http://revistas.userena.cl/index.php/logos/index
Issn Impreso: 0716- 7520
Issn Electrónico: 0719-3262

Nota

Sintaxis Dinámica: un modelo de procesamiento sintáctico-semántico

Dynamic Syntax: a Model of Syntactic-Semantic Processing

Recibido: Agosto 2014 **Aceptado:** Diciembre 2014 **Publicado:** Junio 2015

Nicolás Saavedra

Pontificia Universidad Católica de Chile
Chile

lsaavedrag@uc.cl

Resumen: Se presentan los aspectos básicos de un modelo de procesamiento sintáctico-semántico denominado *Sintaxis Dinámica*. Para cumplir con este objetivo, presentamos definiciones e ilustraciones de los componentes esenciales del metalenguaje, y ejemplificamos este modelamiento dinámico presentando el proceso sintáctico completo de composición del significado de una oración simple. Finalmente, se señalan sucintamente algunas características prominentes del modelo.

Palabras Clave: sintaxis - semántica - procesamiento - composicionalidad

Abstract: We present the basic aspects of *Dynamic Syntax*, a model of syntactic and semantic processing. For the achievement of this goal, we develop several definitions and illustrations of the essential components of the metalanguage, and we exemplify this dynamic modeling through the presentation of the complete syntactic process that produces the meaning composition of a simple sentence. Finally, we signal succinctly some prominent characteristics of the model.

Keywords: syntax - semantics - processing - compositionality

Citación: Saavedra, N.(2015). Sintaxis Dinámica: un modelo de procesamiento sintáctico-semántico. *Logos: Revista de Lingüística, Filosofía y Literatura,* 25(1), 87-97. DOI: 10.15443/RL2507
Dirección Postal: Av. Vicuña Mackenna 4860,Macul,Facultad de Letras, Pontificia Universidad Católica de Chile, Santiago
DOI: dx.doi.org/10.15443/RL2507

1. Introducción

En los siguientes desarrollos introduciremos los fundamentos y el lenguaje descriptivo básico del marco de trabajo denominado Sintaxis Dinámica (SD), cuyas referencias programáticas fundamentales corresponden, a la fecha, a Kempson, Meyer-Viol y Gabbay (2001); Marten (2002); Cann, Kempson y Marten (2005); y a Kempson, Gregoromichelaki y Howes (2011). Una de las características generales más innovadoras de este modelo atañe a la concepción de 'sintaxis' que se propone en él. A saber: que la sintaxis de una determinada lengua natural corresponde al conjunto de estrategias usadas en dicha lengua para construir interpretaciones "paso a paso", a medida que la información se va acumulando incrementalmente. Como veremos, este desplazamiento desde nociones más tradicionales implica también un cambio notable en el lenguaje descriptivo usado; al prescindir de descripciones estructurales "estáticas", la SD ha podido prescindir también de varios de los constructos categoriales que han sido una parte esencial del arsenal descriptivo de buena parte de la teorías sintácticas contemporáneas.

1.1. Una inadecuación descriptiva del análisis sintáctico tradicional

Los análisis tradicionales codifican fielmente el orden de palabras de la oración (e.g. el análisis sintáctico arbóreo en teoría sintáctica chomskiana). Las unidades básicas en estas aproximaciones suelen ser las *palabras*, consideradas en tanto átomos sintácticos de algún sistema particular de categorías (e.g. verbo, nominal o sustantivo, determinante, etc.). Además, estos análisis no codifican las fases del proceso de construcción del producto final, y por esto, cualquier interpretación asignada a la estructura analizada se restringe al objeto sintáctico acabado. Esto no quiere decir que, *a posteriori*, sea imposible consignar o describir la contribución interpretativa de cada palabra incluida en el análisis; lo que se afirma, más bien, es que si existe alguna contribución interpretativa que es efecto específico del *proceso* de construcción del objeto final, entonces las representaciones de este tipo no codifican dicha contribución. Más sintéticamente: los análisis de este tipo representan exclusivamente productos, no procesos.

2. El proceso sintáctico en SD

Para iniciar una caracterización básica de este modelo consideraremos la siguiente representación:

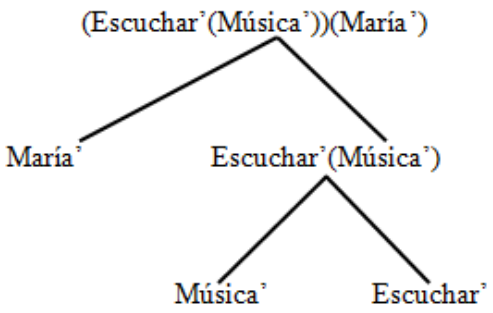

Figura 1. Análisis simplificado en SD de *María escuchó música*.

