Praxis Filosófica
ISSN: 0120-4688
ISSN: 2389-9387
Universidad del Valle;;

Mota, Sergio

Dos concepciones del lenguaje: Wittgenstein y Chomsky en torno a la recursión como “buena” explicación de la naturaleza humana

Praxis Filosófica, núm. 46, 2018, Enero-Junio, pp. 125-149
Universidad del Valle;;

DOI: 10.25100/pfilosofica.v0i46.6154

DOS CONCEPCIONES DEL LENGUAJE: WITTGENSTEIN Y CHOMSKY EN TORNO A LA RECURSIÓN COMO "BUENA" EXPLICACIÓN DE LA NATURALEZA HUMANA

Sergio Mota
Universidad Autónoma de Madrid

Resumen

En este artículo me propongo discutir si la propiedad de la recursión supone una buena explicación de la especificidad humana. Para ello, analizaré dos aproximaciones al estudio del lenguaje humano: las concepciones computacional y antropológica. La principal conclusión de este trabajo es doble. Por un lado, la recursión no es una buena explicación de la esencia humana. Por otro, lo que sí es específicamente humano es la construcción de una mitología, análoga a la metafísica, revestida en este caso con el lenguaje de la ciencia.

Palabras clave: *Wittgenstein; Chomsky; lenguaje; recursión; concepción computacional; concepción antropológica.*

ISSN (I): 0120-4688 / ISSN (D): 2389-9387

Recibido: 12 de diciembre de 2016. **Aprobado:** 25 de noviembre de 2017.

Two conceptions of language: Wittgenstein and Chomsky on recursion as a "good" explanation of human nature

Abstract

My major aim in this paper is to discuss whether the property of recursion provides a good explanation of human specificity. In so doing, I will analyze two approaches to the study of natural language: the computational and the anthropological conceptions. The main conclusion of this work is twofold. On the one hand, I argue that recursion is not a good explanation of human essence. On the other hand, what is, indeed, specifically human is the construction of a mythology with a metaphysical slant, in this particular case, in the guise of the language of science.

Keywords: *Wittgenstein; Chomsky; Language; Recursion; Computational Conception; Anthropological Conception.*

Sergio Mota. Licenciado en Psicología por la Universidad Autónoma de Madrid. Máster en Crítica y argumentación Filosófica por esa misma Universidad, donde también ha obtenido el título de Doctor en Psicología, con una tesis sobre el papel de la recursión en la ciencia cognitiva del lenguaje, llevando a cabo tanto una investigación formal como experimental.

E-mail: sergio.mota.v@gmail.com

DOS CONCEPCIONES DEL LENGUAJE: WITTGENSTEIN Y CHOMSKY EN TORNO A LA RECURSIÓN COMO "BUENA" EXPLICACIÓN DE LA NATURALEZA HUMANA

Sergio Mota
Universidad Autónoma de Madrid

I. Introducción

El presente artículo trata de llevar a cabo una investigación gramatical en el sentido wittgensteiniano del concepto de recursión. Tal investigación consiste en examinar cómo funciona un área de nuestro lenguaje y mostrar aquellos mitos construidos con base en falsas analogías y similitudes superficiales. Así, este trabajo analizará cómo funciona el concepto de recursión y mostrará que su empleo para caracterizar lo que nos hace humanos, i.e. su empleo en la explicación de la esencia humana, está basado en quimeras y mitos. Fundamentalmente en uno que ya ha sido analizado tanto por A. Tomasini (2003) como por P.M.S. Hacker (2014), en sendos artículos que llevaban por título *Dos concepciones del lenguaje* (*Two conceptions of language*). Mi trabajo comparte muchas ideas con estos autores, pero en vez de centrarme en el trabajo de Chomsky de los años 80, en la que tanto Tomasini como Hacker están inmersos, voy a hablar del actual *Programa Minimista* (véase Chomsky, 1995) y su relación con la llamada *Biolingüística* (véase Chomsky, 2006, capítulo 7).

Ciertamente, muchas de sus reflexiones críticas tienen cabida en este nuevo programa, pero también es cierto que dejan fuera aspectos importantes como el papel que Chomsky otorga a la propiedad de la recursión. Dicho papel es crucial, pues pretende explicar lo que nos hace humanos, i.e. la causa primaria de nuestra especificidad humana (cf. Frath, 2014). El objetivo principal de este artículo es, por tanto, analizar la relación entre el lenguaje, la recursión y nuestra naturaleza humana desde dos concepciones del lenguaje, que son, naturalmente, la concepción computacional, representada prominentemente, aunque no sólo, por Chomsky, y la concepción antropológica, adscrita principalmente a Wittgenstein.

No voy a entrar en señalar explícitamente semejanzas y desemejanzas entre Wittgenstein y Chomsky, pues creo que esto se puede encontrar en los trabajos tanto de Tomasini (2003) como de Hacker (2014), pero sí podrán observarse algunos de los paralelismos y diferencias a lo largo de este artículo, que se organiza como sigue: en la siguiente sección presentaré la noción de recursión y su relación con nociones como "computabilidad", "definición inductiva", "inducción matemática" y "auto-referencia", nociones que aparecen frecuentemente asociadas con la recursión pero que son nociones diferentes (i.e. tienen distintos sentidos o usos y se aplican a diferentes casos, con distintos fines). Con ello, quiero apuntar hacia la tercera sección, donde se explicará qué papel juega el concepto de recursión tanto en la definición del procedimiento mecánico finito formulado por Chomsky como en su explicación de la especificidad humana. En la cuarta sección introduciré la perspectiva wittgensteiniana y su concepción antropológica del lenguaje y de la matemática, lo cual permitirá ver el marcado contraste con la concepción chomskiana. En esta sección, también me haré cargo de algunas interesantes críticas realizadas hace ya tiempo en un texto clásico por A. Kenny (1989), y mostraré en qué medida tienen vigencia en la actualidad. Asimismo, presentaré un análisis crítico más actual acerca del uso de la recursión para postular una "buena" explicación acerca de la naturaleza humana. Por último, presentaré una serie de consideraciones finales.

II. La definición por recursión concebida como *técnica*.

De acuerdo con Odifreddi (2011, p. 330), la recursión, en su forma numérica más general, consiste en definir el valor de una función usando otros valores de la misma función. En una línea similar, Cutland (1980, p. 32) indica que la recursión caracteriza un método para definir una función especificando cada uno de sus valores en términos de valores previamente definidos, y posiblemente usando otras funciones previamente definidas. Pero ¿significa "recursión" aquí "definición por recursión" o, por el contrario, significa "computabilidad"? Creo que Soare ha sido uno de los autores que más empeño ha puesto en separar ambas nociones. Así, este autor indica que "recursión" debería significar "definición por recursión", por lo que una función f es recursiva, esto es, está técnicamente definida por recursión, si y sólo si está definida para un argumento x haciendo uso de sus propios valores previamente definidos (digamos, $f(y)$ para $y < x$), pudiendo emplear también otras funciones g más simples y generalmente previamente definidas (véase Soare, 1996). Así, el término 'recursión' se refiere a una función f para la que definimos primero f(0) y después definimos $f(x+1)$ en términos de funciones previamente definidas usando como argumentos (i.e. inputs) x y $f(x)$.

Por su parte, la noción de computabilidad hace referencia a una función o procedimiento capaz de ser verificado o determinado por medio de un proceso matemático de alguna complejidad (Soare, 2012, p. 248). Así, una función efectivamente computable f de números naturales a números naturales, i.e. cuyo *dominio* y *codominio* son los números naturales, es aquella para la cual existe, se ha estipulado, se ha establecido, una lista de reglas explícitamente definida que hace posible computar el valor $f(x_1,\ldots, x_n)$ para los argumentos $x_1,\ldots, x_n$ en un número finito de pasos. Así, una función recursiva es computable, mientras que la conversa, i.e. que una función es computable (si y sólo si) es recursiva, alude a la Tesis de Church.

Para contextualizar y tratar de precisar un poco más estas ideas generales, pensemos en el formalismo de la teoría de conjuntos Zermelo-Fraenkel con el axioma de elección (ZFC, la "C" se refiere a "*choice*" en la expresión inglesa "*the axiom of choice*"). Dentro de este formalismo, y de acuerdo con Hrbacek y Jech (1999, p. 40), el conjunto de los números naturales es un conjunto inductivo; i.e. es *el menor conjunto inductivo*. Un conjunto I se dice inductivo si: (a) $0 \in I$, y (b) si $n \in I$, entonces $n+1 \in \mathrm{I}$. Así, si el conjunto vacío ø pertenece a I, y siempre que un conjunto X pertenezca a I, entonces también el sucesor $S(X)$, obtenido de la unión de X con el conjunto que tiene como único elemento a X; i.e. $S(X) = X \cup \{X\}$. Esto es un ejemplo de definición inductiva, empleada para definir un dominio; en este caso el de los números naturales.

