La jerarquía de Chomsky: una evolución esperable en teoría de la computabilidad

Prof. Dr. Celso Vargas[1]
Instituto Tecnológico de Costa Rica

Resumen: En este artículo se analiza, desde una perspectiva histórica y conceptual, la evolución de la teoría de la computabilidad, comenzando con los resultados de Church y Turing en 1937 hasta la jerarquía de Chomsky in 1963. Esta jerarquía es uno de los resultados más importantes en teoría de la computabilidad, y está relacionada con la manera en la que las gramáticas, los lenguajes, algebras y autómatas pueden ser ordenados de acuerdo con su capacidad generativa. Lo que este resultado muestra es que existe equivalencia entre ciertas clases de autómatas, lenguajes y gramáticas. Tomó un poquito más de dos décadas lograr este importante resultado, lo cual muestra la rapidez con la que evolucionó este campo. En este artículo seguimos la estrategia de presentar los resultados más importantes por décadas, comenzando con la década de los 40, en la cual obtuvimos dos resultados importantes, pero será en la década de los 50 en la que obtengamos los principales elementos de esta jerarquía, cuya consolidación la vemos con claridad durante los tres primeros años de la década de los 60.

Descriptores: Teoría de la computabilidad · Autómatas · Gramáticas · Jerarquía de Chomsky · Historia de la computabilidad.

Abstract: In this paper it is analyzed from a historical and conceptual perspective the evolution of the theory of computability, starting from the results of Church and Turing in 1937 to the Chomsky hierarchy in 1963, the year in which it was completed. This is one of the most important results in the theory of computability, and concerns the way in which grammars, languages, algebras and machines can be ordered according to their generative power. What this shows is the equivalence between certain classes of automata, languages and grammars. It took a little more than two decades to achieve this important result. This suggests how fast evolved the research in this area. The strategy used in this paper is to account by decade the contribution to this goal, starting in the 1940´s in which we found two main contributions: the introduction of finite state automata and the concept of stack (now pushdown). But it was during the 50´s when the progress was spectacular: four main results were obtained. It was during the first three years of the 60´s that this hierarchy reached the current form.

Keywords: Computability theory · Automata · Grammars ·Chomsky hierarchy · History of computability

Recibido: 01/02/ 2013 Aceptado: 30/04/2013

[1] Escuela de Ciencias Sociales. E-mail: celvargas@itcr.ac.cr

Introducción

De 1936 a 1963 observamos una muy rápida evolución del nuevo campo conocido como teoría de la computabilidad. Los dos teoremas de Gödel relacionados con la incompletitud de sistemas axiomáticos que tengan la capacidad expresiva de los *Principia Mathematica* de Russell y Whitehead (fundamentales para la teoría de números), abrieron la posibilidad de avanzar en el desarrollo de métodos generales abstractos para decidir cuándo es posible responder afirmativa o negativamente determinadas preguntas relativas a problemas que admitan una formulación formal. Tanto las máquinas de Turing como la tesis Turing-Church de 1936-37 constituyeron los elementos fundacionales de este importante campo de la computación. Las dos décadas siguientes serán fundamentales para darle consistencia a este nuevo campo del conocimiento. Pero también vemos emerger nuevos campos de investigación interdisciplinaria como la cibernética (Sayre, 1996), la propuesta e implementación de arquitecturas de computadoras por von Neumann y otros, el desarrollo de lenguajes de programación, inteligencia artificial, entre otros.

Uno de los productos más importantes de estas décadas es el establecimiento de lo que se conoce actualmente como la "Jerarquía de Chomsky" formulada por Chomsky a partir de sus trabajos de 1956, 1957 y 1959, y completada en los dos primeros años de la década de los 60. El interés de Chomsky (1928-) era aplicar estos instrumentos formales para comprender y especificar las lenguas naturales, como el español o el inglés. Su pretensión inicial era proponer formalismos lingüísticos que pusieran de manifiesto que este tipo de lenguajes puede ser definido recursivamente. Es decir, mostrar que una lengua natural puede ser generada mediante un conjunto de reglas de tal manera que toda oración con significado en esa lengua pueda ser generada mediante un conjunto de reglas con características matemáticas específicas. Para lograrlo era importante caracterizar formalmente el tipo de lenguaje y sus propiedades, de manera que se mantuviera un compromiso importante entre lo que Chomsky denomina "adecuación descriptiva" y "adecuación prescriptiva", como tendremos ocasión de analizar. Posteriormente, se establecerá la equivalencia entre clases de lenguajes y autómatas, con lo que se alcanzan desarrollos computacionales extraordinarios. A esto contribuirán decisivamente Miller & Chomsky (1963), pero también Schützenberger. Ha sido la computación la que ha obtenido los mayores beneficios de estos desarrollos.

La jerarquía de Chomsky es, pues, un ordenamiento de los tipos de lenguajes según sus propiedades matemáticas y los correspondientes autómatas aceptores o generadores. Su jerarquía está compuesta por cuatro clases generales de lenguajes, conocidos como lenguajes regulares, lenguajes libres de contexto, lenguajes dependientes de contexto y los lenguajes irrestrictos. Desde el punto de vista de

sus aceptores o generadores tenemos los autómatas de estado finito que aceptan únicamente lenguajes regulares, los autómatas con una pila de memoria que aceptan lenguajes libres de contexto, los autómatas de pila linealmente acotados que aceptan los lenguajes dependientes de contexto y las máquinas de Turing que aceptan lenguajes irrestrictos. Como jerarquía mantienen un orden de inclusión propia, como tendremos ocasión de analizar.

En este artículo abordamos, desde el punto de vista histórico y conceptual, la evolución desde las Máquinas de Turing y la Tesis Turing-Church, hasta la jerarquía de Chomsky. Sin embargo, es importante tener presente, como ha puesto de manifiesto John E. Savage (1998), pp.3-7, que estos desarrollos no sólo fueron importantes para lenguajes de programación, compiladores e intérpretes, sino que pronto dieron paso a estudios del desarrollo de la teoría de la complejidad de algoritmos (respecto de espacio y tiempo), computación paralela, circuitos integrados VSLI y otros desarrollos vitales y que han sido integrados en la potenciación de este importante campo de la ciencia y la ingeniería. En todo momento debemos tener presente que las máquinas de Turing y sus equivalentes constituyen los mecanismos de decisión más generales. Nuestro trabajo aquí tiene pues como límite superior estos dispositivos generales, y lo que buscamos es la caracterización de otras clases con menor capacidad expresiva que las máquinas de Turing. La jerarquía de Chomsky constituye una clasificación adecuada a este propósito.

