SOBRE LA NATURALEZA DE LA TESIS DE CHURCH

Jaime Ramos Arenas
Universidad Nacional de Colombia

RESUMEN: La tesis de Church, según la cual una función es efectivamente computable si y sólo si es recursiva, ha recibido considerable atención por parte de filósofos y matemáticos. En particular es incierto cuál es exactamente la naturaleza de la tesis: ¿se trata efectivamente de una conjetura susceptible de ser eventualmente falseada o es una especie de definición de calculabilidad? Deberíamos interpretar la tesis en términos realistas o puramente convencionalistas? Yo sugiero que la interpretación convencionalista trivializa la tesis de Church y que es mucho más interesante entenderla como una verdadera conjetura, según la cual una función sólo es computable, en sentido absoluto, si es recursiva. Por otra parte sugiero que la tesis tiene cierta importancia para la ciencia cognitiva, pero mucho menor de lo que algunos han supuesto.

Mi propósito es hacer aquí una reflexión filosófica, es decir conceptual, sobre el carácter o la naturaleza de la tesis de Church (TC). Uno encuentra en la literatura sobre TC las más variadas aseveraciones que van desde decir que TC no es en verdad una tesis sino una mera definición convencional (Shanker) hasta la afirmación de que TC es una tesis empírica (Nelson). Yo no voy a tratar de resolver la cuestión definitivamente en este escrito, en parte porque como sugiero más adelante, considero que la interpretación que uno le dé a TC depende más o menos directamente de la posición ontológica que uno asuma frente a los objetos matemáticos, y esa es una cuestión muy extensa y difícil que está lejos de ser resuelta. Sin embargo pienso que algunas precisiones sobre TC pueden ayudarnos a entender al menos dónde está el meollo de la polémica en torno a ésta.

A lo largo de este escrito presentaré varias formulaciones de TC que se encuentran en la literatura, aunque existen muchas más, y examinaremos hasta qué punto son o no equivalentes. Empecemos por examinar la formulación de TC que el propio Alonzo Church ofrece al comienzo de su ensayo «*An Unsolvable Problem of Elementary Number Theory*» (1936, p. 90). Según ésta:

(1) una función es efectivamente calculable si es recursiva.

Esto equivale a decir, como señala Church, que una función es efectivamente calculable si es λ-definible.

Esta es una afirmación más bien débil: se afirma solamente que el carácter recursivo de una función es una condición suficiente para establecer su calculabilidad efectiva.

Dado que una función recursiva es aquella cuyos valores pueden ser calculados mediante un número finito de pasos a partir de valores ya conocidos de otras funciones, o más exactamente mediante la aplicación de un número finito de operaciones de composición (sustitución), recursión primitiva y minimización[1], parece claro que una función recursiva es algorítmica y por ende efectivamente calculable.

Lo que se conoce como TC es sin embargo una afirmación más fuerte que la anterior, la cual se encuentra en el mismo ensayo de Church, quien escribe:

> Ahora definimos la noción de función efectivamente calculable sobre enteros positivos, identificándola con la noción de función recursiva de enteros positivos. (1936, p. 100)

TC puede interpretarse ahora en el sentido de que:

(2) Una función es efectivamente calculable si y sólo si es recursiva.

Nótese también que Church describe aquí a TC como una definición, su propósito después de todo es el de dar una definición formal de la noción intuitiva de calculabilidad efectiva. La cuestión que queda pendiente es de qué tipo de definición se trata. Ciertamente esta no es una definición puramente nominal o estipulativa, el tipo de definición usada para definir un símbolo que se introduce en un sistema formal. La noción de calculabilidad efectiva ya existía y Church sólo quiere recoger esa noción intuitiva en una noción formal como la de recursividad.

Básicamente la noción intuitiva de calculabilidad efectiva es la siguiente: una función es efectivamente calculable si es posible construir un algoritmo para computar el valor de la función.