Podemos notar, en primer lugar, que los nodos terminales no codifican el orden de palabras: la oración analizada es *María escuchó música* y no **María música escuchar*. Esto supone, por tanto, que los nodos terminales no están decorados por palabras, sino por algún otro tipo de unidad. Consideremos ahora la siguiente secuencia de esquemas:

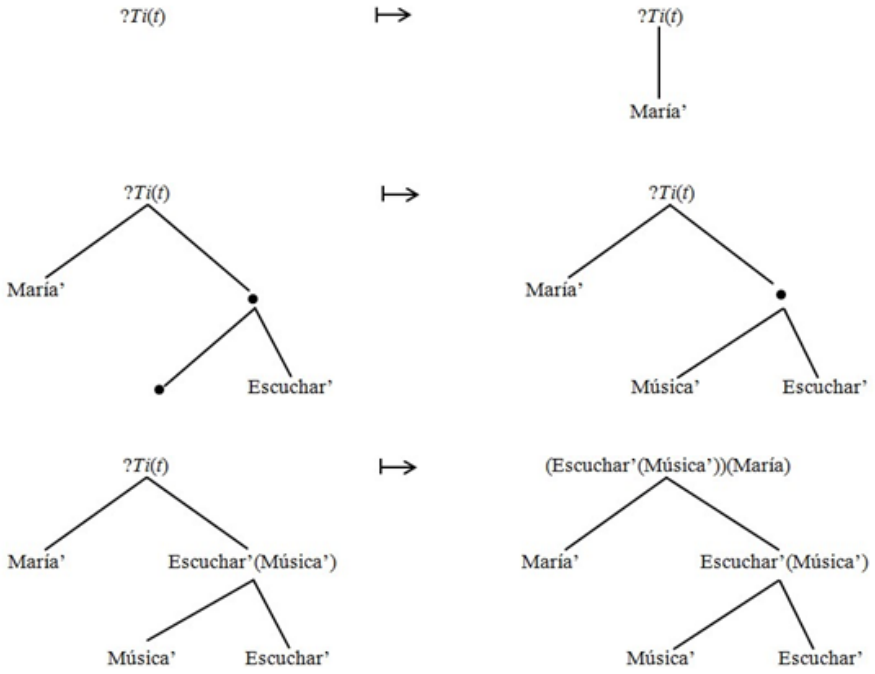

Figura 2. Secuencia simplificada de árboles para *María escuchó música*.

El último árbol de la serie es idéntico al esquema en (1), y la secuencia de árboles que lo precede representa los pasos en la composición de la estructura final. De este modo, la secuencia completa modela el proceso de composición de una cierta estructura conceptual asociada a *María escuchó música*. En este modelamiento dinámico, se asume que el punto inicial de cualquier análisis en contexto o situación de uso corresponde al requerimiento de establecer alguna fórmula proposicional como interpretación (en consonancia con Sperber y Wilson (1995)). Este requerimiento inicial puede proyectar otros sub-requerimientos (o sub-metas) en la medida en que se agrega información nueva (Sperber & Wilson, 1995). Por ejemplo, en (2), el primer paso de la secuencia representa precisamente este requerimiento inicial de construir un árbol con contenido proposicional (requerimiento representado por el símbolo ?*Ti(t)*). En el segundo paso, se asigna una posición estructural (acaso provisoria) a la información provista por *María*, representada como María', y en el tercero se hace lo mismo con *escucha*. En este paso se representa, además, un punto hipotético: la existencia de una posición estructural para un tercer constituyente, que forma una sub-estructura interpretativa junto con la información provista por *escucha*. El cuarto paso confirma esta hipótesis, al incorporar la información provista por *música*, y en los pasos subsiguientes se componen, respectivamente, la sub-estructuras ya mencionadas, a saber: Escuchar'(Música'), y finalmente, la estructura proposicional completa (Escuchar'(Música'))María', que satisface al requerimiento inicial *?Ti(t)*. Puede observarse que los componentes del análisis son unidades informativas o, simplemente, *conceptos* (para incorporar la denominación privilegiada en SD), y no palabras. De este modo, María' corresponde el concepto asignado a la palabra *María,* Escuchar' representa al concepto asignado a *escuchó* (obviando, de momento, consideraciones sobre la flexión temporal), y Música' representa la interpretación asignada a música. También podemos notar que el análisis registra, indirectamente, el orden de palabras en el orden en que la información es agregada en cada paso de la composición. El producto final, no obstante, sólo registra la estructura interpretativa del caso.

