

LÓGICAS PARA LA RED

Logics for the Web

Antonia HUERTAS*

Universitat Oberta de Catalunya, mbuertass@uoc.edu

BIBLID [(0213-3563)8,2006,85-102]

Fecha de aceptación definitiva: 12 de marzo de 2006

RESUMEN

Tim Berners-Lee y sus colegas del World Wide Web Consortium (W3C) llamaron «Semantic Web» al que sería el siguiente estadio de desarrollo de la red (inevitablemente traducido al castellano como Web Semántica). La idea detrás de esta evolución de la red es extender con metadatos y reglas lógicas la red basada en el lenguaje html, con el objetivo de que la infraestructura resultante permita a las máquinas entender los datos de la red de la misma forma que los entendemos los humanos. Así, añadir lógica a la red permitiría a los ordenadores tomar decisiones, hacer inferencias y responder preguntas.

En este texto se pretende revisar las lógicas que han sido propuestas para permitir esta nueva web inteligente y repasar las cuestiones teóricas y pragmáticas más importantes de la corta pero intensa historia de las lógicas para la red.

Palabras clave: web semántica, lógica descriptiva, lenguajes de ontologías, lógicas para la web.

ABSTRACT

Tim Berners-Lee and their colleagues of the World Wide Web Consortium (W3C) called «Semantic Web» the next development of the Web. The idea behind this

* Investigación parcialmente financiada por los proyectos HUM2006-12848-C02-01 y BFF2003-08998-C03.

evolution of the Web is to extend the present network based on the HTML language with metadata and logical rules, with the aim of the resulting infrastructure allowing machines to understand Web data in the same way that human beings understand them. Thus, adding logic to the Web would allow the computers to take decisions, to do inferences and to answer questions.

In this paper it is intended to revise the logics that have been suggested to allow this new intelligent Web and to revise the most important theoretical and pragmatic questions of the short but intense history of logics for the Web.

Key words: semantic web, description logics, ontology languages, logics for the web.

1. INTRODUCCIÓN: LA RED SEMÁNTICA

Internet (world wide web) fue inventada por Tim Berners-Lee [8] a finales de los años 80 del siglo pasado y en dos décadas ha cambiado la forma de comunicarse entre las personas, la forma cómo se propaga, se almacena y se organiza la información, los agentes que controlan ésta, la economía en la sociedad del conocimiento e incluso las formas de delincuencia.

Internet fue pensada para el uso y consumo humano, basada en un contenido que se genera y distribuye desde diferentes bases de datos sin una estructura pre-establecida. Con el aumento de nodos de ésta red y de la ingente cantidad de información que por ella circula, los métodos y sistemas para la búsqueda de información fiable y precisa son de la mayor importancia.

Las herramientas actuales para la búsqueda de información (buscadores como Google o Yahoo) basadas en la búsqueda léxicográfica y en marcajes sintácticos de los documentos muestran a menudo unos resultados poco fiables cuanto poco útiles. Estos buscadores son poco selectivos, dando en muchas ocasiones respuestas no deseadas, y poco precisos, no proporcionando aquello que buscamos.

Por otro lado, los contenidos de las actuales páginas web, escritas en lenguaje html (ver figura 1 y 2), proporcionan básicamente información textual y gráfica sólo comprensibles para humanos. En el momento de aparecer Internet eso era suficiente y no había una necesidad de que otros agentes-máquinas comprendieran el contenido del documento que circulaba por Internet, ¿para qué si su único consumidor tenía que ser el humano que lo descifraba? Hoy en día, sin embargo, los agentes-máquinas buscadores de información necesitan poder acceder de alguna manera a la información sobre el contenido de una determinada página web (buscadores semánticos), así como poder comunicarse fácilmente tanto con los humanos como con las máquinas.

La propuesta actualmente más prometedora es la llamada web semántica [2] (red semántica), propagada por W3C [25]. Sus características son las siguientes:

- Sus páginas web deben tener marcaje semántico de la información (anotaciones con información sobre el contenido de la página web) que

