistema de demostracions. En canvi, la hipòtesi sobre la qual reposa la demostració original de Gödel, en versió Rosser, és simplement la consistència. Una altra diferència significativa és que la demostració de Gödel dóna una fórmula concreta que és indemostrable, mentre que la presentada aquí en dóna infinites però de manera no constructiva.

Agraïments

Vull agrair els comentaris d'en Ricard Gavaldà a una versió preliminar d'aquest article.

Referències

[1] CHAITIN, G. J. «On the length of programs for computing finite binary sequences: statistical considerations». *Journal of the ACM*, 16 (1969), 145–159.

[2] CHAITIN, G. J. «On the simplicity and speed of programs for computing infinite sets of natural numbers». *Journal of the ACM*, 16 (1969), 407–422.

[3] CHAITIN, G. J. «Information-theoretic limitations of formal systems». *Journal of the ACM*, 21 (3) (1974) 403–424.

[4] HOPCROFT, J.; ULLMAN, J. *Introduction to automata theory, languages, and computation*. Addison-Wesley, 1979.

[5] KOLMOGOROV, A. N. «Three approaches to the quantitative definition of information». *Problems of Information Transmission*, 1 (1) (1965) 1–7.

[6] LI, M.;VITÁNYI, P. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag, 1993.

[7] SOLOMONOFF, R. J. «A formal theory of inductive inference, part 1 and part 2». *Information and Control*, 7 (1964), 1–22, 224–254.

[8] TURING, A. M. «On computable numbers, with an application to the Entscheidungsproblem». *Proceedings of the London Mathematical Society*, 42 (1936), 230–265, 1936.

DEPARTAMENT DE LLENGUATGES I SISTEMES INFORMÀTICS
UNIVERSITAT POLITÈCNICA DE CATALUNYA
atserias@lsi.upc.es