De acuerdo con Hrbacek y Jech (1999, p. 46 y ss.), el principio de recursión, o como muchos otros autores prefieren denominarlo, entre ellos Hrbacek y Jech, el teorema de recursión (véase Mac Lane, 1986; Enderton, 2009), es un método general para definir funciones. Este teorema indica que si X es un conjunto, $a \in X$ y g: $X \rightarrow X$ una función (i.e. que toma elementos del conjunto X como argumentos y devuelve como valores elementos del conjunto X), entonces existe una única función f: $N \rightarrow X$ (i.e. que toma elementos del conjunto N como argumentos y devuelve elementos del conjunto X como valores) tal que: (a) $f(0) = a$; (b) $f(n+1) = g(f(n))$ para todo

$n \in N$. Bien, pero esto no es sino una aplicación del esquema V, denominado *definición por recursión*.[1]

Hay otros esquemas por medio de los cuales podemos definir diferentes clases de funciones recursivas (i.e. las primitivas, las generales y las parciales). Veamos cuáles son esos esquemas.

Los esquemas iniciales se corresponden con las llamadas funciones iniciales o básicas (cf. Kleene, 1943, p. 42; Boolos, Burgess y Jeffrey, 2007, p. 64): (I) *función sucesor*: $\sigma(x) = x+1$, es una función unaria ($x+1$ suele expresarse como x'). (II) *función constante*: $\phi(x_1,\ldots, x_n) = c$, la función cero es un ejemplo de función constante, cuyo valor es siempre 0 para cualquier argumento $x_1,\ldots, x_n$. (III) *funciones identidad*: $\phi(x_1,\ldots, x_n) = x_i$, también conocida como la función de la *i-ésima* proyección (o función proyección).

A estas funciones iniciales se aplican otros procesos u operaciones, considerados también como *reglas* de inferencia o de transformación, con el fin de obtener otras funciones recursivas. Es el caso de los siguientes esquemas:

(IV) *definición por sustitución* (o *composición*):

$$\phi(x_1,\ldots, x_n) = \theta(\chi_1(x_1,\ldots, x_n),\ldots, \chi_n(x_1,\ldots, x_n)).$$

(V) *definición por recursión*:

(a) $\phi(0) = c,$

$\phi(y') = \chi(y, \phi(y)).$

(b) $\phi(0, x_1, \ldots, x_n) = \psi(x_1, \ldots, x_n),$

$\phi(y', x_1, \ldots, x_n) = \chi(y, \phi(y, x_1, \ldots, x_n), x_1, \ldots, x_n)$.[2]

---

[1] Alternativamente, la especificación de “recursiva” puede entenderse como sigue. Para ello, voy a seguir a Torretti (1998, p. 94) en su exposición de von Neumann. Éste define inductivamente “argumento” así: (a) *a* es un argumento, (b) si *a* es un argumento, entonces $<a,b>$ es un argumento, (c) si *a* y $<a,b>$ son argumentos, entonces el valor de *f* en *a* (i.e. $[f,a]$) es un argumento. Así, la especificación de recursiva alude al procedimiento de definir o computar un argumento (i.e. el valor de *f* en $n+1$) haciendo uso de los propios valores de *f* –por ejemplo, $f(n)$– previamente computados para argumentos menores (i.e. $f(n\text{-}1)$). Al fin y al cabo $f(n)$, el valor de *f* en *n*, es igual al valor de *g* para $f(n\text{-}1)$ como argumento [i.e. $f(n) = g(f(n\text{-}1))$].

[2] ϕ está definida por recursión con ayuda de ψ y χ si y sólo si para cualesquiera números naturales $y, x_1 \ldots x_n$ ocurre que: $\phi(0, x_1, \ldots, x_n) = \psi(x_1, \ldots, x_n)$; $\phi(y', x_1, \ldots, x_n) = \chi(y, \phi(y, x_1, \ldots, x_n), x_1, \ldots, x_n)$ (Mosterín, 2007, p. 377). En este caso, ejemplifico el esquema con Vb, pero sería extrapolable *mutatis mutandis* a todas las versiones del esquema V. Como señala Mac Lane (1986, p. 46), Va es una versión del teorema de recursión denominada “recursión primitiva”, mientras que Vb es una versión más elaborada, la llamada “recursión con parámetros”. El teorema se podría elaborar todavía más, y seguiría siendo verdadero, con la llamada “doble recursión”; añadiéndola a las anteriores versiones como Vc.

Así las cosas, la clase de las funciones recursivas primitivas se definen mediante los esquemas I-V. Las clases de las funciones recursivas generales y parciales[3] se cierran bajo los esquemas I-VI. El esquema (VI) es como sigue:

(VI) *minimalización* (que hace uso del operador μ, y que se lee "el mínimo…tal que"; cf. Kleene, 1943, p. 45):

$\phi(x_1, \ldots, x_n) = \mu y[\rho(x_1, \ldots, x_n, y) = 0]$.[4]

En el caso de la clase de las funciones recursivas generales, la función definida mediante el esquema VI tiene que cumplir la exigencia de que tiene que tener al menos una solución.[5] Al relajar esta exigencia tenemos las funciones recursivas parciales, pero en este caso una función definida mediante VI no tiene que tener necesariamente una solución (véase Torretti, 1998, para más detalles). Como puede apreciarse, todas las clases hacen uso de la definición por recursión (lo cual *no implica* que lo hagan todos los miembros de cada clase).

Por otro lado, la inducción matemática es el método por medio del cual se pueden probar propiedades de los números naturales y de las funciones definidas para operar sobre ellos. Su esquema general es como sigue (véase Torretti, 1998, p. 506): (a) *el caso base*: la tesis se prueba para $n = 0$; (b) *el paso inductivo*: se prueba que si la tesis es válida para $n \in N$, entonces también es válida para el elemento siguiente $s(n)$. Una manera de expresar este esquema es como sigue: $[P0 \; \& \; (n) \; [Pn \rightarrow PSn] \rightarrow (n)Pn]$. El principio de inducción es empleado, también, para probar ciertas propiedades de las funciones definidas por recursión, como la propiedad conmutativa (ver *infra*). Veamos un ejemplo, pero integrando todo lo que hemos visto hasta ahora. Para ello voy a emplear la llamada estructura inductiva de datos de los números naturales, que se puede representar mediante el siguiente esquema:

[3] Una función parcial es aquella cuyo dominio, i.e. el conjunto de sus argumentos, es algo menor que el conjunto total A (i.e. $\mathrm{dom}(f) \subseteq A$; por ejemplo, si sólo está definido un subconjunto de A). Una función total (como las funciones recursivas primitivas y generales) es aquella cuyo dominio es todo el conjunto A (i.e. $\mathrm{dom}(f) = A$).

[4] Una función ϕ está definida por minimalización a partir de una función ρ en caso normal si y sólo si para cada $x_1, \ldots, x_n$ existe al menos un y tal que $\rho(x_1, \ldots, x_n, y) = 0$, y para cualesquiera números naturales $x_1, \ldots, x_n$ ocurre que: $\phi(x_1, \ldots, x_n) = \mu y[\rho(x_1, \ldots, x_n, y) = 0]$ (cf. Mosterín, 2007, p. 379).

[5] Como señala Mosterín (2007, p. 379), la conocida función de Ackermann es una función recursiva (general); esto es, es una función definible a partir de las funciones iniciales (esquemas I-III) por medio de un número finito de aplicaciones de definiciones por sustitución (esquema IV), por recursión (esquema V), y por minimalización en caso normal (esquema VI). Un ejemplo de tal función es como sigue: $f(0, y) = S(y)$; $f(x+1, 0) = f(x, 1)$; $f(x+1, y+1) = f(x, f(x+1, y))$. La función que se define es f, con ayuda de S, la función sucesor.