pueda ser procesado por buscadores «semánticos» de la red. El contenido de las páginas se organiza conceptualmente y toda esta información debe representarse con lenguajes estándar y expresivos.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
  <head>
    <style>
      a:link           {font:8pt/11pt verdana; color:red}
      a:visited       {font:8pt/11pt verdana; color:#4e4e4e}
    </style>
    <meta HTTP-EQUIV="Content-Type" Content="text-htm; charset=Windows-1252">
    <title>www.uoc.edu</title>
  </head>
  <body bgcolor="white">
    <table width="400" cellpadding="3" cellspacing="5">
      <tr>
        <td id="tableProps" valign="top" align="left"><img id="pagerrorImg"
SRC="res://shdoclc.dll/pagerror.gif"
width="25" height="33"></td>
        <td id="tableProps2" align="left" valign="middle" width="360"><h1
id="textSection1"
style="COLOR: black; FONT: 13pt/15pt verdana"><span id="errorText">No se
puede mostrar la página</span></h1>
        </td>
      </tr>
      <tr>
        <td id="tablePropsWidth" width="400" colspan="2"><font
style="COLOR: black; FONT: 8pt/11pt verdana">La página Web solicitada no
está disponible
en este momento. Puede que el sitio Web tenga problemas técnicos o que
necesite ajustar la
configuración de su explorador.</font></td>
      </tr>
    </table>
  </body>
</html>

```

Figura 1. Página web escrita en lenguaje html.

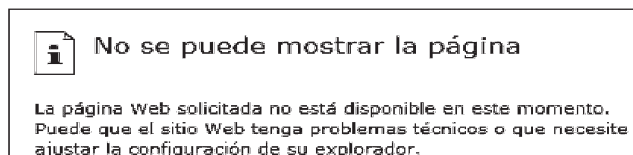


Figura 2. Página web correspondiente a la figura 1 vista con un navegador.

- Sus páginas web pueden ser compartidas y procesadas tanto por humanos como por buscadores (máquinas). La extracción de nuevo conocimiento a partir del existente y el mantenimiento de la consistencia del nuevo conocimiento con el previo debe estar automatizada. Los agentes buscadores de información se basan en sistemas de de preguntas-respuestas en lugar de palabras clave, por tanto deber haber un procedimiento de razonamiento implementable.

El acceso inteligente a la información vía el significado (semántica) del contenido es el rasgo más importante y revolucionario de la web semántica. Inteligente, en este contexto, significa que el sistema puede encontrar contenido o significado implícito a partir de la representación «semántica» del contenido explícito, para ello

serán necesarios diferentes niveles de lenguajes para representación del conocimiento y el razonamiento, y en particular será necesario añadir «lógica» a la red.

En un primer nivel de lenguajes estará el de la ontología del dominio que se quiere representar [9]. Una ontología es un modelo conceptual (colección de definiciones de conceptos y relaciones entre los mismos) que define formalmente un dominio y que permite compartir conocimiento entre los diferentes agentes que usarán una misma ontología.

Un lenguaje para ontologías debe ser compatible con el lenguaje en el que se escriban estas nuevas «páginas web semánticas» y con las herramientas de razonamiento que se usarán en la web semántica. También debe tener una sintaxis intuitiva para los usuarios humanos pero compatible con los lenguajes estándares para la web (que en la web semántica son XML y RDF [24]). Además, la semántica de un lenguaje para ontologías debe poder ser definida formalmente (para permitir a los agentes compartir significado).

El poder expresivo de estos lenguajes debe ser el suficiente como para permitir la definición de los conceptos relevantes; pero no demasiado expresivo ya que haría imposible el razonamiento automático. El razonamiento también es necesario para asegurar la calidad del diseño de la ontología, ya que se usa para demostrar la consistencia entre diferentes conceptos en una ontología y de la jerarquía de los conceptos; así como para poder integrar ontologías diferentes ya que permite probar la consistencia de la integración.

2. LÓGICA PARA LA WEB SEMÁNTICA

La lógica se ocupa de estudiar los razonamientos válidos, también se puede decir que estudia la consecuencia lógica o incluso que se ocupa de «los conjuntos de creencias consistentes» [18]. Para ello la lógica ofrece:

- Lenguajes formales con los que expresar «conocimiento» con rigor y precisión.
- Semánticas formales con las que precisar el significado de «verdad» o «validez» de un razonamiento.
- Razonadores automáticos que pueden deducir y convertir el conocimiento «implícito» en «explícito».
- Pruebas de los razonamientos en un sistema de inferencia (deductivo) que proporcionan «explicaciones» cuando son suficientemente cercanos a la inferencia humana (como por ejemplo los sistemas de deducción natural).

Una lógica depende de la definición de su lenguaje formal, su semántica formal y su procedimiento formal de razonamiento o inferencia. Para cada elección de éstos tres factores tenemos, de hecho, una lógica. Así hay lógicas de proposiciones y de primer orden, de orden superior, bivaloradas, trivaloradas y multivaloradas, fuzzy, modales, dinámicas, temporales, intuicionistas, cuánticas, etc.

