

# El problema del significado: el Teorema de Rice

---

La descripción formal de la noción de tarea efectiva parece reducir este concepto a una colección de manipulaciones de símbolos aparentemente desprovistas de significado. Sin embargo, toda rutina que un ser humano ejecuta de forma consciente –toda rutina encarnada en las acciones de un ser humano, diríamos siguiendo a Searle– posee un significado para la persona que la lleva a cabo. ¿Dónde, en qué momento, se ha perdido la conexión entre el significado o finalidad de una rutina y los símbolos primitivos en que la teoría de la computación descompone esa noción?

Parece que tenemos un problema en el modo en que la Teoría de la Computación se las ve con la noción de significado. Éste problema queda reflejado para Searle en el modo en que es posible mostrar al ocupante de su habitación china ejecutando una rutina cuyas acciones no es capaz de interpretar. O dicho de otro modo, cuyas acciones es incapaz de situar en un contexto suficientemente significativo. Es innegable que siempre nos podemos imaginar ejecutando *ciegamente* una serie de acciones cuya disposición nos parece por entero mecánica. Es más, se trata de una experiencia hasta cierto punto común. En muchas de estas ocasiones acabamos finalmente por entender qué es lo que estamos haciendo al captar el papel que cada acción particular desempeña en la consecución de un cierto fin. En ese momento ya no decimos que tal o cual acción particular carezca de significado, sino que posee o adquiere uno en función del papel al que acabo de aludir.

Esta imagen permite abordar el problema del significado desde un concepto de fundamental importancia en la explicación de la mente: el carácter intencional de todas

sus acciones. Pese a que podamos proceder de forma potencialmente indiscernible a como opera una máquina de Turing cuando actuamos de forma mecánica, poseemos la capacidad de atribuir una finalidad a nuestras acciones, o para ser más exactos, no parece que podamos eliminar fácilmente la impresión de que nuestras acciones poseen una intención, una finalidad. Al conectar el significado con la intencionalidad de nuestras acciones evitamos una lectura de la crítica de Searle que siempre me ha parecido errada.

La crítica a la que me refiero es aquella que insiste en la oposición entre forma y contenido. La descripción formal de una rutina sería precisamente formal por haberse desprovisto de toda traza de significado. De este modo, nunca podríamos atribuir significado a la máquina que describe una rutina salvo que lo hiciéramos explícitamente del mismo modo que un modelo de la lógica asigna un significado a las fórmulas de su lenguaje. Este símil no me parece correcto. Y no lo sería por razones estrictamente lógicas. No es cierto en absoluto que una máquina de Turing, por ejemplo, sea una entidad formal del mismo tipo que lo es una fórmula en un lenguaje lógico. Una fórmula de la Lógica de Primer Orden, por ejemplo, constituye una combinación de símbolos interpretados, las constantes lógicas, y símbolos carentes de significado hasta que un modelo les asigna uno. Una máquina de Turing carece de símbolos no interpretados. No hay nada que pueda ocupar el papel de un modelo en el caso de una máquina de Turing. Por tanto es incorrecto oponer forma y contenido del modo que en ocasiones se hace al pensar en el tipo de manipulación simbólica que conlleva la ejecución de un algoritmo representado por una máquina.

Cada uno de los símbolos que se emplean en una máquina posee un significado previamente asignado: desplazarse a la derecha, hacerlo a la izquierda, imprimir una marca, etc. No es cierto, por tanto, que la Teoría de la Computación, en su análisis de la noción de tarea efectiva, descienda a un nivel de descripción en el que forma y contenido aparezcan dissociados. Eso no llega a suceder nunca. Sí es cierto, si se quiere, que presenta un análisis atómico, en el sentido etimológico, del carácter efectivo de una tarea. La descripción de una tarea efectiva se supondría compuesta de elementos simples no divisibles en elementos menores. El problema

### El problema del significado: el Teorema de Rice

parece entonces explicar por qué ese nivel atómico de descripción permite imaginar coherentemente experimentos mentales del tipo de la habitación china. La única razón por la cual podemos seguir aún pensando en la existencia de un problema con la noción de significado tiene que ver, por tanto, con la aparición de intencionalidad a partir de un cierto punto. Existe un teorema fundamental dentro de la Teoría de la Computación sumamente esclarecedor a ese respecto y que, sin embargo, rara vez ha sido utilizado en este contexto. Me refiero al Teorema de Rice cuyo contenido, demostración y consecuencias veremos en breve.