---

1 Composición: si f es una función de m argumentos y $g_1, ..., g_m$ son funciones de n argumentos, entonces h se obtiene de $f, g_1, ..., g_m$ según el siguiente esquema:

$$h(x_1, ..., x_n) = f(g_1(x_1, ... x_n), ..., g_m(x_1, ..., x_n))$$

Recursión primitiva: la función h se define recursivamente mediante las funciones f y g según el siguiente esquema:

$$h(x,0) = F(x)$$
$$h(x,s(y)) = g(x,y,h(x,y))$$

Minimización: h se obtiene de f mediante la operación de minimización según el siguiente esquema:

$$(h(x_1, ..., x_n) = \begin{cases} \text{El } y \text{ más pequeño para el cual } F(x_1, ..., x_n, y) = 0 \\ \text{Indefinida si no hay un } y \text{ para el cual } F(x_1, ..., x_n, y) = 0 \end{cases}$$

donde F es una función total de n+1 argumentos

TC y la noción de algoritmo

Un algoritmo es un conjunto finito de instrucciones que señalan una serie finita de operaciones para determinar el valor de la función dado un argumento apropiado cualquiera. El algoritmo debe estar descrito mediante un conjunto finito, aunque arbitrariamente grande, de símbolos o de lo contrario no constituye un procedimiento efectivo. No existe tampoco un límite a la capacidad de memoria que se requiere para la computación, mientras ésta sea finita. Esto significa que una función puede ser efectivamente computable aún si no fuese computable en la práctica, debido por ejemplo a que el argumento sea excesivamente grande (Rogers 1967). Se dice que un algoritmo constituye un procedimiento mecánico y efectivo para computar una función. El procedimiento es mecánico en el sentido de no requerir de ningún ingenio o destreza por parte del calculista aparte de comprender las precisas instrucciones dadas en el algoritmo. El procedimiento es efectivo porque lleva indefectiblemente a la determinación del valor de la función dado un argumento apropiado (es decir, el argumento debe estar dentro del dominio de la función).

La noción de algoritmo así descrita es sin embargo intuitiva y no formal, ya que involucra conceptos intuitivos como los de instrucción, procedimiento mecánico, etc. Es decir en esta descripción de algoritmo queda sin establecer exactamente qué puede considerarse como una instrucción, qué tanta comprensión de la instrucciones se requiere y demás.

Resumiendo, una función es efectivamente calculable si es algorítmica, pero dado que "algorítmico" es una noción vaga, Church propone en su lugar la noción formal de *recursividad* (o alternativamente de definibilidad lambda) porque considera que la noción de recursividad incorpora adecuadamente la noción intuitiva de calculabilidad. Según esto TC no puede entenderse como una definición meramente estipulativa de calculabilidad como parecen sostener los convencionalistas.

La interpretación convencionalista de TC

Según la interpretación convencionalista de TC, defendida por ejemplo por Shanker (1987) Church simplemente estipula qué debe entenderse por calculabilidad efectiva, y por tanto la tesis de Church no es ninguna clase de tesis o conjetura que pueda ser falsa o verdadera.

Shanker escribe por ejemplo:

> En verdad, el esquema de Church nos fuerza a concluir, no que es improbable, sino lógicamente imposible descubrir una función sobre los enteros positivos para la cual exista un algoritmo pero que no sea recursiva; cualquier sugerencia en contrario debe, en estos términos, ser considerada

> no como falsa sino como asignificativa... El argumento de Church puede ser mejor entendido en términos convencionalistas donde la inferencia de 'es efectivamente calculable' a 'es recursiva' es estipulada... aún hablar de la verdad de TC (sin prestar atención a la base normativa en que el termino debería ser entendido) es potencialmente desorientador, en la medida que invita a la confusión de que TC es una hipótesis. (p. 624)

La interpretación de Shanker es sin embargo heterodoxa y a mi parecer infundada. No es cierto que Church simplemente estipule que una función es efectivamente calculable sólo si es recursiva. Tal interpretación trivializa totalmente la tesis de Church. Pienso que Shanker tampoco está en lo correcto al afirmar que:

> Si por hipótesis no había una noción matemática pre-existente de función efectivamente calculable entonces no había nada con lo que la nueva regla de Church, y la totalidad de las funciones así definidas, pudiera entrar en conflicto» (p. 625)

Es cierto que Church habla de TC como una definición de calculabilidad efectiva, pero como el mismo Church señala (1936, p. 100), es una definición formal que trata de recoger la noción intuitiva de calculabilidad. Las nociones matemáticas de algoritmo y cálculo son muy antiguas y el trabajo que Church, Turing, Gödel y otros realizaron en los años treintas sólo buscaba precisar y formalizar tales nociones. Church de hecho se sintió compelido ha ofrecer una argumentación en favor de TC, y Turing, quien desarrolló independientemente otra versión de TC como veremos más adelante, también se sintió obligado a ofrecer alguna evidencia en su favor. Por otra parte la aseveración de Shanker de que sería lógicamente imposible encontrar una función que constituya un contraejemplo a TC se funda únicamente en su interpretación puramente convencionalista y es por tanto infundada. Si la recursividad se toma como «criterio», en el sentido wittgensteiniano, de calculabilidad efectiva, entonces si se encontrara un supuesto contraejemplo a TC, una función no-recursiva pero efectivamente calculable, tendríamos que decir que la función en realidad no es efectivamente calculable, algo que Shanker concede (1987, p. 624) pero que resulta injustificado.