Máquinas de Turing y la Tesis Turing-Church

La formalización matemática de la lógica con Frege, Russell y Whitehead, así como los trabajos de Hilbert y sus discípulos, abrieron la posibilidad de ver la lógica como la nueva herramienta para fundamentar todas las matemáticas. Durante el siglo XIX las matemáticas sufren un extraordinario desarrollo tanto en el álgebra como en geometría, sin embargo estos desarrollos no están unificados bajo un marco general. La lógica pretende ofrecer tal marco. Hilbert (1862-1943) jugará un papel de primera línea adelantando un nuevo programa de investigación centrado en la lógica. Es en 1928 que Hilbert plantea sus famosos tres problemas relacionados con la fundamentación de las matemáticas desde el punto de vista finitista:

A. "¿Son las matemáticas completas, en el sentido de que cualquier postulado pueda ser probado o rechazado?"
B. "¿Son las matemáticas consistentes, en el sentido de que nunca se pueda demostrar algo que sea manifiestamente falso?"
C. "¿Son las matemáticas decidibles, en el sentido de que se puede crear un sistema de deducción paso a paso que aplicado a cualquier postulado permita determinar si es cierto o falso?"

Como es ampliamente conocido, las primeras dos preguntas son contestadas negativamente por Gödel, al mostrar que para sistemas con la complejidad de poder expresar la aritmética, se tiene que tomar una decisión entre mantener consistencia y sacrificar completitud o mantener completitud y sacrificar consistencia. Esto claramente, afectó considerablemente al programa hilbertiano. Como ha puesto de manifiesto Irvine (1996), p. 27, el segundo teorema de Gödel (G2) tiene importantes implicaciones filosóficas:

> "Específicamente, se afirma que G2 implica tres corolarios filosóficamente relevantes: primero que cualquier prueba de consistencia para la teoría T, en la cual se mantiene G2, descansará en métodos lógicamente más poderosos que aquellos de la teoría T misma; segundo, que (en cualquier caso significativo) una prueba de consistencia para la teoría T puede no producir ninguna ganancia epistemológica y así no puede proporcionar respuestas satisfactorias al escéptico en relación con la consistencia de T; y tercero, que como resultado de esto, G2, si bien no conlleva un estrepitoso fracaso del programa de Hilbert, al menos indica que son necesarias modificaciones sustantivas"

La teoría de la computabilidad se fundamenta en la respuesta al tercer problema, conocido como "Entscheidungsproblem" (el problema de la decisión). De rebote, se sugiere una respuesta negativa al tercer problema, aun cuando restaba por establecer un procedimiento mecánico general para caracterizar aquello que es "efectivamente computable". Como hemos indicado en Vargas (2012), para este problema encontramos ya en 1935 y 1936 dos posibles formas de respuesta. La primera de ellas de Church, basada en su cálculo lambda en el que logra establecer que una función es efectivamente computable, si y solo si es lambda definible. Por "lambda definible" se entiende una función sobre números enteros positivos para los cuales los valores de la función puede ser calculados mediante un proceso de sustitución repetitiva, es decir, de manera recursiva. La segunda respuesta está basada en lo que se conoce como Máquinas de Turing. Lo que es importante de estos dos resultados es que se logra establecer la equivalencia entre funciones definibles en términos del cálculo lambda y lo que es mecánicamente realizable mediante una máquina de Turing. El problema de la decisión se responde de manera negativa, pero al mismo tiempo se abre el espacio para avanzar en la caracterización más detallada de la clase de funciones que son efectivamente computables.

Nuestra historia comienza aquí. Sin embargo, es importante tener en mente que la misma diferencia o doble perspectiva encontramos durante todo este proceso. Por un lado, los progresos en los desarrollos teóricos sobre la estructura y funcionamiento de los dispositivos de bajo nivel comienzan a desarrollarse a veces de manera independiente de la clase de lenguajes que pueden ser asociados con cada

uno de ellos, y que podríamos denominar de "alto nivel". Como se ha mencionado la equivalencia entre las funciones efectivamente computables y la clase de máquinas (máquinas de Turing) fue establecido por estos autores. Esta misma convergencia es la que vamos a encontrar en los primeros años de la década de los años 60, cuando se logra asociar con cada clase de lenguajes una clase de dispositivos denominados autómatas. Quizá sea conveniente ir analizando los resultados por décadas para ir viendo la manera en la que se va a avanzando en los diferentes aspectos.

Progresos durante la década de los 40

Nos interesa ir analizando aquellos resultados que se mostrarán relevantes para la jerarquía de Chomsky. Utilizamos los conceptos actuales y, cuando sea necesario, los términos que se utilizaron en su momento. Durante esta década dos resultados importantes para nuestro objetivo vieron su nacimiento: a) los autómatas de estado finito, los cuales son originalmente propuestos por McCulloch y Pitts en 1943, y b) Alan Turing en 1947 introduce las pilas ("stack" en ese momento, "pushdown" después), en su discusión sobre los sistemas de almacenamiento en el marco de la propuesta de arquitectura de computador por parte de Charles Babbage.

