Lógica dinámica con operadores ambiguos

Josep Artigas Prous

Grup de Recerca en Lingüística Matemàtica i Enginyeria del Llenguatge (GRLMC)
Universitat Rovira i Virgili
Pl. Imperial Tarraco, 1, 43005 Tarragona
jap@astor.urv.es

Resumen

El almacenaje de Cooper es una técnica que ha sido desarrollada para tratar la ambigüedad semántica causada por el alcance de los cuantificadores. La Ambiguous Predicate Logic (APL) de van Eijck and Jaspars (1996) es una lógica ambigua que utiliza un método inspirado en esta técnica para formular representaciones subespecificadas. En esta comunicación proponemos equipar las teorías dinámicas de la semántica del lenguaje natural, en concreto la Dynamic Predicate Logic (DPL) de Groenendijk and Stokhof (1991), con el método de subespecificación de la APL. El resultado es una lógica dinámica subespecificada y propia.

1 Introducción

Desde una perspectiva computacional, uno de los mayores problemas para el tratamiento del lenguaje natural reside en su ambigüedad, hecho que se traduce en la posibilidad de diversas interpretaciones para una misma oración. Generar representaciones para todas estas interpretaciones resulta demasiado costoso para ser eficiente en los sistemas actuales. Una de las soluciones que se han propuesto es integrar un procedimiento heurístico para determinar la interpretación preferencial en un contexto, cf. Jaspars and Kameyama (1996), Moran (1988) y Poesio (1994). La integración de este procedimiento, pero, tiene como consecuencia un procesamiento no monótono. Supongamos un discurso compuesto por una secuencia de oraciones, la primera de las cuales tiene una interpretación preferencial y una secundaria en este contexto inicial. Si procesamos la segunda oración y modificamos el contexto, podemos vernos inducidos a cambiar las preferencias. Por lo tanto, para procesar un discurso de manera monótona, debemos representar inicialmente ambas interpretaciones.

La otra solución consiste en generar una sola representación que sintetice todas las interpretaciones. Diremos que esta representación está subespecificada si podemos derivar de ella, mediante un procedimiento de desambiguación, una representación específica para cada una de las interpretaciones. Existen dos maneras de llevar a la práctica esta idea: modificar directamente el componente semántico de una gramática, con la consiguiente pérdida de propiedades modelo-teóricas del formalismo, o bien definir un lenguaje formal y posteriormente integrarlo en una gramática, a pesar de la dificultad de su interacción con la sintaxis. La primera es la opción adoptada por la Minimal Recursion Semantics (MRS, Copestake et al., 1995, 1997), concebida explícitamente como una alternativa a la semántica de la HPSG. La segunda orientación es la seguida por la mayoría de teorías basadas en la subespecificación, entre las cuales destacan la Quasi Logical Form (QLF, Alshawi, 1992), la Underspecified DRT (UDRT, Reyle, 1993), la Predicate Logic Unplugged (PLU, Bos, 1996) y la APL (van Eijck and Jaspars, 1996; Jaspars, 1997). Hay diversas estrategias para procesar representaciones semánticas, combinando las teorías anteriores con distintas gramáticas de unificación y formalismos categoriales, cf. Bos et al. (1994), Frank and Reyle (1995) y König (1994).

El propio método de la subespecificación puede considerarse desde diversas perspectivas. En la subespecificación representacional, adoptada por la UDRT y la PLU, una representación se concibe

como un grafo acíclico en los nudos del cual se situan descripciones parciales etiquetadas. Las relaciones de dominio entre las etiquetas se dejan subespecificadas, hecho que permite formular las distintas interpretaciones de una oración ambigua. En la subespecificación operacional, seguida por la QLF y la APL, se representa la ambigüedad mediante una notación que permite utilizar operadores ambiguos como elementos de una fórmula. Se entiende que un operador es una fórmula con una componente subespecificado. Una de las ventajas que ofrece la perspectiva operacional, en concreto la APL, es el hecho que está equipada con un cálculo deductivo completo y, por tanto, las inferencias clásicas pueden derivarse de este cálculo.

En esta comunicación proponemos equipar la DPL con el método de subespecificación de la APL o, dicho de otra manera, formular una versión dinámica de la APL. De esta manera obtenemos un lógica subespecificada con las propiedades dinámicas básicas, como son el ligamiento de las variables más allá del alcance sintáctico del cuantificador, la facilidad para formular representaciones parciales y la interpretación incremental. Organizamos la comunicación de la manera que sigue. En primer lugar, sin ninguna intención de originalidad, recordamos cuales son los elementos de la DPL. Después examinamos las propiedades de un formalismo basado en la subespecificación y de qué modo estas propiedades son aprovechadas en la APL para tratar la ambigüedad semántica. Finalmente presentamos la versión dinámica de este último formalismo.

