Capítulo 3

LÓGICA Y ANÁLISIS DEL LENGUAJE NATURAL

Francisco José Salguero Lamillar

3.1 Lenguaje formal y lenguaje natural.

3.1.1 La lógica de predicados y el lenguaje natural.

Un lenguaje formal es un sistema semiótico similar al lenguaje natural humano. Como tal sistema semiótico, comparte con este último las características propias de los sistemas de signos. La principal diferencia entre los lenguajes formales y el lenguaje natural humano reside en su uso desde el punto de vista de la comunicación: los lenguajes formales han sido diseñados para ser usados en determinados contextos y con fines comunicativos muy concretos, frente a la versatilidad contextual y a la creatividad propias de las lenguas humanas. Por lo demás, los problemas de ambigüedad o de complejidad estructural que se derivan de la comparación entre los lenguajes formales y los lenguajes naturales muestran sólo diferencias de grado que no tienen por qué ser consideradas como características propias de dos realidades esencialmente distintas. En este sentido, Richard Montague propuso en las décadas de 1960 y 1970 un programa de investigación en el que las lenguas naturales eran consideradas desde una perspectiva similar a la de los lenguajes formales de la lógica y de las matemáticas, de modo que en su análisis cabría la posibilidad de utilizar herramientas extraídas a partir de estos últimos.

Un lenguaje formal está compuesto por símbolos que, en un principio, no están interpretados. Estos símbolos se combinan entre sí para obtener expresiones complejas del lenguaje según ciertas reglas que han de ser expuestas con claridad y rigor, no habiendo excepciones para las mismas. Finalmente, es posible interpretar (es decir: dar un significado) a cada una de las expresiones, simples o complejas, del lenguaje formal en función del contexto en el que vaya a ser utilizado y la finalidad para la que se lo requiera. Por lo tanto, en todo lenguaje formal cabe distinguir:

1. Un conjunto de símbolos no interpretados o *vocabulario*.
2. Un conjunto de reglas sintácticas o de formación de expresiones complejas a partir del vocabulario.
3. Un conjunto de *funciones semánticas* para la interpretación de las expresiones del lenguaje.

Ejemplifiquemos las características de los lenguajes formales a través del análisis de un *lenguaje de lógica de predicados de primer orden* al que llamaremos, a partir de ahora, lenguaje L.

El vocabulario de L está compuesto por los siguientes conjuntos de símbolos:

- VAR={x, y, z, x_1, y_1, z_1, x_2, y_2, z_2,....}
- PRED={P, Q, R, S, T, P_1, Q_1, R_1, S_1, T_1,...}
- SIGN={¬, ∧, ∨, →, ∀, ∃}

El conjunto VAR es el conjunto de las variables individuales, el conjunto PRED es el conjunto de los símbolos predicativos y el conjunto SIGN es el conjunto de los signos constantes lógicos. Los conjuntos VAR y PRED tienen cardinalidad infinita, puesto que tienen tantos elementos cuantos números naturales, como puede observarse por el uso de los subíndices. En cambio, el conjunto SIGN es un conjunto finito que en el lenguaje L sólo posee seis elementos.

Las reglas sintácticas para la formación de expresiones complejas de L son las siguientes:

1. Si $\{x_1, x_2,... x_n\}\in$VAR y P∈PRED entonces $Px_1, x_2,... x_n \in$L
2. Si α∈L y β∈L entonces ¬α∈L, α∧β∈L, α∨β∈L y α→β∈L
3. Si x∈VAR y α∈L entonces ∀xα(x)∈L y ∃xα(x)∈L
4. Nada más es una expresión del lenguaje L

En las reglas sintácticas anteriores, α y β son expresiones complejas cualesquiera y nunca elementos del vocabulario de L. De igual modo α(x) y β(x) son expresiones complejas cualesquiera de L que contienen el elemento x del conjunto VAR. A partir de ahora vamos a llamar a las expresiones complejas de L *fórmulas bien formadas* (abreviadamente fbf.).