El concepto de autómata de estado finito por McCulloch y Pitts (1943)

Este trabajo pionero de McCulloch y Pitts titulado "A Logical Calculus of the Ideas Immanent in Nervous Activity" tiene que ver la utilización de la lógica proposicional para lograr una representación lógica del funcionamiento del sistema nervioso central. Estos autores reciben un influencia directa del trabajo de Carnap "The Logical Syntax of Language" en el cual Carnap propone la formalización lógica como uno de los medios importantes para el conocimiento del mundo, y por lo tanto, es un método de investigación filosófica lo mismo que lo es en la matemática y en la física. Estos autores aplican este método en la modelación de la transmisión del estímulo electro-químico utilizado por las neuronas en la sinapsis. Este trabajo es continuado por D. Hebb, quien en 1949 propone explicar el aprendizaje neuronal como el establecimiento de nuevas conexiones entre neuronas y el fortalecimiento del mecanismo de transmisión entre ellas. En el caso de estos autores, dado el carácter de "todo o nada" que exhibe el sistema nervioso central en el sentido de que en ciertas condiciones es posible la excitación y la transmisión del impulso nervioso, y en otras la imposibilidad de la transición o el estímulo inhibitorio, la lógica proposicional es adecuada a esta labor. En este sentido, en cada momento se puede determinar el número de neuronas que están activadas (excitadas) y cuáles no (inhibidas). Si representamos por S_o el conjunto de neuronas que están activas

en el momento 0, S_1 las activas en el momento 1, y así sucesivamente, entonces, S_o, S_1, S_2,…., S_n representa la evolución de la red neuronal en el intervalo temporal {0, 1,…., n}. Por complemento podemos obtener aquellas neuronas que no lo están. Es un modelo temporal en el que la precedencia temporal es importante. Cada una de las neuronas en un momento determinado está excitada o inhibida, de manera que en la representación de esta sistema se utilizan únicamente dos estados; de ahí el nombre de sistemas de estado finito. Así pues, cada una de las neuronas puede representarse como una función cuyos argumentos son una proposición que expresa la propiedad de la neurona, 1 y 0, o verdadero y falso, para expresar si está excitada o no. Claramente, cada momento o estado de la red se puede representar mediante fórmulas conjuntivas o disyuntivas. Los autores presentan dos tipos de modelos de sistemas de estado finito: uno sin ciclos en el que la estructura de la red no cambia con el tiempo, se respeta los periodos de potencial de reposo (es decir, aquellos en los que la neurona no se excita por más intenso que sea el estímulo) y donde el único retraso en la activación de la red es aquel del retraso sináptico. El segundo sistema de estado finito es aquel que utiliza ciclos, es decir, determinadas neuronas pueden tener la propiedad de excitarse un número determinado de veces, ya sea por sí mismas o porque formen un ciclo con otras neuronas y dicho ciclo se repita *n* veces. Para ambos casos, los autores prueban determinados teoremas relativos a estas clases de sistemas, pero que no vamos a analizar aquí. Esta modelación del funcionamiento del sistema nervioso en términos de transiciones entre estados es la primera formulación de un autómata de estado finito.

La introducción del concepto de pila

De acuerdo con Sten Henriksson (2010) el concepto de pila (*stack* y *pushdown*) fue introducida con propósitos técnicos en la década de los 30, cuando se analizaron algunos mecanismos para llevar el control de las proposiciones y conectivas lógicas en la notación polaca de Lukasiewicz. Como se recordará esta notación es prefija, en el sentido de que se introducen los operadores de izquierda a derecha y después de las variables o constantes proposicionales. El ámbito o rango de las conectivas viene dado según la posición: entre más a la izquierda se encuentre la conectiva, mayor es su ámbito. Así las pilas son dispositivos para ir poniendo en cierto orden las conectivas y luego irlas sacando de manera se pueda evaluar las proposiciones complejas. La forma básica de la pila es "last-in, first-out" (LIFO), aunque existen otras posibilidades.

De acuerdo con Donald Knuth (Henriksson 2010, p.4), se debe a Turing la introducción de este concepto en el contexto de su análisis de los dispositivos de almacenamiento en los computadores digitales, en su trabajo de 20 de Febrero

de 1947 ante la London Mathematical Society. Es un trabajo extraordinario en el que el genio de este gran hombre se pone de manifiesto. Después de analizar las ventajas de las computadoras digitales sobre las electrónicas, presenta tres tipos de dispositivos de almacenamiento que estaban siendo considerados en ese momento: a) rebobinado magnético (magnetic wires), b) un dispositivo basado en patrones de carga y c) líneas de retraso acústico (accoustic delay lines) utilizando ondas de compresión en una columna de mercurio. Será respecto de este último en el cual analizara diferentes elementos: los mecanismos de amplificación de las ondas, las condiciones bajo las cuales estas ondas pueden ser utilizadas como dispositivos de almacenamiento, su conversión en representaciones binarias y cómo podemos optimizar su uso, por ejemplo, considerando únicamente aquellos casos en los que se presenta un pulso eléctrico y la manera en la que cada uno de los ellos puede ser expresado de manera discreta (número natural).

La parte que nos interesa mencionar brevemente es aquella que se relaciona con la forma en la que pueden ser utilizados estos dispositivos (columnas de mercurio) en los computadores digitales. Considera que en el caso de este tipo de computadoras la utilización de 200 de estas columnas cada una con una capacidad de 1024 dígitos binarios sería suficiente, lo cual nos daría una memoria total de 204800 dígitos binarios. Una capacidad extraordinaria en ese momento. En este contexto es en el que introduce las pilas como un mecanismo normal para conectar ubicaciones de memoria en un computador y facilitar su recuperación cuando sea necesario. Claramente aquí los procesos de almacenamiento y búsqueda son los que consumen una parte importante del tiempo, de manera que se requiere desarrollar algunos algoritmos de optimización. Propone complementar la computadora digital con un sistema de tarjetas Hollerith, ampliamente utilizado a partir de 1890 al menos. Ilustra su utilización cuando consideramos cada una de éstas como una tabla que contiene información específica sobre una determinada operación, por ejemplo, una posición decimal. Estableciendo "punteros" a cada una de estas tablas, desde la original a las subsidiarias, obtenemos una jerarquía de tablas que facilita este proceso de ubicación y recuperación de información. En este caso, no es necesario utilizar grandes recursos de memoria, sino el mismo principio: en cada tabla se guarda un dígito. Todo lo que hay que tener presente es la posición jerárquica que ocupa la tabla. Pero esta jerarquía no es otra cosa que una pila en la que los elementos que son procesados primero son ubicados en la pila, de manera que se facilita rápidamente su recuperación siguiendo el método de la inversa, es decir, la última tabla que se ha guardado es la primera en ser recuperada. De esta manera vemos aparecer dos conceptos de vital importancia para la teoría de la computabilidad, como veremos más adelante. Sin embargo, la formalización precisa de los mismos deberá esperar todavía mucho más, hasta 1961 y 1963 con Minsky y Evey.