2 Semántica dinámica

La idea subyacente en las aplicaciones de la lógica dinámica a la semántica del lenguaje natural y al análisis de los lenguajes de programación puede resumirse en un eslogan: el significado es una función entre estados de información. Para concretar esta idea, pero, tenemos que precisar qué entendemos por 'estado de información'. En los lenguajes imperativos, un programa puede analizarse como una secuencia de estados de memoria, de manera que el significado de una sentencia será la modificación que causa su ejecución para cambiar de un estado de memoria a otro. En el lenguaje natural, se considera que un estado de información es un contexto y que cada oración de un discurso se interpreta en el contexto formado por las oraciones que la han precedido. El carácter incremental de la interpretación del discurso ha motivado la formulación de esta perspectiva secuencial.

Ahora bien, las diversas teorías semánticas difieren en la manera de llevar a la práctica esta idea común. En la DPL (Groenendijk and Stokhof, 1991) un contexto es modelado como un objeto semántico: una lista de asignaciones totales de valores a las variables. La interpretación dinámica de una fórmula se define mediante un par de asignaciones, de modo que la evaluación veritativa de esta fórmula se consigue teniendo una primera asignación como contexto de entrada y la segunda como contexto de salida. En cambio, en la Discourse Representation Theory (DRT, Kamp, 1981; Kamp and Reyle, 1993) un contexto es una representación y la dinamicidad se consigue mediante el algoritmo de construcción de estas representaciones (DRSs). A partir del análisis sintáctico de un discurso, este algoritmo transforma el contexto precedente en forma de DRS en otro contexto, adjuntando así progresivamente los constituyentes lingüísticos. En la formulación clásica de la DRT, la interpretación dinámica se efectúa a nivel de representaciones, pero sus constituyentes tienen una interpretación estática. Este hecho ocasiona un problema de pérdida de composicionalidad, para el cual se han propuesto diversas soluciones, cf. Zeevat (1989), Muskens (1996) y van Eijck and Kamp (1997).

2.1 Lógica dinámica de predicados

La DPL de Groenendijk and Stokhof (1991) se caracteriza básicamente por el ligamiento dinámico de las variables gracias al cuantificador existencial. En este formalismo, los cuantificadores pueden ligar variables situadas fuera de su alcance sintáctico y, por lo tanto, la siguiente equivalencia resulta válida:

$$(\exists x\phi) \wedge \psi \Leftrightarrow \exists x(\phi \wedge \psi)$$

Para conseguir este efecto, no es necesario modificar el lenguaje de la lógica de predicados de primer orden, sinó la interpretación de sus fórmulas. Ahora bien, la versión que formulamos aquí, inspirada en van Eijck (1999), es una extensión de la DPL en la

cual la conectiva $\wedge$ es reemplazada por una conectiva ; con la finalidad de conseguir una composición secuencial.

El vocabulario no lógico de $\mathcal{L}_{DPL}$ consta de un conjunto P de predicados y un conjunto C de constantes individuales, mientras que el vocabulario lógico está compuesto por un conjunto V de variables $x, y, z, v_1, \ldots, v_n$ y algunas de las constantes lógicas habituales. En cuanto a las metavariables, entendemos que $c_1, \ldots, c_n$ pertenecen al rango de C y que $\delta_1^i, \ldots, \delta_n^i$ pertenecen al rango de P. También asumimos un conjunto T de términos $t_1, \ldots, t_n$.

Definición 1 (Sintaxis)
El lenguaje $\mathcal{L}_{DPL}$ es un conjunto que contiene al menos una fórmula ϕ. Definimos recursivamente una fórmula ϕ de la manera siguiente:

$$\begin{array}{lll} t & ::= & c \mid v \\ \phi & ::= & \delta(t_1, \ldots, t_n) \mid v \doteq t \mid \neg\phi \mid \\ & & \phi;\psi \mid \phi \rightarrow \psi \mid \exists v\phi \end{array}$$

Observemos que la connectiva ; refleja el hecho que, desde una perspectiva dinámica, la conjunción actúa como la composición secuencial y, por lo tanto, es asociativa pero no conmutativa. Por otra parte, hemos omitido el cuantificador universal, que formularemos de acuerdo con la siguiente abreviatura:

$$\forall x\phi \quad := \quad \neg\exists x\neg\phi$$

La interpretación de una fórmula en la DPL se define como un subconjunto del producto del conjunto de asignaciones totales: $[\![\cdot]\!] \subseteq \mathcal{G} \times \mathcal{G}$. Sea $\mathcal{D}$ el dominio de un modelo, una asignación es una función del conjunto de variables al dominio, de modo que el conjunto de asignaciones se define como $\mathcal{G} = \{g \mid g : V \longmapsto \mathcal{D}\}$.