Sintaxis	**_Axiomas_**
zero: Nat	_iszero(zero) = true_
suc: Nat→Nat	_iszero(suc(x)) = false_
iszero: Nat→Bool	_pred(suc(x)) = x_
pred: Nat→Nat	_sum(x,y) = if(iszero(x), y, suc(sum(pred(x),y)))_[6]
sum: Nat×Nat→Nat	_prod(x,y) = if(iszero(x), zero, sum(x,prod(pred(x),y)))_
prod: Nat×Nat→Nat	

De acuerdo con Bertot y Komendantskaya (2009), esta estructura inductiva de datos se construye mediante la introducción de dos *constructores* (que son funciones), *zero* y *succesor*, los cuales generan elementos del nuevo tipo. Además de los constructores, hay dos *operadores* (dos funciones recursivas primitivas), la *suma* y el *producto*. Así, esta estructura inductiva de datos consiste en un conjunto de elementos y un conjunto de operaciones. Así, una vez que el tipo *inductivo* está definido, podemos definir por recursión funciones que operan sobre el conjunto de elementos inductivamente definido (sean números, listas, árboles). Queda claro, por tanto, que la recursión se reserva a la definición de funciones.[7]

Bien. Aplicando la definición anterior del método de la inducción matemática, uno puede probar la tesis T de que si m y n son cualesquiera elementos de N, entonces $m + n = n + m$:

Base: $m + 0 = 0 + m$.

Paso inductivo: si $m + n = n + m$, entonces $m + (n+1) = (n+1) + m$:

$m + (n+1) = (m+n)+1 = (n+m)+1 = n + (m+1) = n + (1+m) = (n+1) + m$.

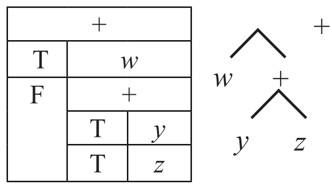
Podríamos seguir señalando que si $m + (n+1) = (n+1) + m$, entonces $(m+1) + (n+1) = (n+1) + (m+1)$; véase Hrbacek y Jech (1999, p. 53) y Boolos et al. (2007, p. 213) para más detalles.

En este sentido, es más adecuado hablar de estructura inductiva que de estructura recursiva dado que, aunque a veces se empleen de manera intercambiable, la definición inductiva y la definición por recursión son ciertamente diferentes. Por otro lado, en algunos trabajos en ciencia de la computación (i.e. *Computer Science*) se define y ejemplifica una estructura recursiva de datos de una manera distinta, *pace* las semejanzas, a la

[6] En algunos ejemplos, el caso base de la definición de la función suma o adición es: a+1 = a′ y el paso recursivo es: a+n′ = (a+n)′. Así, la definición de la función suma *presupone* la función sucesor como función básica y *previamente definida*.

[7] Es posible encontrarse en la literatura que “esquema de inducción” (*induction scheme*, en inglés), se refiera tanto al esquema de recursión como al esquema de inducción matemática. Para evitar malentendidos innecesarios, recomiendo emplear “esquema de recursión” (i.e. *recursion scheme*) cuando hablemos del método para definir funciones, mientras que se puede emplear la noción de esquema de inducción cuando se hable de la inducción matemática, como hacen, por ejemplo, Boolos et al. (2007, p. 214).

indicada más arriba; un uso que, en mi opinión, induce a error. Así, Wirth (1986) ha indicado que hay una analogía entre una estructura recursiva y un procedimiento recursivo, en cuanto valores (i.e. componentes) de la estructura contienen uno o más componentes (i.e. otros valores) que pertenecen a la misma estructura de datos (lo que en ocasiones se ejemplifica con listas dentro de listas o árboles dentro de árboles); en analogía con los procedimientos que contienen una o más llamadas de sí mismos. Un ejemplo de tal estructura es, de acuerdo con Wirth:



Esta estructura subyace a expresiones como $w + (y + z)$. En otras ocasiones se dice que una estructura es recursiva porque se menciona ella misma en su propia definición, en analogía con las funciones recursivas. Así, el tipo de dato inductivo de los números naturales sería recursivo porque se menciona en su propia definición como sigue: *zero*: *Nat; suc*: *Nat→Nat* (recuérdese que estos eran los dos constructores). Sin embargo, creo que estos son buenos ejemplos de lo saturada y recargada que está la noción de recursión, lo que lleva a proponer analogías engañosas. La noción de recursión se aplica, en todos estos casos, a la definición de funciones, y no hay que confundirla ni con la organización interna de expresiones particulares, lo que también lleva a confundirla con la auto-inclusión (i.e. constituyentes dentro de constituyentes del mismo tipo, sean esquemas, árboles,…), ni con la definición inductiva.[8]

La última distinción que voy a mencionar es sobre recursión y auto-referencia. Estas nociones no deberían identificarse sin más, i.e. no sería adecuado decir que un X es recursivo porque es auto-referente. Si hay un aspecto auto-referente en la recursión cuando definimos el valor de una función para un argumento dado usando sus propios valores previamente definidos para argumentos menores (i.e. se usan *sus propios* valores), hay que tener presente que una definición más técnica de función recursiva (total o parcial) alude a las funciones que son caracterizadas mediante una cadena de definiciones por composición, recursión y minimalización, partiendo de

[8] De acuerdo con Kleene (1952, p. 217), la definición inductiva y la definición por recursión son dos métodos *análogos* pero *diferentes*. Así la definición inductiva *justifica* tanto la definición recursiva de una función sobre un dominio establecido mediante la definición inductiva, como las formas correspondientes de las pruebas por inducción matemática, aunque no son sinónimos (véase Kleene, 1952, pp. 258-261).

funciones básicas iniciales. Si vemos un aspecto auto-referente aquí, es claro que no es suficiente para hablar de recursión, aunque pueda ser necesario. En todo caso, hay funciones recursivas iniciales en las que no hay auto-referencia, por tanto es discutible que sea incluso necesaria.

Es fácil notar, por otro lado, que no hay nada de paradójico en una función recursiva, o dicho en términos más generales, una definición recursiva, precisamente por tener base en una definición inductiva (pensemos por ejemplo en los números naturales) no es ni circular ni paradójica, algo que queda oscurecido al afirmar que una función es recursiva porque se define en términos de sí misma, o al señalar que la recursión es la propiedad auto-referente que subyace a diferentes clases de funciones. Estas formas de expresión son, aunque muy usadas, imprecisas e incorrectas.

La auto-referencia está presente en las llamadas paradojas de auto-referencia, como la paradoja de Russell y su eliminación o bloqueo no tiene nada que ver con eliminar la recursión. Además, hay otros ejemplos mucho más triviales de auto-referencia, como por ejemplo “esta oración tiene cinco palabras”, en donde ninguna función recursiva es computada.

En este sentido, podemos trazar de forma general, aunque esencial, una distinción entre recursión y auto-referencia, que atañe al hecho de que la recursión es *aparentemente circular*, mientras que otros fenómenos auto-referenciales, como la paradoja recién mencionada, pueden llevar a círculos viciosos. Así, y de acuerdo con Ferreirós (2006, pp. 40-41), un círculo vicioso nunca nos hace avanzar, “[p]ero si el movimiento de revolución vuelve sobre un plano distinto, si se trata de una hélice, entonces tiene lugar un avance”.

Por otra parte, las funciones recursivas pueden emplearse para modelar casos de auto-referencia, pero tenemos que distinguir entre ambos. Así, con la ayuda tanto de la aritmetización de la sintaxis[9], como con la del teorema de puntos fijos (también conocido como el lema diagonal), se pueden modelar fenómenos auto-referenciales.[10] Un caso conocido es la famosa proposición de Gödel para $T{:}\vdash_T G \leftrightarrow \neg Bew(\ulcorner G \urcorner)$. Esta proposición dice de sí misma que es indemostrable (Gödel, 1931, p. 9), lo que constituye una instancia específica de auto-referencia (si bien no es paradójica). En este caso, G puede ser visto como un punto fijo cuando x es substituido por G en $\neg Bew(\ulcorner x \urcorner)$, y el teorema de puntos fijos es usado para mostrar que un punto fijo para G es indemostrable en una teoría consistente. Así, el predicado $Bew\ x$ no

---

[9] Una técnica introducida por Gödel (1931) que consiste en codificar expresiones de un sistema formal T por medio de números.

[10] Un punto fijo para f es cualquier x tal que $f(x) = x$; de tal suerte que x permanece igual, sin variación, bajo f.

puede ser capturado (aunque sí es expresado) en *T*; i.e. no es recursivo.[11] Por tanto, no hay ninguna función recursiva primitiva ni ningún otro tipo de procedimiento algorítmico para decidir qué es un teorema de *T*.

El teorema de puntos fijos también es empleado en la prueba del teorema de Tarski sobre la indefinición de la verdad (véase Enderton, 2009, p. 236; Odifreddi, 2011, p.347), pero con el objetivo de mostrar que el predicado veritativo no es expresable en ningún sistema aritmético consistente y suficientemente potente (i.e. el predicado veritativo no es ni efectivamente numerable ni recursivo).