Los progresos en la década de los 50

Durante esta década se darán pasos realmente significativos hacia el establecimiento de la jerarquía de Chomsky tal y como la conocemos actualmente. Son cuatro los resultados principales. Primero, se avanza considerablemente en la comprensión y el establecimiento de las propiedades de los autómatas de estado finito, que ya vimos aparecer con el trabajo de McCulloch y Pitts (1943), principalmente por el trabajo de Kleene (1951/6), en el que propone una forma de construir lenguajes (álgebras) basado en operaciones simples, y ciertas correspondencias entre los lenguajes generadores y los autómatas de estado finito. Segundo, los trabajos de Mealy (1955) y de Moore (1956) para la representación de circuitos lógicos secuenciales y en los que tenemos de forma explícita la descripción de los autómatas de estado finito. Tercero, Chomsky en esta década establece una parte importante de la jerarquía de Chomsky, que tiene que ver con jerarquización de los lenguajes que puede ser obtenida a partir de gramática; esta jerarquización se establece según su capacidad generativa. Encontramos tres trabajos relevantes aquí: el primero de ellos de 1956, otro de 1957, y el último publicado en 1959 en el que se establece claramente esta jerarquía. Sin embargo, la equivalencia entre los lenguajes generados por las gramáticas o las álgebras no se correlaciona directamente con los autómatas, en ese momento. Este trabajo será realizado en los primeros años de la década siguiente. Finalmente, en 1959, Newell, Shaw & Simon introducen los autómatas de pila en computación en su trabajo sobre el GLS (General Logical solver), con lo cual se inicia una nueva era en la década de los 60.

Las álgebras de Kleene y los autómatas de estado finito

Stephen Kleene ha jugado un papel muy importante en el desarrollo de la teoría de la computabilidad. Trabajó bajo la dirección de Church desde 1934 y extiende los resultados de Church utilizando la propuesta más general de funciones recursivas desarrollada por Gödel. Pero el trabajo que nos interesa para los propósitos de este artículo es el que escribe en diciembre de 1951 bajo el título de "Representation of events in nerve nets and finite automata", y publicado en 1956 en *Automata Studies* de la Universidad de Princeton. Como su nombre lo indica, Kleene explora en este trabajo las consecuencias tanto prácticas como teóricas (más teóricas) del trabajo propuesto por McCulloch y Pitts en 1943, en particular para el campo de la robótica. Primero, Kleene muestra que el comportamiento de una red de McCulloch y Pitts puede ser descrita en términos de un conjunto de "eventos regulares". Dado que los procesos de activación neuronal en este tipo de modelos toma tiempo, podemos expresar, como ya se mencionó, cuáles neuronas están

activas y cuáles no en un momento determinado. Así pues, el comportamiento de la red para un intervalo suficientemente grande, puede representarse en términos de un conjunto de eventos regulares para cada uno de los instantes en lo que hay activación neuronal, es decir, es la sumatoria de los eventos regulares durante ese intervalo. Si hacemos abstracción de los factores temporales, los elementos de esta representación son los siguientes: E es un evento regular; E*, que significa que se da cero y más eventos regulares E; EE_1 que es la concatenación de dos eventos; $E+E_1$, la disyunción de los eventos E y E_1. Se utilizan paréntesis para desambiguar eventos regulares. De esta manera, (E)* significa que no ocurre E o se da un número ilimitado de veces. Dado un conjunto $\Sigma = \{ E_1, E_2,\ldots, E_n\ n\geq 1\}$ de eventos, podemos construir diferentes álgebras utilizando las siguientes reglas: i) $e \epsilon \Sigma^*$ (el evento regular vacío) pertenece al monoide libre Σ^*; ii) Todos los elementos de $\Sigma \epsilon \Sigma^*$; iii) Si α y $\beta \epsilon \Sigma^*$, entonces, también pertenecen $\alpha\beta$, $\alpha+\beta$ y α^*. iv) Únicamente son elementos de Σ^* aquellos que se obtengan mediante la aplicación de estas reglas. Así pues, el comportamiento de estas redes neuronales puede ser captado mediante este conjunto de eventos regulares para utilizar el término utilizado por Kleene (teorema 6 en este escrito de Kleene).

El segundo aspecto que nos interesa mencionar es que el inverso del resultado anterior también se mantiene. Para cada evento regular existe un autómata de estado finito o red que lo acepta. Este resultado Kleene lo establece como un corolario del resultado anterior. De esta manera, tenemos la correspondencia entre expresiones regulares y autómatas, un resultado que forma parte estructural de la jerarquía de Chomsky como veremos más adelante. Kleene se percata de que hay eventos que no pueden ser representados por medio de una autómata de estado finito (lo analiza en el anexo 3), con lo cual visualiza que este tipo de álgebras son un subconjunto de otras estructuras matemáticas de complejidad mayor.

Jerarquización de los lenguajes

Noam Chomsky es una figura fundamental en nuestra historia, y por tal razón merece una mayor extensión en este artículo que el proporcionado a otros autores. Es fundamental en dos sentidos diferentes: por lo revolucionario de sus puntos de vista en el campo de la teoría lingüística y en otros campos, y segundo, por sus contribuciones en el perfilamiento de la jerarquía que lleva su nombre. Sobre este último aspecto, como tendremos ocasión de mencionar con mayor detalle, son varias las equivalencias que se pueden establecer dentro de esta jerarquía. Sin embargo, antes de avanzar en la primera propuesta de jerarquía de lenguajes y sus respectivas gramáticas, es importante situar el contexto en el cual introduce esta jerarquía que revolucionará varios campos del conocimiento y de la ingeniería.