Definición 2 (Modelo)
Un modelo de primer orden $\mathcal{M}$ es un par $\langle \mathcal{D}, \mathcal{I} \rangle$ en el cual $\mathcal{D}$ es un conjunto no vacío que delimita el dominio de $\mathcal{M}$ y $\mathcal{I}$ una función de interpretación tal que, para cada constante individual c, $\mathcal{I}(c) \in \mathcal{D}$, y para cada predicado δ con aridad i, $\mathcal{I}(\delta) \subseteq \mathcal{D}^i$.

Entendemos que $h[v]g$ indica que la asignación h difiere de g al menos respecto al valor asignado a la variable v.

Definición 3 (Semántica)
Definimos la interpretación de una fórmula de la manera que sigue (en general, omitimos los índices cuando no inducen a error):

$$[\![c]\!] := \mathcal{I}(c)$$

$$[\![v]\!] := g(v)$$

$$[\![\delta(t_1, \ldots, t_n)]\!] := \{\langle g, h\rangle \mid h = g \wedge [\![t_1]\!], \ldots, [\![t_n]\!] \in \mathcal{I}(\delta)\}$$

$$[\![v \doteq t]\!] := \{\langle g, h\rangle \mid h = g \wedge [\![v]\!] = [\![t]\!]\}$$

$$[\![\neg\phi]\!] := \{\langle g, h\rangle \mid h = g \wedge \neg\exists k : \langle h, k\rangle \in [\![\phi]\!]\}$$

$$[\![\phi;\psi]\!] := \{\langle g, h\rangle \mid \exists k : \langle g, k\rangle \in [\![\phi]\!] \wedge \langle k, h\rangle \in [\![\psi]\!]\}$$

$$[\![\phi \rightarrow \psi]\!] := \{\langle g, h\rangle \mid h = g \wedge \forall k : \langle h, k\rangle \in [\![\phi]\!] \Rightarrow \exists j : \langle k, j\rangle \in [\![\psi]\!]\}$$

$$[\![\exists v\phi]\!] := \{\langle g, h\rangle \mid \exists k : k[v]g \wedge \langle k, h\rangle \in [\![\phi]\!]\}$$

La interpretación dinámica $[\![\phi]\!]$ de una fórmula ϕ se define en términos de un conjunto de pares de asignaciones tales que $\langle g, h\rangle \in [\![\phi]\!]$ ssi la asignación h es una posible salida evaluada en relación a g como asignación de entrada. De todo ello se deducen fácilmente las definiciones que siguen.

Definición 4 (Verificación)
La fórmula ϕ es verdadera en $\mathcal{M}$ de acuerdo con g, notación $\mathcal{M}, g, h \models \phi$, ssi $\exists h : \langle g, h\rangle \in [\![\phi]\!]^{\mathcal{M}}$.

Definición 5 (Consecuencia)
La fórmula ψ es una consecuencia del conjunto de premisas $\phi_1, \ldots, \phi_n$, notación $\phi_1, \ldots, \phi_n \models \psi$, ssi $\forall\mathcal{M}\forall g_i, h : \langle g_1, g_2\rangle \in [\![\phi_1]\!]^{\mathcal{M}} \wedge \ldots \wedge \langle g_n, h\rangle \in [\![\phi_n]\!]^{\mathcal{M}} \Rightarrow \exists k : \langle h, k\rangle \in [\![\psi]\!]^{\mathcal{M}}$.

La idea que sintetiza la dinamicidad es que una oración que contiene un posible antecedente para una anáfora se manifiesta como una fórmula cuantificada existencialmente $\exists v\phi$, y que los valores de v que verifican ϕ constituyen el contexto en el cual se evalua la oración que sigue en el discurso. Si la fórmula ψ que representa esta oración contiene alguna ocurrencia libre de la misma variable v, entonces se interpreta de manera referencial gracias al cuantificador precedente. No es difícil observar la consecuencia de esta idea:

Observación 1 (Composición secuencial)

$$(\phi;\psi);\chi \;\Leftrightarrow\; \phi;(\psi;\chi)$$

$$\exists x\phi;\psi \;\Leftrightarrow\; \exists x(\phi;\psi)$$

Como corolario a todo lo anterior, la observación siguiente pone de relieve que el cuantificador existencial en posición de antecedente en una implicación tiene fuerza universal, y que puede ligar las ocurrencias libres de la misma variable que se encuentran en el consecuente:

Observación 2 (Implicación)

$$(\phi;\psi) \rightarrow \chi \;\Leftrightarrow\; \phi \rightarrow (\psi \rightarrow \chi)$$

$$\exists x\phi \rightarrow \psi \;\Leftrightarrow\; \forall x(\phi \rightarrow \psi)$$

3 Ambigüedad y subespecificación

El lenguaje natural se diferencia de los lenguajes formales básicamente en el hecho que sus expresiones son ambiguas y que esta ambigüedad alcanza todos los niveles del lenguaje. De manera general, entendemos que una oración es ambigua sintácticamente si podemos asignarle por lo menos dos árboles sintagmáticos, y que es ambigua semánticamente si podemos derivar de un árbol sintagmático por lo menos dos representaciones semánticas. La desambiguación sintáctica se basa habitualmente en métodos estadísticos (cf. Hindle and Rooth, 1993), mientras que los procedimientos de desambiguación semántica utilizan actualmente representaciones subespecificadas.

Diremos que $\mathcal{L}$ es un lenguaje subespecificado si permite formular representaciones compactas que sinteticen las diversas interpretaciones de una oración ambigua y, además, tiene un procedimiento de desambiguación que pueda extraer una representación específica para cada una de estas interpretaciones. En el caso que nos ocupa, las representaciones compactas (i.e. subespecificadas) son fórmulas de la ADPL, mientras que las representaciones especificadas son fórmulas de la DPL. Por otra parte, diremos que $\mathcal{L}$ es un lenguaje completo si el procedimiento de desambiguación produce todas y cada una de las posibles interpretaciones de una oración, y que es un lenguaje cerrado si formula estas interpretaciones únicamente mediante representaciones de $\mathcal{L}$. En este sentido, observemos que las fórmulas del lenguaje especificado son también válidas en el lenguaje subespecificado. Finalmente, diremos que $\mathcal{L}$ es un lenguaje propio si es completo y cerrado.

Existen muchas formas de ambigüedad en el lenguaje natural, algunas de las cuales han sido estudiadas en profundidad desde una perspectiva computacional, como las ambigüedades léxicas y sintácticas, mientras que otras han sido objeto de pocos análisis, como las ambigüedades semánticas causadas por el alcance de los cuantificadores, las cuales, sin embargo, son abundantes en el lenguaje natural, cf. van Deemter and Peters (1996). Observemos un ejemplo de este tipo de ambigüedad:

(1) Every student read a book

Esta oración ofrece dos interpretaciones. En el primer caso, cada estudiante lee un libro, el cual no tiene que ser necesariamente el mismo que leen los demás estudiantes. En el segundo caso, existe solamente un libro, que lee cada estudiante. Formulamos respectivamente ambas interpretaciones en la lógica de predicados y en la DPL para facilitar la comparación:

(2) a. $\forall x(Sx \rightarrow \exists y(By \wedge Rxy))$

b. $\exists x\, Sx \rightarrow \exists y\, By; Rxy$

(3) a. $\exists y(By \wedge \forall x(Sx \rightarrow Rxy))$

b. $\exists y\, By; \exists x\, Sx \rightarrow Rxy$

Puede sorprender el tratamiento del cuantificador universal y la conectiva condicional en la DPL, pero no es más que la aplicación de la equivalencia en términos del cuantificador existencial (cf. Observación 2).

Como vemos, el orden de los cuantificadores es crucial para diferenciar las dos interpretaciones. La técnica del almacenaje fue desarrollada por Cooper (1983) para solucionar ambigüedades de alcance no local de este tipo. Esencialmente, es una manera de representar y manipular cuantificadores, la elección del alcance de los cuales puede ser postergado. La idea básica consiste en asociar a los nudos de un

árbol sintagmático un almacén, el cual contiene una representación semántica junto con los cuantificadores asociados a los nudos inferiores del arbol. El almacén se utiliza para generar representaciones con el alcance resuelto después de analizar una oración.

En la APL se utiliza un operador en lugar de un almacén. Sintácticamente, un operador es una fórmula que tiene un hueco vacío para alojar otra fórmula. Una representación subespecificada es un árbol formado por un conjunto de operadores que dominan una fórmula, pero en el cual las relaciones de dominio entre estos operadores no están especificadas. Este método permite formular representaciones más concisas que utilizar meramente una lista de todas la desambiguaciones. El árbol de la Figura 1 es una representación subespecificada en términos de la ADPL para la oración ambigua (1), las líneas punteadas del cual indican las relaciones de dominio subespecificadas.