En cuanto a la interpretación del vocabulario y de las fbfs. del lenguaje L, tradicionalmente se ha usado la lógica de predicados de primer orden para formalizar ciertas oraciones enunciativas del lenguaje natural con el fin de descubrir las relaciones lógicas que se dan entre ellas. Por lo tanto, la interpretación de las expresiones simples y complejas de L se encuentra en consonancia con ciertas intuiciones que poseemos como hablantes de una lengua en relación con la interpretación de los enunciados de dicha lengua. En primer lugar, entendemos que dichos enunciados hacen referencia a determinados individuos del mundo; en segundo lugar, que estas oraciones enunciativas establecen propiedades y relaciones entre los individuos referidos y, finalmente, entendemos que estos enunciados se corresponden o no con la realidad del mundo que describen por lo que son verdaderos o falsos, respectivamente. Todo ello está presente en la siguiente interpretación del vocabulario y de las fórmulas bien formadas del lenguaje L:

Sea D el dominio del discurso, es decir, el conjunto de los individuos a los que se refieren las oraciones enunciativas formalizadas mediante L. Sea *f* una función semántica que otorga una interpretación a cualquier expresión simple o compleja de L de modo que si σ es una expresión de L de cualquier tipo, $f(\sigma)=||\sigma||$, siendo $||\sigma||$ el significado denotacional de la expresión σ. Entonces:

1. Si $x \in VAR$, $||x||=a$, donde $a \in D$.
2. Si $P \in PRED$ y P admite n variables individuales como argumento (es decir: si P es un predicado n-ádico) entonces $||P||=\{<a_1,a_2,...a_n>, <b_1,b_2,...b_n>,... <c_1,c_2,...c_n>\}$, donde $a_1,a_2,...a_n,b_1,b_2,...b_n,...c_1,c_2,...c_n \in D$ y los signos < y > indican que se trata de elementos ordenados del conjunto.
3. Si $P \in PRED$ y $x_1, x_2,... x_n \in VAR$ entonces $||Px_1, x_2,... x_n||=1$ si y sólo si (abreviadamente sii) $<||x_1||, ||x_2||,... ||x_n||> \in ||P||$. En otro caso $||Px_1, x_2,... x_n||=0$
4. Si $\alpha \in L$, $||\neg\alpha||=1$ sii $||\alpha||=0$. En otro caso $||\neg\alpha||=0$
5. Si $\alpha \in L$ y $\beta \in L$ entonces $||\alpha\wedge\beta||=1$ sii $||\alpha||=1$ y $||\beta||=1$. En otro caso $||\alpha\wedge\beta||=0$
6. Si $\alpha \in L$ y $\beta \in L$ entonces $||\alpha\vee\beta||=1$ sii $||\alpha||=1$ o $||\beta||=1$. En otro caso $||\alpha\vee\beta||=0$
7. Si $\alpha \in L$ y $\beta \in L$ entonces $||\alpha\rightarrow\beta||=1$ sii $||\alpha||=0$ o $||\beta||=1$. En otro caso $||\alpha\rightarrow\beta||=0$
8. Si $\alpha(x) \in L$, $||\forall x\alpha(x)||=1$ sii para todo individuo $a \in D$ $||\alpha(a)||=1$. En otro caso $||\forall x\alpha(x)||=0$
9. Si $\alpha(x) \in L$, $||\exists x\alpha(x)||=1$ sii hay al menos un individuo $a \in D$ tal que $||\alpha(a)||=1$. En otro caso $||\exists x\alpha(x)||=0$

Las reglas de interpretación 8 y 9 son formas abreviadas a las que hay que añadir la condición de que el valor de las variables que haya en α y que no sean la variable x permanezca constante. Eso significa que 8 y 9 han de parafrasearse de la siguiente manera:

- Si $\alpha(x) \in L$, $f(\forall x\alpha(x))=1$ sii para toda función g igual a f excepto, a lo sumo, por el valor semántico de la variable x, $g(\alpha(x))=1$. En otro caso $f(\forall x\alpha(x))=0$
- Si $\alpha(x) \in L$, $f(\exists x\alpha(x))=1$ sii hay al menos una función g igual a f excepto, a lo sumo, por el valor semántico de la variable x, tal que $g(\alpha(x))=1$. En otro caso $f(\exists x\alpha(x))=0$

El lenguaje formal que acabamos de definir puede concebirse como una imagen de un fragmento de una lengua natural: el fragmento correspondiente a las oraciones enunciativas de esa lengua. Por lo tanto, podemos establecer una serie de criterios para analizar este fragmento de una lengua natural cualquiera (v. gr.: el español) y explicar ciertos fenómenos lingüísticos que se encuentran en la frontera entre la sintaxis y la semántica de los lenguajes naturales.

3.1.2 Forma lógica de las oraciones. Algunos problemas de ambigüedad oracional.

Algunos de los fenómenos lingüísticos que pueden ser estudiados con ventaja usando el lenguaje L son, por ejemplo, los relacionados con la ambigüedad oracional. Tradicionalmente, el problema del significado de las palabras ha monopolizado la atención de los lingüistas dedicados a la semántica del lenguaje natural. Se daba por supuesto que el significado de las oraciones no era más que una composición del significado del léxico que las constituye; pero este supuesto, que por un lado es trivial, demuestra ser insuficiente para explicar el significado oracional. Enunciado de esta forma tan simple, de él se deduciría que las dos oraciones siguientes son sinónimas, contra toda evidencia lingüística:

(1) Un perro mordió a Juan.

