

Otros modelos: normalización

La descripción del modelo de la efectividad que representan las funciones incluidas en [Frp] no era, como ya he aclarado, el objetivo principal de Gödel. A lo sumo se trata de un medio destinado a un fin mucho más específico: el análisis de la derivabilidad de la consistencia absoluta de la Teoría elemental de números. No obstante, tal vez sea inapropiado considerar un medio a algo que realmente es un *canon*. La clase de las funciones recursivas primitivas permitió fijar una interpretación razonable de la efectividad de un procedimiento útil, por ejemplo, para entender en qué sentido una demostración formal debía ser efectiva. Aplicado ese criterio a la demostrabilidad de una formalización suficientemente potente de la Teoría de números, el resultado fue, y pudo ser, el ya conocido. Desde esta interpretación del problema es fácil ver el inmediato impacto que el trabajo de Gödel tuvo sobre el programa Formalista. El aparato inferencial que aporta la Lógica a una doctrina matemática concreta no conduce a la demostración efectiva de todos sus teoremas. Al menos, si *efectivo* se interpreta en el sentido de [Frp] y la teoría en cuestión contiene suficiente potencia expresiva como para generar esa clase de funciones en su interior.

El uso que Gödel hace de [Frp] le permite soslayar abiertamente el problema de si dicha clase contiene realmente todas las funciones numéricas que son efectivamente calculables. Este es el problema que nos va a ocupar, precisamente, a lo largo de este capítulo. Pero veamos primero cómo funciona el formalismo de las funciones recursivas primitivas un poco más de cerca.

En ocasiones no resulta fácil imaginar de qué modo unas funciones con tan poco contenido matemático como las funciones iniciales –cfr. [12] cap.2.1– pueden dar lugar a la definición de otras tan características de la aritmética como la suma o el

producto. Y esto por mencionar tan sólo casos muy básicos. Quizá viendo sus definiciones podamos hacernos idea de lo que hacen realmente estas funciones.

[1] Funciones recursivas primitivas (suma y producto):

i. suma:

$$\text{sum}(0,y)=p_1^1(y)$$

$$\text{sum}(x',y)=s(p_2^3(x,\text{sum}(x,y),y))$$

ii. producto:

$$\text{prod}(0,y)=0$$

$$\text{prod}(x',y)=\text{sum}(\text{prod}(x,y),y)$$

La idea de efectividad que se expresa en la clase de las funciones recursivas primitivas es muy elemental. Para calcular los sucesivos valores de una función, basta con mostrar de manera efectiva cómo obtener ese valor a partir de los precedentes fijando el valor para $x=0$ de manera directa. La variable sobre la que opera la recursión, x en este caso, recibe el nombre de variable de recurrencia y sólo se admite una de estas variables en cada función definida por recursión primitiva. A partir de la definición para [Frp] resulta absolutamente inmediato observar lo siguiente:

[2] Teorema: Cada función recursiva primitiva está definida de manera efectiva para cada posible valor de sus argumentos.

Esquema de la demostración. Basta observar que las funciones iniciales establecen de manera efectiva el valor que arrojan para todos sus posibles argumentos. Y que las reglas de composición y recursión preservan ese comportamiento si las funciones sobre las que operan lo hacen a su vez.

Otros modelos. Normalización

Las funciones que arrojan valor para todos y cada uno de sus argumentos reciben en ocasiones el nombre de *funciones totales* o *funciones totalmente definidas*. Las funciones recursivas primitivas están formadas sólo por funciones totales.

¿Puede considerarse que la clase de funciones numéricas definida por [Frp] contiene todas las funciones numéricas que es posible definir mediante un algoritmo, es decir, de manera efectiva? Gödel conoce la existencia de funciones que poseen una definición efectiva inobjetable y que, sin embargo, no tienen definición posible dentro del formalismo de [Frp]. Ackermann establece en 1928 una función -conocida como función de Ackermann- la cual, aunque calculable de manera efectiva, no lo es mediante usos, digamos simples, del método de recursión. El problema surge porque la función de Ackermann sólo es definible por recursión doble, esto es, mediante una aplicación de la regla de recursión sobre dos variables de recurrencia. Esta opción está expresamente excluida, como ya he dicho, en la definición de [Frp] de Gödel.