La preocupación principal de Chomsky fue la lingüística. En la época en que Chomsky introduce su revolución lingüística, el "paradigma" dominante, si podemos usar este término, era el estructuralismo lingüístico, es decir, aquella visión según la cual el lenguaje tiene varios niveles estructurales. Esta visión de la lingüística se inicia con Ferdinand de Saussure y es fortalecida con el advenimiento de la filosofía del positivismo lógico, que tiene su auge durante la segunda y tercera década del siglo XX; pero mientras aquélla comienza a entrar en crisis casi inmediatamente después de su introducción, la filosofía analítica, de la que el positivismo lógico puede verse como su versión radical, tendrá un extraordinario y duradero efecto que se extiende hasta nuestros días.

Como recordamos, el positivismo lógico mantiene la tesis básica de que todo enunciado con sentido es o bien un enunciado analítico (lógico o matemático) o bien es un enunciado sintético, es decir, que se refiere la experiencia. El único método legítimo para obtener conocimiento empírico (de la experiencia) es por medio de la inducción, aunque podría decirse que hay otros enunciados que pueden ser reducidos directamente a enunciados sobre experiencia, por ejemplo, aquellos que involucran entidades teóricas como el átomo. Uno de los métodos más interesantes para llevar a cabo esta reducción son las "oraciones Ramsey". En la lingüística dominante durante la década de los 30 y hasta los 50 del siglo pasado, se tomó muy en serio el proceso de construcción de teorías lingüísticas siguiendo el camino trazado por el empirismo lógico. Como admirablemente ha puesto de manifiesto Newmeyer (1980), las teorías lingüísticas deben proceder de la fonemática, primer nivel estructural, conformado por el conjunto de alófonos identificables como propios en una determinada lengua, y la organización de éstos en términos de unidades independientes (fonemas), avanzar hacia la estructura morfo-fonemática (la organización de los fonemas en agrupamientos con ocurrencias en el orden de la palabra o en el contexto de las palabras), a la sintaxis (estructura y orden de las palabras para formar oraciones) y, finalmente, al discurso (la organización de las oraciones y sus conectores para formar agrupamientos complejos). De esta manera, el gran edificio de la construcción de la estructura de una lengua determinada está fuertemente soportada por la fonética, la fonología y por el principio de inducción. Dado un *corpus* lingüístico, es decir, un conjunto de registros de una lengua (por ejemplo grabaciones u otros tipos de registros), el método lingüístico debe comenzar por el recuento de los alófonos; mediante la aplicación del método de distribución complementaria (aquellos sonidos aparecen en contextos diferentes y complementarios) o por variación libre, obtenemos los fonemas de esta lengua. El punto y el modo de articulación son fundamentales en este proceso de discriminación; procedemos luego a identificar agrupamientos mayores (morfemas) y su posición regular en esa lengua, y así sucesivamente. En todo caso, mis hipótesis sobre una

determinada propiedad deben ser sometidas a la validación inductivamente, primero tomando en consideración el conjunto de registro inicial; y segundo, recurriendo a aquellas ampliaciones o registros adicionales que permitan verificar mi hipótesis. De esta manera, no es aceptable ninguna generalización sobre una lengua determinada si no está soportada por un conjunto de cuerpos lingüísticos. La lingüística estructural avanzó enormemente en fonética, fonología y morfo-fonología pero casi nada en los demás niveles estructurales de descripción lingüística.

Chomsky no podría encontrar satisfactorio este tipo de aproximaciones. En efecto, Chomsky está interesado en una serie de fenómenos que no pueden ser abordados desde esta perspectiva y que ponen de manifiesto la inadecuación de la lingüística estructural como teoría lingüística. Primero, la creatividad lingüística. Parte del hecho de que este fenómeno se observa en todos los niveles lingüísticos indicados anteriormente, por ejemplo, en el cambio lingüístico en todos los niveles, se observa que a nivel individual existe una constante creación de nuevas oraciones, nuevos sonidos, nuevos morfemas y palabras, siempre dentro de un marco general que le proporciona la lengua. Esta gramática que se esconde detrás de todos estos procesos es lo que debe ser el objeto principal de la investigación lingüística. Segundo, para nuestro autor hay fuertes indicios de la existencia de esa gramática en el reconocimiento de oraciones o palabras con sentido y sin sentido por parte de los distintos hablantes que conforman una comunidad lingüística. Es esta misma gramática la que guía tanto la creación lingüística como la construcción de estos enunciados, palabras y sonidos sin mucho significado dentro de la lengua. Tercero, una característica que exhiben los hablantes de una lengua es que su gramática es generativa, es decir, es un proceso de producción y de creatividad guiado por reglas, las cuales debe "descubrir" el lingüista. Cuarto, ciertos sistemas matemáticos juegan un papel muy importante en la comprensión, explicación y descripción del lenguaje. Desde este punto de vista, el objetivo fundamental de Chomsky es proporcionar una definición recursiva de lo que es una oración en una lengua natural.

Sobre esta base, Chomsky en el trabajo escrito en 1956 "Three Models for the Description of Language" y en su obra "Syntactic Structures" publicada un año después, analiza tres mecanismos generativos y discute su adecuación para reunir las condiciones generales descritas en el párrafo anterior. El primer modelo, es el de "procesos de estado finito de Markov", ahora conocidos como gramáticas regulares. Estos mecanismos generativos comienzan con un estado inicial, un conjunto de transiciones entre estados, en cada una de las cuales escribe el símbolo generado hasta alcanzar un estado final. Consideremos un ejemplo simple: Sea $\Sigma = \{a, b\}$ y sea $Q = \{S_1, S_2\}$ dos estados, S_1 estado inicial y S_2 el estado final, $\rightarrow$ una función binaria de transición entre estados ($X \rightarrow Y$ se lee "X se reescribe como Y", con $Y \epsilon (\Sigma U Q)$). Una gramática regular tendría las siguientes reglas:

1. $S_1 \rightarrow a\ S_1$
2. $S_1 \rightarrow a\ S_2$
3. $S_2 \rightarrow b\ S_2$
4. $S_2 \rightarrow b$

Una posible derivación es la siguiente: $S_1 \rightarrow a\ S_1 \rightarrow aaS_2 \rightarrow aabS_2 \rightarrow aabb$. Esta secuencia resulta de la aplicación primero de la regla 1 (regla de inicio); luego la regla 2, 3 y regla 4 (estado final). Dado que son reglas recursivas, el conjunto de derivaciones es infinito pero numerable.