Por tanto, (I) la razón por la que una función es recursiva no es la auto-referencia, (II) dado que la auto-referencia es una noción más amplia que la recursión, hay fenómenos auto-referentes que no son recursivos y (III) la afirmación de que X es recursivo por la razón exacta de que X contiene una auto-referencia es falsa.

El empeño puesto en esta sección también tiene que ver con el hecho de que Chomsky ha reconocido en distintos trabajos, como se verá en la sección siguiente, la importancia de hacer uso de estas herramientas conceptuales y técnicas para el estudio, ni más ni menos, que del lenguaje natural.

III. La concepción computacional del lenguaje: Chomsky

Chomsky ha insistido, en distintos momentos de su carrera, en la importancia de la teoría de la computabilidad y la lógica matemática (véase por ejemplo Chomsky, 1955/1975, p. 39), dado que, al menos desde un punto de vista conceptual, la teoría de la gramática puede ser vista como el estudio de una clase especial de funciones recursivas (Chomsky, 1959, p. 143). En esta línea, Chomsky (2007a) indica que uno de los principales factores que han impulsado el desarrollo de la Biolingüística ha sido precisamente el trabajo realizado dentro de la teoría de la computabilidad, que ha permitido estudiar más seriamente los mecanismos formales de la gramática generativa. En un trabajo reciente, ha señalado que el estudio del lenguaje-I, entendido como un sistema dotado de infinitud discreta, cae

[11] Se dice que el predicado *Bew x* es recursivamente numerable (i.e. el rango de una función recursiva parcial), pero no es recursivo (cf. Boolos et al., 2007). De acuerdo con Post (1944, p. 306), un conjunto de enteros positivos es recursivamente numerable, i.e. computablemente o efectivamente numerable, si hay una función recursiva (total o parcial) $f(x)$, de una variable para entero positivo, cuyos valores, para enteros positivos como valores de x, constituyen tal conjunto, i.e. es el rango de $f(x)$. Por otro lado, un conjunto recursivo (i.e. decidible) S es aquel para el que se ha estipulado un procedimiento efectivo de decisión (i.e. una función recursiva total), que permite decidir si un entero positivo n pertenece o no a S (cf. Post, 1944, p. 311). Así, un conjunto recursivo es efectivamente numerable, mientras que la conversa no se sostiene, esto es, hay conjuntos efectivamente numerables que no son recursivos.

dentro de la teoría de la computabilidad, i.e. máquina de Turing, funciones recursivas,…(Chomsky, 2015).

Veamos en qué consiste este estudio dentro del *Programa Minimista*. En este programa, Chomsky (1995) expurga conceptualmente sus propuestas anteriores y establece que un lenguaje-I es un procedimiento computacional, *alias* 'sistema generativo' (i.e. un algoritmo), que genera recursivamente una ordenación (o serie) infinita de expresiones jerárquicamente estructuradas mediante la operación denominada Merge (cf. Chomsky, 1995, p. 226). Como es bien sabido, las expresiones u objetos sintácticos generados por Merge son *interpretados* por dos sistemas, a saber: el sensorio-motor y el conceptual-intencional (véase para más detalles Hauser Chomsky y Fitch, 2002). Así, Merge puede entenderse como una función gramatical que opera sobre un dominio de objetos sintácticos (cf. Mota, 2014, 2015a, 2015b). Es el momento de aplicar las nociones de definición inductiva y definición recursiva precisadas en la sección anterior para definir el dominio y la función, respectivamente.

Así, la definición inductiva de "objeto sintáctico" es como sigue: (a) el ítem léxico α es un objeto sintáctico; (b) si α es un objeto sintáctico, entonces el par $\{\alpha, \beta\}$ es un objeto sintáctico; (c) el objeto $K = \{\gamma, \{\alpha, \beta\}\}$ es un objeto sintáctico, donde γ es la etiqueta (i.e. label) de K en el ejemplo de Chomsky (1995, p. 243), pero que en versiones posteriores es omitida; (d) todo objeto sintáctico satisface (a), (b), o (c).

Una vez definido el dominio, presentaré la definición recursiva de Merge. Podemos expresar su forma general de la siguiente manera: $[N, O_s, M(O_s)]$ (véase Mota, 2015a, 2015b). N hace referencia a las piezas léxicas (o ítems léxicos) y objetos sintácticos que configuran la numeración (o repertorio seleccionado de objetos que entran en una derivación sintáctica), O_s hace referencia a un objeto sintáctico cualquiera de la serie generada (que puede tomar n-valores: si $n = 1$, entonces $O_s = X$; si $n = 2$, entonces $O_s =$ X, Y, y así sucesivamente) y $M(O_s)$ hace referencia a un objeto sintáctico nuevo generado a partir de O_s mediante la aplicación de M ( ) –esto es, *Merge*. Como he indicado en otros trabajos, este procedimiento genera recursivamente objetos sintácticos, independientemente de su organización interna –esto es, presenten o no auto-inclusión–. Tal forma general puede definirse recursivamente como sigue: $M^{(0)}{}'(O_s) = O_s$, Def; $M^{(n)}{}'(O_s) = M'M^{(n-1)}{}'(O_s)$ Def. En su forma binaria, Merge toma dos objetos sintácticos como argumentos (por ejemplo X,Y) y devuelve otro objeto sintáctico (por ejemplo Z) como valor (i.e. M: $Os \times Os \rightarrow Os$).

En un trabajo reciente, Chomsky (2014, p. 1), señala que podemos pensar en la recursión como enumeración de objetos discretos por un procedimiento computable finitario *P*. Así, tomando tal procedimiento recursivo *P* como una función sobre enteros positivos, tenemos que su rango, i. e. $R = \{P(n)\}$, es el conjunto enumerado por *P*. En este caso, el conjunto de objetos discretos es efectivamente numerable. Si, además, Merge permite decidir en un número finito de pasos si una expresión es o no gramatical, entonces el conjunto es decidible (i.e. recursivo), y Merge puede ser considerada como una función recursiva total.

Por otro lado, Chomsky (2007b, p. 6; 2008, p. 139) ha insistido en que el modo de operar (cómo procede, o puede implementarse de manera abstracta) de este procedimiento computacional consiste en aplicar iterativamente la operación Merge. Esta distinción entre una definición recursiva y una implementación iterativa tiene también un reflejo en las ciencias formales. Siguiendo a Moschovakis (1998, 2009; cf. Moschovakis y Paschalis, 2008) en la distinción entre *iteradores* y *recursores*, podemos decir que una máquina de Turing (véase Turing, 1936) es un iterador (i.e. un modelo de implementación o de computación; un tipo especial de algoritmo), lo cual no significa que no pueda definirse recursivamente.[12] Veámoslo con más detalle.

Iterador	***Recursor***
Para dos conjuntos X y Z un iterador ϕ: X→Z queda definido por la siguiente quíntupla: (i, S, σ, T, o); donde: (1) i: X → S es la función *input* y toma los inputs desde X. (2) S es un conjunto no vacío, el conjunto de los estados de la máquina. (3) σ es la función transición de un estado a otro. (4) T es el conjunto de los estados terminales. (5) o: T → Z es la función *output*, que devuelve el output computado por la máquina.	Un recursor α: X→W puede ser definido desde un conjunto parcialmente ordenado (i.e. *partially ordered set*; *poset*) X a un conjunto W, como sigue: (D, map_{TA}, V); donde: (1) D es un conjunto inductivo parcialmente ordenado, el dominio de α. (2) map_{TA} es la función transición de α. (3) V es una función parcial que devuelve el valor de α.

Por una parte, la computación del iterador ϕ para un input x dado, $x \in$ X, es una secuencia de estados que puede ser definida recursivamente como sigue: $S_0(x) = x$ Def., $S_{n+1}(x) = S_n(x)$, si $S_n(x) \in T$ (i.e. si es el estado

[12] De acuerdo con Gödel (1964, pp. 71-72), al trabajo de Turing se debe una adecuada definición del concepto general de sistema formal, así como un análisis del concepto de procedimiento mecánico (*alias* "algoritmo" o "procedimiento computacional"). De acuerdo con Shanker (1987), un algoritmo puede concebirse como una secuencia definida de reglas (operaciones) que especifican cómo producir un resultado (*output*) desde un *input* en un número finito de pasos.

final) Def.; $S_{n+1} = \sigma(S_n(x))$ Def. (i.e. si la secuencia de estados continúa; véase Mota, 2015c, para más detalles). Así, la definición recursiva del iterador ϕ no debe confundirse con que su implementación sea iterativa (cf. Luuk y Luuk, 2011 en esta misma línea). Por otra parte, y de modo ilustrativo, las funciones recursivas expuestas en la sección precedente son recursores, i.e. una descripción recursiva construida tomando como base un conjunto de operaciones tomadas como primitivas (Mota, 2015c, p. 322). Así, para Moschovakis (2009), la teoría de los algoritmos es la teoría de las funciones recursivas. Desde luego, esto último es compatible con la definición de Merge, que no sólo hace uso del esquema V, sino que toma como funciones básicas y previamente definidas a la función identidad y a la función predecesor.