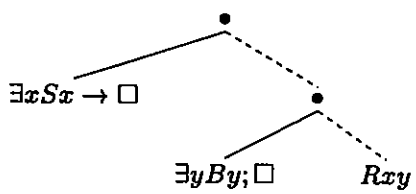

Figura 1: Representación subespecificada

Podemos definir una notación lineal simplemente recogiendo en una lista todos los operadores que dominan una fórmula: $(\exists x Sx \rightarrow \Box, \exists y By; \Box) Rxy$. La razón por la cual se utilizan listas es suministrar un modo de representar la ordenación de fórmulas parciales. En este sentido, el fragmento $\exists x Sx \rightarrow \Box$ es un operador que contiene un cuantificador existencial, el orden del cual es libre respecto al cuantificador incorporado en el otro operador $\exists y By; \Box$.

4 ADPL

La sintaxis de $\mathcal{L}_{ADPL}$ es la misma que la sintaxis de la APL. Las fórmulas son las de la lógica de predicados, pero con la posibilidad de contener operaciones subespecificadas (i.e. una lista de operadores seguida por una fórmula). Las otras expresiones son operadores y contextos. Un operador α es una fórmula con un hueco, mientras que un contexto Δ es una lista de operadores.

Definición 6 (Sintaxis)
El lenguaje $\mathcal{L}_{ADPL}$ es un conjunto que contiene al menos una fórmula ϕ. Definimos recursivamente una fórmula ϕ de la manera siguiente:

$$
\begin{array}{lll}
t & ::= & c \mid v \\
\phi & ::= & \delta(t_1, \ldots, t_n) \mid v \doteq t \mid \neg\phi \mid \phi;\psi \mid \\
 & & \phi \rightarrow \psi \mid \exists v\phi \mid \Delta^{\rightarrow}_{1,n}\phi \\
\alpha & ::= & \Box \mid \neg\alpha \mid \alpha;\phi \mid \phi;\alpha \mid \alpha \rightarrow \phi \mid \\
 & & \phi \rightarrow \alpha \mid \exists v\alpha \mid \Delta^{\rightarrow}_{1,n}\alpha \\
\Delta & ::= & \alpha \mid \Delta\alpha
\end{array}
$$

Utilitzaremos $\Delta^{\rightarrow}_{i,j}$ para simplificar la expresión $(\Delta_i, \Delta_{i+1}, \ldots, \Delta_{j-1}, \Delta_j)$, sean i, j números positivos tales que $i \leq j$.

Diremos que una fórmula ϕ tiene forma especificada $\mathsf{EF}(\phi)$ si no contiene un operador ambiguo, es decir, si no tiene forma $\Delta^{\rightarrow}_{1,n}\phi$. La semántica de la ADPL se define en términos de desambiguaciones parciales en EF. En este sentido, observemos que una fórmula en EF es justamente una fórmula de la DPL y, por lo tanto, su semántica es la misma. En el caso de fórmulas con operadores ambiguos, la interpretación puede definirse de la manera siguiente:

Definición 7 (Semántica subespecificada)

$$[\![\Delta^{\rightarrow}_{1,n}\phi]\!]^{\mathcal{M}}_{g,h} := \bigcup\{[\![\psi]\!]^{\mathcal{M}}_{g,h} \mid \psi \in \mathsf{EF}(\Delta^{\rightarrow}_{1,n}\phi)\}$$

Como vemos en esta definición, un operador ambiguo se interpreta como la unión de todas las posibles permutaciones de las operaciones de desambiguación. Ahora bien, en una oración ambigua puede ocurrir que no todas las interpretaciones posibles sean verificadas en $\mathcal{M}$. Por todo ello, debemos refinar la idea de verificación y definir un procedimiento de desambiguación para obtener estas interpretaciones.

4.1 Desambiguación

El procedimiento para obtener las desambiguaciones es un proceso de reescritura. Supongamos un

contexto formado por una lista de operadores α que dominan una fórmula ϕ. Para obtener una fórmula en EF, podemos desplazar recursivamente los operadores fuera de la lista y alojar la fórmula en el hueco $\Box$ del operador desplazado. Si agotamos todas las posibles permutaciones de esta operación, obtenemos el conjunto de desambiguaciones de la representación subespecificada. La substitución de una fórmula ϕ por el hueco $\Box$ en un operador α, notación $\alpha/\phi/$, se define de la manera que sigue:

$$\begin{aligned}
\Box/\phi/ &:= \phi \\
(\neg\alpha)/\phi/ &:= \neg\alpha/\phi/ \\
(\alpha;\psi)/\phi/ &:= \alpha/\phi/;\psi \\
(\psi;\alpha)/\phi/ &:= \psi;\alpha/\phi/ \\
(\alpha\rightarrow\psi)/\phi/ &:= \alpha/\phi/\rightarrow\psi \\
(\psi\rightarrow\alpha)/\phi/ &:= \psi\rightarrow\alpha/\phi/ \\
(\exists v\alpha)/\phi/ &:= \exists v\alpha/\phi/ \\
(\Delta^{\rightarrow}_{1,n}\alpha)/\phi/ &:= \Delta^{\rightarrow}_{1,n}\alpha/\phi/
\end{aligned}$$