LA INTERPRETACIÓN ESTRUCTURALISTA DE TC

Quizás sea más acertado interpretar TC, siguiendo a Stewart Shapiro, como una definición real de calculabilidad, en el sentido aristotélico de definición real. Es decir que si TC es adecuada la noción de recursividad devela la esencia del concepto de computabilidad. De acuerdo con Shapiro (1981) la calculabilidad sería una propiedad matemática de las funciones, que aunque pre-formal, tendría una extensión fija y determinada, la cual puede o no ser recogida por una definición formal. Shapiro inscribe su interpretación de TC dentro de una corriente estructuralista en la filosofía de las matemáticas. De acuerdo con esta corriente, los objetos del universo no-matemático presentan estructuras matemáticas subyacentes en sus interrelaciones. Esta postura supone una especie de platonismo según la cual los objetos y propiedades matemáticas tendrían un carácter objetivo independiente de nuestra actividad matemática.

La cuestión de la realidad objetiva e independiente de los entes matemáticos, claramente rebasa los límites de este trabajo. Lo que queda en claro es que la interpretación de la tesis de Church a que uno llegue depende en últimas de una posición filosófica acerca de la ontología del universo matemático. Es decir, si uno es un realista, esto es platónico respecto a los entes matemáticos, entonces seguramente conciba la calculabilidad como una propiedad inherente a ciertas funciones, en cuyo caso es o bien verdadero o bien falso que el conjunto de funciones calculables sea coextensivo con el conjunto de funciones recursivas. Según esto, TC tendría un valor de verdad, lo que la haría algo mucho más cercano a una tesis o conjetura de lo que los convencionalistas estarían dispuestos a aceptar. Quizás describir una definición como verdadera o falsa pueda resultar un tanto extraño para algunos, pero esto sucede para los que entienden las definiciones como convenciones, pero si existen definiciones reales entonces puede decirse que una definición es verdadera si captura la esencia del concepto definido y falsa en caso contrario. Vale la pena advertir que el esencialismo es una doctrina que sufrió un duro golpe a manos de Hume, Wittgenstein y varios otros filósofos y es dudoso que pueda ser salvada. En cualquier caso veamos la formulación de TC que sugiere Shapiro:

(3) los conjuntos de funciones calculables y funciones recursivas son necesariamente coextensivos

Entendida así, TC se convierte prácticamente en una tesis con un valor de verdad, el cual es además desconocido puesto que no existe, ni puede existir, una prueba de la verdad de TC, ni un contraejemplo aceptado que permita establecer su falsedad.

La interpretación estructuralista, o platónica, de TC es atractiva y sobre todo hace de TC una tesis importante y de gran contenido, pues siguiendo esta interpretación, si TC es verdadera entonces una función no-recursiva no es efectivamente calculable en un sentido absoluto, es decir la función no es efectivamente calculable no sólo por nosotros sino por cualquier ser calculante. El carácter no algorítmico de la función residiría en la función misma, no en las limitaciones de nuestros algoritmos. Aquí vale la pena hacer una aclaración: decir que una función no es efectivamente calculable quiere decir que no es posible construir un procedimiento general y uniforme para calcular el valor de la función para un argumento cualquiera que esté dentro del dominio de la misma, pero no quiere decir que para un argumento específico dado no sea posible calcular el valor de función. Así por ejemplo un matemático hábil puede ser capaz de determinar si una máquina de Turing se detiene o no dado cierto dato de entrada en la cinta de la maquina, pero no es posible construir un algoritmo universal para determinar si cualquier máquina de Turing dada se detendrá o no dado un dato de entrada cualquiera, por lo cual la función de la parada no es efectivamente calculable.

¿ES TC UNA TESIS PURAMENTE MATEMÁTICA?