Hasta aquí, hemos definido un cálculo, y la definición por recursión es una regla muy útil para lograr la definición de una función computable, que es lo que Chomsky quiere. En este sentido, este cálculo no se diferencia sustancialmente de la definición por recursión de otras funciones con dominio y codominio en los números naturales. Dado que para Chomsky el lenguaje, entendido como lenguaje-I, es *interno* al individuo, y está considerado en intensión, lo que significa que está interesado en la naturaleza del sistema, más que en sus manifestaciones externas (véase Chomsky, 2015, p. 92), su estudio no se puede quedar en un nivel formal o abstracto. En un trabajo reciente, Berwick, Friederici, Chomsky y Bolhuis (2013, p. 90), nos indican que el lenguaje humano está asentado sobre un mecanismo computacional particular *realizado neurológicamente*. En el plano neurológico, tal mecanismo generativo/computacional debe diferenciarse de ambos sistemas de interfaz; i.e. el sensorio-motor y el conceptual-intencional y estos autores presentan diferentes regiones identificadas tanto con el mecanismo computacional como con los sistemas de interfaz, conectadas vía tractos específicos (véase Berwick et al., 2013 para más detalles).

¿Qué sucede entonces con la propiedad de la recursión? Bajo este planteamiento, la recursión no sea concebida meramente como una técnica para definir funciones, sino como una propiedad que define la naturaleza humana, pues para Chomsky el componente recursivo de la facultad del lenguaje parece carecer de cualquier análogo en la comunicación animal (cf. Hauser, Chomsky y Fitch, 2002, p. 1571).

La recursión es, así, considerada como una propiedad del mundo real, que ya está ahí fuera, y que debemos descubrir en nuestra actividad científica. No es de extrañar, por tanto, que las reglas sean concebidas como internas a la mente/cerebro, lo que implica tanto (I): que las reglas son independientes del formalismo empleado para caracterizarlas (i.e tales reglas están ya ahí y tenemos que *descubrirlas*, no están constituidas por la teoría), como (II): que tales reglas surgen de la interacción de tres factores (Chomsky, 2005), a saber, un primer factor que tiene que ver con la dotación genética (i.e. el estudio de la gramática universal); un segundo factor que es la experiencia y que establece la variación entre lenguas; y un tercer factor que tiene que ver con principios que no son específicos de la facultad del lenguaje, por ejemplo los principios de computación eficiente (Chomsky, 2005 para más detalles). Por tanto, el mecanismo computacional chomskiano, además de contar con principios de organización estructural (derivados de los factores arriba mencionados) y restricciones que no son compartidos en gran parte por otras facultades mentales, lleva a cabo sus procesos y computaciones de una manera automática y obligatoria, y es función del científico descubrir y describir las reglas que constituyen tales computaciones y procesos.

Esta concepción del lenguaje, aunque ampliamente compartida y no carente de interés a muchos niveles, puede contrastarse, con cierto provecho, con otra concepción caracterizada por un punto de vista antropológico. Veamos en qué consiste tal concepción.

IV. La concepción antropológica del lenguaje: Wittgenstein

Wittgenstein, por su parte, tampoco fue ajeno a los desarrollos en la teoría de la computabilidad (cf. Shanker, 1987). En diferentes trabajos encontramos referencias explícitas a la propiedad de la recursión.

Empezando por el *Tractatus*, la noción clave en torno al concepto de recursión es la noción de forma general de una serie de formas. El término general de una serie de formas es un concepto formal y por tanto expresa una variable (4.1273). Además, una serie de formas es una serie que está ordenada por relaciones internas (4.1252) y tanto la forma general de la proposición (cf. 5.2, 6) como la forma general de un entero positivo (4.1252) son instancias de formas generales de series de formas. Dicho de otro modo, tanto $[\bar{p}, \bar{\xi}, N(\bar{\xi})]$ como $[0, \xi, \xi + 1]$ son instancias de $[a, x, O'x]$ (5.2522). Estas expresiones entre corchetes pueden definirse mediante un sistema de ecuaciones recursivas como sigue (Mota, 2014, 2015a, 2015c para más detalles):

$[a, x, O'x]$	$O^{(0)}{}'(a) = a$, $O^{(n)}{}'(a) = O'O^{(n-1)}{}'(a)$.
$[\bar{p}, \bar{\xi}, N(\bar{\xi})]$	$N^{(0)}{}'(\bar{\xi}) = \bar{\xi}$, $N^{(n)}{}'(\bar{\xi}) = N'N^{(n-1)}{}'(\bar{\xi})$.
$[0, \xi, \xi + 1]$	$\xi + 1 = \xi'$ $[a+1 = a']$, $a+(\xi + 1) = (a + \xi)+1$ $[a+(b + 1) = (a + b)+1]$.

El primer término expresa el comienzo de la serie, sus valores representan un dominio. El segundo término es un término cualquier de la serie, el cual pertenece al dominio. Si estuviéramos hablando de una función parcial, cabría la posibilidad de que no perteneciera al dominio de dicha función. El segundo término, por tanto, haría las veces de *input*. El tercer término alude a dos aspectos que en cierto modo aparecen juntos: por un lado tenemos que *O'x* es un valor (i.e. un output) y por otro tenemos que *O'* es la operación de transición que permite pasar de un término a otro. Como se puede apreciar, hay grandes paralelismos entre la forma general de Merge y la forma general de una serie de formas, y es que a mi juicio, Merge genera series de formas. Así, creo que Wittgenstein ya advirtió algo de gran importancia tanto para la noción de algoritmo (cf. Mota, 2015c), como para la operación Merge, tan relevante para la formulación de Chomsky (Mota, 2014, 2015a). Por otra parte, y como he señalado en Mota (2015c), la noción de serie de formas recoge lo que es común tanto a las definiciones de iteradores y recursores (ver *supra*), como a la definición generalmente empleada de algoritmo, presentada unas líneas más arriba de la mano de Shanker (1987). Como puede apreciarse, la técnica de la definición por recursión, aunque no fue mencionada como tal en el *Tractatus*, subyace a tales formas generales.

En obras posteriores, Wittgenstein fue mucho más explícito a este respecto (véase por ejemplo, 1975, §§163-167; 1974, 29-33, 36; 2005, 126-130, 133). Así, en las *Observaciones Filosóficas* (§163) dirá de una definición recursiva que es una regla fundamental del sistema que nos indica cómo proceder y, como tal, no se puede aseverar o negar, es decir, no son *descripciones*. Abundando en esta misma idea, Wittgenstein, en la *Gramática Filosófica* (36) nos indica que la definición recursiva "es una regla para la construcción de reglas de sustitución, o también el término general de una serie de definiciones [formas]". Esto nos lleva directamente al esquema presentado unas líneas más arriba, donde se ilustra con claridad este punto.

Por tanto, para Wittgenstein, la definición por recursión no alude más que a una *técnica*, a una regla para definir un cálculo, no a nada que nos comprometa ontológicamente con una propiedad del mundo que tenemos que descubrir y que, además, podría, ni más ni menos, que favorecer una explicación naturalista de la esencia de la especie humana. Para Wittgenstein, "el matemático crea *esencias*" (1978, I, §32, cursivas en el original). Por tanto, no es nada extraño que Wittgenstein señale que el matemático no es un descubridor, sino un inventor –o un constructor– (1978, I, §168; II, §2, §38; V, §11). De ahí, se desprende que la matemática, después de todo, es un fenómeno antropológico (1978, VII, §33).

Al reflexionar sobre cómo funciona el concepto de recursión, podemos ver diferencias claras e importantes entre las concepciones computacional y antropológica. Desde la segunda, hablar un lenguaje, así como computar una función, es una *práctica* normativa, dominar un lenguaje es aprender *técnicas* para usar palabras en el ámbito comunicativo humano. De acuerdo con esto, Wittgenstein está lejos de señalar que un lenguaje de un individuo aislado socialmente sea inconcebible; uno puede seguir reglas *privadamente*. A lo que Wittgenstein se opone es que uno pueda seguir *reglas privadas*, reglas que nadie en principio puede entender o seguir (cf. Hacker, 1986, pp. 252-253), pero que operan de manera automática y obligatoria en una suerte de estructura mental que caracteriza a la especie humana. La idea de que hay reglas para el uso correcto de las palabras *que necesitan ser descubiertas* no tiene sentido (cf. Hacker, 2014); y menos sentido tiene todavía si se piensa que es mediante el método científico como hay que descubrirlas.