Estos dispositivos presentan varias limitaciones desde el punto de vista lingüístico, siendo el más importante la imposibilidad de representar adecuadamente la relativización (anidamiento de oraciones), es decir, aquellos casos en los que hay más de una oración incrustada dentro de la oración principal. Por ejemplo, "el hombre que vino de Japón, que compra objetos usados, es calvo". En este caso, tenemos tres oraciones: "el hombre es calvo", "el hombre vino de Japón" y "el hombre compra objetos usados". Lo que es importante de señalar es que en las tres oraciones, "el hombre" tiene la misma referencia. Esta misma referencia no puede ser controlada con este tipo de mecanismos.

Así pues, pasemos al segundo tipo de gramáticas generativas, conocida como de estructura fraseal. Estas gramáticas tienen como punto de partida "O" de oración (recuérdese que el propósito de Chomsky es proporcionar una definición recursiva de lo que es una oración), y se descompone en constituyentes fraseales (siendo las principales FN (frase nominal), FV (frase verbal, denominados no terminales), los no-terminales se descomponen en terminales o no terminales, y así sucesivamente. Un ejemplo sencillo de este tipo de gramáticas es el siguiente: Sea Σ = { el, hombre, es, calvo} y sea Q = { O, FN, FV, Art, N, V}, y sus reglas las siguientes:

1. $O \rightarrow FN + FV$
2. $FN \rightarrow (Art) + N$ (es opcional del artículo)
3. $FV \rightarrow V + FN$
4. $Art \rightarrow el$
5. $N \rightarrow \{hombre, calvo\}$
6. $V \rightarrow ser$

La derivación de la oración "el hombre es (ser) calvo", se obtiene de la siguiente manera:

O → FN+ FV (1) → Art+N+ FV(2) → Art+N+ V+FN (3) → el N+ V+FN(4) → el hombre V+FN (5) → el hombre ser FN (6) → el hombre ser N (2) → el hombre ser calvo

Desde el punto de vista más formal, este tipo de gramáticas (después conocidas como libres de contexto) tienen la siguiente forma:

Σ = {a, b, c,…} el vocabulario terminal y sea Q = {A, B, C,…} el vocabulario no terminal, y sea A un símbolo distinguido (oración), conocido como axioma. Entonces, las reglas de este tipo de gramática son las siguientes (presentadas en la forma de Chomsky):

A → x β, donde x ϵ Σ y β ϵ Q* (conocido como cerradura de Kleene, de la que ya hemos hablado).

Lo que muestra Chomsky es que este tipo de gramáticas tienen la gran ventaja de permitir manejar anidamientos de oraciones, llevando un control preciso de los elementos que deben ser *instanciados* en cada anidamiento; sin embargo, no son suficientes para explicar fenómenos como la concordancia de número y género en español, ni tampoco las correspondencias entre las oraciones activas y las pasivas, la relativización y otros fenómenos lingüísticos. De esta manera, introduce un tercer tipo de gramáticas, denominadas transformacionales, las cuales tienen la propiedad de transformar una oración en otra oración o un componente *frasal* en otro componente. Este tipo de gramáticas debe aplicarse sobre las salidas de las gramáticas de estructura *frasal*, de manera que una gramática para una lengua natural está compuesta, desde el punto de vista sintáctico, de dos estructuras gramaticales. De hecho las salidas de la gramática *frasal* son denominadas "estructuras profundas", mientras que las generadas por la gramática transformacional se las denomina "estructuras superficiales". Son estas últimas las que coinciden con las oraciones que producimos como hablantes nativos.

Una regla transformacional toma la siguiente forma:

[N(sing) + ser] → [N(sing) + es]

Estas reglas se leen de la siguiente manera: Siempre que se da un Nombre singular y el verbo "ser", se aplica la transformación del verbo a "es". Resulta fácil la extensión de esta transformación para todos los otros casos de concordancia entre nombre y verbo en español.

En general, dado un conjunto O de oraciones generadas por la gramática G_o, y dado un conjunto T_i de transformaciones, obtenemos un nuevo conjunto O_i de oraciones, tales que T_i (O)= O_i. Las transformaciones pueden llevar a cabo varios cambios en las oraciones de entrada. Primero, borrar piezas léxicas, sustituir elementos de la oración por otros, insertar nuevas piezas léxicas o invertir su orden (como en el caso de las pasivas). Para una presentación formal de las transformaciones véase Peters y Ritchie (1973).

Chomsky habla de dos tipos de capacidad generativa. La primera de ellas, la capacidad generativa débil, indica que el conjunto de reglas propuestas (sintácticas, transformacionales y morfo-fonológicas) corresponden con un corpus determinado de una lengua específica. Sin embargo, esta generación no implica que sea considerada como la gramática adecuada a esa lengua. Por ello, se requiere la adecuación generativa fuerte que es la discriminación de la mejor gramática y con mayor poder explicativo. La expectativa de Chomsky es que una gramática de estructura *frasal* y un componente transformacional proporcionarán explicaciones profundas sobre las gramáticas de los lenguajes naturales.

Como una nota al margen, en 1973, Peters y Ritchie probaron que las gramáticas transformacionales tienen la misma capacidad generativa que una gramática sensible de contexto, con lo cual el objetivo de alcanzar gramáticas con alto grado de explicación se ven fuertemente limitadas. De hecho, Chomsky mismo propuso un nuevo formalismo conocido como "gobierno y ligamiento" en los últimos años de la década de los 70 e inicios de los 80.