Con la definición de esta clase, Gödel se introduce en el análisis interno del concepto de efectividad. ¿Qué operaciones pueden considerarse como primitivas a la hora de generar otras más complejas aunque igualmente efectivas? El hecho de que Gödel no considere en esa ocasión todas las funciones que son objetivamente consideradas como efectivamente calculables, sino tan sólo una clase formada exclusivamente por funciones que sí lo son, reviste una importancia manifiesta. La noción de tarea efectiva, aunque elemental, desde el punto de vista de nuestras capacidades cognitivas, no parece ser simple.

Casi al mismo tiempo que Gödel demuestra sus famosos resultados de limitación, en Princeton se concentra un grupo de investigadores dirigidos por A. Church. Church, que es por entonces un joven profesor interesado por la Lógica, trabaja en el diseño de un formalismo capaz de definir funciones numéricas mediante lo que es una expresión directa del modo de calcular sus valores. El sistema resultante, el cálculo- λ , es expuesto por primera vez en 1933 y, tras dos años de un intensa práctica, provoca en quienes lo conocen la impresión de que sus recursos son suficientes como para permitir expresar toda función numérica que resulte

efectivamente calculable. S. Kleene y B. Rosser, participan por entonces como estudiantes de Church en esta intensa aventura resolviendo, muchas veces como trabajos de tipo escolar, la reducción de multitud de funciones numéricas al formalismo del cálculo- λ . Parece ser que en este proceso de rodaje tuvo especial importancia – por razones que no podemos discutir ahora- la expresión de la *función antecesor* en términos del formalismo de ese cálculo. Por no mencionar el problema identificado y resuelto por Kleene y Rosser que hacía del cálculo- λ un sistema inconsistente.

Es este grupo el que concibe por primera vez la posibilidad de contar con un formalismo capaz de expresar todas las funciones numéricas definibles sobre los números naturales. Es preciso llamar la atención sobre el hecho de que la idea que subyace al cálculo- λ difiere sustancialmente de aquella que vemos recogida en la clase de las funciones recursivas primitivas: la efectividad se explica en esta ocasión como un sistema simbólico que procede a combinar signos sin un contenido matemático inmediato de forma reiterada. Es difícil determinar si con ello estamos o no ante una explicación de la efectividad realmente distinta, o si sólo se trata de una modificación más o menos aparente. En cualquier caso, y a primera vista, la idea que nos podemos formar de una tarea efectiva cuando se dice que tiene expresión en el cálculo- λ , y cuando se dice que es expresable mediante una definición recursiva no es, desde luego, la misma.

Parece, por tanto, que el estudio de la noción de tarea efectiva no viene acompañado, precisamente, de un consenso inicial. La insistencia ante la posibilidad de hallar un formalismo que capture exactamente todas las funciones –tareas- que pueden ser calculadas de manera efectiva resulta meritoria porque demuestra una fe en la potencia del análisis formal que no corre pareja al curso de los acontecimientos.

¿Qué situaciones son esperables cuando de entrada se cuenta con varias descripciones formales del mismo concepto intuitivo o informal? Evidentemente la primera opción es que todos estos modelos coincidan. La segunda, que no lo hagan. En caso de que no coincidan, podemos encontrar que las distintas descripciones

Otros modelos. Normalización

formales arrojan intersecciones no vacías que difieren en algunos detalles o, por el contrario, podemos verlas formando una jerarquía en la cual se va aumentando el rango de recursos sin alcanzar un máximo que pueda ser considerado como descripción óptima.