(2) Juan mordió a un perro.

El principio de composicionalidad en su versión más rigurosa se debe al matemático alemán G. Frege y podemos parafrasearlo de la siguiente manera: “El significado de una expresión compleja es una función del significado de sus partes y de las reglas sintácticas mediante las cuales se combinan”. El significado oracional depende, por tanto, no sólo del significado del léxico involucrado, sino también de su sintaxis. El lenguaje formal L nos ofrece la forma lógica de las oraciones enunciativas (1) y (2) en la que se observa sin ambigüedad cómo se combinan los elementos léxicos de ambas oraciones:

(1') $\exists x(\mathrm{PERRO}(x)\wedge \mathrm{MORDER}(x,j))$

(2') $\exists x(\mathrm{PERRO}(x)\wedge \mathrm{MORDER}(j,x))$

La diferencia entre ambos enunciados es trivial, pero es interesante hacer algunas consideraciones sobre ello. En primer lugar, vamos a convenir para mayor claridad en la exposición que los predicados del conjunto PRED se puedan parafrasear como predicados del español, como en el caso de PERRO y MORDER, dos predicados monádico y diádico, respectivamente, que se corresponden con el nombre común “perro” y con el verbo transitivo “morder”. En segundo lugar, vamos a introducir nombres propios en nuestro lenguaje para referirnos a elementos del dominio; este es el caso de la variable autointerpretada “juan” (que puede ser abreviada como “j”), a la que vamos a denominar *constante individual*. Finalmente, es interesante comprobar que la única diferencia entre (1') y (2') se encuentra en la combinación de los argumentos del predicado MORDER. Tanto la expresión MORDER(x,j) como la expresión MORDER(j,x) son fórmulas bien formadas de L. Su interpretación dependerá, por tanto, de la interpretación de sus componentes más simples: la variable cuantificada *x*, la constante *j* y el predicado MORDER. Pero la interpretación de morder, que es un predicado diádico, será un elemento del conjunto $\wp(D^2)$, es decir, un conjunto de pares ordenados, por lo que el orden en la combinación de los elementos constituyentes de ambas fórmulas será determinante en su interpretación. Esta característica propia del lenguaje L que hace que la interpretación de una fórmula dependa de su sintaxis la podemos y la debemos trasladar también al fragmento del lenguaje natural que estamos estudiando. Partimos pues de la hipótesis de que el principio de composicionalidad, tal y como quedó enunciado, es válido en idénticos términos para el lenguaje formal L y para el lenguaje natural (en este caso, el fragmento de las oraciones enunciativas del español).

El ejemplo aportado es ciertamente trivial y puede parecer que lo dicho hasta ahora no tiene ninguna relevancia para el análisis del lenguaje natural más allá de la complicación de la sintaxis del español con la forma lógica de las oraciones enunciativas. Sin embargo esto no es así. Hay numerosos casos de ambigüedad oracional que tan sólo pueden explicarse apelando a la forma lógica de las oraciones. Sea por ejemplo la oración:

(3) La ley castiga a quien ofende.

Esta oración tiene dos posibles interpretaciones. En una de ellas, el verbo transitivo “ofender” tiene como sujeto a “la ley” y como objeto directo a la referencia de la cláusula de relativo “a quien”. Podríamos parafrasearla del siguiente modo:

(4) La ley castiga a quien la ley ofende.

En la otra, “a quien” es el sujeto de “ofender” y el objeto directo no queda explícito en la oración. Su paráfrasis podría ser la siguiente:

(5) La ley castiga al ofensor.

Ambas interpretaciones se realizan sobre el significado unívoco de los términos léxicos constituyentes, por lo que tan sólo podemos explicar la ambigüedad apelando a criterios sintácticos, lo que queda implícito en las paráfrasis (4) y (5). Esto se puede ver con una gran claridad en las formas lógicas correspondientes:

(4') $\exists x(LEY(x) \wedge \forall y(OFENDER(x,y) \rightarrow CASTIGAR(x,y)))$

(5') $\exists x(LEY(x) \wedge \forall y(\exists z OFENDER(y,z) \rightarrow CASTIGAR(x,y)))$

Estas formas lógicas se corresponden con el análisis sintáctico estructural de (3). Para cada una de sus dos interpretaciones posibles hay un indicador sintagmático diferente que muestra las diversas relaciones entre los componentes de la oración de modo similar a lo que ocurre con el análisis sintáctico de (4') y (5') en el lenguaje L. Con lo que se establece una correspondencia entre la interpretación de la fórmula que declara la forma lógica de una oración enunciativa y la interpretación semántica de dicha oración; y, por tanto, también se establece esta relación entre las formas lógicas de las oraciones y sus indicadores sintagmáticos.