Hay otro aspecto de la tesis de Church que quiero mencionar y es el de si TC tiene un carácter puramente matemático, y por ende su verdad o corrección puede ser establecida por medios puramente matemáticos, o si por el contrario involucra conceptos no matemáticos, por lo cual no podría ser resuelta por una prueba matemática, como presumiblemente sería el caso, por ejemplo, con la conjetura de Goldbach.

En la forma como Church formula TC parecería que la cuestión es eminentemente matemática, pues la calculabilidad y la recursividad son propiedades de funciones de números. Sin embargo como Nelson (1987) y Shapiro han sugerido, TC puede leerse como una afirmación acerca de computadores y funciones.

Shapiro señala que:

> El estructuralista supone o bien que existe una estructura matemática común a todos los aparatos mecánicos de computación o que todas las mentes tienen una estructura matemática común en virtud de la cual tienen el potencial de comprender y ejecutar algoritmos, y por ende, la habilidad de computar (...) La computabilidad es una propiedad que las funciones tienen en relación con una o ambas de estas estructuras. (1981, p. 355)

La recursividad es claramente una propiedad matemática de ciertas funciones, y puesto que, según TC, la computabilidad es reducible a la recursividad entonces la computabilidad es también una propiedad matemática. Pero por otra parte computar una función es un proceso o actividad llevado a cabo por seres racionales y, según algunos, por máquinas (se puede pensar, sin embargo, que las máquinas no computan porque no siguen algoritmos, *nosotros* computamos usando las máquinas) por lo que la computabilidad puede verse como una propiedad de las funciones relativa a la capacidad de un ente calculador. Según esto habría una diferencia entonces entre la computabilidad y lo que podríamos llamar propiedades matemáticas puras, por ejemplo la propiedad de ser par en el caso de los números o la propiedad de ser total o parcial en el caso de las funciones. Aparentemente estas no son propiedades que los objetos matemáticos tengan con relación a un calculista como sí lo es la computabilidad (esta es una distinción que al menos el platónico debería hacer).

LA VERSIÓN DE TURING DE TC

La versión de TC formulada independientemente por Turing es más conducente a la anterior interpretación de computabilidad.

Como es bien sabido, casi al mismo tiempo que Church, Turing (1937) propuso una definición equivalente de calculabilidad, o computabilidad como la llama Turing, según la cual:

(4) una función es computable si y sólo si es Turing-computable.

Una función es Turing-computable si es posible construír una máquina de Turing tal que dado el argumento de la función como dato de entrada a la máquina ésta produce el valor de la función como dato de salida.

Dado que está probado que todas las funciones recursivas son Turing-computables, y viceversa, la definición de computabilidad de Turing se considera extensionalmente equivalente a la de Church, por lo que en ocasiones se habla de la tesis Church-Turing. Las dos definiciones de calculabilidad no son sin embargo intensionalmente equivalentes dado que recursividad y Turing-computabilidad no son términos sinónimos. Al contrario fue algo así como una feliz sorpresa el que tipos de algoritmos distintos como las máquinas de Turing, las definiciones recursivas, los algoritmos de Markov y el cálculo lambda resultaran todos equivalentes, en el sentido de que si una función es computable siguiendo uno de estos tipos de algoritmos entonces es computable por cualquiera de los otros, y si no es computable por alguno de estos tipos de algoritmos tampoco lo es por ninguno de los otros. En el caso más prominente Church y Turing establecieron independientemente, siguiendo distintos tipos de algoritmos, que el famoso problema de la decisión de Hilbert (el *Entscheidungsproblem*) es insoluble. Hilbert caracterizó el problema de la decisión como la pregunta de si es posible construir un procedimiento general de decisión que permita determinar si una fórmula cualquiera dada del cálculo de funciones de primer orden es probable o no (es un teorema del cálculo o no). Una manera de expresar esta insolubilidad en términos de funciones es establecer que la siguiente función no es efectivamente computable:

$$F(n) = \begin{cases} \textit{1 si n es probable,} \\ \textit{0 si n es no es probable} \end{cases}$$

(Donde *n* es el número gödeliano de una fórmula del cálculo de Hilbert-Ackermann. Probable se usa aquí en el sentido de ser susceptible de prueba).

Church y Turing ofrecen por tanto definiciones formales de la noción intuitiva de computabilidad (o calculabilidad), la cuestión es entonces establecer si la tesis de Church y la tesis de Turing son, digámoslo así, epistemológicamente equivalentes.