Las lógicas adecuadas para la web semántica caen dentro del ámbito de la «representación del conocimiento», que se ocupa de los formalismos que son a la vez epistemológica y computacionalmente adecuados para expresar el «conocimiento» acerca de un determinado dominio o contexto. Durante la década de 1970 se desarrollaron los llamados sistemas «frame-based» y los «semantic networks», en los cuales el conocimiento se representa caracterizando clases de objetos y las relaciones entre ellos. Como esos sistemas no fueron definidos formalmente y el énfasis, en ese momento, se puso sobre todo en la implementación de los sistemas de inferencia asociados a ellos no fue posible proporcionarles procedimientos deductivos completos y correctos con respecto a su semántica (carecían de semántica formal). Por tanto, añadir lógica a esos sistemas que representaban la semántica de un dominio y que eran computacionalmente bastante eficientes se convirtió en una necesidad y en un reto. La respuesta fue el comienzo de la investigación en los llamados «frame-based description systems» que daría origen a la familia de las lógicas descriptivas.

Las lógicas descriptivas [4] se han convertido en las mejores candidatas para añadir razonamiento a la red. Su sintaxis formal permite describir conceptos de las nociones importantes de un dominio o universo, relaciones entre éstos y constructores de nuevos conceptos. Tienen, además, una semántica formal pero intuitiva. Son lógicas formales y por tanto permiten razonar sobre la base de conocimiento así definida. Veremos en la próxima sección que son variantes de la lógica de primer orden clásica y que su razonamiento es también heredero de los sistemas de inferencia de ésta.

Desde 1989 hay un workshop dedicado a las lógicas descriptivas y desde 1997 hay un grupo de trabajo, el «DL Group», que ha originado la aparición de *The Description Logic Handbook* [4], el mejor texto para un primer estudio riguroso de éstas lógicas.

3. LÓGICAS DESCRIPTIVAS

Las principales características formales de las lógicas descriptivas son:

- Modelan conceptualmente las ontologías, esto es, proporcionan descripciones de los dominios formalizando los elementos terminológicos o descriptivos de su ontología. Es fundamental el hecho de que su sintaxis y semántica son formales (sin ambigüedades).
- Tienen un formalismo descriptivo: conceptos, roles y constructores.

Veamos el ejemplo de formalización del concepto «es un animal que tiene todos los progenitores humanos»:

Animal $\sqcap \forall$ *tiene.Hijo.Humano*. Donde los diferentes elementos que aparecen son:

Un concepto primitivo: *Animal*.

Un rol o relación: \forall *tieneHijo.Humano* («todo hijo es humano»).

Un constructor de nuevos conceptos: \sqcap (conjunción de conceptos).

- Tienen un formalismo terminológico: axiomas terminológicos que introducen descripciones complejas y propiedades de la «terminología descriptiva»:

Ejemplos:

Mujer \sqsubset *Persona* («una mujer es una persona»).

Hombre \equiv *Persona* \sqcap \neg *Mujer* («un hombre es una persona no mujer»).

- Tiene un formalismo asertivo que introduce propiedades de individuos.

Ejemplo: *maria:Mujer* («el individuo María es una mujer»).

(jesus,maria):tieneHijo («el individuo María tiene hijo el individuo Jesús»).

Nótese que el uso de mayúsculas y minúsculas es convencional y opuesto al uso en el lenguaje natural (mayúsculas para conceptos, minúsculas para individuos).

- Son capaces de inferir nuevo conocimiento a partir de conocimiento dado. Tienen, por tanto, algoritmos de razonamiento (cálculos) que son decidibles. Las técnicas de razonamiento deberían permitir una implementación automatizable. Veremos que ya se han implementado en sistemas muy eficientes (uso efectivo).

Los elementos centrales del alfabeto del lenguaje de las lógicas descriptivas son:

- Nombres de concepto (*concept name*): asignan un nombre a un grupo de objetos. Ejemplo: *Mujer*, *Persona*... (es un elemento primitivo o atómico del alfabeto de la lógica).
- Nombres de rol (*role name*): asigna un nombre a una relación entre objetos. Ejemplo: *tieneHijo* (atómico o elemento primitivo).
- Constructores (*constructor*): relaciona nombres de conceptos y nombres de roles, y también crea conceptos complejos a partir de los atómicos (complex concepts).
Ejemplo: \sqcap (conjunción), \sqcup (disyunción).
- Definiciones de conceptos complejos: se usa el símbolo \equiv
Ejemplo: *Madre* \equiv *Mujer* \sqcap \exists *tieneHijo.Persona*

La semántica de las lógicas descriptivas se basa en interpretaciones del lenguaje sobre un conjunto Δ no vacío (de los objetos o individuos del dominio). La idea intuitiva es que una interpretación es una descripción completa del mundo o dominio.

- Formalmente una interpretación *I* es una función que asigna a los símbolos del lenguaje, conceptos y roles, subconjuntos y relaciones sobre Δ , respectivamente, (un modelo del dominio en el sentido de la lógica clásica).

El sistema ALC

Hay diferentes sistemas de lógica descriptiva. ALC es la lógica descriptiva básica. En la tabla 1 hay una presentación formal de este sistema.

TABLA 1.
Sintaxis y semántica del sistema ALC

ALC	<i>Sintaxis</i>	<i>Semántica</i> (<i>I</i> es una interpretación de los símbolos de la sintaxis)
Nombres de conceptos atómicos	A, B, \dots	<i>Predicados unitarios</i> $I(A), I(B), \dots$ son subconjuntos de Δ (dominio de interpretación)
Nombres de roles atómicos	R, S, \dots	<i>Predicados binarios</i> $I(R), I(S), \dots$ son relaciones binarias sobre Δ
Conceptos universal y vacío	\top \perp	$I(\top) = \Delta$ Universal: describe el universo del dominio $I(\perp) = \emptyset$ Vacío: describe lo contradictorio.
Conceptos complejos (se obtienen a partir de los conceptos y roles atómicos usando constructores)	(C) $\neg C$	<i>Complementario: concepto con objetos que no son de C</i> $I(\neg C) = \Delta - I(C)$
	$C \sqcap D$	<i>Intersección de conceptos: concepto con objetos de C y D</i> $I(C \sqcap D) = I(C) \cap I(D)$
