COMENTARIOS SOBRE LA RELACION ENTRE LA PROGRAMACION LOGICA Y LAS LOGICAS NO MONOTONAS

Agustín ARRIETA*

ABSTRACT

My purpose in this paper is to show the evolution of the relationship between nonmonotonic logic and logic programming. I think that there are two periods in the evolution of this relationship. The first one is the point of contact between these two fields that had been developed independently. In the second period, as I will show, the motivation to propose three-valued nonmonotonic logic comes from the study of the *relationship* between these two fields, and not from the study of nonmonotonic logic itself.

Introducción

El objetivo de este trabajo es rastrear la evolución seguida en los estudios sobre las relaciones entre las lógicas no monótonas y la programación lógica, para mostrar que si en un principio los puentes entre estos dos campos se han establecido teniendo en cuenta aspectos desarrollados en uno y otro de manera independiente, al proponerse las lógicas no monótonas trivalentes las motivaciones provienen fundamentalmente de los estudios que relacionan ambas disciplinas, y no tanto de las motivaciones asociadas con el estudio de las lógicas no monótonas.

El trabajo se divide en cuatro secciones. En la primera de ellas situamos la circunscripción y la lógica autoepistémica en el "mapa" de las lógicas no monótonas. Posteriormente se tratan algunos problemas computacionales asociados a dichas lógicas, para, en la tercera sección, presentar de manera concisa la "evolución" en el establecimiento de semánticas para clases de programas lógicos cada vez más generales. Finalmente, analizamos la relación existente entre los aspectos desarrollados en la primera y tercera sección.

1. Lógicas no monótonas: circunscripción y lógica autoepistémica

La lógica clásica cumple la propiedad de monotonía. Decimos que una lógica es monótona cuando verifica la siguiente propiedad sintáctica:

Sean Γ y Δ dos conjuntos de fórmulas cualesquiera. Sea α una fórmula. Si $\Gamma \vdash \alpha$, entonces $\Gamma \cup \Delta \vdash \alpha$.

THEORIA - Segunda Época - Vol. X
1995, Nº 24, 181-198

O, alternativamente (equivalentemente si hay corrección y completud), cuando cumpla la siguiente propiedad semántica:

Sean Γ y Δ dos conjuntos de fórmulas cualesquiera. Sea α una fórmula. Si $\Gamma \models \alpha$, entonces $\Gamma \cup \Delta \models \alpha$.

Los sistemas formales estándar son acumulativos: cuanto más conocimiento incorporemos al conjunto que define el conocimiento básico más conclusiones podremos[1] obtener y, en cualquier caso, lo que sí está asegurado es que las conclusiones anteriormente establecidas siguen siendo consecuencias del sistema. No hay ningún tipo de revisión sobre las consecuencias anteriores. Es debido a la presuposición que hacemos de disponer de un conocimiento completo, en el sentido de que no se consideran asunciones no explícitas sobre las que descansan conclusiones que posteriormente pueden abandonarse por dar lugar a inconsistencias. En el razonamiento monótono el nuevo conocimiento, plasmado en nuevos axiomas, no tiene ningún efecto sobre las consecuencias anteriores, ya que estas últimas se realizaron sobre la seguridad que proporciona el conocimiento completo. Así pues, cualquier esquema de razonamiento que considere principios diferentes, como la suposición de que no siempre tenemos conocimiento completo o, de manera más extrema, la suposición de que siempre conocemos de manera incompleta, deberá asumir otra serie de características para el concepto de consecuencia. Este es el problema al que se debe enfrentar la lógica cuando aborda el análisis del razonamiento ordinario. Este último es un razonamiento que, en general, descansa sobre *asunciones* y *conjeturas* implícitas para, precisamente, completar el conocimiento explícito, resultando, por ejemplo, perfectamente posible que algunas asunciones sobre las que se han extraído conclusiones, ante la adquisición de nuevo conocimiento[2], devengan falsas, con lo cual las conclusiones anteriores dejan de ser consecuencias del sistema. Este es un rasgo natural del razonamiento ordinario. Por esta razón, autores como Minsky y McCarthy[3] coinciden al afirmar la necesidad existente en el campo de la inteligencia artificial de estudiar en profundidad el razonamiento no monótono. Un logicista como McCarthy irá más allá y planteará la necesidad de investigar en torno a lógicas no monótonas, en sentido amplio.

Son numerosas las diferentes estrategias que se han planteado a la hora de llevar a cabo un tratamiento "lógico" de la no monotonía. En la actualidad, además existen diversas obras[4] que hacen una recopilación de todas ellas. Para estas estrategias, proponemos la siguiente clasificación:

(1) Lógicas ligadas a procesos de minimización.
(2) Lógicas con técnicas de razonamiento por defecto.
(3) Lógicas no monótonas con operadores modales.