Algunos autores, por ejemplo Nelson (1987), han sostenido que TC es una tesis empírica. En principio parece un tanto extraño que un criterio de caracterización de las funciones matemáticas en computables y no computables pudiera en algún sentido ser empírica. Sin embargo hay aspectos de TC que pueden llevar a pensar que se trata efectivamente de una tesis empírica.

En primer lugar debe tenerse en cuenta que que la verdad de TC no puede establecerse mediante una prueba meramente matemática. No puede probarse TC ni ahora ni en el futuro, porque es siempre posible inventar nuevos tipos de algoritmos para computar funciones, los cuales, si TC es verdadera, deberán resultar equivalentes a los que tenemos hoy. TC está fundada en argumentos que pueden considerarse empíricos, los más importantes de los cuales son los siguientes:

(a) Un argumento que se ha presentado en favor de TC es que todas las funciones efectivamente computables descubiertas hasta ahora son recursivas, y viceversa. Church mismo(1936) hace mención a esta evidencia inductiva, la cual se usa frecuentemente para fundamentar hipótesis empíricas de las ciencias naturales. Sin embargo, también en matemáticas se ha apelado a argumentos inductivos en el caso de conjeturas para las cuales no hay una prueba definitiva. Por ejemplo en los casos del último teorema de Fermat y la conjetura de Goldbach se han utilizado computadores para verificar si en el caso de números muy grandes las conjeturas se siguen cumpliendo. Pero nadie diría que por este solo hecho el teorema de Fermat y la conjetura de Goldbach sean empíricos.

(b) Otro argumento importante en favor de TC es el de la invariancia. En este siglo se han construido varias formas de caracterizar funciones computables todas las cuales han resultado ser equivalentes, es decir, que si una función es computable según una de ellas también lo es según todas las demás. Entre éstas se encuentran el cálculo lambda, las funciones recursivas, las máquinas de Turing y los algoritmos de Markov. Este hecho sugiere que esos algoritmos han recogido apropiadamente la noción intuitiva de computabilidad. En otras palabras, que estas definiciones formales de computabilidad identifican una "clase natural" de funciones, suponiendo que exista tal cosa. Debe resultar claro que este es también un argumento empírico.

(c) Otro argumento en favor de TC es el presentado por Turing, el cual apela a nuestra intuición de lo que es calcular. Turing (1937) señala que los calculistas tienen ciertos rasgos característicos que determinan qué funciones pueden ser computadas:

1) El cómputo debe rerquerir solamente de un número finito de símbolos porque la mente humana no puede manejar una infinidad de ellos
2) En cada paso del proceso de cálculo el calculista está en un cierto "estado mental", y el paso subsiguiente está determinado por el símbolo que estrá considerando en ese momento y el estado mental en que se encuentrre.
3) El número de estados mentales distintos de un calculista es finito
4) Todas las operaciones matemáticas pueden descomponerse en un número finito de operaciones simples (mecánicas).

De todo lo anterior se sigue que el proceso de cálculo es un proceso discreto. Si esta caracterización de la forma como la mente humana opera es correcta, entonces puede mostrarse que todas las funciones computables por un ser humano son también computables por una máquina de Turing. Este argumento está fundado en una presuposición psicológica, y por ende empírica, de la forma como opera la mente humana. Nótese que si Turing está en lo correcto respecto a la forma como la mente calcula, entonces TC debe ser verdadera. Sin embargo, la inversa no es válida: TC puede ser verdadera y la hipótesis mecanicista de Turing de una mente que opera discretamente como una máquina, puede ser falsa.

Es en todo caso significativo el hecho de que los tres argumentos más importantes que se han ofrecido en favor de TC sean todos empíricos, pero esto por sí solo no establece que la tesis lo sea. Creo que la cuestión sigue en todo caso abierta. Lo que sí es claro es que si TC es empírica, entonces la computabilidad no sería una propiedad puramente matemática de las funciones, sino una propiedad que tienen las funciones con respecto a ciertos entes físicos, como por ejemplo nosotros mismos.

Una variante de TC inspirada en Turing es la siguiente:

(5) Una función es computable algorítmicamente por un sujeto si y sólo si es computable por una máquina de Turing

Esta formulación es lógicamente equivalente (tiene necesariamente el mismo valor de verdad) que la formulación de Church (una función es efectivamente calculable si y sólo si es recursiva). Mejor dicho se trata de una formulación distinta de la misma tesis no de una tesis distinta. Sin embargo esta formulación sugiere más claramente que TC se refiere a la capacidad computacional de los sujetos.