	(U) $C \sqcup D$	<i>Unión de conceptos: concepto con objetos de C o D</i> $I(C \sqcup D) = I(C) \cup I(D)$
	(E) $\exists R.C$	<i>Restricción existencial: el concepto cuyos objetos son los que están relacionadas por R con los de C</i> $I(\exists R.C) = \{b \in \Delta / \text{existe } c \in \Delta [(b,c) \in I(R) \text{ y } c \in I(C)]\}$
	$\forall R.C$	<i>Restricción universal: el concepto cuyos objetos se relacionan por R sólo con objetos de C</i> $I(\forall R.C) = \{b \in \Delta / \text{para todo } c [(b,c) \in I(R) \text{ implica } c \in I(C)]\}$

Ejemplos de conceptos bien formados en ALC:

Animal $\sqcup \forall \text{tieneHijo.Humano}$: «es animal o bien todos sus hijos son humanos».

Persona $\sqcap \neg \text{Mujer}$: «es persona y no mujer»

Mujer $\sqcap \exists \text{tieneHijo.Persona}$: «es mujer y tiene un hijo persona»

Se llama *AL* al lenguaje descriptivo «mínimo» que consiste en *ALC* excepto los constructores unión (*U*) y complementario (*C*) y con una versión limitada de la restricción existencial (*E*), en la que sólo se permite $\exists R. \top$ para un rol cualquiera.

Por tanto el lenguaje de la tabla 1 puede considerarse el resultado de añadir *U, C*, y *E* al lenguaje básico *AL* y podría escribirse *ALUCE*. Sin embargo, del hecho que $C \sqcup D$ es equivalente a $\neg(\neg C \sqcap \neg D)$ y $\exists R.C$ es equivalente a $\neg \forall R. \neg C$ resulta

que U y E pueden ser expresados en ALC (el resultado de añadir sólo C a AL), y por ello se denota ALC al lenguaje de la tabla 1.

4. INFERENCIA CON LÓGICAS DESCRIPTIVAS

Las lógicas descriptivas son algo más que lenguajes para formalizar conceptos, deben representar la ontología de un dominio y permitir razonar sobre él. Para ello se introducen nuevos elementos del lenguaje y la semántica necesarios para formalizar las propiedades de los individuos del dominio y de las relaciones entre conceptos y roles, son las llamadas bases de conocimiento.

Una **base de conocimiento** está formada por un par de conjuntos:

- Una **terminología** o TBox, que es un conjunto de axiomas terminológicos (inclusiones o igualdades de conceptos o roles) que se ilustran en la tabla 2.

Las **definiciones** son igualdades que se utilizan para introducir nombres simbólicos para descripciones complejas (conceptos complejos). Ejemplos de definiciones de conceptos complejos:

$$\begin{aligned} \text{Hombre} &\equiv \text{Persona} \sqcap \neg \text{Mujer}, \text{Madre} \equiv \text{Mujer} \sqcap \exists \text{tieneHijo. Persona} \\ \text{Padre} &\equiv \text{Hombre} \sqcap \exists \text{tieneHijo. Persona}, \text{Progenitor} \equiv \text{Madre} \sqcup \text{Padre} \end{aligned}$$

- Una **descripción del universo** o ABox, que es un conjunto de axiomas asertivos (instanciaciones de conceptos o de roles) que se ilustran en la tabla 2.

TABLA 2.
TBox y Abox

	<i>Sintaxis</i>	<i>Semántica</i> (I es una interpretación de los símbolos de la sintaxis)
Nombres de individuos	o, p, \dots	<i>Objetos</i> $I(o), I(p), \dots$ son <i>elementos</i> de Δ (dominio de interpretación)
TBox	$C \sqsubseteq D$ $R \sqsubseteq S$	<i>Inclusiones de conceptos o roles</i> $I(C) \subseteq I(D)$ $I(R) \subseteq I(S)$
	$C \equiv D$ $R \equiv S$	<i>Igualdades de conceptos o roles</i> $I(C) = I(D)$ $I(S) = I(S)$
ABox	$\bar{o} : C$ $(o, p) : R$	<i>Instanciación de concepto: a es del concepto C</i> $I(o) \in I(C)$ <i>Instanciación de rol: b está relacionado con a por R</i> $(I(o), I(p)) \in I(R)$

Ejemplo de TBox:

{Mujer \sqsubseteq Persona, Madre \equiv Mujer \sqcap \exists tieneHijo. Persona}

Ejemplo de Abox:

{jesus:Persona, maria:Persona, marta:Persona, (jesus,maria):hijo, (maria,marta):hijo}

Un modelo del TBox (respectivamente del ABox) es una interpretación del lenguaje que hace verdaderas todas las fórmulas del TBox (respectivamente del ABox), exactamente igual que en la semántica de primer orden.

Una base de conocimiento (TBox y ABox) es equivalente a un conjunto de axiomas de la lógica de primer orden, y por tanto se puede definir un cálculo o sistema de inferencia que permite derivar «conocimiento» implícito a partir del «explícito» de la base de conocimiento.

A continuación vamos a presentar los problemas más importantes de razonamiento o inferencia para las lógicas descriptivas, es decir aquello que es importante poder inferir. Veremos que hay un problema de inferencia principal, la verificación de la consistencia para ABoxes, al cual todas las otras inferencias se pueden reducir.

Razonando con conceptos

Supongamos que tenemos un lenguaje descriptivo para un dominio, por ejemplo de *ALC*, y que se ha definido una TBox (axiomas terminológicos) para modelar un dominio. Si se define un nuevo concepto es importante saber si es consistente o contradictorio con el TBox. Esta propiedad se conoce como ser el concepto **satisfacible** (o respectivamente insatisfacible) con respecto al TBox. También puede ser necesario saber si un concepto es más general que otro, si son equivalentes o si son disjuntos. La formalización de estas propiedades es la siguiente.

Supongamos que *T* es un TBox, y *C* y *D* conceptos:

- *C* es **satisfacible** respecto a *T* si existe un modelo *I* de *T* tal que $I(C) \neq \emptyset$
- *C* es **subsumido** por *D* respecto a *T* si para todo modelo *I* de *T*, $I(C) \subseteq I(D)$. Se escribe $T \models C \sqsubseteq D$

Razonando con ABoxes