3.2 De la lógica de predicados al lenguaje intensional de Montague.

3.2.1 Estratificación y tipos lógicos.

La lógica de predicados ofrece, por tanto, un análisis en profundidad de un fragmento importante de una lengua natural, permitiendo explicar elegantemente ciertos casos de ambigüedad no trivial, como ocurre con la ambigüedad derivada de la cuantificación. Pero el lenguaje natural es ambiguo a causa de otros factores diferentes de la ambigüedad léxica o de la ambigüedad cuantificacional. La confusión de niveles en el lenguaje natural da lugar, seguramente, a las formas más habituales de ambigüedad. En los lenguajes formales, en cambio, esta confusión de niveles no se da, gracias a la teoría de los tipos. Y es precisamente de esta observación de donde nace el presupuesto básico del que partió R. Montague para proponer su análisis formal de un fragmento del inglés. Nosotros vamos a hacer una caracterización de los tipos categoriales para el lenguaje formal L que más tarde aplicaremos al análisis del fragmento de oraciones enunciativas del español que estamos tratando. Para ello vamos a definir un tipo de la siguiente manera:

1. *e* y *t* son tipos.
2. Si A y B son tipos entonces <A,B> es un tipo.
3. Nada más es un tipo.

Los tipos fundamentales *e* y *t* son los tipos propios de las entidades individuales y de las expresiones susceptibles de ser verdaderas o falsas, respectivamente. El tipo <A,B> es una función de los elementos del dominio de tipo A (D_A) en los elementos del dominio de tipo B (D_B), lo que da lugar al conjunto de entidades de tipo <A,B> ($D_{<A,B>}$). Por lo tanto, definimos el conjunto de las entidades de tipo <A,B> como el conjunto de todas las funciones de D_A en D_B: $D_b^{D_a}$.

La Tabla 1 muestra el tipo de algunas de las expresiones del lenguaje formal L, atendiendo a sus propiedades combinatorias dentro de la fórmula.

Expresiones del lenguaje L	**Tipos**
Fórmulas	t
Variables y constantes individuales	e
Predicados monádicos de primer orden	<e,t>
Predicados diádicos de primer orden	<e,<e,t>>
Predicados triádicos de primer orden	<e,<e,<e,t>>>
Predicados monádicos de segundo orden	<<e,t>,t>
Predicados diádicos de segundo orden	<<e,t>,<<e,t>,t>>
Cuantificadores y negador	<t,t>
Conectores lógicos	<t,<t,t>>

Tabla1

3.2.2 La reinterpretación de la predicación mediante λ-abstracción.

Mediante este concepto de los tipos categoriales es posible reinterpretar el fragmento del español analizado en términos formales, para lo cual es necesario introducir un nuevo operador que va a permitir abstraer sobre las diferentes clases de expresiones del lenguaje y de esta manera determinar su categoría y sus propiedades combinatorias. Es el denominado operador lambda o λ-abstracto. La sintaxis del operador lambda es similar a la de los operadores cuantificacionales, aunque de su interpretación semántica se colige en seguida que una expresión cuyo signo principal sea el operador lambda no tiene por qué ser ella misma una fórmula bien formada. La semántica del operador lambda queda definida del siguiente modo:

> Dada una variable u de tipo A y una expresión α de tipo B, $||\lambda u\alpha||^{V}$ es una función f de D_A en D_B tal que para cualquier $\sigma \in D_A$, $f(\sigma)=||\alpha||^{V'}$, donde v'=v excepto porque v'(u)=σ.

Es evidente por la propia definición de tipo y por la semántica del operador lambda que el tipo de la expresión $\lambda u\alpha$ antes descrita es <A,B>. Un ejemplo lo aclarará. Sean el predicado monádico de primer orden P y la variable individual x y la fórmula Px. Esta fórmula, como se puede comprobar en la Tabla 1, es de tipo t, en tanto que el predicado P es de tipo <e,t> y la variable x es de tipo e. Al abstraer sobre la fórmula Px podemos hacerlo sobre cualquiera de sus dos variables, por lo que podemos obtener cualquiera de las siguientes expresiones con sus tipos correspondientes:

(6) λx[Px] (cuyo tipo es <e,t>)

(7) λP[Px] (cuyo tipo es <<e,t>,t>)

Es decir, las expresiones (6) y (7) no son fórmulas puesto que no son de tipo t, sino que se trata de dos predicados monádicos de primer y segundo orden, respectivamente. Ahora bien, ambas dan lugar a sendas fórmulas de la lógica de predicados de segundo orden si se añade alguna expresión de la que puedan predicarse. En el caso de (6), la expresión debe ser de tipo e, es decir, el nombre de un individuo, como por ejemplo “juan”:

(8) λx[Px](j).