Pero será en la publicación de Chomsky de 1959 "On Certain Formal Properties of Grammars" donde vemos un gran avance en la comprensión de las propiedades matemáticas de las gramáticas y su clasificación según la capacidad generativa, es decir, según el tipo de lenguajes que pueden ser generados por cada una de las clases de gramáticas. La estrategia utilizada por Chomsky es analizar el tipo de restricciones que pueden ser aplicadas a las reglas de las gramáticas. Sea Σ = {a, b, c,…} el vocabulario terminal y sea Q = { A,B,C,…} el vocabulario no terminal, denotaremos con Σ^* al conjunto de concatenaciones que podemos obtener a partir de Σ, es decir, Σ^* = {ɜ, a,…, ab, ac,…..}, donde ɜ representa la hilera vacía, y sea Q* construido de la misma manera que Σ^*, entonces, para ϕ y $\psi \in \Sigma^* U Q^*$. Una regla toma la siguiente forma:

1. $X \phi Y \rightarrow X \psi Y$. Se lee de la siguiente manera: en el contexto X _ Y, ϕ se reemplaza o reescribe como ψ.

Obtenemos la clase más general de gramáticas si no imponemos ninguna restricción sobre X, Y, ϕ o ψ, es decir, estos pueden ser incluso la hilera vacía, pero también, si ϕ = BC, podría darle el caso de que ψ = CB, o ψ = B, es decir, elimina la C, pero también ψ =

BBϕCC. Como puede observarse las operaciones que pueden realizarse sobre una hilera son de muy diverso tipo, pero corresponden a aquellas realizadas por una máquina de Turing. En efecto, estas gramáticas, denominadas irrestrictas, tienen la misma capacidad que estos dispositivos mecánicos generales. Son gramáticas tipo 0 (G_{t0}).

Obtenemos un segundo tipo de gramáticas, las tipo 1, si imponemos a la regla anterior la siguiente restricción:

2. $X \phi Y \rightarrow X \psi Y$, siempre que $\phi \in Q^+$ y $X, Y, \psi \in \Sigma^+ U\ Q^+$ y $|\phi| \leq |\psi|$. Es decir, no se da el caso de que X, Y, ϕ o ψ sean vacíos, y al menos la longitud de ϕ tiene que ser igual a ψ.

Este tipo de gramáticas son denominadas tipo 1 o gramáticas sensibles a contexto (G_{t1}). Claramente, constituyen un subconjunto propio de G_{t0}, es decir, $G_{t1} \subset G_{t0}$, aun cuando Chomsky establece una condición más débil y es que sea simplemente subconjunto. Podemos obtener una nueva clase de gramáticas si imponemos varias restricciones, comenzando con la eliminación del contexto en el que se debe llevara a cabo la reescritura, es decir, eliminamos X e Y. Las reglas de esta clase de gramáticas es la siguiente:

3. $\phi \rightarrow \psi$, $\phi \in Q$ y $\psi \in \Sigma^+ U\ Q^+$, además $|\phi| < |\psi|$. Este tipo de gramáticas son conocidas como gramáticas libres de contexto. Podemos obtener dos formas normales, si imponemos algunas condiciones adicionales. Una primera, si establecemos que $\psi \in \Sigma X\ Q^+$, es decir, el símbolo terminar debe aparecer al lado izquierdo de los no terminales en ψ. Se conoce como gramáticas en forma normal de Greibach. La segunda forma normal se debe al mismo Chomsky y es que $\psi \in Q^+$, o $\psi \in \Sigma$.

Finalmente, podemos obtener una clase más restringida si permitimos reglas únicamente de la siguiente forma:

4. $\phi \rightarrow \psi$ donde $\phi \in \Sigma$ y $\psi \in \Sigma X\ Q$ o $\psi \in \Sigma$. Es decir, las reglas de este tipo de gramáticas es de la forma $A \rightarrow aB$ o $A \rightarrow a$, como ya indicamos. La gramáticas tipo 3 o regulares son también de dos tipos: a) de ramificación a la derecha, como la indica, pero también, puede ser de ramificación de izquierda, si contiene reglas de la forma, $A \rightarrow Ba$ o $A \rightarrow a$

Está clasificación de las gramáticas y de los lenguajes generados tuvo enormes consecuencias para el desarrollo de la computación, sobre todo en la construcción de lenguajes de programación como indicaremos más adelante.

Los primeros tres años de la década de los 60.

Nuestra historia concluye con la completación de la jerarquía de Chomsky que vimos aparecer en el artículo de Chomsky de 1959, en relación a la clasificación de las gramáticas y su correspondencia con algunas clases de autómata. Como se mencionó anteriormente, Mealy y Kleene lograron una adecuada conceptualización de los autómatas de estado finito y su relación con el tipo de álgebras que puede generar (expresiones regulares). De hecho, Chomsky mismo hace referencia a los autómatas de estado finito (procesos de estado finito de Markov), indica claramente que era conocido de los investigadores del campo esta equivalencia. Lo mismo podemos decir de las máquinas de Turing, de las funciones recursivamente enumerables y de las gramáticas irrestrictas, cuya equivalencia vemos que Chomsky ya establece. Únicamente nos quedan dos resultados: a) la equivalencia entre las gramáticas libres de contexto y los autómatas con una pila de memoria ya sugerida por Chomsky en 1959, y b) la equivalencia entre las gramáticas dependientes de contexto y los autómatas de pila linealmente acotadas.

El primer desarrollo en este sentido fue realizado por Minsky en 1961, en el que introduce los autómatas de estado finito con dos pilas de memoria como una variedad de máquinas de Turing. En 1962 Chomsky y Schutzenberger establecen uno de los teoremas más importantes de las gramáticas libres de contexto, pero al mismo tiempo se establece la equivalencia entre gramáticas libres de contexto y autómatas con una pila de memoria. En ese mismo año (aunque publicado en año siguiente), Evey (1963) establece también esta equivalencia.