Para establecer la conexión entre la discusión anterior y este resultado voy a plantear un pequeño problema de apariencia en principio trivial. ¿A qué tipo de entidad nos referimos cuando hablamos, por ejemplo, de la *suma*? Desde el punto de vista del álgebra abstracta, por ejemplo, parece claro que estaríamos hablando de una operación definida sobre la base del álgebra que tiene en el  $0$  su elemento neutro y que es conmutativa y asociativa. Sin embargo, parece que aquello que entendemos como la operación suma es algo que necesariamente queda subdeterminado por una descripción puramente algebraica. Podemos recurrir, como de hecho se hace, a la descripción de un *algoritmo de sumar*. Sumar sería, por tanto, aquello que hacemos cuando desarrollamos este algoritmo concreto sobre un par de enteros positivos. Si atendemos bien a lo que aquí se dice, pronto apreciamos que lo que hemos hecho al proceder de este modo, es ligar la suma a un algoritmo particular, de modo que un algoritmo distinto deberá dar lugar a una operación distinta. Es obvio, sin embargo, que tampoco actuamos así. No es determinante para la función particular que se compute el algoritmo concreto, siempre que escojamos uno que arroje siempre los mismos valores que otro dado como referencia. Al aceptar esta observación, por otra parte trivial, estaremos concediendo implícitamente que aquello a que denominamos función suma es, en realidad:

[1] *Definición computacional de una función numérica:*

- $f(x_1, x_2, \dots, x_n) = \{\varphi^i / \forall x_1, x_2, \dots, x_n \varphi^i(x_1, x_2, \dots, x_n) = \theta(x_1, x_2, \dots, x_n)\}$ , donde  $\theta(x_1, x_2, \dots, x_n)$  es una función numérica computable previamente fijada mediante su índice.

Para huir de una concepción excesivamente restrictiva de aquello en que consiste una función, la cual haría del algoritmo –máquina- el referente inmediato de cada una de ellas, nos vemos obligados a una sutil combinación de criterios constructivos y estipulaciones conjuntistas sobre cuyo carácter poco sabemos. Si el conjunto de funciones computables a que se alude en [1] fuese un conjunto que pudiésemos construir de manera efectiva, siempre sería posible escoger un representante canónico al cual atribuiríamos la representación ideal de la función analizada. Pero, ¿qué significa poder obtener ese conjunto de manera efectiva? Supongo que nada distinto de localizar un algoritmo que establezca la pertenencia de una función computable a esa clase, o lo que es lo mismo, un algoritmo que determine si un entero positivo es o no el índice de una función que coincida para n-tupla con otra tomada como testigo. Si tales conjuntos resultasen ser todos ellos conjuntos recursivos –ya sabemos lo que eso significa-, estaría bastante claro cuál es el referente de cada función numérica: se trataría de la clase de equivalencia a la que pertenece, clase que aparecería designada mediante un representante de nuestra elección. Me parece obvio que una situación de estas características permitiría determinar un nivel en el cual resolver, o al menos encauzar adecuadamente, el problema de la intencionalidad antes aludido. Es posible que mientras ejecuto los pasos de una cierta rutina, mi percepción puede estar oscurecida por una especie de momentánea pérdida de significado. Esta situación resultaría paliada en el momento en que establezco, o se me informa, de la pertenencia del algoritmo que ejecuto a su correspondiente clase de equivalencia. Ese sería en trasfondo en el cual resolver el problema de la intencionalidad. De ser así, el significado o intención de un algoritmo siempre podría ser fundamentado en los propios aspectos internos de la Teoría de la