Una vez definida una TBox, al definir la ABox, las propiedades más importantes que habrá que verificar son las de la consistencia del ABox y el TBox, y la derivación de una instanciación a partir de la ABox. Veamos formalmente estos conceptos.

Supongamos que *T* es un TBox, *A* es un ABox, *C* un concepto y *a* un nombre de individuo:

- *A* es **consistente** con respecto a *T* si existe una interpretación que es modelo de *T* y de *A*
- $o : C$ se deriva de *T* y *A* si todo modelo *I* de *T* y *A* cumple $I(o) \in I(C)$
($T, A \models o : C$)

Para un estudio en profundidad de la inferencia con TBoxes y ABoxes recomendamos [7]. Para la introducción que pretendemos aquí es suficiente con las definiciones anteriores y comprender que son estos los tipos básicos de inferencia

que pretendemos poder realizar en el sistema de lógica descriptiva para la representación de la ontología, y que, por tanto, los algoritmos que nos interesan serán fundamentalmente los que resuelven los problemas de satisfacibilidad y subsumición de conceptos, y el de consistencia para ABoxes. De hecho todos los problemas de inferencia relevantes se pueden reducir al de consistencia para ABoxes en una lógica descriptiva con intersección y complementario de conjuntos. Lo mínimo que le pediremos a la lógica descriptiva es que estos tres problemas sean decidibles y mejor si es a costa de una complejidad baja.

5. EXTENSIONES DE ALC

El poder expresivo de una lógica descriptiva es la capacidad para representar «conocimiento» a cerca del dominio y depende de la riqueza de su lenguaje. Por ejemplo, en la ontología de un árbol genealógico de una familia, sería necesario poder representar todos los conceptos, roles (atómicos y complejos) e individuos necesarios.

Por otro lado, las lógicas descriptivas fueron pensadas como sistemas formales para representar conocimiento [10], y ello significa ir más allá de almacenar terminologías y descripciones, en particular significa poder derivar hechos implícitos a partir de los datos. Por este motivo la implementación de procesos de inferencia debe tener en cuenta la posibilidad de usar algoritmos de inferencia óptimos. En el estudio de tales algoritmos el punto de partida es conocer su complejidad computacional. Es sabido que hay una relación inversa entre el poder expresivo de una lógica y la complejidad computacional de sus algoritmos de inferencia.

El poder expresivo de la lógica descriptiva para ontologías debe ser, por tanto, el suficiente como para permitir definir la terminología relevante pero no demasiado expresivo como para hacer imposible el razonamiento. Por esta razón, en el desarrollo de las lógicas descriptivas el estudio del razonamiento ha ido ligado al de la complejidad computacional y la expresividad.

El poder expresivo del lenguaje de la lógica *ALC* no es suficiente. Para comprobarlo basta ver estos ejemplos de «información» básica sobre un dominio sencillo no expresable en *ALC*:

- «Una mujer que tiene exactamente dos hijos» (no es posible expresar restricciones numéricas).
- «Todo hombre es hijo de una mujer» (no es posible expresar el inverso de un rol).
- «La madre del padre es la abuela» (no es posible expresar composición de roles).

Es necesario extender el lenguaje de *ALC*, pero añadiendo los elementos necesarios de forma que no se dispare la complejidad computacional, ya que queremos poder razonar con esa lógica y, en particular, disponer de los algoritmos mínimos de satisfacibilidad, subsumición y consistencia. Veamos los constructores más importantes utilizados para extender el lenguaje de *ALC* y también algunos de los sistemas obtenidos extendiéndola.

Constructores de roles

Intersección ($R \sqcap S$), unión ($R \sqcup S$), complementario ($\neg R$), composición ($R \circ S$), cierre transitivo (R^+), inversa (R^-), identidad (Id). La interpretación semántica de estas expresiones es la siguiente. Sean R y S nombres de roles y C nombre de concepto:

$$\begin{aligned} I(R \sqcap S) &= I(R) \cap I(S), I(R \sqcup S) = I(R) \cup I(S), I(\neg R) = \Delta - I(R) \\ I(R \circ S) &= \{(a, b) \in \Delta \times \Delta / \exists c ((a, c) \in I(R) \text{ y } (c, b) \in I(S))\} \\ I(R^+) &= \bigcup_{i \geq 1} (I(R))^i \text{ con } (I(R))^i = I(R) \times \Lambda^{(i)} \Lambda \times I(R) \\ I(R^-) &= \{(b, a) \in \Delta \times \Delta / (a, b) \in I(R)\} \\ I(C) &= \{(a, a) \in \Delta \times \Delta / a \in I(C)\} \end{aligned}$$

Ejemplos:

tieneMadre \sqcup tienePadre es el rol equivalente a tieneProgenitor
 tienePadre \circ tieneMadre es el rol equivalente a tieneAbuelo-materno
 tieneHijo⁺ es el rol equivalente a tieneDescendiente
 tieneHijo⁻ es el rol equivalente a tieneProgenitor
 «La madre del padre es la abuela» ahora se puede expresar con:
 tienePadre \circ tieneMadre \sqsubset tiene.Abuela
 «Todo hombre es hijo de una mujer» ahora se puede expresar con:
 Hombre $\sqsubset \exists$ tieneHijo⁻. Mujer

ALC_{reg} extiende ALC con los constructores de roles unión, composición, cierre transitivo, y identidad. La presencia en el lenguaje de una lógica descriptiva de estos constructores de roles «regulares» se especifica con el subíndice $_{reg}$.

$ALCI_{reg}$ extiende ALC_{reg} con los constructores de roles inverso.

Restricciones numéricas (N)

La restricción numérica máxima ($\leq n R$) se interpreta como el concepto «individuos tales que existen como máximo n individuos relacionados con ellos por R »

$$I(\leq n R) = \{a \in \Delta / \text{cardinal}\{ b \in \Delta / (a, b) \in I(R)\} \leq n \}$$