Estas opciones se limitan a describir aquello que podría haber pasado y ante lo cual hay que estar preparado cuando se trata de crear o definir una clase de objetos mediante el uso de nuestro ingenio matemático o formal. En los años en que Church, Kleene, Turing y algunos otros se encuentran inmersos en el intento de comprender el comportamiento de la efectividad se observa un predominio del segundo tipo de fenómeno. Es decir, de aquel por el cual el tratamiento formal de un objeto de nuestra intuición deja de alcanzar el estatus de una auténtica descripción del mismo. Por una parte, ciertos resultados de limitación –especialmente los de Gödel- parecen confirmar el carácter esquivo de ciertos dominios de nuestro ingenio matemático y formal. Habría ciertos conceptos intuitivos, incluso perfectamente claros en ese nivel, que nunca podrían ser capturados consistentemente en todos sus detalles por medio de una descripción formal rigurosa. Por otra parte, la aparición de resultados de compatibilidad, de consistencia mutua diríamos, en ciertos dominios invita a pensar en el carácter creativo que tienen las teorías formales. El caso paradigmático es el de la geometría. Tras considerarse durante siglos que la geometría euclídea era el prototipo de descripción rigurosa –axiomática- de nuestra intuición espacial, bastaron unos pocos años para proponer modelos alternativos consistentes diferentes de aquella: las llamadas geometrías no-euclídeas. ¿Cuál era la descripción correcta? ¿Cuál de todas ellas describía la geometría real y cuáles de ellas era pura invención de la mente? Esta es, obviamente, una pregunta carente de sentido.

Es llamativo que en un contexto en el que proliferan ejemplos en los que la descripción formal de un objeto parece subdeterminar la realidad a la que aparentemente se refiere, Church y Kleene crean poder lograr una descripción óptima –completa- de concepto tan esquivo como el de tarea efectiva. Parece ser que, a parte del éxito mostrado por el cálculo- λ en la reducción de funciones efectivamente

calculables a su propio formalismo, una de las razones que vino a alimentar esa especie de fe pudo ser la inmunidad demostrada por el cálculo- λ a la técnica de diagonalización. De esta técnica ya he hablado al presentar el modelo clásico de la computación. La posibilidad de aplicar este argumento bajo unas condiciones realmente generales hacía difícil imaginar la existencia de dominios que pudieran resistirse a su efecto. Aplicado sobre una colección L de objetos, este procedimiento es capaz de demostrar la existencia de otro objeto digno de hallarse en ella y que, sin embargo, no se encuentra inicialmente incluido. El efecto de la técnica de diagonalización viene a ser entonces el de una demostración de incompletitud sobre la descripción de la clase L . En el caso que nos afecta, el éxito de una prueba de diagonalización nos habría llevado a concluir la existencia de funciones efectivamente calculables que no son definibles en el formalismo del cálculo- λ . Las razones por las que Kleene y Church opinan que este formalismo no diagonaliza son complejas y, en cierto modo, han quedado superadas por otros elementos considerados de mayor peso. No obstante, es un hecho que su argumentación entorno a este punto tuvo una gran importancia a la hora de mantener su fe en momentos críticos.

¿Qué elementos son los que permitieron que prosperase la idea de que era posible agrupar bajo una única definición formal toda la clase de funciones numéricas –y con ello de tareas de cualquier tipo– que resultan calculables de manera efectiva? Fue precisamente la relativa variedad de métodos que llegaron a converger en un cierto momento lo que finalmente prestó apoyo a esa posición inicialmente tan poco plausible. La existencia de una variedad de descripciones formales de factura realmente distinta que vienen a describir –demostrablemente– la misma clase de objetos –funciones numéricas– prestó un apoyo a esa tesis infundiendo unas dosis de optimismo sólo matizadas por algunas limitaciones inherentes a estos tratamientos formales. Los formalismos que participaron en este movimiento de convergencia son los siguientes:

Otros modelos. Normalización

[3] *Modelos históricos (hasta 1936):*

- i. Cálculo- λ .
- ii. Funciones recursivas generales.
- iii. Funciones recursivas por minimalización.
- iv. Máquinas de Turing.

Junto a estos aún es posible añadir una lista más o menos larga de modelos posteriores que han tenido algún impacto en la literatura. Estos últimos no llegaron a participar realmente en la constitución del modelo clásico de la computación, al menos en un nivel teórico.