### El problema del significado: el Teorema de Rice

Computación. Llegaríamos, en definitiva, a una *reducción* de la noción de significado, entendida como la necesidad de asignar al concepto intuitivo de función algo distinto de un algoritmo particular, a los propios recursos de la Teoría de la Computación. Este criterio se podría expresar mediante un principio:

[2] *Reducción computacional del significado*: El significado de una función numérica  $\theta$  vienen dado por un conjunto *recursivo* que definimos como:

$$\Theta = \{i/\varphi^i = \theta, \text{ para cada } n\text{-tupla de enteros positivos}\}.$$

¿Qué sucedería si para algún caso particular, o para todos, si se quiere, no se cumpliera esta reducción? Creo que en ese caso siempre habría lugar a afirmar que la atribución de significado a un algoritmo tendría que estar basada en procedimientos que no se sustentan plenamente en principios computacionales. Esta afirmación afecta al problema planteado al comienzo en la medida en que parece indicar la persistencia de una dificultad evidente a la hora de explicar la intencionalidad en términos puramente computacionales. Si es así, y en la medida en que parece imposible negar que la disposición del ser humano ante las tareas que ejecuta es netamente intencional, la crítica de Searle continuaría siendo uno de los puntos fuertes en que basar la resistencia al avance del mecanicismo.

Pero, ¿qué dictamina finalmente la Teoría de la Computación acerca de la hipótesis establecida en [2]? El Teorema de Rice nos pone ante la peor de todas las posibles soluciones: ningún conjunto de índices puede ser a la vez recursivo y responder a un criterio conjuntista de pertenencia. Veámoslo.

El Teorema de Rice -H. G. Rice 1953- suele ser considerado una consecuencia más o menos inmediata del Teorema de los parámetros aunque sin situarse por ello dentro de la categoría menor de corolario. En realidad, podemos considerar que el

Teorema de Rice se sirve del papel protagónico que el Teorema de los Parámetros atribuye al conjunto  $K$  para responder de una sola vez a la insolubilidad de una buena colección de problemas. El Teorema de Rice, así entendido, constituye un resultado de indudable interés al paliar en alguna medida la disparidad de criterios y técnicas utilizados hasta ahora para establecer la insolubilidad de muchos de los problemas que aquí se han planteado. Obsérvese que salvo la *reducibilidad* entre conjuntos  $A, B$  introducida en la sesión anterior, la cual reconoce la existencia de una cierta estructura en el grado de complejidad del problema de la decisión predicado de subconjuntos de  $\mathbb{N}$ , nada hay que indique la existencia de relaciones apreciables y sistematizables entre dos problemas que resulten finalmente insolubles.

Antes de abordar la demostración del teorema, introducimos un par de funciones útiles en lo que sigue:

[3] i.  $\varphi^i$  simboliza la *función vacía* cuya característica evidente es permanecer indefinida para cualquier valor de sus argumentos.

ii.  $\theta_K(x)=1$  si  $x \in K$ ,  
indefinida en otro caso.

[4] Teorema de Rice: Sea  $\mathcal{F}$  un conjunto cuyos miembros son una cierta colección de funciones computables del tipo  $\psi(x)$ . Sea  $I^{\mathcal{F}}$  por su parte el subconjunto de  $\mathbb{N}$  formado por los índices de las funciones en  $\mathcal{F}$ .  $I^{\mathcal{F}}$  es recursivo si  $\mathcal{F}$  es el conjunto vacío o bien es el conjunto de los índices de todas las funciones computables unarias  $\psi$ .

### El problema del significado: el Teorema de Rice

#### Esquema de demostración:

**Parte I:** Si  $I^3$  es el conjunto vacío o bien es el conjunto de todas las funciones computables unarias entonces su carácter recursivo se determina de forma trivial. Obsérvese que todo subconjunto finito de  $\mathbb{N}$  es recursivo mientras que, por otra parte, la recursividad de las funciones de cifrado y descifrado de máquinas garantiza la recursividad del conjunto de los índices todas las funciones del tipo indicado.