Aprender un lenguaje es aprender a jugar juegos de lenguaje, a participar en los juegos de lenguaje característicos de la cultura o forma de vida en la cual uno ha nacido; de ahí, la regularidad necesaria para el lenguaje es un acuerdo *en la forma de vida* (Wittgenstein, 1988, §241); una conexión esencialmente cultural (más que biológica) entre el lenguaje y los sistemas complejos de prácticas y actividades que mantiene junta a una comunidad. La noción de regla privada, desconectada del ámbito de la práctica, simplemente no tiene sentido, dado que el concepto de regla es el concepto de una cierta forma de práctica. Así, la idea de que hay reglas que operan en un mecanismo mental interno (una abstracción del cerebro) de forma automática y obligatoria, no tiene sentido.

En esta línea, Kenny (1989) llevó a cabo una crítica de algunas formulaciones en *Reglas y Representaciones* (Chomsky, 1980) que merece la pena traer a colación. De acuerdo con Kenny (1989, pp. 205-206), hay que distinguir entre poseedores, capacidades (o habilidades) y vehículos. Así, por ejemplo, en relación con el lenguaje, que sería la habilidad o capacidad –una abstracción de la conducta, dice Kenny (Ibid., p. 205)–, distinguimos entre sus poseedores (las personas, no sus cerebros o sus mentes) y los vehículos de la habilidad, i.e. la parte de su poseedor en virtud de la cual éste es capaz de ejercer la habilidad. Dadas estas distinciones conceptuales (no metafísicas u ontológicas), Kenny encuentra que la caracterización chomskiana del lenguaje en términos de "órganos mentales" entraña una confusión filosófica (véase Chomsky, 1980, p. 39). Como dice Kenny (1989, p. 203), "la descripción que da Chomsky de las estructuras mentales que investiga, introduce un elemento metafísico irrelevante en el interfaz entre fisiología y psicología". A lo que Kenny se refiere es a la tendencia de

Chomsky (cf. 1980, pp. 3, 49) a señalar que nuestras habilidades lingüísticas están basadas sobre estructuras mentales (i.e. el mecanismo computacional y las representaciones mentales u objetos sintácticos sobre los que opera), que sirven como vehículos para el ejercicio de tales capacidades.[13] Ese es justamente el elemento metafísico.

En *Los Cuadernos Azul y Marrón*, Wittgenstein nos invita suponer que construimos un modelo de mente como resultado de las investigaciones psicológicas, un modelo que, como si dijéramos, explicara la acción de la mente. Tal modelo, descrito, por ejemplo, en términos computacionales, sería parte del *simbolismo* (Wittgenstein, 2009, p. 32). Pero cuando nos preocupamos por la *naturaleza* del pensamiento, del lenguaje, de la memoria, etc…, corremos el riesgo de interpretar erróneamente que nos estamos preguntando por la *naturaleza*, computacional o no, de un medio, de un vehículo, cuando, en realidad, estamos ante una confusión causada por el uso desconcertante de nuestro lenguaje (cf. Wittgenstein, 2009, p. 33). Por ello, y de acuerdo con McGinn (1997, pp. 54-55), uno de los principales blancos de Wittgenstein en las *Investigaciones* es el modelo de la mente como un mecanismo interno. De acuerdo con Hacker (2007), la concepción computacional, en relación con la cual se adscriben al cerebro, en mayor o menor grado de abstracción, representaciones simbólicas y operaciones computacionales (i.e. reglas) sobre ellas, está inspirada, al menos en parte, en una confusión acerca del procesamiento de la información por parte de los ordenadores. De ahí que Hacker (2014, p. 1286) opine que es quimérica la idea de que entender una palabra implique cálculos, computaciones o un proceso de derivación del tipo que sea. Es este tipo de confusión el que lleva a señalar que el uso sistemático que hace Chomsky del término “mente/cerebro” y su comparación de la facultad del lenguaje con los órganos del cuerpo refleja una visión general de la relación entre la mente y el cerebro que se puede describir en los siguientes términos (Cain, 2005, p. 11): describir la mente de un individuo es describir su cerebro en términos mentales; es describir su cerebro en un nivel de abstracción más alto que en el que lo hacen los neurocientíficos. Alternativamente, describir el cerebro en términos neurocientíficos es describir la mente en un nivel de abstracción más bajo que en el que lo hacen los psicólogos. Primero, definir y describir la función Merge es describir un cálculo, y no el cerebro a ningún nivel de abstracción. Describir el algoritmo que computa una calculadora no es describir el nivel electrónico en un nivel de abstracción mayor (cf. Mota, 2014, 2015b). A esto

[13] Como Chomsky (1980, p. 3) mismo afirma, está interesado en varios aspectos relacionados con las capacidades cognitivas humanas y las estructuras mentales que sirven como vehículos para el ejercicio de tales capacidades.

se refiere Hacker (2014, pp., 1287-1288), cuando habla del encantamiento de una forma de representación. Las formas de representación son, por otro lado, convencionales, guiadas por la normatividad de nuestras reglas, creadas o inventadas. En una línea crítica muy similar, Tomasini (2003, p. 154) indica que "como la filosofía en el filósofo, la ciencia también hace que el científico en ocasiones se dedique a perseguir quimeras". Y continúa, "la idea de un "órgano mental" es un ejemplo paradigmático de dicha desorientación" (Tomasini, Ibid., p. 154).

Dicho esto, sería un error tachar a Wittgenstein de reduccionista y conductista (como hizo Fodor, 1984, p. 24). Se puede aceptar, con Hacker (2007), que hay paralelismos entre Wittgenstein y el conductismo psicológico y el conductismo lógico, pero en ningún caso se puede hablar de reduccionismo. Así, Wittgenstein no reduce lo mental ni a la conducta ni al cerebro/fisiología (i.e. no es materialista). Asimismo, para Wittgenstein lo mental no tiene ningún epíteto metafísico, sino lógico (cf. Mota, 2015d), lo cual significa, entre otras cosas, que Wittgenstein no busca *explicar* lo mental postulando mecanismos o entidades mentales (por ejemplo estructuras mentales) que ejerzan de mediadores, interfaces o vehículos de las capacidades humanas; más bien, lo mental es una forma de hablar (compuesta por varios usos que damos a diferentes conceptos) sobre el conjunto de capacidades intelectuales humanas, gramaticalmente, i.e. conceptualmente, unidas con, pero no reducidas a meras formas de comportamiento.[14] En este sentido, la relación entre los conceptos psicológicos y la conducta se revela mediante una investigación gramatical sobre cómo nuestros conceptos funcionan (i.e. son usados). Así, lo mental no es la descripción del cerebro (un órgano) en un nivel mayor de abstracción. Tampoco sería correcto caracterizar a Wittgenstein como pragmatista (cf. Fodor, 2008), *pace* las semejanzas (véase Bouwsma, 1986; cf. Hacker, 2010). Por supuesto,

[14] Según McGinn (1997, p. 95), Wittgenstein no debe verse como un anti-realista respecto a los procesos mentales. Esto depende de qué se entiende por 'mente'. Es claro que Wittgenstein rechaza lo mental entendido como un mecanismo interno, en analogía con el modelo o la figura de un mecanismo físico. Esta falsa analogía entre objetos mentales y físicos subyace al uso que alguien puede hacer de la expresión "esto es una rodilla" como si el negar que exista la mente concebida como un mecanismo interno implicara negar la existencia de las rodillas. A esta imagen de la mente no le corresponde nada, y Wittgenstein puede verse como un anti-realista respecto a estas concepciones de la mente como algoritmo (un algoritmo es una construcción o invención matemática, no un descubrimiento de una entidad existente en una realidad, sea ésta empírica o platónica), como estructura mental que vehicula las capacidades intelectuales. Esta imagen es una quimera y un mito. Por el contrario, Chomsky sí mantiene una actitud realista respecto al mecanismo computacional, el cual es un sistema generativo natural (Chomsky, 2010, p. 52). Tal mecanismo es un objeto de comparación (un modelo), no de investigación.

también sería un error implementar a Wittgenstein en otras *teorías* como la denominada "*embodied cognition*". De acuerdo con Hacker (2007), no vemos a los demás seres humanos como mentes encarnadas sino seres vivos con capacidades perceptivas, afectivas y volitivas que se comportan en un trasfondo de normas sociales y valores (i.e. una forma de vida). Así, el sujeto de los predicados psicológicos no es la mente ni el cerebro, sino la persona como un todo. Por último, desde la perspectiva antropológica no se intenta construir ninguna teoría de ningún tipo, por lo que el vocabulario psicológico no es parte de un vocabulario teórico, sino describir la red conceptual que está en juego en nuestro discurso sobre el lenguaje.

