Matemática computable

Antonio Montalbán.
U. de Chicago

Coloquio Uruguayo de Matemática. Diciembre, 2009

Qué es un algoritmo?

Definición: Un conjunto $A \subseteq \mathbb{N}$ es *computable* si existe un programa que, $\forall n$, decide si $n \in A$.

Tesis de Church-Turing: Esta definición es independiente del lenguaje, y captura exactamente la noción de algoritmo.

Ejemplos: Los siguientes conjuntos son computables:

- El conjunto de números pares.
- El conjunto de números primos.
- El conjunto de los programas escritos correctamente.

P: ¿Porque restringirnos solamente a $\mathbb{N}$?
Todo objeto finito puede ser codificado con un número.
Ej: $\langle a_1, ..., a_n \rangle$ puede ser codificada como $2^{a_1} \cdot 3^{a_1} \cdot ... \cdot p_n^{a_n}$.

Ejemplos de conjuntos no computable

El problema de la palabra: Consideren los grupos definidos a partir de una cantidad finita de generadores y de relaciones entre los generadores. El conjunto de pares (generadores, relaciones), de groupos **no-triviales** no es computable.

Variedades simplemente conexas: El conjunto de triangulaciones de variedades **simplemente conexas** no es computable.

El décimo problema de Hilbert: El conjunto de polinomios en $\mathbb{Z}$ con varias variables y con **soluciones en** $\mathbb{Z}$ no es computable.

El problema de la parada:
El conjunto de programas que **paran** no es computable.

Computabilidad relativa

Dados $A, B \subseteq \mathbb{N}$, *$A$ es computable en B*, ($A \leqslant_T B$), si hay un programa φ^B que usa a B como *oráculo*, y t.q.

$$\varphi^B(n) = 1 \Leftrightarrow n \in A$$

A es *Turing-equivalente* a B, ($A \equiv_T B$) si $A \leqslant_T B$ y $B \leqslant_T A$.

Ejemplo: Los siguientes conjuntos son Turing-equivalentes.

1. {pares (generadores, relaciones) de grupos no-triviales};
2. $\{p \in \mathbb{Z}[X_1, X_2, ...] : p$ tiene raíces en $\mathbb{Z}\}$;
3. {programas que paran}.

Grados de Turing

Def: Las clases de equivalencia de $\equiv_T$ se llaman *grados de Turing*.

Obs:

- Dado A, el conjunto $\{B \subseteq \mathbb{N} : B \leqslant_T A\}$ es numerable.
- Cada grado de Turing es numerable.
- Hay una cantidad no-numerable de grados de Turing.
- $0 = \{\text{conjuntos computables}\}$ es el $\leqslant_T$-menor grado de Turing.

El salto de Turing

Sea $\varphi_0, \varphi_1, \varphi_2, \ldots$ una enumeración de los programas que usan un oráculo.
Dado $A \subseteq \mathbb{N}$,
φ_e^A es el programa y la función obtenida usando oráculo A.

Def: El *salto de Turing* de A es

$$A' = \{(e, n) : \varphi_e^A(n) \downarrow\}$$

Lemma:

- $A \leqslant_T B \Rightarrow A' \leqslant_T B'$,
- $A \equiv_T B \Rightarrow A' \equiv_T B'$,
- $A <_T A'$.

Pricipales Sub-areas de teoría de la computabilidad

1. Teoría de la computabilidad pura.
2. Matemática computable.
3. Matemática reversa.
4. Aleatoriedad effectiva.

Objetivos de la Matemática computable

Estudiar la complejidad de

- construcciones,
- procesos,
- demostraciones, y
- estructuras matemáticas.

Relacionar propiedades computacionales
con propiedades estructurales.

Ejemplo 1: Grafos

Consideren un grafo $G = (V, E)$ infinito numerable.
Podemos suponer que $V \subseteq \mathbb{N}$.

Def: G es *computable* si $V \subseteq \mathbb{N}$ y $E \subseteq \mathbb{N}^2$ son computables.

Def: Dado $v \in V$, la *componente conexa* de v es

$$C(v) = \{w \in V : \exists x_0, ..., x_k \in V \\ (v, x_0) \in E \ \& \ (x_0, x_1) \in E \ \& \ ... \ \& \ (x_{k-1}, x_k) \in E \ \& \ (x_k, w) \in E\}.$$

P: ¿Si G es computable, es $C(v)$ computable?
P: ¿Hay algún oráculo que calcule $C(v)$ para todo G computable?

Complejidad de la componente conexa

Lema 1: Para todo G computable y $v \in V$, $C(v) \leqslant_T 0'$.

Lema 2: [Folclore]
Existe un grafo computable G y $v \in V$, tal que $0' \leqslant_T C(v)$.