**Parte II:** Se trata de probar ahora que si  $I^3$  es recursivo entonces  $3$  es, o bien el conjunto vacío, o bien el conjunto de todas las funciones computables unarias. Parece evidente que esta parte dista mucho de ser trivial. Asumiremos que  $I^3$  es recursivo y mostraremos a continuación que si existe una función computable unaria  $\psi$  tal que  $\psi \notin 3$ , entonces  $3$  es el conjunto vacío.

Es obvio que esta última afirmación es equivalente a aquella otra que establece que si  $3$  no es el conjunto vacío entonces toda función  $\psi$  pertenece a  $3$ , con lo cual bastará ver probada la primera de nuestras afirmaciones para obtener el teorema al completo. Puesto que es independiente nuestra elección de  $3$  y cualquier afirmación que establezcamos acerca de este conjunto se puede traducir apropiadamente a su complemento, tomaremos  $3$  de tal modo que la función vacía  $\varphi^i$  no pertenezca a  $3$ . Con eso garantizamos que  $3$  no es el conjunto de todas las funciones computables unarias. Supongamos entonces que contrariamente a lo establecido previamente  $3 \neq i$ . Sea ahora  $\psi$  una función en  $3$ . A continuación conjeturamos la existencia de una función computable totalmente definida  $h(x)$  la cual utilizamos para localizar otra función computable  $\varphi^{h(x)}$  cuya conducta es la siguiente:

$\varphi^{h(x)}$  es:      la función  $\psi$  si  $x \in K$ ,  
                          $\varphi^i$  en otro caso.

De aquí y del hecho de que hayamos escogido  $\psi$  de modo que la función  $\psi \in \Sigma$  pero sucediendo al mismo tiempo que  $\varphi^i \notin \Sigma$  se sigue que:  $x \in K$  si y sólo si  $h(x) \in \Sigma$ , donde  $h(x)$  es una función computable totalmente definida.

Pero esto significa que  $K$  es 1-reducible a  $\Sigma$ ,  $K \leq_1 \Sigma$  en símbolos, de donde se sigue que si  $\Sigma$  es recursivo  $K$  también lo es. Esta contradicción permite concluir que  $\Sigma$  no es recursivo lo cual basta para establecer el enunciado del teorema.

Toda esta argumentación queda pendiente de que la función computable  $h(x)$  definida anteriormente exista realmente. Definimos la función binaria  $\varphi^e(x,y)$  en términos de:

$$\varphi^e(x,y) = \theta_K(x) \cdot \psi(y),$$

Es obvio ahora que si  $x \in K$ , la función  $\theta_K$  definida previamente se comporta de forma que  $\theta_K(x) = 1$ , de donde se sigue que si  $x \in K$  entonces  $\varphi^e(x,y) = \psi(y)$  para todo  $y$ . En otras palabras,  $\varphi^e$  es  $\psi$ . Si  $x \notin K$  entonces sucede que  $\theta_K(x)$  está indefinida para todo  $y$ , de donde  $\varphi^e$  resulta ser la función vacía  $\varphi^i$ . Puesto que estos resultados se obtienen tomando la variable  $x$  como parámetro, aplicamos el teorema del mismo nombre sobre  $\varphi^e(x,y)$  obteniendo:

$$\varphi^{s(e,x)}(y) = \varphi^e(x,y), \text{ para una función } s(e,x) \text{ computable totalmente definida.}$$

Bastará tomar ahora la función  $h(x)$  de tal modo que  $h(x) = s(e,x)$ , resultando de ahí que:

$$\varphi^{h(x)}(y) = \varphi^{s(e,x)}(y) = \varphi^e(x,y),$$



### El problema del significado: el Teorema de Rice

con lo cual  $h(x)$  existe y con ello queda probado el teorema.

Se trata de una demostración prolija cuyo interés consiste casi exclusivamente en apreciar el uso del Teorema de los Parámetros. Entiendo, no obstante, que se trata de detalles con un fuerte sabor matemático y no me entretendré en ello.