[4] *Modelos posteriores (a partir de 1936):*

- i. Algoritmos de Markov
- ii. Sistemas de Post.
- iii. Ábacos y Máquinas de Registros.
- iv. Máquinas de acceso aleatorio.
-

Se trata de una lista abierta en la que las diferencias entre variantes notacionales y modelos distintos es difícil de establecer. Máxime cuando su presencia en la lista garantiza su identidad extensional.

La coincidencia de los modelos listados en [3] adquiere un definitivo valor formal y epistemológico cuando el último de sus componentes, las Máquinas de Turing, vienen a aportar en componente heurístico hasta entonces ausente en las demás propuestas. De los modelos que aquí denomino históricos sólo he desarrollado en profundidad uno de ellos que, además, da la casualidad de haber sido el último en el tiempo. El primero es, sin duda, el cálculo- λ . Este es, con diferencia, el que ofrece

un tratamiento más abstracto de la noción de tarea efectiva, al punto de ser difícil caracterizarlo siquiera de forma muy general. Siempre se ha dicho que el móvil del cálculo- λ es el de comprimir en la expresión literal de una función numérica una descripción del algoritmo que permite calcular sus valores, y seguramente sea así. Pero también podría decirse, y no sería menos correcto, que este formalismo intenta desarrollar el paso que lleva de un proceso de iteración simbólica dentro de una expresión –mediante el uso del llamado *abstractor* λ - a la lectura del valor numérico de una función. La dificultad de manejo de este formalismo llevó a que su propio autor, Church, y Kleene, alumno suyo y buen conocedor del mismo, lo abandonasen por el mucho menos farragoso procedimiento descrito por Gödel conocido bajo el nombre de recursividad general.

La Clase de las funciones recursivas generales [FRG] tienen su presentación pública en 1934 a lo largo de un curso que Gödel imparte en Princeton. Como él mismo indica, se trata de desarrollar una idea sugerida por el matemático francés J. Herbrand. La idea se refiere a admitir como medio para definir funciones numéricas todo tipo de substituciones preservadoras de la identidad que quepa ejecutar sobre una colección finita de ecuaciones. Parece, por tanto, que la idea subyacente es de distinta factura a la que Church considera para definir el cálculo- λ .

Pese a aceptar la posibilidad de que esa clase de funciones reúna en su interior todas las recursiones, Gödel parece poco inclinado a aceptar la sugerencia que a ese respecto le hace Church, y que, obviamente va más allá. En efecto, Church sugiere la posibilidad de considerar que las funciones definibles en el cálculo- λ , o que son recursivas generales, agotan todas aquellas que están asociadas a un mecanismo efectivo de cómputo. Para minar las reticencias que Gödel manifiesta ante la identificación de un concepto intuitivo –calculabilidad efectiva- con otro formal –recursión general- Church se prepara para demostrar que tales versiones son equivalentes.

Otros modelos. Normalización

El que da los pasos que resultan decisivos para alcanzar ese resultado es S. Kleene convencido también de la hipótesis que Church comunica a Gödel en ese periodo. Para establecer la identidad entre cálculo- λ y recursividad general, Kleene desarrolla un tercer formalismo, el de la recursión por minimalización, $[Fr\mu]$, al que se reducen esas clases. Merece la pena describir su aspecto. La clase de las funciones recursivas por minimización se obtienen a partir de una ampliación de los recursos con que se caracteriza una idea muy próxima, la de la recursión primitiva.

[5] *Base para la recursión por minimalización:*

Funciones iniciales:

F1) $s(x)=x'$

F2) $c(x_1, \dots, x_n)=k$, donde $k \in \mathbb{N}$

F3) $p_n^i(x_1, \dots, x_i, \dots, x_n)=x_i$.