V. Consideraciones finales

¿Qué se quiere decir cuando se habla de recursión? En primer lugar el concepto de recursión alude a un fenómeno antropológico, i.e. a la técnica construida para definir funciones que llamamos "computables". Por tanto, y de acuerdo con Frath (2014), podemos rechazar, con Wittgenstein, los sistemas metafísicos construidos sobre hipótesis ontológicas, lo que en este caso en concreto viene a significar que Wittgenstein rechaza explicaciones sobre la naturaleza humana con base en hipótesis ontológicas sobre mecanismos mentales. Así, el concepto de recursión también se usa con base en falsas analogías y similitudes superficiales. Un ejemplo de una falsa analogía es cuando representamos la recursión mediante el modelo de cosas dentro de cosas del mismo tipo (véase más arriba la definición recursiva del iterador ϕ, en donde se pasa de un estado a otro y la noción de auto-inclusión no está implicada en absoluto). Así, la recursión no es una propiedad global de las estructuras, y esta caracterización está basada en una falsa analogía con respecto al uso que de ese concepto hacemos al definir funciones, i.e. el establecimiento de reglas que constituyen un modo de proceder. Un ejemplo de similitud superficial lo encontramos cuando se dice que un X es recursivo porque es auto-referente. La afirmación de que X es recursivo por la razón exacta de que X contiene una auto-referencia es inadecuada.

La utilidad y adecuación de la recursión para definir funciones seduce a Chomsky hasta un punto tal que construye un mito. Naturalmente, si entiende que el lenguaje es un mecanismo computacional cuya operación básica se define por recursión, y si además dicho componente recursivo parece carecer de cualquier análogo en la comunicación animal, es evidente que va a situar a esa propiedad en la punta de lanza de una buena explicación de la especificidad humana. Sin embargo, como he analizado previamente, esto se basa en una figura desorientadora, que consiste en entender el lenguaje como una estructura mental interna, cuando el lenguaje es una *práctica* normativa; dominar un lenguaje es aprender *técnicas* para usar palabras, no alcanzar un estado mental.

Este tipo de confusiones muestra aquello que Wittgenstein señaló en las *Investigaciones Filosóficas* (1988, §109); esto es, que el lenguaje embruja nuestro entendimiento: el modelo de lo mental ha causado una suerte de confusión o mito que ha llevado a buscar en el mundo una propiedad cuyo descubrimiento supondrá la confirmación de una teoría sobre la naturaleza (computacional) del lenguaje, cómo funciona, y la esencia humana. Además muestra una mala comprensión del papel en el lenguaje de las proposiciones metafísicas.[15] Lo que, por otro lado, sí es una característica definitoria de los seres humanos el engañarnos con y en nuestro propio lenguaje al reflexionar sobre el lenguaje mismo, embrujarnos con el uso metafísico que hacemos de nuestro lenguaje, sin reparar en las construcciones conceptuales que nosotros mismos confeccionamos. En otras palabras, todo esto es una desorientación concerniente con la proyección de un simbolismo sobre el mundo, creyendo que tales sombras son objetos reales del mundo. Caemos, así, en la metafísica de la mente, en la imagen de la mente considerada como un mecanismo causal interno, y hacemos, de acuerdo con Reguera (1992), "una traslación ilícita de la metodología científica dura a un objeto blando que no se ve, del que no se tiene prueba alguna, aunque guste imaginárselo". Esos mecanismos causales son las pretendidas estructuras mentales, los vehículos de los que hablaba en la sección precedente, de los que se pretende que la recursión sea su propiedad definitoria.

De acuerdo con Wittgenstein en las *Investigaciones Filosóficas* (1988, XIV), "[e]n efecto, en psicología existen métodos experimentales y *confusión conceptual*". No obstante, la presencia del método experimental no debe hacer creer que ya se disponen de los medios para liberarnos de los problemas, dado que, de acuerdo con Wittgenstein, problemas y métodos pasan de largo sin encontrarse. Por tanto, a la cuestión de si la recursión es

[15] Lejos de expresar verdades factuales, metafísico-ontológicas (independientes del lenguaje), o ideales, las proposiciones metafísicas exhiben relaciones y transiciones desde un concepto a otro. Por ello, es una *ilusión* considerar la metafísica como una investigación sobre las posibilidades y las necesidades de la realidad. La metafísica, por tanto, no puede considerarse descriptiva, ni de un mundo empírico, ni de un mundo ideal, platónico o abstracto. Por otro lado, Rhees (2005, pp. 66, 90-92, 141, 149, 166) niega, y estoy de acuerdo con él, que lo que hace posible un juego de lenguaje *no son ciertos hechos*, sino que *no cuestionamos* ciertos hechos, es decir, "no estamos diciendo que haya cierta clase de proposiciones o hechos que no puedan ser dudados. Más bien lo contrario" (Rhees, 2005, p. 66). Así, nuestros juegos de lenguaje (y de ahí, nuestras prácticas) no están determinados por nada como *la estructura del mundo* ni son dependientes de nada como *la naturaleza de la constitución humana*; lo que hace a un juego de lenguaje *un* juego de lenguaje no son ciertos hechos que sean básicos.

una propiedad que explica nuestra especificidad le rodea toda una nube de filosofía y psicología que, al final, se condensa en una gotita de gramática (cf. Wittgenstein, 1988, XI). Este (pseudo-)problema, que resulta de la confusión conceptual, y que tiene apariencia de problema científico, no puede por menos que condensarse en una gotita de gramática. Su disolución no supone el empleo del método científico (no necesitamos hacer ningún descubrimiento), sino la clarificación conceptual.[16]

En esta clarificación conceptual, que nada tiene que ver con el reemplazo de una *teoría* sobre qué es y cómo funciona el lenguaje por otra, consiste, precisamente, la investigación gramatical de Wittgenstein; esto es, tal investigación consiste en examinar un área de nuestro lenguaje que se ha convertido en punto principal para el mito filosófico y la confusión, debido, por ejemplo, a falsas analogías y similitudes superficiales como las que he señalado más arriba. Esto sucede, en relación con la propiedad de la recursión, cuando se produce un traslado ilícito de una propiedad empleada para definir funciones hasta otorgarle un papel explicativo en la especificidad de la esencia (parece que computacional) humana, y este traslado tiene base en un estilo de pensamiento que nos aleja cada vez más de entender, incluso, en qué consiste la recursión. La importancia de la recursión reside, más que en la idea vacía y mitológica de *explicar* qué nos hace humanos, en la práctica cotidiana de definir funciones que son computables. Así, alguien entiende este concepto cuando utiliza esta regla para construir otras reglas de sustitución.

Agradecimientos:

Quiero expresar mi gratitud a los profesores Eduardo Barrio, José Ferreirós, Roberto Torretti y José Manuel Igoa por su ayuda en relación con una versión previa de este trabajo.

Referencias bibliográficas

Bertot, Y. & Komendantskaya, E. (2009). Using structural recursion for corecursion. *In Types for Proofs and Programs, International Conference, TYPES 2008*, Vol. 5497, 220-236.

Berwick, R., Friederici, A.D., Chomsky, N. & Bolhuis, J. (2013). Evolution, brain, and the nature of language, *Trends in Cognitive Science*, 17, 89-98.

[16] Uno tiene la sensación que ante preguntas como "¿Qué es la mente?", "¿Qué es el significado?", "¿Qué es la recursión?", tiene que proporcionar *explicaciones*, *descubrir* algo, en analogía con cuestiones científicas. Sin embargo, esta tendencia es la fuente de la metafísica. Tratar de responder estas preguntas mediante métodos científicos no lleva a un mayor entendimiento de los mismos, sino a una completa oscuridad.

Boolos, G., Burgess, J. y Jeffrey, R. (2007). *Computability and logic*. Cambridge, UK: Cambridge University Press.

Bouwsma, O.K. (1986). *Wittgenstein: Conversations 1949-1951*. Edited by J.L. Craft and R.E. Hustwit. Indianapolis, Ind.: Hackett Publishing [Spanish edition (2004): Últimas conversaciones, trad. Miguel Ánguel Quintana Paz. Salamanca: España].