De esta manera, siguiendo a Book (1976), p.1, podemos establecer el teorema Chomsky-Schützenberger de la siguiente manera: Para todo lenguaje libre de contexto L, existe un conjunto regular R, y homomorfismos h_1 y h_2, con h_1 preservando longitud, tal que $L = h_1 (h_2^{-1}(D_2) \cap R)$, donde (D_2) es un conjunto Dick sobre dos letras. Varias consecuencias derivan de este teorema. Primero, siempre podemos encontrar un conjunto que sea anidadamente balanceado, y ésta es precisamente una de las propiedades de los lenguajes libres de contexto. Segundo, que esta característica de auto-anidamiento no puede ser expresada o aceptada por un autómata de estado finito, pero sí si agregamos un mecanismo de almacenamiento que lleve el control de los anidamientos (una pila de memoria), Evey (1963). Tercero, una propiedad de la ya hemos hecho referencia y es que toda gramática libre de contexto puede convertirse en una que sea auto-anidada, excepto en aquellos casos en los que no se tengan ciclos repetidos. Cuarto, el que podamos construir gramáticas no ambiguas para determinados lenguajes. Este aspecto es fundamental pues podemos hacer afirmaciones muy fuertes sobre la naturaleza de una lengua o para la construcción

de analizadores sintácticos de lenguajes de programación. En efecto, son los únicos formalismos en los que se puede garantizar univocidad en la derivación de una hila u oración, de manera que se convierten en las herramientas más potentes en cuestiones de predictividad. Un rasgo realmente muy valorado, tanto científica como filosóficamente.

La equivalencia entre los lenguajes libres de contexto y los autómatas de pila forma ya parte de los textos normales en teoría de la computabilidad. Una buena referencia es la obra de Aho, Hopcroft y Ullman (1979). Es usual utilizar la forma normal de Greibach que, como se recordará, las reglas de producción inician con un terminal seguido de un conjunto de vacío o no de no terminales. Solo se requiere una pila para simular el comportamiento de una gramática libre de contexto. Ya para 1963 se ha logrado completar la jerarquía de Chomsky con lo cual tenemos una serie de equivalencias y relaciones de inclusión entre lenguajes y autómatas. Por un lado, es equivalente formular un lenguaje en términos algebraicos, o en términos de la gramática que lo genera, o bien en términos del autómata que lo acepta. Por otro lado, los lenguajes pueden ser jerarquizados dependiendo de su capacidad generativa, comenzando por los lenguajes regulares, le siguen los libres de contexto, los dependientes de contexto y los irrestrictos.

Las implicaciones de la jerarquía de Chomsky para el desarrollo de la computación han sido realmente extraordinarias. La jerarquía permitió que la construcción de lenguajes de programación con propiedades matemáticas conocidas, compiladores e intérpretes, en particular, los lenguajes de programación tendieran a utilizar dos tipos de autómatas: los autómatas de estado finito para llevar a cabo el reconocimiento léxico (la formación de *tokens*) propios del lenguaje de programación y los autómatas de pila como analizadores sintácticos, es decir, para analizar las expresiones bien formadas del lenguaje.

BIBLIOGRAFÍA

Aho, A., Hopcroft, J. & Ullman, J.D. (1979): *Introduction to Automata Theory, Languages, and Computation.* Reading, MA: Pearson-Addison-Wesley.

Angus, A. & Kozen, D. (2001): "Kleene algebra with tests and program schematology". Technical Report TR 2001-1844, Cornell University: Department of Computer Science.

Book, R. V. (1976): "On the Chomsky-Schützenberger Theorem". Research Report #33. University of Yale Publications (http://www.cs.yale.edu/publications/techreports/tr33.pdf).

Chomsky, N. (1959) "On Certain Formal Properties of Grammars", *Information and Control* 2, pp. 137-167

Chomsky, N. (1957) Syntactic Structures. The Hague, The Netherlands: Mouton & Co.

Chomsky, N. (1956) Three Models for the Description of Language. I. R. E. Transactions on information theory, Proceedings of the Symposium on Information Theory, volume 2, Nº 3, pp. 113–124, September.

Evey R. J. (1963): "Applications of pushdown store machines". Proceedings of the AFIPS–Fall Joint Computer Conference, Montreal 1963: AFIPS Press.

Henriksson, S. (2010): "A brief history of the stack". Conference Paper. Pittsburg 2009, Sigcis workshop.

Irvine, A.D. (1996): "Philosophy of logic". En Parkinson and Shanker (1996); pp 9-49.

Kleene, S. (1951/6): Representation of events in nerve nets and finite automata. Memoria del Proyecto RAND. USA (Automata Studies (Shannon, C. ed.). *Annals of Mathematics Studies* 34, 1956, pp. 3-41).

McCulloch & Pitts (1943) "A Logical Calculus of the Ideas Immanent in Nervous Activity" en: *Bulletin of Mathematical Biophysics*, Vol. 5; pp.115-133.

Mealy, G. (1955) "A Method for Synthesizing Sequential Circuits", en *The Bell System Technical Journal*, Septiembre 1955.

Miller, G. & Chomsky, N. (1963): "Finitary models of language users". R. D. Luce, R. Bush, E. Galanter, eds. Handbook of mathematical psychology, vol. 2, pp. 419-491. New York: Wiley.

Minsky, M. (1961) "Recursive Unsolvability of Post's Problem of "Tag" and other Topics in Theory of Turing Machines". *The Annals of Mathematics*, 2nd Ser., Vol. 74, No. 3. (Nov., 1961), pp. 437-455.

Moore, E. (1956): "Gedanken-experiments on sequential machines", *Automata Studies*, 34, pp. 129-153, Princeton 1956,, N. J.: Princeton University Press.

Newell, A., Shaw, J.C., & Simon, H.A (1959) "A Variety of Intelligent Learning in a General Problem Solver". *Self Organizing Systems*, Yovits & Cameron (Eds.), p.153: Pergamon Press.

Newmeyer, F. (1980): *Linguistic Theory in America. The First Quarter Century of Transformational Generative Grammar.* New York: Academic Press.

Parkinson and Shanker eds. (1996): *Philosophy of Science, Logic and Mathematics in the Twentieth Century*. Routledge History of Philosophy, vol IX, London: Routledge.

Peters. P.S. & Ritchie, R.W. (1973): "On the Generative Power of Transformational Grammars" *Information Sciences* 6, pp. 49-83.

Savage, J. E. (1998): *Models of Computation. Exploring the Power of Computing*. Reading, MA: Addison-Wesley.

Sayre, K, M. (1996): "Cybernetics". En Parkinson and Shanker (1996), pp. 292-314.

Vargas, C. (2012): "Alan Turing: Máquinas e Inteligencia. En conmemoración de los 100 años de su nacimiento". *Aporía · Revista Internacional de Investigaciones Filosóficas*, Nº4 (2012), pp. 43-63.