3. Árboles en crecimiento: las herramientas de la SD

El dinamismo descrito en la sección previa es representado a través de esquemas arbóreos *en crecimiento*. Esto significa que, en SD, es particularmente importante disponer de definiciones claras acerca de qué significa ser un árbol, qué significa ser un árbol parcial, y qué significa que exista crecimiento de un árbol a otro (Cann *et al.*, 2005).

3.1. Nodos y anotaciones

En particular, las representaciones de conceptos (simples o complejos) que decoran nodos, reciben el nombre de *Fórmulas*, y se expresan como valores del predicado *Fo*. Así, *Fo*(Cantar'(Juan')) representa información semántica que podría ser expresada haciendo uso de la oración *Juan cantó* (nuevamente: sin considerar, de momento, el tratamiento del tiempo gramatical). Las fórmulas son parte del conjunto de *Rótulos (Labels)* con que un nodo puede aparecer anotado. Además del rótulo correspondiente a una fórmula particular, existe un rótulo que entrega información sobre el *Tipo* de la fórmula. El tipo es la categoría semántica de la fórmula, y la asocia a una clase particular de denotación (Cann et al., 2005; Marten, 2002). Una fórmula de tipo *t* denota un valor de verdad: *Fo*(Cantar'(Juan')) y *Fo*((Escuchar'(Música'))(María)) son ejemplos de fórmulas de tipo *t*. En tanto, *Fo*(María') es una fórmula de tipo *e*, que denota alguna entidad. A *Fo*(Cantar') podemos asignarle el tipo $e \rightarrow t$, correspondiente a un predicado monádico, pues cuando es combinado con un término individual (e.g. como el expresado por *Fo*(Juan')), desarrolla una proposición (e.g. como la expresada por *Fo*(Cantar'(Juan'))); además, se trata de un predicado que denota un conjunto (e.g. el conjunto de entidades que cantan). Los tipos como $e \rightarrow t$, que poseen un formato condicional, reciben el nombre de tipos *funtores*, y los tipos como *t* y *e*, reciben el nombre de tipos *argumentales*. La tabla que aparece a continuación (basada en Cann *et al.*, 2005) contiene una lista de los tipos más comúnmente usados en SD, junto con sus respectivas descripciones y ejemplos:

Tipo	Descripción	Ejemplos
Ti(e)	Término individual	*Fo*(María'), *Fo*(ε, x, Estudiante'(x))
Ti(t)	Proposición	*Fo*(Cantar'(Juan')), *Fo*(Escuchar'(Música')(María))
Ti(e → t)	Predicado (monádico)	*Fo*(Escuchar'(Música')), *Fo*(Correr')
Ti(e → (e → t))	Predicado (diádico)	*Fo*(Escuchar'), *Fo*(Entregar-a'(Juan'))
Ti(e → (e → (e → t)))	Predicado (triádico)	*Fo*(Entregar'), *Fo*(Poner')
Ti(t→ (e → t))	Predicado (proposicional)	*Fo*(Creer'), *Fo*(Decir')
Ti(cn)	Nominal (nombre común)	*Fo*(x, Estudiante'(x)), *Fo*(y, Padre'(Juan')(y))
Ti(cn → e)	Cuantificador	*Fo*(λP. ε, P)

Tabla 1. Tipos semánticos comúnmente usados en SD

3.2. Requerimientos y crecimiento

En general, cualquier componente que motiva el crecimiento sucesivo de un árbol en SD es denominado *Requerimiento*. Esta motivación corresponde, más concretamente, a la necesidad de agregar información sub-especificada. En términos propiamente representacionales, un requerimiento establece el imperativo de proveer una instancia de un rótulo en el nodo en desarrollo, y se muestra como un símbolo de interrogación junto al rótulo que se requiere

instanciado (Cann et al., 2005; Marten, 2002). Así, por ejemplo, el requerimiento *?Ti(t)* en un determinado nodo puede leerse como "desarrolla contenido proposicional en este punto" o, representacionalmente, "deriva una fórmula de tipo t en este punto". En cambio, un rótulo *Ti(t)* en un determinado nodo puede leerse como "en este punto hay contenido proposicional" o, representacionalmente, "se ha derivado una fórmula tipo t en este punto de procesamiento".

Una estrategia común para satisfacer un determinado requerimiento inicial en SD, es descomponer este requerimiento en sub-metas cuya satisfacción sucesiva produce la satisfacción del requerimiento inicial. Así:

> en cualquier etapa de la derivación de un árbol (...), alguna información podrá estar ya establecida, mientras otras metas siguen pendientes. La derivación está completa si, después de que toda la información del léxico se ha incorporado en el árbol, todos los requerimientos han sido satisfechos (Cann et al., 2005).