Este resultado afirma literalmente la imposibilidad de localizar un solo conjunto recursivo no vacío integrado por índices de funciones computables que sea estrictamente menor que el conjunto de todas las funciones computables. En otras palabras, nuestra capacidad para reconocer los índices de una colección de funciones identificadas a través de una cierta propiedad queda más allá de los límites de los procesos computables. Esta observación se traduce en un corolario que agrupa algunos de los resultados de indecidibilidad previamente demostrados y muchos otros del mismo estilo y entre ellos aquellos que más directamente afectan al tratamiento computacional del significado.

[5] Corolario: Ninguno de los conjuntos que a continuación se indican es recursivo (decidible):

- i.  $\{i/\varphi^i \text{ es total}\}$
- ii.  $\{i/\varphi^i = \varphi^j \text{ para algún } j \text{ fijado previamente}\}$
- iii.  $\{i/\varphi^i \text{ dom}(\varphi^i) \text{ es finito}\}$
- iv.  $\{i/\varphi^i \text{ } k \in \text{dom}(\varphi^i) \text{ para algún } k \text{ fijado previamente}\}$

El estudio del Teorema de Rice produce siempre una impresión que recuerda en algo la que provocan las paradojas o en general cualquier rompecabezas con soluciones inesperadas. ¿Cómo es posible que la herramienta -las funciones computables- que empleamos para generar subconjuntos propios no vacíos de  $\mathbb{N}$  garantizando su

recursividad no tenga aplicación cuando los elementos en uno de esos conjuntos son interpretados como índices de funciones computables? Aparentemente nada impide que dado un conjunto recursivo cualquiera procedamos a tomar sus elementos como índices de funciones numéricas violando de este modo el enunciado del teorema: lo que en realidad sucede entonces es que hay un cierto entero positivo que es el índice de una cierta función computable el cual tendría derecho a estar en ese conjunto pero que, sin embargo, no está.

El impacto del Teorema de Rice sobre la estructura de cualesquiera subconjuntos recursivos de  $\mathbb{N}$  es evidente a partir del comentario anterior. Una forma extraordinariamente directa de resumir la forma en que este resultado impide obtener un tratamiento satisfactorio del significado desde un punto de vista computacional sería el siguiente:

[6] Corolario: Sea  $\Gamma$  un subconjunto recursivo propio y no vacío de  $\mathbb{N}$ . Hay entonces sendos enteros positivos  $i$  y  $j$  tales que:

- i.  $i \in \Gamma, j \notin \Gamma$  y
- ii.  $\varphi^i = \varphi^j$ .

Si consideramos ahora un conjunto recursivo  $\Gamma$  como el invocado en los comentarios que preceden a este teorema encontramos que la interpretación de los elementos de dicho conjunto como índices de funciones computables da como resultado lo siguiente: para cualquier posible propiedad predicable de una colección de funciones computables existe una función con esa propiedad cuyo índice no está en  $\Gamma$  -siempre que  $\Gamma$  sea un subconjunto propio no vacío y recursivo de  $\mathbb{N}$ . Otra forma de interpretar este resultado consiste en afirmar que nunca podremos formar de manera efectiva -recursiva- clases de máquinas de Turing utilizando como único criterio la identidad de las funciones que éstas computan.

### *El problema del significado: el Teorema de Rice*

Es muy posible que todos estos resultados, consecuencias más o menos directas del problema de parada y del Teorema de los Parámetros, no afecten de manera determinante el poder explicativo de los modelos computacionales de la mente. Tal vez tampoco disminuyan los derechos de las tesis mecanicistas, pero tampoco cabe duda que lo complican todo un tanto. Porque, ¿de donde procede la capacidad nuestra capacidad para dotar de significado a tareas que, en realidad, no somos capaces de reconocer de manera efectiva mediante su descripción o algoritmo?

**Orientación bibliográfica.**

Este resultado está bastante bien caracterizado en **[Bridges, 1994]** secc.5. También tiene un tratamiento pormenorizado en **[Alonso, 1996]**. En **[Salomaa, 1994]** secc. 4.4 se hace una rápida demostración de la que no se extraen comentarios de interés.