En el caso de (7), la expresión de la que se predique ha de ser de tipo <e,t>, es decir, un predicado monádico, como por ejemplo “correr”:

(9) λP[Px](CORRER).

Mediante el proceso denominado λ-conversión, que consiste en sustituir la variable ligada por el operador lambda por la expresión de la que se predica el λ-abstracto, se obtienen las fórmulas correspondientes de la lógica de predicados:

(8') Pj

(9') CORRER(x).

Es fácil demostrar que

$$||\lambda x[Px](j)||=||Pj||$$

y que

$$||\lambda P[Px](CORRER)||=||CORRER(x)||$$

atendiendo a la semántica del operador lambda, por lo que la relación entre la lógica de predicados de primer orden y el lenguaje con abstracción y teoría de los tipos introducido es evidente.

Esto nos permite establecer la denotación del Sintagma Nominal de una oración (a partir de ahora SN) en relación con el Sintagma Verbal (SV) mediante la abstracción del predicado en la fórmula que se corresponde con la forma lógica de una oración enunciativa cualquiera. Valga como ejemplo la siguiente oración simple en la que su SN es un nombre propio del que se predica una acción mediante un verbo intransitivo:

(10) Juan corre.

La oración enunciativa (10) puede formalizarse mediante un predicado monádico de primer orden y una constante individual. de hecho su forma lógica es:

(10') CORRER(juan)

Si en (10') abstraemos sobre el predicado obtenemos:

(11) $\lambda P[P(juan)](CORRER)$

Por λ-conversión sabemos que la denotación de (10') y (11) es la misma, es decir, que $||CORRER(juan)||=||\lambda P[P(juan)](CORRER)||$. Si en (11) prescindimos del objeto sobre el que se abstrae, el predicado CORRER, y nos quedamos sólo con el λ-abstracto, tendremos que convenir que $||juan||=||\lambda P[P(juan)]||$, o lo que es lo mismo, que la denotación de un nombre propio es idéntica a la denotación del conjunto de sus propiedades. Por lo tanto, el tipo categorial de los nombres propios de individuos del discurso es el mismo que el de los predicados monádicos de segundo orden: $<<e,t>,t>$. Más adelante veremos que esto no entra en contradicción lógica con la asignación de tipo que se hizo en primer lugar a los términos que denotan individuos, sino que es una consecuencia lógica de la misma.

La λ-abstracción permite, por tanto, reinterpretar la predicación en lenguaje natural. Esta reinterpretación se halla en la base de lo que R. Montague denominó el *lenguaje intensional*, fundamental para el análisis formal del fragmento de los enunciados de un lenguaje natural. En pocas palabras, lo que Montague propone es un lenguaje intermedio entre las estructuras sintácticas del fragmento del lenguaje natural estudiado y sus correspondientes formas lógicas. Este lenguaje intermedio es el lenguaje intensional, en el que a cada estructura sintáctica le corresponde una interpretación no ambigua en un modelo matemático y para el que se puede definir un cálculo similar al cálculo de secuentes de Gentzen: el cálculo Lambek.

3.3 El cálculo Lambek.

3.3.1 Categorías sintácticas y tipos categoriales. Reglas de un cálculo Lambek básico.

Mediante la λ-abstracción, Montague redefine la relación entre el SN sujeto y el SV predicado de modo que todos los componentes sintácticos de uno y otro tengan asignados unos tipos categoriales en relación con su denotación y con sus propiedades combinatorias dentro de la oración. Esto permite asignar tipos a las categorías sintácticas fundamentales, tal y como queda expuesto en la Tabla 2.

Categorías sintácticas	**Tipos**
Oración (O)	t
Nombre Propio (NP)	e (o bien <<e,t>,t>)
Nombre Común (NC)	<e,t>
Verbo Intransitivo (VI)	<e,t>
Verbo Transitivo (VT)	<e,<e,t>>
Determinante (Det)	<<e,t>,<<e,t>,t>>
Adjetivo (Adj)	<<e,t>,<e,t>>
Adverbio (Adv)	<<e,t>,<e,t>>
Sintagma Nominal (SN)	<<e,t>,t>
Sintagma Verbal (SV)	<e,t>

Tabla 2

A partir de aquí es posible definir un cálculo de categorías que permite deducir estructuras sintácticas básicas. Las siguientes reglas definen un cálculo Lambek de esta clase.