Tenemos que señalar, en primer lugar, que esta clasificación es muy relativa (bien podría presentarse una clasificación dicotómica distinguiendo propuestas manifiestamente sintácticas y propuestas de marcado carácter semántico), ya que son numerosos los resultados obtenidos que ponen en relación estos formalismos que en un principio se nos presentaban heterogéneos. Basta pensar en la caracterización general que hace Shoham en (sho88) en términos de modelos preferenciales o también en los teoremas que relacionan formalismos pertenecientes a grupos distintos de nuestra clasificación. Nos referimos a los resultados de interrelación de Konolige en lo que a la lógica por defecto y a la lógica autoepistémica se refiere[5], o a los resultados de Gelfond, Przymusinski y Przymusinska[6] relacionando la lógica autoepistémica, la circunscripción y la negación como fallo y la asunción del mundo cerrado. Hoy en día, a pesar de la importante proliferación de formalismos lógicos no monótonos, existe una tendencia contrarrestadora, que se materializa en las ideas anteriormente señaladas, cuyo objetivo es "poner orden" en este mundo en plena emergencia, de tal forma que cabe pensar en la posibilidad de una futura teoría general del razonamiento no monótono. A pesar de esta tendencia a la unificación y a la generalización, creemos que es posible hacer distinciones entre los mismos, sobre todo teniendo en cuenta la intención diferente que subyace a cada propuesta.

En el primer grupo incluimos aquellas formalizaciones de razonamiento no monótono que conllevan de manera inmediata un proceso de minimización. Nos referimos de modo general a la "asunción del mundo cerrado" y a la 'circunscripción". En el primero minimizamos con respecto a los enunciados atómicos, haciendo que la teoría obtenida sea completa[7] con respecto a los mismos. La idea de la minimización queda recogida en la semántica que se obtiene para dicha asunción, ya que son los modelos minimales los que determinan dicha semántica. El conocido problema de la inconsistencia de algunas teorías bajo dicha asunción hace que se hayan planteado modos más sofisticados de la misma (asunción del mundo cerrado generalizada, asunción del mundo cerrado extendida).

Si trabajamos con la circunscripción, en su primera modalidad (circunscripción predicativa), lo que suponemos es una minimización con respecto a los objetos que caen bajo determinado predicado: sólo los objetos que explícitamente caen bajo ese predicado en el marco de la teoría básica son los objetos que cumplen dicho predicado. Si la minimización se lleva a cabo sobre todos los predicados y además asumimos el "cierre del dominio" y la "unicidad del nombre"[8], la circunscripción predicativa coincide con la asunción del mundo cerrado. Esto nos da una idea de la estrecha relación existente entre la idea de minimización y estas dos estrategias para la obtención de sistemas no monótonos, aunque hay una diferencia interesante entre ambos formalismos: la circunscripción presenta una mayor flexibilidad. La estrategia inherente a la circunscripción proporciona, de por sí, la posibilidad de pensar en una minimización de sólo algunos predicados o de minimizar algunos permitiendo variar otros, así como de establecer prioridades en el proceso de minimización. Así surgen diferentes métodos de circunscripción. Existen, además, teoremas que establecen la estrecha relación existente entre las distintas formas de circunscribir y versiones más sofisticadas de la asunción del mundo cerrado.[9]

La idea semántica subyacente a la circunscripción se traduce sintácticamente añadiendo a la teoría un axioma formulado en la lógica de segundo orden[10], para posteriormente obtener las consecuencias de la teoría siguiendo procesos inferenciales análogos a los de la lógica elemental. La inclusión de este axioma es la causa del carácter no monótono de la lógica. La circunscripción constituye una estrategia que comparte los mecanismos inferenciales asociados a la lógica clásica, aunque, por supuesto, resulta una lógica diferente de ésta, hecho que queda plasmado en su semántica y en las consecuencias que de ello se derivan.