Reglas:

R1) Composición: $\phi(x_1, \dots, x_n)=\psi(\theta_1(x_1, \dots, x_n), \dots, \theta_m(x_1, \dots, x_n))$

R2) Recursión primitiva: $\phi(0, x_1, \dots, x_n)=\psi(x_1, \dots, x_n)$
 $\phi(y', x_1, \dots, x_n)=\theta(y, \phi(y, x_1, \dots, x_n), x_1, \dots, x_n)$

R3) Minimalización: $f_i^n(x_1, \dots, x_n)= \mu y[g_j^{n+1}(x_1, \dots, x_n, y)=0]$, si sucede que
 $\exists y[g_j^{n+1}(x_1, \dots, x_n, y)=0]$,
indefinida en otro caso,

donde la expresión a la derecha de la identidad significa *el menor de los y tales que* $g_j^{n+1}(x_1, \dots, x_n, y)=0$, y supuesto además que $g_j^{n+1}(x_1, \dots, x_n, y)$ es una función totalmente definida.

[6] *Funciones recursivas por minimización*: Se trata de la menor clase de funciones que satisfacen las siguientes condiciones:

- i. es alguna de las funciones F1-F3 indicadas en [3], o bien,
- ii. ha sido obtenida a partir de una función ya incluida en esa clase por medio de una serie finita de aplicaciones de las reglas R1-R3.

La clase de las funciones recursivas primitivas se obtiene eliminando como procedimiento admisible la regla R3 de minimización. Es obvio, no obstante, que tanto la recursión primitiva como la recursión por minimización parecen responder al mismo patrón intuitivo. Se trata de presentar una colección finita de funciones cuyo cálculo es efectivo de una forma que podríamos calificar de inmediata e intuitiva. Como ya hemos visto, son todas ellas trivialmente efectivas. Para obtener nuevas funciones también efectivas, bastará con introducir una serie de reglas de composición sobre funciones que preserven la efectividad en aquellas que se definen con su concurso. No es difícil apreciar que todas ellas poseen ese requisito. No obstante, en esta definición he prescindido ya de un requisito común por entonces: que las funciones resultantes estén totalmente definidas. Esta exigencia se mostraría excesiva y perniciosa a lo hora de configurar un modelo satisfactorio de la calculabilidad.

Parece evidente que el cálculo- λ , las funciones recursivas generales, las máquinas de Turing y la recursión por minimización se dirigen al mismo objetivo, la caracterización de aquellas tareas sobre números que cabe ejecutar de manera efectiva. Pero resulta igualmente obvio que no cabe ver en ellas variantes mínimas de una idea común, y por tanto, tampoco hay por qué presuponer nada acerca de su equivalencia. Esta equivalencia se traduce en un resultado muy concreto. Hay que demostrar que toda función definible en el cálculo- λ es recursiva general, recursiva por minimización, computable mediante una máquina de Turing, etc.

Otros modelos. Normalización

Hemos dicho que Kleene introduce la recursión por minimización para reducir a este procedimientos otros ya conocidos. ¿Dónde reside tan fabulosa capacidad? ¿Cómo llega Kleene a darse cuenta de ello? La idea pudo ser más o menos como sigue. ¿Cuál es el rasgo que todos estos procedimientos –aunque históricamente las máquinas de Turing aún no estaban en el mercado las incluiremos en estas consideraciones- tienen en común? Kleene piensa acertadamente que todos ellos definen sus objetos característicos –funciones- mediante un sistema simbólico finito. El cálculo del valor de una función tiene entonces el aspecto general de la aplicación de una serie de transformaciones finitas definidas para las expresiones inicialmente aportadas. Esas transformaciones pueden llegar a arrojar una forma, toda, o parte de la cual, consideraremos el valor de la función. Esa cadena de símbolos tiene, por otra parte, un aspecto perfectamente identificable en el proceso –recuérdense las condiciones impuestas en el caso de las máquinas de Turing sobre el estado de parada, por ejemplo-. Si podemos hablar de esta forma sin entrar en el detalle de cada procedimiento, ¿por qué no tomar esto mismo como definición general, precisa, de la efectividad? Para ello sólo sería preciso poder referirse a cualquier procedimiento simbólico del tipo descrito de forma indeterminada. ¿Es posible hacer esto? Kleene sabe muy bien que sí. Gödel ha hecho buen uso de este tipo de estrategia pocos años antes al operar sobre códigos de expresiones en el interior de la Teoría elemental de números. Si fuera posible referirse a una expresión simbólica finita mediante un código, y a una transformación definida sobre ese tipo de expresiones mediante una operación trivialmente efectiva sobre códigos, entonces toda la diferencia entre los distintos formalismos propuestos se reduciría al punto de hacerse insustancial: existirían transformaciones regulares definibles entre códigos y operaciones que conducirían de un formalismo a otro sin solución de continuidad. Toda la matemática que se precisa para establecer este sistema de codificación se encuentra dentro de la clase de las funciones recursivas primitivas. Con su concurso afirma Kleene, es posible definir un predicado metateórico que resulta ser recursivo primitivo. El predicado es el siguiente:

- [7] $T(e, x_1, \dots, x_n, y)$ es la relación definida sobre n -tuplas de enteros positivos en la que $\langle e, x_1, \dots, x_n, y \rangle \in T$ si y sólo si e es el código de un sistema finito de cadenas de símbolos el cual al iniciar sus transformaciones cuando se le suministra la n -tupla $\langle x_1, \dots, x_n \rangle$ como input alcanza en su y -ésima transformación una configuración final.

Como ya he dicho, Kleene demuestra que este predicado numérico es recursivo primitivo. El código de la configuración final posee la información relativa al valor numérico que cabe asociar a la n -tupla que se toma como input. Sólo hace falta considerar un procedimiento de descodificación. La función que se ocupa de este trámite se representa como $U(y)$ –*upshot* o resultado de y -. Para representar funciones numéricas en general asociadas a un procedimiento de cálculo basado en manipulaciones finitas de símbolos del tipo más general que quepa imaginar sólo hay que añadir un recurso para aludir a la *existencia* de una configuración final, si es que existe. Obviamente, no podemos presumir que tal situación exista a riesgo de caer de nuevo bajo el alcance de la diagonalización. Es obvio que esta parte le corresponde a la operación de minimalización. Todo junto lleva al sorprendente resultado que exponemos bajo la denominación de *teorema de normalización de Kleene*.

- [8] *Teorema de normalización de Kleene*: Toda función numérica $\varphi(x_1, \dots, x_n)$ recursiva general, λ -definible, Turing-computable, recursiva por minimalización, etc, puede ser expresada equivalentemente de la siguiente forma:

$$\varphi(x_1, \dots, x_n) = U(\mu y \, T(e, x_1, \dots, x_n, y)), \text{ para algún entero positivo } e.$$

Nótese que en la medida en que $T(e, x_1, \dots, x_n, y)$ es recursivo primitivo está siempre definido y cabe aplicar la regla de minimalización.

Otros modelos. Normalización

No es de extrañar que Gödel reconociera años más tarde que uno de los pocos resultados no triviales que pudo ver en aquellos primeros días del desarrollo de la Teoría de la Computación fuera precisamente éste. Sólo hay que lamentar que su carácter abstracto, junto con el hecho de establecer un resultado que, por una vez, no desbarata la confianza del ser humano en sus capacidades formales, le haya privado de la fama que otros resultados tienen. A mi juicio estamos ante uno de los momentos cumbre del pensamiento matemático de todos los tiempos comparable en belleza a otros episodios mucho más conocidos por legos y profanos.

Este resultado parece estar disponible ya durante 1935, aportando unas garantías a la identificación propuesta por Church entre calculabilidad efectiva y formal hasta entonces insospechadas. Aunque la identidad que Turing establece entre su propio formalismo y el del cálculo- λ fue un resultado muy bienvenido en esos momentos, la exposición que aquí he hecho del teorema de Kleene parece restar importancia a ese dato, pero esto es algo de lo que hablaremos algo más adelante.