La restricción numérica mínima ($\geq n R$) se interpreta

$$I(\geq n R) = \{a \in \Delta / \text{cardinal}\{ b \in \Delta / (a, b) \in I(R)\} \geq n \}$$

Ejemplo. Ahora sí que podemos expresar «una mujer que tiene exactamente dos hijos»:

$$\text{Mujer} \sqcap \leq 2 \text{tieneHijo} \sqcap \geq 2 \text{tieneHijo}$$

$ALCN$ extiende ALC con estos constructores.

Restricciones numéricas calificadas (Q)

La restricción numérica calificada máxima ($\leq n R.C$) se interpreta como el concepto «individuos tales que existen como máximo n individuos de C relacionados con ellos por R ».

$$I(\leq n R.C) = \{a \in \Delta / \text{cardinal}\{b \in \Delta / (a, b) \in I(R) \text{ y } b \in I(C)\} \leq n\}$$

La restricción numérica calificada mínima ($\geq n R.C$) se interpreta

$$I(\geq n R) = \{a \in \Delta / \text{cardinal}\{b \in \Delta / (a, b) \in I(R) \text{ y } b \in I(C)\} \geq n\}$$

Añadiendo estos constructores a $ALCI_{reg}$ obtenemos $ALCQI_{reg}$.

Restricciones funcionales sobre roles atómicos y sus inversos (F)

R es un **rol funcional** para una interpretación I si $(a, b) \in I(R)$ y $(a, c) \in I(R)$ implica que $b = c$. La restricción funcional sobre roles y sus inversos es un concepto que se contruye a partir del rol y que significa «los objetos para los cuales el rol es funcional».

$$I(\leq 1 R) = \{a \in \Delta / \text{cardinal}\{b \in \Delta / (a, b) \in I(R)\} \leq 1\}$$

$ALCFI_{reg}$ es la extensión de $ALCI_{reg}$ con este nuevo constructor.

Nominales (O)

Supongamos que $O = \{o, p, \dots\}$ es el conjunto de nombres de individuos del dominio.

Una interpretación I se puede extender añadiendo $I(o) \in \Delta$ para todo $o \in O$.

Podemos definir las instataciones como las fórmulas $C(o)$ o $R(o, p)$ con C un concepto, R un rol atómico y o, p individuos de O . Con estas nuevas fórmulas del lenguaje una ABox es un conjunto de dichas instalaciones.

$ALCQO_{reg}$ resulta de añadir nominales a $ALCQ_{reg}$.

$ALCIO_{reg}$ resulta de añadir nominales a $ALCI_{reg}$.

Dominios concretos

Un dominio concreto D es un conjunto $\Delta(D)$ (el dominio) más un conjunto $\text{Pred}(D)$ de los nombres de predicado de D . Cada nombre de predicado P de D se asocia con una ariedad n y un predicado n -ario de $\Delta(D)$.

Ejemplo: el dominio concreto N , tiene como dominio el conjunto ∞ de los números naturales y $\text{Pred}(N)$ el conjunto de los predicados binarios $<, \leq, >, \geq$.

Sistemas SH

$SHIQ$ es $ALCQI$ + roles transitivos + inclusión roles. Esta es otra notación muy utilizada para algunos sistemas de lógica descriptiva. La importancia de esta lógica, como veremos después, es que es la que actualmente se está implementando.

SHOIQ es *SHIQ* + nominales. Se demuestra también que *SHOIQ* es *SHOIN* extendida con restricciones cualificadas.

AHOIN(D) es *ALCIN* + nominales + dominios concretos (**D**).

Aunque extender una lógica con dominios concretos la dota de una expresividad muy valorada para representar ontologías, fácilmente puede llevar a la indecidibilidad. Veremos, sin embargo, que *SHOIN(D)* es decidible y es base para el lenguaje de ontología actualmente más aceptado.

Complejidad de las extensiones de ALC

El objetivo, recordemos, es encontrar algoritmos de decisión para los problemas de inferencia, satisfacibilidad, subsumición y consistencia en ABoxes para las lógicas más expresivas y con la menor complejidad posible, de forma que la implementación computacional sea afrontable.

La búsqueda de estos procedimientos de decisión ha sido uno de los objetivos fundamentales en el desarrollo de las lógicas descriptivas. Una de las maneras de obtenerlos es estudiando la conexión de las lógicas descriptivas con otras lógicas conocidas. Es el caso de la decidibilidad en *ALC* y en todas sus extensiones que se obtienen añadiendo constructores que en la lógica de primer orden se pueden expresar con 2 variables [3]. Es una consecuencia del hecho de que éstas son traducibles a un fragmento de la lógica de primer orden con dos variables, que se sabe que es decidible [15]. Sin embargo, la complejidad del procedimiento de decisión obtenido de esta manera es normalmente mayor del que realmente se necesita; por ejemplo el problema de satisfacibilidad para la lógica de primer orden con dos variables es NEXPTIME (que es una complejidad muy grande, aunque todavía es decidible) mientras que en *ALC* es PSPACE-hard [12,5] (es una complejidad menor). Otra manera de estudiar la complejidad es usando la conexión con las lógicas modales proposicionales [20].

Los algoritmos basados en tableaux son usados con frecuencia para estudiar la decidibilidad y complejidad de las lógicas descriptivas. Además, la casi totalidad de los razonadores automáticos para lógicas descriptivas se basan en algoritmos de tableaux.

En la tabla 3 se presentan las principales extensiones de *ALC*, especificando la o las nuevas propiedades expresables en la extensión y los límites para la complejidad computacional¹ del sistema resultante respecto a los problemas de inferencia que nos importan [11,13 y 22].

1. PSPACE (también llamada ESPACIOP) es la clase de los problemas de decisión que pueden ser resueltos por una máquina de Turing determinista en espacio polinomial y tiempo ilimitado. EXPTIME

TABLA 3.
Extensiones de AAX

LD	Propiedad expresable en la lógica	Complejidad
ALC	Lógica descriptiva básica	PSPACE
ALCN	+ restricciones numéricas no calificadas (N)	PSPACE
ALC_{reg}	+ expresiones regulares sobre roles (unión, composición, cierre reflexivo-transitivo) ($_{reg}$)	EXPTIME EXPTIME