En lo que al segundo grupo se refiere, cabe describir la estrategia que se sigue de la siguiente manera: se trata de introducir reglas de inferencia que de algún modo conllevan consideraciones metalógicas. Cabría hablar, incluso, de metarreglas. Una regla de inferencia en una lógica por defecto tiene una estructura: α: β/γ, donde γ se obtiene a partir de α, si α pertenece a la teoría y β es consistente con la misma. Esto último constituye, precisamente, la consideración metalógica a la que arriba hacíamos referencia. Semánticamente esto se traducirá también en una selección llevada a cabo entre los modelos de la teoría primitiva, pero, a diferencia de la circunscripción y de la asunción del mundo cerrado, los modelos que determinan su semántica no llevan la idea de minimalidad. Esta estrategia concebida a partir de reglas de inferencia, que denominaremos *defectos*, también tiene por objetivo completar el conocimiento de partida, pero lo hace de una manera más flexible. Esta mayor flexibilidad, también presente en la circunscripción, es más bien una contrapartida al carácter excesivamente rígido de la asunción del mundo cerrado que provoca la emergencia de teorías inconsistentes bajo la misma, debido al imperativo que subyace a ella: "hay que conseguir una teoría completa en lo que a los hechos básicos se refiere". En un principio la primera dificultad que cabe plantear a las lógicas por defecto es la referencia a cuestiones metalógicas que se hace en la formulación de la lógica en el nivel del lenguaje objeto. Esto entraña una circularidad evidente, ya que obtener algo como consecuencia de la lógica supone tomar decisiones sobre los enunciados que son consecuencia de la lógica. Por ello, el conjunto de consecuencias de una teoría por defecto se define recurriendo a la técnica del punto fijo. Frente a la circunscripción, la lógica por defecto nos desvía del abordaje semántico clásico. El recurso al concepto de extensión para la lógica por defecto es una prueba de ello.

La lógica autoepistémica, perteneciente al tercer grupo aquí propuesto por incorporar un operador modal, se plantea en el marco de una discusión[11] sobre la lógica no monótona de McDermott y Doyle en (mcdo80). Esta última, a su vez, constituye un puente entre la lógica por defecto y la lógica autoepistémica, a pesar de que Moore considera que la lógica autoepistémica no representa una modalidad de razonamiento por defecto. La lógica no monótona presentada en (mcdo80) pretende eliminar el problema analizado anteriormente, cuando señalábamos que en la noción de regla de inferencia se hacían consideraciones metalógicas. Con este objetivo McDermott y Doyle introducen en el lenguaje de su lógica un operador modal que representa esa propiedad metalógica de consistencia a la cual hemos hecho referencia. Esto hace que, posteriormente, la lógica obtenida tenga que ser estudiada semánticamente a través de las nociones kripkeanas. En (mcdo80) se plantea por primera vez la posibilidad de abordar el problema de la no monotonía a través de lógicas modales. La traducción que McDermott y Doyle proponen para un defecto, "α: β/γ", viene dado por el siguiente enunciado: $(\alpha \wedge k\beta) \rightarrow \gamma$, siendo k el operador modal de "consistencia".

En cualquier caso la perspectiva de Moore es diferente, ya que él concibe la lógica autoepistémica como un sistema que representa a un agente racional ideal que reflexiona sobre sus propias creencias. El autor considera que esta perspectiva impide ver la lógica autoepistémica como un sistema que funciona con inferencias por defecto. Además, Moore va más allá, ya que cree que McDermott y Doyle están equivocados cuando interpretan la lógica que ellos han ideado como una lógica que funciona a través de inferencias por defecto. Según Moore, una lógica que funciona con inferencias por defecto es no monótona porque sus inferencias son defectibles (defeasible).[12] Se trata de una especie de razonamiento prototípico, donde las conclusiones, por diferentes razones según el caso, tienen alta probabilidad de ser verdaderas. Contrariamente, la lógica presentada en (mcdo80), así como la lógica autoepistémica de Moore, es no monótona, a pesar de ser un sistema de inferencias válidas. La no monotonía viene motivada por el carácter indéxico del operador "k" (o "B" en su caso). Este operador cambia de "significado" según el contexto o, si se prefiere, según la evolución de la teoría. Estos cambios hacen que el conjunto de consecuencias de la teoría varíen con la misma, de tal manera que antiguas consecuencias, en un momento posterior, dejan de serlo, aunque siempre en un sistema de inferencias válidas. Este operador puede ser interpretado como una especie de "toma de conciencia" que el agente ideal hace sobre sus propias creencias. Tanto sobre lo que cree, como sobre lo que no cree.[13] Shoham, en (sho88), no comparte esta idea de Moore que hace referencia a las distintas naturalezas de la lógica por defecto y la lógica autoepistémica y, por lo tanto, a diferentes formas de inferencia no monótona. Shoham interpreta que Moore considera que las inferencias por defecto son útiles para entornos con "hechos estadísticos", mientras que las inferencias autoepistémicas se efectúan sobre lagunas en algún conocimiento particular. Considera que hay que distinguir entre el significado (meaning) de los enunciados (y el problema de su posible referencia a hechos de naturaleza estadística) y la utilidad (utility) de una lógica no monótona. La utilidad de una lógica no monótona se establece según criterios computacionales y es una cuestión relativamente independiente de la del significado de los enunciados.