A cada etapa en una derivación corresponde el desarrollo de un determinado nodo. El nodo en desarrollo se identifica haciendo uso de un *puntero*, representado por el símbolo '◊'. Tanto la denominación 'puntero' como el símbolo '◊' forman parte del lenguaje del que hace uso la SD para describir árboles (Kempson et al., 2001). La interpretación de este artefacto descriptivo es la siguiente: cuando el puntero se sitúa junto a un nodo que posee un determinado requerimiento, se está estableciendo que tal requerimiento corresponde a la tarea llevada a cabo en ese punto actual de desarrollo arbóreo (Cann et al., 2005; Marten 2002). Por ejemplo, cuando el estado de desarrollo de un árbol corresponde al requerimiento inicial *?Ti(t),* entonces podemos describir esa etapa como *?Ti(t),* ◊. En el siguiente esquema es posible observar varios requerimientos activos, destacados en gris. Obsérvese la posición del puntero en el nodo funtor inmediatamente dominado por el nodo raíz:

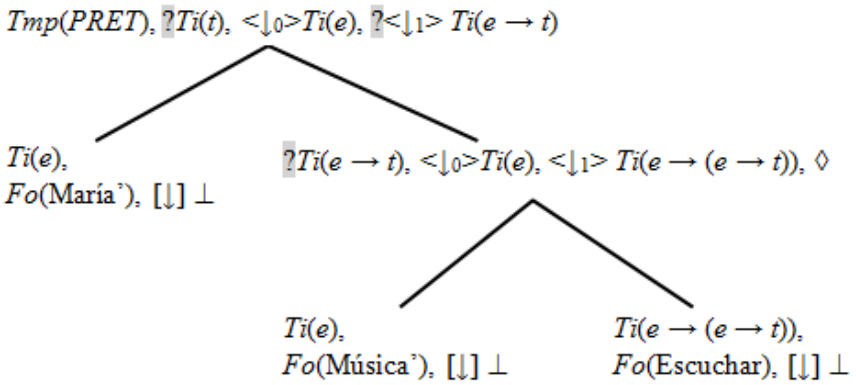

Figura 3. Requerimientos activos en un árbol en desarrollo.

3.3. Lógica arbórea

Los principios y las reglas que guían y restringen el crecimiento arbóreo en SD, están basadas en el metalenguaje formal denominado *Lógica de Árboles Finitos* (LDAF) (Blackburn & Meyer-Viol, 1994; Kempson *et al.*, 2001). En este marco de trabajo, cada nodo puede ser identificado por un índice numérico. Por convención, el nodo hija (*daughter node*) izquierdo de un nodo *n* recibe una asignación numérica *n0* (e.g. terminado en 0) y la hija derecha recibe una asignación numérica *n*1 (e.g. terminado en 1), como puede apreciarse en el siguiente esquema:

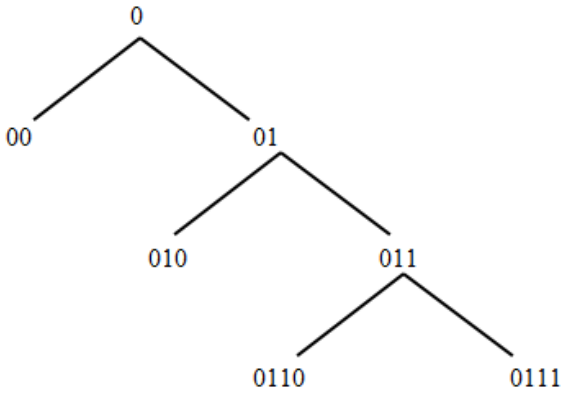

Figura 4. Identificación numérica de nodos.

Las relaciones de dominancia son aquí las acostumbradas. Por ejemplo: el nodo 0 domina a todos los demás nodos del árbol, pero *sólo domina inmediatamente* a los nodos 00 y 01. Esta información numérica puede formar parte de los rótulos de un árbol, junto con las anotaciones de tipo y fórmula. La anotación numérica se expresa usando el predicado *Tn* (de *tree node*, tr.: *nodo arbóreo*) que toma como valor algún índice numérico (Marten, 2002).

En general, la siguiente convención se adopta para todos los análisis en SD:

3.4. Convención de distribución de argumentos y funtores:

Un nodo hija izquierdo está siempre anotado con una fórmula argumental y un nodo hija derecho está siempre anotado con una fórmula funtor (Cann *et al.,* 2005).