Es esta lectura de TC la que inspiró al propio Turing y a muchos otros, especialmente en el campo de la ciencia cognitiva, a adoptar un filosofía mecanicista de la mente. Se argumenta que dado que todo procedimiento algorítmico es mecanizable, y se asume que el pensar es un procedimiento fundamentalmente algorítmico, entonces es posible en principio construir una máquina que piense. Algunos consideran que TC constituye la condición de posibilidad del paradigma computacional de la ciencia cognitiva. Nelson (1987) llega a sostener que:

> la tesis es una proposición empírica de la ciencia cognitiva, que está abierta a confirmación, corrección o abandono, y que según la evidencia presente parece ser verdadera (p. 581).

Esta aseveración parece exagerada, el hecho de que una tesis o un resultado matemático tenga incidencia sobre otras ciencias no la hace por sí misma empírica. Por otra parte la formulación de TC no se fundó en investigaciones empíricas o psicológicas, al menos no en el caso de Church, sino en el problema lógico planteado por Hilbert de si era posible en principio resolver todos los problemas matemáticos de una manera axiomática. Aun en el caso de Turing sus resultados matemáticos parecen haberlo conducido a una filosofía mecanicista y no a la inversa.

En cualquier caso es claro que TC por sí sola no puede constituir un pilar sobre el cual fundar una filosofía mecanicista o un modelo computacional de la mente. Aunque Turing estableció efectivamente que todo procedimiento algorítmico es mecanizable, esto es irrelevante para la ciencia cognitiva a menos que se pueda establecer independientemente que los procesos cognoscitivos son esencialmente algorítmicos, algo que se considera hoy mucho más dudoso que hace digamos treinta años, después de los estruendosos fracasos por encontrar algoritmos universales, por ejemplo para la resolución de problemas. Pero para que TC pudiera convertirse en una piedra fundacional de la inteligencia artificial ni siquiera bastaría establecer que los proceso cognoscitivos, o al menos un buen número de ellos, son en verdad algorítmicos, sino que se requeriría mostrar que ese carácter algorítmico o formal constituye la característica fundamental del pensar racional. El modelo formalista de los procesos mentales deja completamente inexplicado el carácter intencional de los estados mentales, como el propio Fodor ha señalado convincentemente.

Por otra parte es exagerado sostener que TC es absolutamente irrelevante para la ciencia cognitiva. Los modelos computacionales de la mente presuponen la verdad de TC, y si TC resultara falsa y el conjunto de funciones humanamente computables fuera mayor que el conjunto de funciones Turing-computables entonces nuestro poder deductivo sería mayor que el de cualquier máquina posible. De hecho, algunos autores han creído que el teorema de incompletitud de Gödel muestra precisamente esto, que para cada sistema formal hay una fórmula que podemos saber que es verdadera, y que no puede ser probada en el sistema mismo. Sin embargo, es posible que haya una fórmula que sea indecidible para cada uno de nosotros y sin embargo sea deducible en algún sistema formal. Dicho simplemente, la verdad de TC puede verse como una condición necesaria pero no suficiente para la posibilidad de los modelos computacionales de la mente.

Bibliografía

Church, Alonzo (1936): "An Unsolvable Problem of Elementary Number Theory". En Davis (ed.) (1965)

Davis, Martin (ed.) (1965): *The Undecidable: Basic Paper on Undecidable Propositions, Unsolvable Problems and Computable Functions*. New York, Raven Press.

Nelson, R.J. (1987): "Church's Thesis and Cognitive Science". *Notre Dame Journal of Formal Logic*, Vol. 28 No. 4, pp. 581-609

Rogers, Hartley (1967): *Theory of Recursive Functions and Effective Computability*. New York, Mac Graw-Hill

Schanker, S.G. (1987): "Wittgenstein vs. Turing on the Nature of Church's Thesis". *Notre Dame Journal of Formal Logic*, Vol. 28 No. 4, pp. 615-649

Shapiro, Stewart (1981): "Understanding Church's Thesis". *Journal of Philosophical Logic*. Vol. 10, pp. 353-365

Turing, A.M. (1937): "On Computable Numbers, with an Application to the Entscheidungsproblem", en Davis (ed.) (1965)