En este apartado hemos pretendido situar la circunscripción y la lógica autoepistémica en el mapa de las lógicas no monótonas. Tanto la circunscripción como la lógica autoepistémica han sido los primeros formalismos lógicos no monótonos que han sido puestos en relación con la programación lógica. En el primer caso por coincidir en aspectos semánticos y en el segundo por la posible interpretación del operador B como la negación propia de la programación lógica. Ambos formalismos abren la puerta a una interesante línea de investigación que trata de establecer puentes entre dos disciplinas que se han desarrollado de forma paralela e independiente: nos referimos a la programación lógica y al razonamiento no monótono. En lo que resta nos ocuparemos de este tema, aunque nos centraremos, en lo que al razonamiento no monótono se refiere, únicamente de la circunscripción. En cualquier caso, nuestros comentarios son aplicables a la lógica autoepistémica.

2. Cuestiones complejo-computacionales

Uno de los marcos donde más se ha profundizado en las lógicas no monótonas ha sido y es en inteligencia artificial (IA). Desde el mismo momento en el que se propone la lógica como marco para la representación y tratamiento de las cuestiones relativas al conocimiento en IA, una de las críticas más habituales a dicho planteamiento es el que hace referencia a aspectos computacionales. Todo autor que haga una reflexión sobre las posibilidades de la lógica en inteligencia artificial, tarde o temprano, deberá confrontar el problema.[14] Podemos distinguir dos niveles en los "problemas complejo-computacionales". El primer nivel se refiere al problema de la decidibilidad. Sabemos que la lógica elemental es indecidible, es decir, no existe método algorítmico que nos permita decidir, en general, cuándo una fórmula de primer orden es teorema. Por lo tanto, todo formalismo que descanse sobre la lógica elemental arrastrará ineludiblemente el problema de la indecidibilidad. El método de resolución de Robinson constituye un procedimiento de semidecidibilidad para la lógica elemental, en el sentido de que dicho procedimiento puede proporcionar una respuesta, si la misma es afirmativa, y en caso contrario, en general, no responde.

Al problema del segundo nivel lo denominaremos "el problema de la intratabilidad". Efectivamente, todo mecanismo de inferencia que tenga como sustrato básico, como mínimo, la lógica proposicional se enfrenta con este problema añadido. Téngase en cuenta que el problema de la satisfacibilidad de una fórmula en la lógica de enunciados es un problema "situado" en la clase de complejidad NP (es un problema NP-completo), siendo la misma una clase de problemas intratables.[15] A partir de aquí, según vayamos introduciendo cuantificadores, el mismo problema, ya para lenguajes de mayor poder de expresión, va incrementándose en lo que a complejidad se refiere. Si añadimos operadores modales a la lógica proposicional el problema, según el sistema, pertenecerá, como mínimo, a NP y con la inclusión de cuantificadores, si conseguimos salvar el problema de la decidibilidad, deberemos de enfrentarnos, en todo caso, con el problema de la alta intratabilidad.

Los problemas complejo-computacionales, vistos desde la perspectiva de los dos niveles que hemos distinguido, provocan un pesimismo con respecto a la posibilidad de "implementación" y de "utilidad práctica" que presenta la lógica cuando pensamos en una lógica para la formalización del razonamiento ordinario. Ante dichos problemas caben varias posibilidades. Destacamos dos. La primera de ellas hace referencia a la reducción del poder de expresión de los lenguajes, identificando sublenguajes más tratables, mientras que la segunda trata de evaluar la complejidad para los casos habituales, con lo cual el mecanismo, en teoría, no tratable puede mantener interés. Téngase en cuenta que en teoría de la complejidad se establecen las clasificaciones sobre "el caso peor".[16]

Las lógicas no monótonas, por razones obvias, también se ven afectadas por dichos problemas. Por ejemplo, en lo que a la computación se refiere, la circunscripción[17] tiene un alto grado de dificultad, originada fundamentalmente por el hecho de que al circunscribir una teoría de primer orden obtenemos una teoría de segundo orden, con lo que este salto supone. Papadimitriou y Kolaitis, tras caracterizar complejo-estructuralmente algunos problemas en el marco de la circunscripción, señalan:

> is there a different formalization of common-sense reasoning which on the one hand is computationally more tractable than circumscription, and on the other hand retains most salient features of it?[18]

Los trabajos de Lifschitz y Rabinov[19] son una muestra de una de las posibles líneas de trabajo planteables a partir de estos resultados tan negativos. Ambos autores identifican lenguajes que, proporcionando un poder de expresión "interesante", hacen que la circunscripción de sus fórmulas permanezca en "primer orden".