En SD, la posibilidad de describir relaciones entre nodos es importante por dos motivos: primero, porque permite establecer que cierta información se encuentra en un determinado nodo madre o hija, *desde el "punto de vista" de un nodo distinto*, y segundo, porque permite "expresar requerimientos que necesitan ser satisfechos en un nodo distinto al nodo en desarrollo" (Cann *et al.*, 2005: 39). Esta relaciones son descritas haciendo uso de dos modalidades básicas: la *relación de hija (daughter relation)* es descrita haciendo uso del símbolo '<↓>', "abajo", y la *relación de madre (mother relation)* es descrita haciendo uso del símbolo '<↑>', "arriba". Estos símbolos pueden usarse en el mismo formato en que han sido recién introducidos, o bien, pueden ir acompañados de un subíndice numérico. Este último recurso es necesario si queremos distinguir entre nodos argumentales o funtores. Por ejemplo, $<\downarrow_0>$ refiere a una hija argumental, y $<\downarrow_1>$ refiere a una hija funtor. Lo mismo aplica para la relación de madre: $<\uparrow_0>$ refiere a la madre de un nodo argumental, y $<\uparrow_1>$ refiere a la madre de un nodo funtor (Cann *et al.*, 2005; Marten, 2002). Supongamos que en un nodo n encontramos una expresión como $<\downarrow_0>Ti(e)$. El modo de leer esta descripción es el siguiente: "existe un nodo hija argumental que el nodo *n* domina inmediatamente, y ese nodo hija argumental está anotado con la información de tipo *Ti(e)*". Por otra parte, los operadores modales <↓> y <↑> pueden ser iterados: <↓><↓>, <↓><↑>, etc. Esto permite identificar las propiedades de un nodo desde otro nodo que se encuentra a una distancia mediata. En el siguiente esquema se destacan en gris los operadores modales:

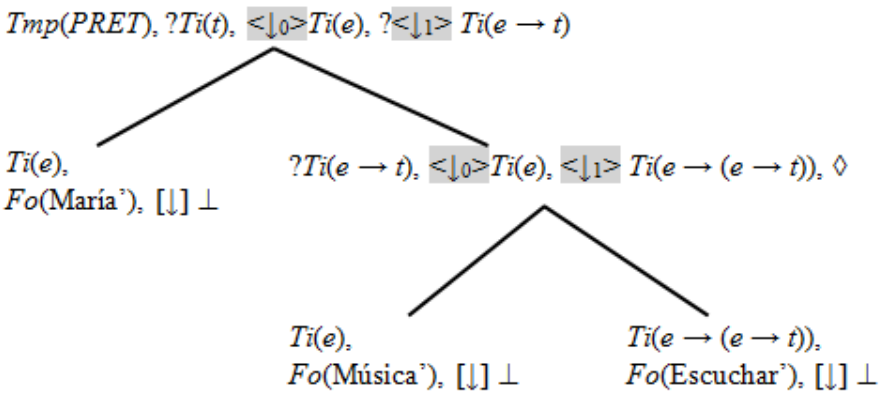

Figura 5. Operadores modales en un árbol en desarrollo.

3.4. Otros símbolos

Otros símbolos usados en SD para la descripción de árboles son los siguientes:

Restricción de base: [↓] ⊥
La restricción de base, anotada en un determinado nodo, "cierra" el crecimiento arbóreo en dicho punto, estableciendo que el nodo en cuestión es terminal (Cann *et al.*, 2005).

"Notas promisorias": *Tmp(PRET), Indef(+), +Q*, entre otras
De vez en cuando se usan símbolos provisionales que "valen" por estructuras no definidas. Por ejemplo, más adelante encontraremos el símbolo *Tmp*(PRET), el cual, anotado en el nodo raíz, indica que estamos ante una predicación de pretérito simple. En SD, se considera a dichos símbolos como "notas promisorias", esto es: rótulos que sirven para indicar, informalmente, algún tipo de contenido que debiera ser descrito en términos explícitos (Cann *et al.*, 2005). A modo de ejemplo, para el caso de *Tmp*(PRET), y otras notas temporales posibles, se ha propuesto reemplazar dichas notas con esquemas temporales basados en el trabajo de Reichenbach (e.g. Cann, 2011). El siguiente esquema ilustra el uso de estos símbolos especiales (incluyendo el puntero):

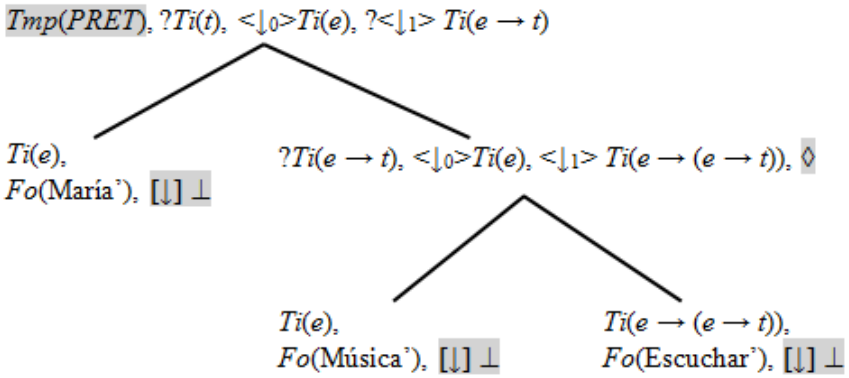

Figura 6. Símbolos especiales en un árbol en desarrollo.