1. APLICACIÓN: $$\frac{A \qquad \langle A,B\rangle}{B}$$

2. COMPOSICIÓN: $$\frac{\langle A,B\rangle \qquad \langle B,C\rangle}{\langle A,C\rangle}$$

3. ASCENSO: $$\frac{A}{\langle\langle A,B\rangle,B\rangle}$$

4. DESCENSO: $$\frac{\langle\langle\langle A,B\rangle,B\rangle,C\rangle}{\langle A,C\rangle}$$

5. PARAMETRIZACIÓN: $$\frac{\langle B,C\rangle}{\langle\langle A,B\rangle,\langle A,C\rangle\rangle}$$

Mediante estas reglas del cálculo Lambek se puede demostrar la corrección sintáctica de ciertas reglas de estructura sintagmática propias de una gramática del fragmento del español que estamos analizando. Algunos ejemplos de estas reglas son los siguientes:

O → SN SV
SN → NP
SN → Det NC
NC → NC (Adj)

La corrección de estas reglas de estructura sintagmática viene dada porque el tipo de los constituyentes sintácticos que aparecen a la izquierda de la flecha coincide en cada caso con el tipo de la secuencia que aparece a la derecha, lo que se puede demostrar fácilmente mediante las reglas anteriores y la correspondencia establecida en la Tabla 2 entre las categorías sintácticas y sus tipos. De hecho, algunos de los casos son triviales, como la correspondencia entre los tipos del SN y del NP.

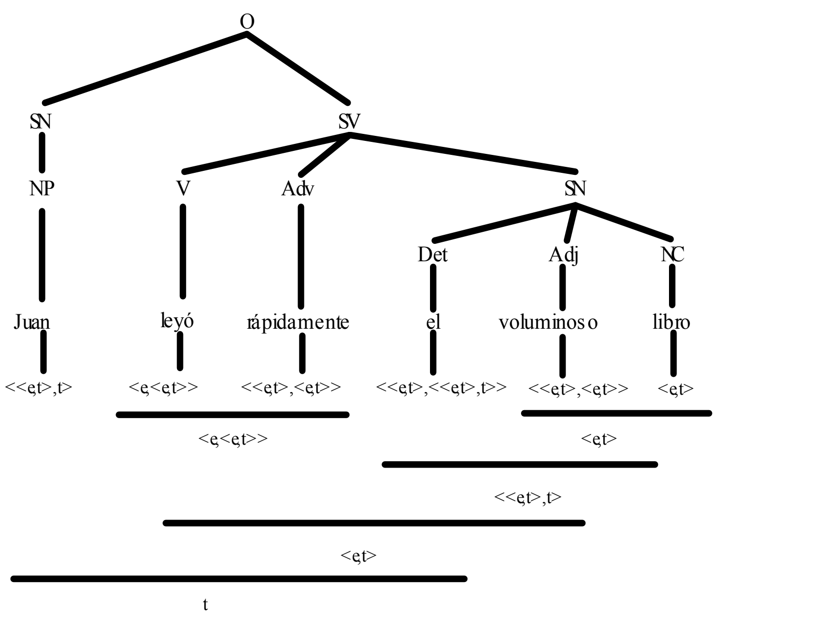

Figura 1

La figura 1 muestra una deducción mediante las reglas de este cálculo a partir de un indicador sintagmático previo de la oración *Juan leyó rápidamente el voluminoso libro*.

3.3.2 Interpretación funcional de los tipos categoriales y bases de datos léxicas.

Los tipos categoriales anteriores pueden interpretarse funcionalmente, siendo asimilables a la implicación lógica. Como ya vimos, el tipo <A,B> se interpreta como una función de D_A en D_B. Esta función puede caracterizarse como una fórmula de lógica de proposiciones ($A \rightarrow B$), y por lo tanto las reglas del cálculo Lambek se pueden reinterpretar también como reglas de un cálculo deductivo natural de lógica de proposiciones. Así, las reglas de Aplicación y de Composición se corresponden con el Modus Ponens y con la regla de la Transitividad de la Implicación, respectivamente. Las reglas de Ascenso, Descenso y Parametrización son fácilmente derivables en la lógica proposicional clásica. La regla de ascenso, por ejemplo, se corresponde con la siguiente ley de la lógica proposicional:

$$(e \rightarrow ((e \rightarrow t) \rightarrow t))$$

Esto explica el "ascenso de tipo" de los nombres propios de individuos desde el tipo original *e* hasta el tipo <<e,t>,t>.