A partir de esta definición, podemos especificar una relación de reescritura para las fórmulas que se aplica recursivamente:

$$\begin{aligned}
(\alpha)\phi &\Rightarrow \alpha/\phi/ \\
(\Delta\alpha)\phi &\Rightarrow \Delta\alpha/\phi/ \\
\Delta^{\rightarrow}_{1,n}\phi &\Rightarrow (\Delta^{\rightarrow}_{1,i-1},\Delta^{\rightarrow}_{i+1,n})\alpha/\phi/ \\
&\quad \text{si } n>1,\ \Delta_i=\alpha \\
\Delta^{\rightarrow}_{1,n}\phi &\Rightarrow (\Delta^{\rightarrow}_{1,i-1},\Delta,\Delta^{\rightarrow}_{i+1,n})\alpha/\phi/ \\
&\quad \text{si } n>1,\ \Delta_i=\Delta\alpha
\end{aligned}$$

La aplicación del procedimiento de desambiguación en la representación subespecificada de (1) produce las interpretaciones que siguen:

$$\begin{aligned}
&(\exists xSx\rightarrow\Box,\exists yBy;\Box)Rxy \\
&\Rightarrow (\exists xSx\rightarrow\Box)\exists yBy;Rxy \\
&\Rightarrow \exists xSx\rightarrow\exists yBy;Rxy
\end{aligned}$$

$$\begin{aligned}
&(\exists xSx\rightarrow\Box,\exists yBy;\Box)Rxy \\
&\Rightarrow (\exists yBy;\Box)\exists xSx\rightarrow Rxy \\
&\Rightarrow \exists yBy;\exists xSx\rightarrow Rxy
\end{aligned}$$

Como no existen otras posibilidades de reescritura, podemos definir el conjunto de fórmulas en EF de esta representación:

$$\begin{aligned}
&\mathsf{EF}((\exists xSx\rightarrow\Box,\exists yBy;\Box)Rxy) = \\
&\{\exists xSx\rightarrow\exists yBy;Rxy,\exists yBy;\exists xSx\rightarrow Rxy\}
\end{aligned}$$

4.2 Verificación

Supongamos la representación $\Delta^{\rightarrow}_{1,n}\phi$ para una oración ambigua. Para verificar el conjunto de fórmulas $\Psi=\{\psi \mid \psi\in\mathsf{EF}(\Delta^{\rightarrow}_{1,n}\phi)\}$ que representan las desambiguaciones de esta oración necesitamos definir también la posibilidad que sean falsas. Este hecho obedece a tres situaciones de interpretación:

- Todas las fórmulas de Ψ son verdaderas en $\mathcal{M}$. Notación: $\mathcal{M},g,h\models\Psi$.
- Todas las fórmulas de Ψ son falsas en $\mathcal{M}$. Notación: $\mathcal{M},g,h\dashv\Psi$.
- Al menos una fórmula $\psi\in\Psi$ es falsa en $\mathcal{M}$, y al menos una fórmula $\psi\in\Psi$ es verdadera en $\mathcal{M}$. Notación: $\mathcal{M},g,h\not\models\Psi$ y $\mathcal{M},g,h\not\dashv\Psi$.

Observemos que la verificación y la falsificación de todas las fórmulas de Ψ equivalen respectivamente a verificar y falsificar $\Delta^{\rightarrow}_{1,n}\phi$. De acuerdo con la noción dinámica de secuencia de contextos, hemos definido la verificación de una fórmula como una transición entre un contexto de entrada y un contexto de salida (cf. Definición 4). Por lo tanto, podemos definir la falsificación como el error en la transición para un determinado contexto.

Definición 8 (Verificación y falsificación) *Definimos recursivamente la verdad y la falsedad de una fórmula en $\mathcal{M}$ de la manera que sigue:*

1. $\mathcal{M},g,h\models\delta(t_1,\ldots,t_n)$ ssi $h=g$ y $[t_1]^{\mathcal{M}},\ldots,[t_n]^{\mathcal{M}}\in\mathcal{I}(\delta)$.
2. $\mathcal{M},g,h\dashv\delta(t_1,\ldots,t_n)$ ssi $h=g$ y $[t_1]^{\mathcal{M}},\ldots,[t_n]^{\mathcal{M}}\notin\mathcal{I}(\delta)$.
3. $\mathcal{M},g,h\models v\doteq t$ ssi $h=g$ y $[v]^{\mathcal{M}}=[t]^{\mathcal{M}}$.
4. $\mathcal{M},g,h\dashv v\doteq t$ ssi $h=g$ y $[v]^{\mathcal{M}}\neq[t]^{\mathcal{M}}$.
5. $\mathcal{M},g,h\models\neg\phi$ ssi $h=g$ y $\mathcal{M},h,k\dashv\phi$.