Esta narración de los hechos es la que permite dividir los modelos formales de la efectividad en las clases [3] y [4]. Sucede, simplemente, que hay un momento a partir del cual la expectación desaparece dejando lugar a móviles distintos, generalmente asociados a necesidades prácticas. El modelo de Post es anómalo en ese sentido, ya que vuelve a tratarse de un formalismo muy abstracto que define colecciones de transformaciones sobre palabras. Pero Post fue, en muchos sentidos, un personaje muy anómalo. Los algoritmos de Markov representan un modelo muy similar al que propone Post.

Completamente distinta es la historia que rodea a los modelos que aparecen bajo los rótulos de máquinas de registros, ábacos o máquinas de acceso aleatorio –*random access machines*-. Todas ellas se parecen considerablemente entre sí y a su vez, guardan una estrecha relación con las máquinas de Turing. De hecho, se suele considerar que su definición es el resultado de especializar la cinta de cálculo que aparece en la definición de una máquina de Turing. Las máquinas de registros, por

ejemplo, considerar un número en principio ilimitado de registros, que podemos representar mediante $[n]$, donde n es un numeral, en los que se pueden almacenar cantidades en principio ilimitadas de marcas –bolas, piedras, o cualquier recurso que podamos imaginar-. Las acciones básicas son las añadir o sustraer una marca a un registro dado. Definir a partir de ahí operaciones más complejas es un juego en el que podemos desarrollar nuestro ingenio a placer. La idea es notable porque favorece una tendencia cognitiva a la que estamos muy habituados: especializar tareas dentro de la clase general de procedimientos. Dicho de otra forma, podemos asignar cierto significado a cada registro obteniendo así un rendimiento intuitivo muy superior a aquel que vemos en las máquinas de Turing. En el caso de las máquinas de acceso aleatorio, la novedad es que cada cadena de símbolos posee una dirección –*address*– sobre la que es posible operar. Las operaciones recuerdan en mucho a aquellas que figuran en lo que se denomina código de máquina de los modernos procesadores. Como es obvio no es casualidad.

No hay límites a las variantes que podemos imaginar. Unas serán modificaciones más o menos substanciales de modelos ya existentes, y aún cabe que otras sean novedosas en algún sentido. Sea como fuere, siempre que respetemos la inexistencia de limitaciones de espacio o memoria y pretendamos capturar deliberadamente toda la extensión de tareas que pueden ser descritas en los términos que establece el teorema de normalización, sabemos que no iremos más allá de lo que ahora tenemos. En otras palabras, el teorema de normalización puede ser interpretado como un aviso acerca del tipo de condiciones que deberían ser superadas por algún otro procedimiento que pretendiese dar una expresión formal de ciertas tareas efectivas que no puedan ser expresadas en términos de los procedimientos ya existentes. ¿Es posible imaginar algo así?

Orientación bibliográfica.

[**Boolos, 1974**] ofrece la exposición de una serie de variantes históricas de la computabilidad. Es una referencia fácil de seguir y muy bien expuesta. Útil por lo que hace a funciones recursivas y máquinas de registros (ábacos). Además desarrolla una demostración de equivalencia que no es frecuente exponer en los manuales. [**Alonso, 1996**] contiene una detallada demostración del modo en que las funciones Turing-computables y $[Fr\mu]$ coinciden.

[**Salomaa, 1985**] añade versiones más abstractas de las que apenas diremos nada aquí: sistemas de Post y algoritmos de Markov. Para el cálculo- λ no hay muchas opciones. El manual clásico de referencia es [**Barendregt, 1981**], aunque como breve introducción histórica más que otra cosa podría valer [**Rosser, 1984**]. En esa misma dirección se puede citar [**Penrose, 1989**] pp.100 y ss. que resulta en esta ocasión especialmente claro y bien motivado.

Para el teorema de normalización valen cualquiera de los manuales citados en la sesión anterior. Me atrevo a sugerir la propia interpretación que de este resultado hace Kleene en [**Kleene, 1952**] en las secc. 59 y 60. No obstante se trata de unas

páginas extremadamente difíciles con lo que sólo se recomienda como lectura tutelada.