Teorema: $0'$ es necesario y suficiente para calcular $C(v)$ para todo G computable y $v \in V$.

Teorema de König

Defs

- 2^* es el conjunto de secuencias finitas de 0s y 1s.
- Un *árbol binario* es un subconjunto $T \subseteq 2^*$ tal que
$$\sigma \in T \ \& \ \tau \subseteq \sigma \Rightarrow \tau \in T.$$
- Un *camino* por T es $f : \mathbb{N} \to \{0, 1\}$ tal que $(\forall n)\ f \upharpoonright n \in T$
donde $f \upharpoonright n = (f(0), f(1), ..., f(n-1)) \in 2^*$.

Teorema de König: Todo árbol binario infinito tiene un camino.

P: ¿Si T es computable, hay algún camino computable?
P: ¿Hay algún oráculo que encuentre caminos para T computable?

Complejidad de los caminos

Lema 1:
Todo árbol binario, infinito y computable tiene un camino $\leqslant_T 0'$.

Lema 2:
Existe un árbol bin.inf.comp, que no tiene caminos computables.

Lema 3: [Jockusch, Soare]
Todo árbol bin.inf.comp, tiene un camino $<_T 0'$.
Hasta se puede encontrar un camino f tal que $f' \leqslant_T 0'$.

Lema 4:
Existe $A \subseteq \mathbb{N}$, $0 <_T A <_T 0'$, tal que $A' \equiv_T 0'$, y
que puede computar caminos en cualquier árbol bin.inf.comp.

Ideales primos y maximales

Sea $(R; 0, 1, +_R, \times_R)$ un anillo conmutativo infinito numerable.
Podemos suponer $R \subseteq \mathbb{N}$.

Def: R es *computable* si $R \subseteq \mathbb{N}$, $+_R \subseteq \mathbb{N}^3$ y $\times_R \subseteq \mathbb{N}^3$ son computables.

Def: Un *ideal* en R es un subconjunto $I \subseteq R$ tal que

- $x, y \in I$ implica $x +_R y \in I$,
- $x \in I$, $y \in R$ implica $x \times_R y \in I$.

Def: $I \subsetneq R$ es un *ideal maximal* si $\nexists$ ideal J, $I \subsetneq J \subsetneq R$.
Def: $I \subsetneq R$ es *primo* si $x \times y \in I$ implica que $(x \in I)$, o $(y \in I)$.

P: ¿Qué tan difícil es encontrar un ideal maximal o primo en un anillo computable?

Ideales maximales

[Friedman, Simpson, Smith]

Lema 1: Todo anillo computable tiene un ideal maximal $\leqslant_T 0'$.

Lema 2: Hay un anillo computable R tal que $0'$ es computable en cualquier ideal maximal de R.

Teorema: $0'$ es suficiente y necesario para calcular ideales maximales en anillos computables.

Ideales primos

[Friedman, Simpson, Smith]

Lema 1: Para todo anillo computable R, existe un árbol binario, infinito y computable cuyos caminos pueden calcular ideales primos en R.

Lema 2: Para todo árbol binario, infinito, computable T, existe un anillo computable cuyos ideales primos pueden calcular caminos en T.

Teorema: El problema de buscar ideales maximales en anillos es equivalente al problema de buscar caminos en árboles binarios infinitos.

König versus el problema de la parada

Teorema: Sea $A \subseteq \mathbb{N}$. Los siguientes son equivalentes.

(1) $0' \leqslant_T A$;
(2) A puede decidir si un grupo dado por (generadores, relaciones) es no-trivial;
(3) A puede decidir si un polinomio en $\mathbb{Z}[X_1, X_2, ...]$ tiene raíces en $\mathbb{Z}$.
(4) $C(v) \leqslant_T A$ para todo grafo computable G, $v \in G$;
(5) todo anillo computable tiene un ideal maximal $I \leqslant_T A$;
(6) toda secuencia creciente acotada y comp. en $\mathbb{Q}$ tiene limite $\leqslant_T A$;

Teorema: Sea $A \subseteq \mathbb{N}$. Los siguientes son equivalentes.

(1) todo árbolo bin.inf.comp. tiene un camino $\leqslant_T A$;
(2) todo anillo computable tiene un ideal primo $\leqslant_T A$;
(3) Si $\{(a_i, b_i) : i \in \mathbb{N}\}$ es una familia de intervalos con $a_i, b_i \in \mathbb{Q}$ tal que $\forall n, [0,1] \not\subseteq \bigcup_{i<n}(a_i, b_i)$,
existe un numero real $r \in [0,1], r \not\in \subseteq \bigcup_{i\in\mathbb{N}}(a_i, b_i)$ computable en A.