6. $\mathcal{M}, g, h \dashv \neg\phi$ ssi $h = g$ y $\mathcal{M}, h, k \models \phi$.

7. $\mathcal{M}, g, h \models \phi;\psi$ ssi $\mathcal{M}, g, k \models \phi$ y $\mathcal{M}, k, h \models \psi$.

8. $\mathcal{M}, g, h \dashv \phi;\psi$ ssi $h = g$, y
   - $\mathcal{M}, h, k \dashv \phi$, o bien
   - para toda k en $\mathcal{M}, h, k \models \phi$ se mantiene que $\mathcal{M}, k, j \dashv \psi$.

9. $\mathcal{M}, g, h \models \phi \rightarrow \psi$ ssi $h = g$ y para toda k en $\mathcal{M}, h, k \models \phi$ se mantiene que $\mathcal{M}, k, j \models \psi$.

10. $\mathcal{M}, g, h \dashv \phi \rightarrow \psi$ ssi $h = g$, y
    - $\mathcal{M}, h, k \dashv \phi$, o bien
    - $\mathcal{M}, h, k \models \phi$ y $\mathcal{M}, k, j \dashv \psi$.

11. $\mathcal{M}, g, h \models \exists v\phi$ ssi existe k tal que $k[v]g$ y $\mathcal{M}, k, h \models \phi$.

12. $\mathcal{M}, g, h \dashv \exists v\phi$ ssi $h = g$ y para toda k con $k[v]g$ se mantiene que $\mathcal{M}, k, j \dashv \phi$.

13. $\mathcal{M}, g, h \models \Delta^{\rightarrow}_{1,n}\phi$ ssi $\mathcal{M}, g, h \models \psi$, para toda $\psi \in \mathsf{EF}(\Delta^{\rightarrow}_{1,n}\phi)$.

14. $\mathcal{M}, g, h \dashv \Delta^{\rightarrow}_{1,n}\phi$ ssi $\mathcal{M}, g, h \dashv \psi$, para toda $\psi \in \mathsf{EF}(\Delta^{\rightarrow}_{1,n}\phi)$.

Ahora bien, la posibilidad que una fórmula subespecificada sea indeterminada (es decir, $\mathcal{M}, g, h \not\models \Psi$ o bien $\mathcal{M}, g, h \not\dashv \Psi$) dificulta la definición de la relación de consecuencia. Si entendemos el conjunto de premisas como una conjunción y las conclusiones como una disyunción, podemos definir una relación de consecuencia ambigua de la manera que sigue:

Definición 9 (Consecuencia)
Las fórmulas $\Psi_1, \ldots, \Psi_n$ *son consecuencia de las premisas* $\Phi_1, \ldots, \Phi_n$, *notación* $\Phi_1, \ldots, \Phi_n \models \Psi_1, \ldots, \Psi_n$, *ssi para todo* $\mathcal{M}, g, h_i, j$:

- si $\mathcal{M}, g, h_{[1,\ldots,n]} \models \Phi_1, \ldots, \Phi_n$, entonces $\mathcal{M}, h_n, j \models \Psi_i$, i

  si $\mathcal{M}, g, h_{[1,\ldots,n]} \dashv \Psi_1, \ldots, \Psi_n$, entonces $\mathcal{M}, h_n, j \dashv \Phi_i$.

- si $\mathcal{M}, g, h_{[1,\ldots,n]} \not\models \Phi_1, \ldots, \Phi_n$, entonces $\mathcal{M}, h_n, j \not\models \Psi_i$, i

  si $\mathcal{M}, g, h_{[1,\ldots,n]} \not\dashv \Psi_1, \ldots, \Psi_n$, entonces $\mathcal{M}, h_n, j \not\dashv \Phi_i$.

El valor de una consecuencia se fija de manera dinámica en relación a la asignación de salida de las premisas. Del mismo modo que la verificación, el hecho que un conjunto de fórmulas específicas Ψ sea consecuencia de unas determinadas premisas equivale a que también lo sea $\Delta^{\rightarrow}_{1,n}\phi$.