$ALCI_{reg}$	+ inverso de roles (I)	EXPTIME
$ALCFI_{reg}$	+ restricciones funcionales sobre roles atómicos (F)	EXPTIME
$ALCQI_{reg}$	+ restricciones numéricas calificadas (Q)	EXPTIME
$ALCQO_{reg}$	+ un alfabeto para los objetos del dominio (O)	EXPTIME
$ALCIO_{reg}$		EXPTIME
SHIQ	ALCQI + roles transitivos + inclusión roles	EXPTIME
SHOIN	+ restricciones numéricas no calificadas (N)	EXPTIME
SHOIQ	SHIQ + nominales	EXPTIME
SHOIN(D)	SHOIN + dominios concretos	EXPTIME

6. ALGO DE HISTORIA

Los hechos más significativos de la historia de las lógicas descriptivas se pueden agrupar en cuatro fases:

1. 1970-80: *Semantic networks y frames*. Son los antecedentes de las lógicas descriptivas, aunque estos sistemas usados para la representación del conocimiento no pueden considerarse aún lógicas ya que no tienen una semántica formal.
2. 1980-90: *Aparecen las lógicas descriptivas*. Básicamente se implementan sistemas, como KLONE, K-REP, BACH, LOMM y usan sobre todo los llamados algoritmos estructurales. En esta etapa se normalizan las descripciones de conceptos, se comparan las diferentes sintaxis de las lógicas descriptivas normalizadas en uso, pero aún no existe una semántica formal estándar. Sólo se obtienen sistemas completos para lógicas descripti-

(también llamada EXP) es la clase de los problemas que pueden ser resueltos en una máquina de Turing determinista en tiempo $O(2^{p(n)})$, donde $p(n)$ es una función polinomial sobre n . PSPACE \subseteq EXPTIME.

NEXPTIME es la clase de los problemas de decisión que pueden ser resueltos en una máquina de Turing no-determinista en espacio $O(2^{p(n)})$, donde $p(n)$ es una función polinomial sobre n .

Para una introducción a la teoría de la complejidad se recomienda Sipser, M.: *Introduction to the Theory of Computation, Second Edition*. Course Technology (Thomson), 2005.

vas inexpressivas (que no detectan todas las subsumiciones e instanciaciones). Comienzan los estudios formales de la complejidad.

3. 1990-2000: *Algoritmos basados en tableaux*. Se implementan sistemas usando esos algoritmos (KRIS, CRACK, FACT, RACE) y se desarrollan procedimientos de inferencia automática para lógicas descriptivas muy expresivas. Continúa el análisis de la complejidad. Se estudia la relación con otras lógicas, en particular con las lógicas modales y fragmentos de la lógica de primer orden.
4. 2000-2005: Desarrollo de sistemas basados en tableaux para lógicas descriptivas muy expresivas. Comienzan las aplicaciones importantes a la Web semántica (lenguajes de ontologías) y la bioinformática.

7. UN LENGUAJE DE ONTOLOGÍAS: OWL DL

Desde 1990 la lógica descriptiva se vio ya como la candidata para proporcionar lenguajes de ontologías: es una lógica y por tanto posee una semántica formal y tiene potentes herramientas de razonamiento.

En ese momento inicial, sin embargo, había un desajuste entre las bases de conocimiento (TBox y ABox) grandes que necesitaban las ontologías y el poder expresivo de las lógicas descriptivas eficientes, pero en los últimos diez años se ha reducido ese desajuste.

El lenguaje de ontología basado en lógica descriptiva que actualmente se está imponiendo es una iniciativa del W3C [23], en particular hay que citar a Deborah L. MacGuinness y Frank van Harmelen. Es el lenguaje OWL DL.

Su sintaxis está basada en RDF (compatible con la web) y en frames (comprensible para los humanos). Su semántica se define por traducción a la lógica descriptiva. OWL DL se basa en el sistema de lógica descriptiva *SHOIN(D)*. En este lenguaje es posible expresar restricciones numéricas no cualificadas (N), axiomas terminológicos complejos, roles inversos y transitivos, y subroles (importantes en la comunidad ontológica). Es un sistema muy expresivo y además es decidible, su complejidad es EXPTIME (alto coste) pero existe un sistema razonador, FACT, que en la práctica funciona bien.

El sistema *SHOIQ* se ha barajado también como la lógica descriptiva detrás del OWL LD pero es decidible en NEXPTIME y difícil de implementar en la práctica. Nótese, sin embargo la proximidad entre *SHOIQ* y *SHOIN(D)*.

Otro lenguaje para ontología basado en lógica descriptiva y que también es utilizado es DAM + OIL . Aunque el hecho de que W3C haya apostado por OWL-LD lo hace por el momento el más importante en la carrera hacia un estándar para lenguaje de ontología.

En la tabla 4 se muestran estos dos lenguajes para ontologías con las lógicas descriptivas en las que se basan y las características de las mismas.

TABLA 4.
Lenguajes de ontologías y sus lógicas descriptivas

Lenguaje	Lógica Descriptiva	Complejidad e Implementación
DAM+OIL	SHIQ	dedidible, EXPTIME, FACT (razonador automático)
OWL DL	SHOIN(D)	dedidible, EXPTIME, FACT (razonador automático)

8. EXTENSIONES DE LA LÓGICA DESCRIPTIVA

En el proceso de formalización de los sistemas basados en marcos (frame-based) se dieron muchos casos de constructores que no podían formalizarse en primer orden, por ejemplo los asociados a nociones de incerteza o falta de información [14]. Entonces se pensó en utilizar algún tipo de razonamiento no monotónico. Siguiendo esta idea, autores como Baader y Hollunder [5] han adaptado herramientas de la lógica no-monotónica a las lógicas descriptivas, pero esta adaptación no es trivial dado que las LD no son, habitualmente, lógicas proposicionales.