Cain, M. (2005). Confronting philosophical objections to Chomskyan linguistics, *European Journal of Analytic Philosophy*, *1*, 5-24.

Chomsky, N. (1955/1975). *The logical structure of syntactic theory*. New York: Plenum Press.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, 2, 137-167.

Chomsky, N. (1980). *Rules and representations*. New York: Columbia University Press.

Chomsky, N. (1995). *The minimalist program*. Cambridge. MA: MIT Press.

Chomsky, N. (2005). Three factors in language design, *Linguistic Inquiry*, *36*, 1-22.

Chomsky, N. (2006). *Language and mind*. Cambridge: Cambridge University Press.

Chomsky, N. (2007a). Of minds and language, *Biolinguistics*, *1*, 9-27.

Chomsky, N. (2007b). Approaching UG from below. In U. Sauerland & H. M. Gärtner (eds.), *Interfaces + Recursion = Language?* (pp. 1-30). Berlin: Mouton.

Chomsky, N. (2008). On phases. In R. Freidin, C. Otero, & M. L. Zubizarreta (*Eds*.), *Foundational issues in linguistic theory* (pp. 133-166). Cambridge. MA: MIT Press.

Chomsky, N. (2014). Minimal recursion: exploring the prospects. In T. Roeper & M. Speas (*Eds*.), *Recursion: Complexity in Cognition* (pp. 1-15). Heidelberg/ New York/Dordrecht/London: Springer.

Chomsky, N. (2015). Some core contested concepts, *Journal of Psycholinguistic research*, 44, 91-104.

Cutland, N. (1980). *Computability: an introduction to recursive function theory*. Cambridge: Cambridge University Press.

Enderton, H. (2009). *A mathematical introduction to logic*. San Diego, CA.: Harcourt Academic Press.

Ferreirós, J. (2006). *Introducción a G. Cantor, Fundamentos para una teoría general de conjuntos*. Barcelona: Crítica.

Fodor, J.A. (1984). *El lenguaje del pensamiento*. Madrid: Alianza.

Fodor, J.A. (2008). *LOT 2. The language of thought revisited*. Oxford: Oxford University Press.

Frath, P. (2014). There is no recursion in language. In F. Lowenthal & F. Lefebvre (eds.), *Language and recursion* (pp. 181-191). New York: Springer.

Gödel, K. (1931). On formally undecidable propositions of the Principia Mathematica and related systems. I. Reprinted in M. Davis (2004) (ed.), *The undecidable* (pp. 4-38). New York: Raven Press.

Gödel, K. (1964). Postscriptum to Gödel 1931. In M. Davis (*Ed.*), *The undecidable* (pp. 71-73). New York: Raven Press.

Hacker, P. M. S. (1986). *Insight and illusion. Themes in the philosophy of Wittgenstein*. Oxford: Clarendon Press.
Hacker, P. M. S. (2007). The relevance of Wittgenstein's Philosophy of Psychology to the Psychological Sciences. *Proceedings of the Leipzig Conference on Wittgenstein and Science*.
Hacker, P. M. S. (2010). A normative conception of necessity: Wittgenstein on necessary truths of logic, mathematics and metaphysics. In V. Munz, K. Puhl, and J. Wang (eds.), *Language and the World, Part One: Essays on the Philosophy of Wittgenstein, Proceedings of the 32nd International Ludwig Wittgenstein Symposium in Kirchberg, 2009* (pp. 13-34). Frankfurt: Ontos Verlag.
Hacker, P. M. S. (2014) Two conceptions of language, *Erkenntnis*, 79, pp. 1271-1288.
Hauser, M., Chomsky, N. & Fitch, T. (2002). The faculty of language: What is, who has it, and how did it evolve?, *Science*, 298, 1569-1579.
Hrbacek, K. & Jech, T. (1999). *Introduction to Set Theory*, New York: Marcel Dekker, Inc.
Kenny, A. (1989). *El legado de Wittgenstein*. Madrid: Siglo XXI.
Kleene, S. C. (1943). Recursive predicates and quantifiers, *Transactions of the American Mathematical Society*, 53, 41-73.
Kleene, S. C. (1952). *Introduction to metamathematics*. Amsterdam: North-Holland Publishing.

Luuk, E. & Luuk, H. (2011). The redundancy of recursion and infinity for natural language, *Cognitive Processing*, 12, 1-11.
Mac Lane, S. (1986). *Mathematics. Form and function*. New York: Springer-Verlag.
McGinn, M. (1997). *Wittgenstein and the Philosophical Investigations*. London: Routledge.
McGinn, M. (2009). Wittgenstein and internal relations, *European Journal of Philosophy*, 18, pp. 495-509.
Moschovakis, Y. & Paschalis, V. (2008). Elementary algorithms and their implementations. In S.B. Cooper, B. Löwe, & A. Sorbi (*Eds.*), *New computational paradigms* (pp. 87-118). Berlin: Springer.
Moschovakis, Y. (1998). On founding the theory of algorithms. In H.G. Dales, & G. Oliveri (*Eds*.), *Truth in mathematics* (pp. 71-104). Oxford, UK: Oxford University Press.
Moschovakis, Y. (2009). What is an algorithm? in B. Engquist & W. Schmid (Eds.), *Mathematics Unlimited: 2009 and Beyond* (pp. 919-936). Berlin: Springer.
Mosterín, J. (2007). *Los lógicos*. Madrid: Austral.
Mota, S. (2014). *Sobre el anti-realismo de Wittgenstein y su aplicación al programa chomskiano, Metatheoria*, (en proceso de edición)
Mota, S. (2015a). Sobre el concepto de recursión y sus usos, *Praxis Filosófica*, 40, 153-181.
Mota, S. (2015b). ¿Por qué se usa 'recursión' cuando se quiere significar 'autoinclusión'?: Clarificaciones conceptuales sobre la recursión en el programa chomskiano, *Revista de Lingüística Teórica y Aplicada*, 53, 171-191.

Mota, S. (2015c). ¿Qué es un algoritmo? Una respuesta desde la obra de Wittgenstein, *ÉNDOXA*. Series Filosóficas, 36, 317-328.

Mota, S. (2015d). Wittgenstein en torno a los conceptos, *Análisis. Revista de Investigación Filosófica*, 2, 195-219.

Odifreddi, P. (2011). Recursive functions: an archeological look. In S.B. Cooper, &, A. Sorbi (Eds.), *Computability in context. Computation and logic in the real world* (pp. 329-349). London: Imperial College Press.

Post, E. (1944). Recursively enumerable sets of positive integers and their decision problems. Reprinted in M. Davis (2004) (ed.), *The undecidable* (pp. 305-337). New York: Raven Press.

Reguera, I. (1992). *Introducción a L. Wittgenstein, Lecciones y conversaciones sobre estética, psicología y creencia religiosa*. Barcelona: Paidós

Rhees, R. (2005). *Wittgenstein's On Certainty. There –like our life*. Oxford: Blackwell Publishing.

Shanker, S.G. (1987). Wittgenstein versus Turing on nature of Church's thesis, *Notre Dame Journal of Formal Logic*, *28*, 615-649.

Soare, R. (1996). Computability and recursion, *The Bulletin of Symbolic Logic*, 2, 284-321.

Soare, R. (2012). Interacting computing and relativized computability. In B.J Copeland, C.J. Posy, & O. Shagrir (Eds.), *Computability. Turing, Gödel, Church, and Beyond* (pp. 203-260). Cambridge, MA.: The MIT Press.

Tomasini, A. (2003). *Estudios sobre las filosofías de Wittgenstein*. México, D.F.: Plaza y Valdés.

Torretti, R. (1998). *El paraíso de Cantor*. *La tradición conjuntista en la filosofía matemática*. Santiago de Chile: Editorial Universitaria.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. Reprinted in M. Davis (2004) (ed.), *The undecidable* (pp. 116-151). New York: Raven Press.

Wirth, N. (1986). *Algorithms and data structures*. Hemel Hempstead, UK: Prentice Hall.

Wittgenstein, L. (1922). *Tractatus logico-philosophicus*. London: Routledge.

Wittgenstein, L. (1988). *Investigaciones filosóficas*. Barcelona: Crítica.

Wittgenstein, L. (1974). *Philosophical grammar*. Oxford: Blackwell.

Wittgenstein, L. (1975). *Philosophical remarks*. Oxford: Blackwell.

Wittgenstein, L. (1978). *Remarks on the foundations of mathematics*. Oxford: Blackwell.

Wittgenstein, L. (2005). *The big typescript: TS213*. Oxford: Wiley-Blackwell.

Wittgenstein, L. (2009). *Los cuadernos azul y marrón*. Madrid: Tecnos.