Esta interpretación funcional de los tipos permite crear un cálculo basado en bases de datos cuyos componentes son entradas léxicas de una lengua natural acompañadas de su correspondiente tipo funcional. Mediante unas pocas reglas básicas, este cálculo sirve para reconstruir el proceso de codificación y decodificación en fragmentos amplios de discurso. Para definirlo es necesario definir previamente un marco lógico al que denominamos Sistema Deductivo Etiquetado (L*abelled* D*eductive* S*ystem*).

Definimos un LDS como un par $<L_t,\Gamma>$, donde L_t es una determinada lógica y Γ es un álgebra en la que se han definido algunas operaciones sobre etiquetas. La lógica L_t es aquel sistema formal que más se adecua a nuestros propósitos de análisis: el fragmento implicacional de una lógica proposicional lineal de la relevancia. De este modo, una *fórmula bien formada* (fbf.) de la lógica L_t se define como toda aquella fórmula que satisface las siguientes condiciones:

1. *e* y *t* son fbfs.
2. Si A y B son fbfs. entonces $A \rightarrow B$ es una fbf.
3. Nada más es una fbf.

Una *expresión correcta* (o simplemente una *expresión*) es un par α:A, donde $\alpha \in \Gamma$ y A es una fbf. Una *base de datos* es un conjunto de expresiones correctas.

Una deducción Δ en LDS consiste en obtener a partir de ciertas asunciones del tipo α_1:A_1,...α_n:A_n, donde α_1,...α_n son etiquetas ψ A_1,...A_n son fbfs., una expresión de la forma $\phi(\alpha_1,...\alpha_n)$:$\tau$, donde $\phi(\alpha_1,...\alpha_n)$ es una etiqueta obtenida mediante procesos combinatorios a partir de α_1,...α_n y τ es la fórmula que representa el tipo categorial τ obtenido a partir de A_1,...A_n mediante algunas reglas lógicas definidas en L_t.

Todas las asunciones deben ser usadas en una deducción en LDS (requisito de la relevancia), debiendo ser usadas solamente una vez (requisito de la linealidad). Cuando obtenemos mediante deducción una expresión de la forma α:τ en una base de datos δ_κ y los requisitos de relevancia y linealidad se cumplen decimos que la base de datos δ_κ está cerrada. Es posible abrir en cualquier momento de la deducción una nueva base de datos mediante la pertinente asunción. En ese caso decimos que la nueva base de datos se encuentra *anidada* en el seno de la base de datos previa.

LDS permite construir bases de datos en las que toda la información léxica de una lengua natural está disponible en un determinado momento para el hablante/oyente. Esta información consiste principalmente en una etiqueta y su correspondiente tipo funcional (y a veces también un orden) por cada entrada léxica. De este modo, dada una base de datos cualquiera, es posible manipular la información que contiene mediante reglas definidas en función de la lógica elegida, incrementando de esta forma la propia base de datos. Y como cada tipo funcional es interpretado como un tipo categorial lógico, podemos decir que LDS provee un análisis del lenguaje natural que asigna como interpretación de cada secuencia de palabras gramaticalmente correcta una base de datos estructurada.

Las bases de datos concebidas de este modo son parecidas en muchos aspectos a los denominados *conjuntos modelos,* contrapartida restringida de los mundos posibles en lógica modal. Por ejemplo, se las puede considerar como momentos, interpretando las bases de datos de una manera temporal, o, incluso, como estados de conocimiento, sirviendo como modelo epistemológico de la comunicación lingüística. En cualquier caso, las expresiones pertenecientes a una base de datos en LDS se comportan como los teoremas de un conjunto de enunciados *à la Hintikka*, por lo que diremos que para cada expresión α_i:A_i que aparezca en una base de datos δ_κ, $\delta_\kappa \vdash \alpha_i$:$A_i$.