4. Computación sintáctica en SD

4.1. Información léxica

La información léxica en SD es de naturaleza esencialmente procedimental. Más específicamente: las entradas léxicas en SD son computaciones sintácticas (e.g. sub-rutinas en la derivación de estructura). Esto posibilita que el procesamiento de una palabra pueda agregar información a nodos no terminales, agregar nuevos requerimientos, o construir árboles parciales, y que incluso pueda inducir la creación de estructura proposicional completa (Cann et al., 2005; Marten, 2002). Las entradas léxicas poseen el siguiente formato general:

SI	?*Ti(X)*	Activador (Trigger)
ENTONCES	…	Acciones
EN OTRO CASO (E.O.C)	…	Enunciado E.O.C (Elsewhere Statement)

Tabla 2. Formato de las entradas léxicas en SD.

La primera línea describe el contexto apropiado para el procesamiento de una palabra, o *Activador* (e.g. en este caso, corresponde al requerimiento de una fórmula de tipo *X*, *?Ti(X)*). Si el contexto requerido por el Activador existe, entonces el consecuente del condicional (segunda línea) ejecuta una serie de *Acciones Léxicas*. Si el contexto requerido por el Activador no es satisfecho, el enunciado E.O.C de la tercera línea induce acciones alternativas. La acción alternativa más usual corresponde a Abortar, que interrumpe el proceso sintáctico. Las Acciones Léxicas incluyen, típicamente, los siguientes predicados: **crear(…):** crear un nuevo nodo; **ir(…):** mover el puntero al nodo especificado (desplazarse de un punto a otro), y **anotar(…):** anotar un nodo con la información especificada. El siguiente ejemplo corresponde a la entrada léxica de *María*:

María	SI	*?Ti(e)*	Activador (Trigger)
	ENTONCES	anotar *Ti(e)*, *Fo*(María'), [↓] ⊥	Acciones: anotación
	EN OTRO CASO (E.O.C)	Abortar	Error: Enunciado E.O.C (Tr. *Elsewhere Statement*)

Tabla 3. Entrada léxica de *María* en SD.

4.2. Reglas computacionales básicas

Las reglas computacionales básicas del proceso sintáctico corresponden a: Introducción, Predicción, Simplificación, Completación y Eliminación (Cann *et al.*, 2005; Marten, 2002). La regla de **Introducción** expande un requerimiento original en sub-requerimientos, dentro de un mismo nodo. Por ejemplo: la expansión del requerimiento original *?Ti(t)*, en los sub-requerimientos ? < $\downarrow_0$> *Ti(e)* y ? <$\downarrow_1$> *Ti(e → t)*. La regla de **Predicción** crea nodos hija y los anota con los sub-requerimientos de tipo agregados por Introducción: *?Ti(e)* y *?Ti(e → t)*. La regla de **Simplificación** elimina requerimientos que ya han sido satisfechos. Por ejemplo: si un nodo está anotado, al mismo tiempo, con el requerimiento *?Ti(e)* y la anotación de tipo que lo satisface, *Ti(e)*, Simplificación elimina el requerimiento, y deja sólo la anotación de tipo: *Ti(e)*. La regla de **Completación** mueve el puntero desde un nodo hija ya completo al nodo madre, y anota el nodo madre con la información de tipo del nodo hija. La regla de **Anticipación** mueve el puntero desde un nodo madre que ha sido anotado con información de tipo de un nodo hija, hasta un nodo hija que posee un requerimiento pendiente. La regla de **Eliminación** compila información reunida en un par de nodos terminales para satisfacer el requerimiento de un nodo superior. Por ejemplo: si un nodo con un requerimiento *?Ti(t)* está anotado con la siguiente

información : $\langle\downarrow_0\rangle$ *Fo(x), Ti(e)*, $\langle\downarrow_1\rangle$ *Fo(P),Ti(e → t)*, Eliminación realiza una aplicación funcional sobre las fórmulas *Fo(x)*, y *Fo(P)*, obteniendo *Fo(P(x))*, y efectúa modus ponens sobre los tipos *Ti(e)* y *Ti(e → t)*, obteniendo *Ti(t)*. La información resultante, *Fo(P(x))*, *Ti(t)*, que satisface el requerimiento *?Ti(t)*, queda anotada en el nodo madre.