Otras extensiones interesantes de las lógicas descriptivas consisten en añadir operadores modales como *obligación*, *posibilidad*, *temporalidad*, *creencia*. En muchos sentidos estas extensiones pueden considerarse como las correspondientes de la lógica de primer orden, y entonces se pueden heredar muchos de los problemas de las interpretaciones de las sentencias modales cuantificadas [17]. Baader y Ohlbach [6] estudian estas extensiones, que heredan los problemas de expresividad e inferencia propios de lenguajes descriptivos muy especializados.

La incerteza es otro importante aspecto del conocimiento a representar. Dos de las formas más exitosas de tratarla, la de la lógica probabilística y la de la lógica fuzzy han sido propuestas para incorporar a las lógicas descriptivas. Para el caso de las nociones probabilísticas, Heinsohn [16] y Jaeger [19], usan axiomas terminológicos probabilísticos, y las tareas de razonamiento consisten en medir probabilísticamente las instataciones y subsumiciones. Las lógicas descriptivas fuzzy [26] pueden caracterizar nociones como «muy» o «bastante», que no se asocian a valores numéricos, sino más bien a funciones graduadas. También hay algoritmos de razonamiento para la computación fuzzy de la subsumición usando métodos basados en tableaux.

9. CONCLUSIÓN

Las lógicas descriptivas son capaces de expresar las estructuras de conceptos y roles con operadores simples y reglas sencillas de construcción. Se pueden añadir o eliminar los operadores para obtener diferentes lenguajes con diferentes expresividades. El resultado es la familia de lenguajes más utilizados en la representación del conocimiento y base para lenguajes de ontologías, en particular para las ontologías de la web. Han sido muy estudiados desde la década de 1990 y se

han propuesto diferentes sistemas de inferencia y sus implementaciones. Su complejidad computacional está también bien estudiada.

La característica más importante de estas lógicas para la red ha sido, quizás, la cercanía entre teoría y práctica. El lado pragmático y la implementación de los sistemas han ido en paralelo al lado formal y teórico, lo cual es raro en la inteligencia artificial por un lado y en la lógica por otro. La necesidad de dotar a la web semántica con lenguajes interpretables a la vez por las máquinas y los humanos y que permitan un razonamiento eficaz automático ha hecho desarrollarse en los últimos diez años estas lógicas para la red. Pero esta historia no ha hecho más que comenzar.

BIBLIOGRAFÍA

- [1] ANTONIOU, G., *Nonmonotonic Reasoning*, MIT Press, 1997.
- [2] ANTONIOU, G. y VAN HARMELEN, F., *A Semantic Web Primer*, The MIT Press, 2004.
- [3] ARECES, C., *Logic Engineering (the case of description and hybrid logics)*, Amsterdam, ILLC-publications, 2000.
- [4] BAADER, F., *et al.* (eds.), *The Description Logic Handbook*, Cambridge U.P., 2003.
- [5] BAADER, F. y HOLLUNDER, F., «Embedding defaults into terminological knowledge representation formalisms», *Journal of Automated Reasoning*, 14: 149-180, 1995.
- [6] BAADER, F. y OHLBACH, H., A multi-dimensional terminological knowledge representation language. *Journal of Applied Non-Classical Logics*, 5:153-197, 1995.
- [7] BAADER, F. y NUTT, W., *Basic description logics*. En [4].
- [8] BERNERS-LEE, T., *Weaving the Web*, Harper Business. 2000.
- [9] BORGIDA, A. and BRACHMAN, R. J., *Conceptual Modeling with Description Logics*, In [4], pp.
- [10] BRACHMAN, R. J., LEVESQUE H. J., (eds.), *Readings in Knowledge Representation*, Los Altos, Morgan Kaufmann, 1985.
- [11] CALVANESE, D. y DE GIACOMO, G., *Expressive description logics*. In [4].
- [12] DOMINI, F. M., *Complexity of Reasoning*. In [4].
- [13] DOMINI, F. M.; LENZERINI, M.; NARDI, D. y Nutt, W., The complexity of concept languages. *Information and Computation*, 134:1-58, 1997.
- [14] GABBAY, D. *et al.* (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, Clarendon Press, 1994.
- [15] GRÄDEL, E.; KOLAITIS, G. and VARDI, M. Y. «On the decision problem for two-variable first-order logic», *Bulletin of Symbolic Logic*, 3(1):53-69, 1997.
- [16] HEINSOHN, J., «Probabilistic description logics», en PÉREZ DE MONTARAS, R. y POOL, D. (eds.), *Proc. Of the 10th Conf. On Uncertainty in Artificial Intelligence*, pp. 311-318, Seattle, Whashington, 1994. Los Altos, Morgan Kaufmann.
- [17] HUERTAS, A., *Modal Logic and Non-Classical Logic*. PhD thesis, Universidad de Barcelona, 1994.
- [18] HUERTAS A. y MANZANO, M., *Lógica para principiantes*, Alianza, 2004.

- [19] JAEGER, M., «Probabilistic reasoning in terminological logics». En TORASSO, P. *et al.* (eds.), *Proc. Of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94)*, pp. 305-316, 1994.
- [20] SCHILD, K., «A correspondence theory for terminological logics». In *Proc. Of the 12th Int. Joint Conf. on Artificial Intelligences (IJCAI'91)*, pp. 466-471, 1991.
- [21] SCHMIDT-SCHAUBB, M. y SMOLKA, G., «Attributive concept descriptions with complements», *Artificial Intelligence*, 48 (1):1-26, 1991.
- [22] TOBIES, S., *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, LuFG Theoretical Computer Science, Germany, RWTH-Aachen, 2001.
- [23] *W3C OWL Web Ontology Language*. <http://www.w3.org/TR/owl-features/>.
- [24] *W3C RDF/XML Syntax Specification*. <http://www.w3.org/TR/rdf-syntax-grammar>.
- [25] *W3C Semantic web*. <http://www.w3.org/2001/sw/>.
- [26] YEN, J., «Generalizing term subsumption languages to fuzzy logic». En RAITER, R. y MYOPOULOS, J. (eds.), *Proc. Of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pp. 472-477, 1991.