5 Conclusiones

La ADPL es una lógica dinámica propia que permite formular representaciones subespecificadas de oraciones ambiguas. Los beneficios de la subespecificación para el procesamiento del lenguaje natural han sido aprovechados en aplicaciones de traducción automatizada como Verbmobil (cf. Wahlster, 1993) y en algunos sistemas de propósito general como CLE (cf. Alshawi, 1992) y LexGram (cf. König, 1994). La integración del lenguaje en una gramática de unificación es facilitada por el uso de listas para representar los operadores ambiguos.

La tarea de la semántica computacional no consiste sólo en formular lenguajes de representación y algoritmos capaces de soportar un procesamiento eficiente, sinó también técnicas de inferencia para razonar con estas representaciones. En este sentido, una extensión del trabajo será equipar el lenguaje con un cálculo deductivo dinámico.

Finalmente, otra propuesta de trabajo es trasladar el método de subespecificación de la APL a la DRT, la cual tiene una sintaxis más rica que permite una mayor expresión de fenómenos lingüísticos.

Referencias

Alshawi, H. (ed.) (1992). *The Core Language Engine*, MIT Press, Cambridge.

Bos, J. (1996). Predicate Logic Unplugged, *Proceedings of the 10th Amsterdam Colloquium*, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, pp. 133–142.

Bos, J., Mastenbroek, E., McGlashan, S., Millies, S. and Pinkal, M. (1994). A compositional DRS-based formalism for NLP applications, *Proceedings of the International Workshop on Computational Semantics*, Tilburg, pp. 21–31.

Cooper, R. (1983). *Quantification and Syntactic Theory*, Reidel, Dordrecht.

Copestake, A., Flickinger, D. and Sag, I. A. (1997). *Minimal Recursion Semantics. An Introduction*, Ms, Center for the Study of Language and Information (CSLI), Stanford.

Copestake, A., Flickinger, D., Malouf, R., Riehemann, S. and Sag, I. (1995). Transfer and Minimal Recursion Semantics, *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, pp. 124–136.

Frank, A. and Reyle, U. (1995). Principle based semantics for HPSG, *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, pp. 9–16.

Groenendijk, J. and Stokhof, M. (1991). Dynamic Predicate Logic, *Linguistics and Philosophy* **14**(1): 39–100.

Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations, *Computational Linguistics* **19**(1): 103–120.

Jaspars, J. (1997). *Minimal Logics for Reasoning with Ambiguous Expressions*, Report CLAUS Nr. 94, Department of Computational Linguistics, University of Saarbrücken.

Jaspars, J. and Kameyama, M. (1996). Preferences in dynamic semantics, *Proceedings of the 10th Amsterdam Colloquium*, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, pp. 445–464.

Kamp, H. (1981). A theory of truth and semantic representation, *in* J. Groenendijk, T. Jansen and M. Stokhof (eds), *Formal Methods in the Study of Language*, Mathematisch Centrum, Amsterdam, pp. 277–322.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Kluwer, Dordrecht.

König, E. (1994). *A Study in Grammar Design*, Arbeitspapiere des Sonderforschungsbereichs 340, Bericht Nr. 54, Institute for Computational Linguistics (IMS), University of Stuttgart.

Moran, D. (1988). Quantifier scoping in the SRI Core Language Engine, *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, pp. 33–40.

Muskens, R. (1996). Combining Montague semantics and discourse representation, *Linguistics and Philosophy* **19**(2): 143–186.

Poesio, M. (1994). *Discourse Interpretation and the Scope of Operators*, PhD thesis, University of Rochester, Department of Computer Science, Rochester.

Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation and deduction, *Journal of Semantics* **10**(2): 123–179.

van Deemter, K. and Peters, S. (eds) (1996). *Semantic Ambiguity and Underspecification*, Lecture Notes No. 55, CSLI Publications, Stanford.

van Eijck, J. (1999). Axiomatising dynamic logics for anaphora, *Journal of Language and Computation* **1**(1): 103–126.

van Eijck, J. and Jaspars, J. (1996). *Ambiguity and Reasoning*, Report CS-R9616, Centre for Mathematics and Computer Science (CWI), Amsterdam.

van Eijck, J. and Kamp, H. (1997). Representing discourse in context, *in* J. van Benthem and A. ter Meulen (eds), *Handbook of Logic and Language*, Elsevier Science, Amsterdam, pp. 179–238.

Wahlster, W. (1993). VERBMOBIL: Traslation of face-to-face dialogs, *Proceedings of the 3rd European Conference on Speech Communication and Technology*, Berlin, pp. 29–38.

Zeevat, H. (1989). A compositional approach to DRT, *Linguistics and Philosophy* **12**(2): 95–131.