4.3. Ejemplo de un proceso sintáctico en SD: María escuchó música

A continuación presentamos los pasos (en algunos casos, abreviados) del proceso sintáctico gatillado por la agregación lineal de los componentes *María, escuchó* y *música*:

$$?Ti(t),\ \Diamond$$

Figura 7a. Estado inicial correspondiente al requerimiento "derivar una fórmula tipo t".

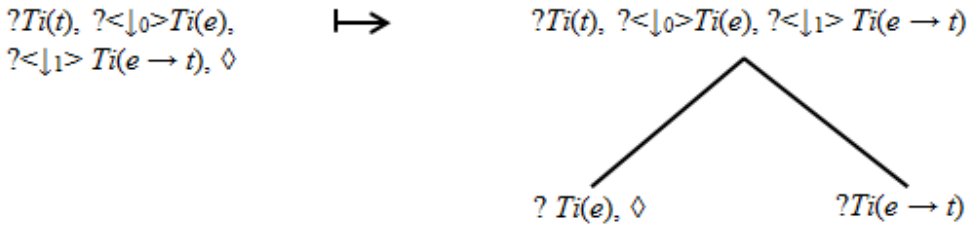

Figura 7b. Aplicación de secuencia de reglas: Introducción y Predicción.

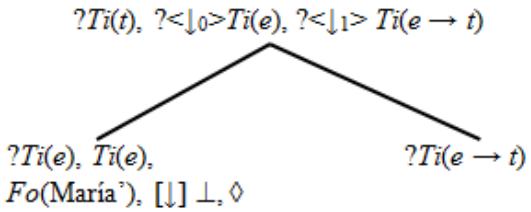

Figura 7c. Información léxica: procesamiento de María.

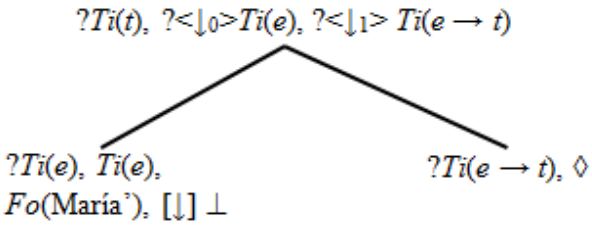

Figura 7d. Aplicación de secuencia de reglas: Simplificación, Completación, Simplificación y Anticipación. Árbol resultante.

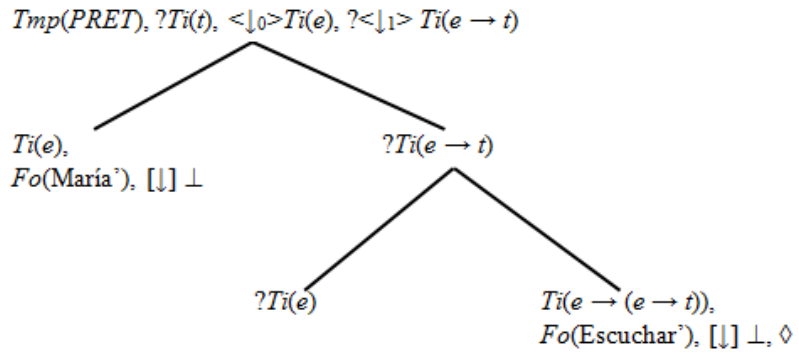

Figura 7e. Información léxica: procesamiento de escuchó.

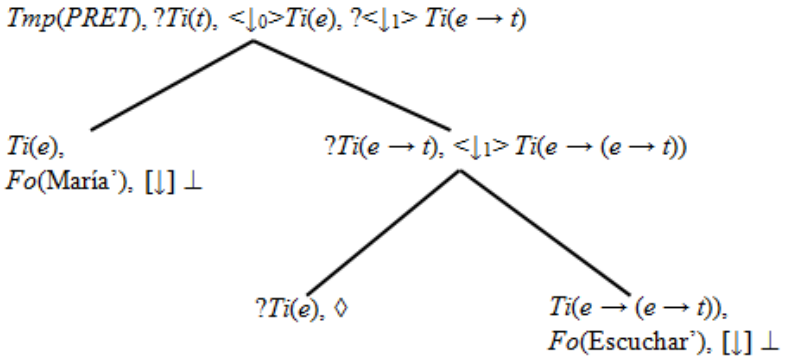

Figura 7f. Aplicación de secuencia de reglas: Completación, y Anticipación. Árbol resultante.