Varias bases de datos pueden estar relacionadas en una deducción. Esta relación es similar a la relación de accesibilidad (o alternatividad) entre conjuntos modelos en lógica modal. Sea $\mathfrak{R}$ dicha relación. Sea $\Delta_e(\delta_k)$ el dominio de la base de datos δ_k, es decir, el conjunto de referentes de las expresiones de tipo *e* en δ_k. Decimos que dadas dos bases de datos relacionadas mediante ka relación $\mathfrak{R}$, la población de la primera bases de datos es heredada por la segunda. Este requisito se conoce en lógica modal como *requisito de dominios anidados* y puede ser definido simbólicamente de la siguiente manera:

$$\forall \delta_i \delta_k [\delta_i \Re \delta_k \Rightarrow \Delta_e(\delta_i) \subseteq \Delta_e(\delta_k)]$$

Hechas todas estas consideraciones, podemos establecer las reglas de inferencia básicas del cálculo en LDS de la siguiente manera:

R1. Aplicación: Dadas dos expresiones $\alpha{:}A \rightarrow B \in \delta_k$ y $\beta{:}A \in \delta_n$, podemos añadir a la actual base de datos δ_i una expresión de la forma $\alpha(\beta){:}B$ sii o bien $\delta_k \Re \delta_n \Re \delta_i$ o bien $\delta_n \Re \delta_k \Re \delta_i$.

R2. Composición: Dadas dos expresiones $\alpha{:}A \rightarrow B \in \delta_k$ y $\beta{:}B \rightarrow C \in \delta_n$, podemos añadir a la actual base de datos δ_i una expresión de la forma $(\alpha \quad \beta){:}A \rightarrow C$ sii o bien $\delta_k \Re \delta_n \Re \delta_i$ o bien $\delta_n \Re \delta_k \Re \delta_i$.

R3. λ-Abstracción: Dadas dos expresiones $x{:}A \in \delta_n$ y $\alpha(x){:}B \in \delta_n$, donde $x{:}A$ es la única asunción de δ_n y $\alpha(x){:}B$ ha sido derivada en δ_n, si $\delta_k \Re \delta_n$ entonces $\lambda x \alpha(x){:}A \rightarrow B \in \delta_k$ y δ_n se cierra.

R4. λ-Conversión: Si $\lambda x[\alpha(x)](\beta){:}A \in \delta_k$ entonces $\alpha(\beta){:}A \in \delta_k$.

R5. Reutilización: Dada una expresión de la forma $\alpha{:}A \in \Delta_e(\delta_k)$ podemos añadirla a otra base de datos δ_n sii $\delta_k \Re \delta_n$.

Las reglas de Aplicación y λ-Abstracción son las formas etiquetadas de las reglas de la lógica clásica proposicional de eliminación de la implicación (Modus Ponens) e introducción de la implicación (Teorema de la Deducción), mientras que la regla de λConversión opera exclusivamente sobre la etiqueta sin modificar la fórmula correspondiente. La regla de Composición corresponde al principio de la transitividad de la implicación lógica.

Es importante la caracterización de la relación $\Re$ para hacer que el sistema sea más o menos potente. La mejor caracterización de $\Re$ para nuestros propósitos es hacer de esta relación un orden parcial, es decir: $\Re$ es una relación reflexiva, antisimétrica, transitiva y conexa. Esta definición acerca el concepto de las bases de datos en LDS al concepto kripkeano de *estado de información*, de modo que una base de datos contiene no solamente la información que aportan las expresiones que la componen, sino también la información que se deriva de estas expresiones y de las expresiones pertenecientes a aquellas bases de datos relacionadas con la actual. Un ejemplo simple de deducción en LDS nos lo da la figura 2.

δ_1

1. juan': e	Asunción
2. leyó': $(e \rightarrow (e \rightarrow t))$	Asunción
3. rápidamente': $((e \rightarrow t) \rightarrow (e \rightarrow t))$	Asunción
4. el': $((e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t))$	Asunción
5. voluminoso': $((e \rightarrow t) \rightarrow (e \rightarrow t))$	Asunción
6. libro': $(e \rightarrow t)$	Asunción
7. voluminoso(libro)': $(e \rightarrow t)$	Aplicación 5,6
8. el(voluminoso(libro))': $((e \rightarrow t) \rightarrow t)$	Aplicación 4,7
9. (leyó rápidamente)': $(e \rightarrow (e \rightarrow t))$	Composición 2,3
10. (leyó rápidamente)(el(voluminoso(libro)))': $(e \rightarrow t)$	Composición 9,8
11. (leyó rápidamente)(el(voluminoso(libro)))(juan)': t	Aplicación 10,1

Figura 2

En la deducción de la figura 2 toda la información necesaria se encuentra codificada en una única base de datos δ_1. Sin embargo, en el discurso es habitual que la información no sea tan directa como en este caso. LDS permite tratar con fragmentos del discurso más extensos que una única oración en los que la información se encuentra en bases de datos relacionadas. Esto hace que sea una herramienta eficaz para el tratamiento adecuado de las relaciones de presuposición que pueden darse en el discurso en lenguaje natural, ofreciendo un análisis formal de fragmentos de discurso en los que aparecen preguntas y respuestas, anáforas o elipsis, por ejemplo.