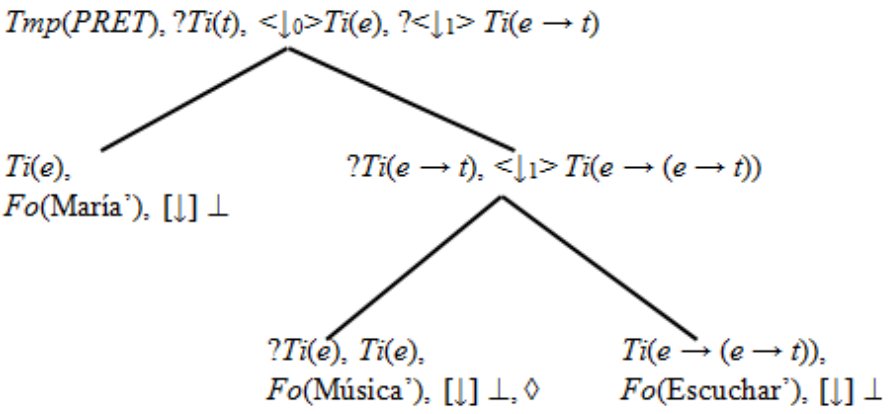

Figura 7g. Información léxica: procesamiento de música.

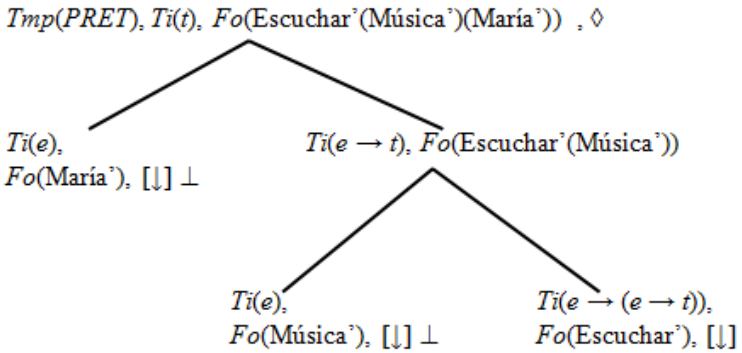

Figura 7h. Secuencia de reglas: Simplificación, Completación, Eliminación, Simplificación, Completación, Simplificación, Eliminación, y Simplificación. Árbol resultante (y final).

5. Comentarios finales

Este modelo supone una concepción innovadora de la sintaxis como proceso que opera directamente sobre conceptos. Además, se concibe el componente léxico del lenguaje como un repertorio de rutinas de procesamiento. Este último factor hace posible que el modelo prescinda de las tradicionales "clases de palabras", en favor de un tratamiento más económico y homogéneo del léxico y la sintaxis, que, sin embargo, no desdibuja la distinción entre ambos estratos. En este sentido, se trata de un enfoque fuertemente minimista, que elimina por completo el aparato categorial de la gramática en la derivación sintáctica, en favor de un sistema recursivo y funcional de tipos léxicos. Además, al tratarse de un modelo de reciente formulación, las aplicaciones de éste al análisis de las lenguas naturales se han restringido a unas pocas lenguas, entre las que se cuentan el inglés, el japonés y el griego. Las aplicaciones al castellano (y a las lenguas románicas en general) han sido escasas, y el presente trabajo ha contribuido también a superar, muy preliminarmente, esta carencia.

Bibliografía

Blackburn, P., & Meyer-Viol, W. (1994). Linguistics, Logic, and Finite Trees. *Bulletin of the IGPL*, 2: 3-29. doi: 10.1093/jigpal/2.1.3

Cann, R. (2011). Towards an Account of the English Auxiliary System, en R. Kempson, E. Gregoromichelaki & C. Howes (2011). *The Dynamics of Lexical Interfaces*. Stanford: CSLI.

Cann, R., Kempson, R., & Marten, L. (2005). *The Dynamics of Language: An Introduction*. Oxford: Elsevier.

Kempson, R., Gregoromichelaki, E., & Howes, C. (2011). *The Dynamics of Lexical Interfaces*. Stanford: CSLI.

Kempson, R., Meyer-Viol, W., & Gabbay, D. (2001). *Dynamic Syntax*. Oxford: Blackwell.

Marten, L. (2002). *At the Syntax-Pragmatics Interface. Verbal Underspecification and Concept Formation in Dynamic Syntax*. Oxford: OUP. doi: 10.1093/acprof:oso/9780199250639.001.0001

Sperber, D., & Wilson, D. (1995). *Relevance: Communication and Cognition*. Oxford: Blackwell.