Otra posible vía se configura en el marco del estudio de las relaciones existentes entre los formalismos no monótonos y la programación lógica. Estas dos disciplinas no tienen, en principio, objetivos comunes. La programación lógica aborda el problema de diseño de lenguajes de programación declarativos inspirados en la lógica, mientras que en los estudios sobre las lógicas no monótonas no son de principal relevancia las cuestiones relativas al diseño de lenguajes.

El carácter no monótono de la "negación como fallo" hace que la definición de la semántica de los lenguajes de programación lógica resulte especialmente dificultosa, ya que nos enfrentamos con semánticas no-clásicas. En cualquier caso esta noción de negación, a la vez que genera dificultades, tiene su lado positivo, si pensamos que la programación lógica, según se vayan definiendo con mayor precisión su semántica declarativa y procedimental, puede proporcionar un marco adecuado y dotado de cierta eficiencia, donde se puedan representar e implementar formalismos no monótonos. Al considerar la programación lógica como el marco para la tratabilidad de los formalismos no monótonos, estamos también reconociendo sublenguajes que pueden resultar tratables, ya que en los lenguajes de programación lógica se trabaja, en el caso más sencillo, con cláusulas Horn y, según se avanza en estos estudios, con otros sublenguajes que analizaremos posteriormente. En cualquier caso todos estos lenguajes son de un poder de expresión limitado frente al de la lógica elemental.

La relación "programación lógica/lógica no monótona", además de proporcionar un posible marco a partir del cual cabe encontrar una salida al pesimismo computacional planteado sobre la lógica no monótona, nos permite estudiar características estructurales comunes a todos los formalismos propuestos (lógica por defecto, autoepistémica, circunscripción etc) ya que todos ellos pueden ser estudiados en un lugar común. Así lo señala Przymusinski:

> They may also contribute to a better understanding of relations existing between various forms of non-monotonic reasoning and to the eventual discovery of deeper underlying principles of non-monotonic-reasoning.[20]

3. Semánticas canónicas dentro de la programación lógica

Es sabido que el lenguaje de programación PROLOG se concibe en el marco de un sublenguaje de la lógica de predicados de primer orden. Nos referimos al lenguaje que acepta como fórmulas únicamente cláusulas con, a lo sumo, un literal positivo, esto es, cláusulas de Horn. Independientemente de las peculiaridades del lenguaje PROLOG, dentro de la programación lógica se investiga en torno a lenguajes más ricos, en lo que a poder de expresión se refiere. El problema fundamental que se plantea a la hora de llevar a cabo dichas extensiones es el de determinar cuál es la semántica correspondiente a las mismas. El estudio de estas semánticas constituye, en la actualidad, una línea de investigación muy importante dentro de la programación lógica. Paralelamente a este trabajo se plantea el problema del establecimiento de técnicas sintácticas, muy ligadas al método de resolución, que permitan atrapar, digámoslo así, mecánicamente dichas semánticas declarativas sobre la base de modelos canónicos[21] o modelos "in mente" del que propone una determinada teoría en el marco de la programación lógica. En la literatura estas técnicas son conocidas como las semánticas procedimentales asociadas a un programa. En este trabajo no haremos referencia a las semánticas procedimentales.

3.1. Semántica del modelo mínimo para programas positivos

Consideramos un programa positivo como un programa cuyas reglas son de la forma: $A \leftarrow A_1, A_2,..., A_m$, siendo $m \geq 0$ y $A_1, A_2,..., A_m$ y A átomos cualesquiera.

Una de las semánticas que se han propuesto para los programas lógicos es la que se obtiene bajo la compleción de Clark (COMP). Con la perspectiva que esta semántica proporciona las relaciones que en el programa se definen débilmente a través de una cláusula "si-entonces" pasan a ser definidas mediante una expresión "si y sólo si". Al ver P como COM(P), se produce una divergencia con respecto a la semántica clásica:

Ejemplo 1

Sea el programa $\{Pa \leftarrow Pa, Rb\}$

Si consideramos la compleción de Clark del programa, $(\forall x(Rx \leftrightarrow x{=}b) \land \forall x(Px \leftrightarrow (x{=}a \land Pa)))$, comprobamos que obtenemos como consecuencia $\neg Ra$, hecho que no es posible sobre la base del concepto clásico de consecuencia. También se comprueba que si añadimos a nuestra teoría el átomo Ra, $\neg Ra$ deja de ser consecuencia de COM(P). Estamos ante un ejemplo de la no monotonía de COM(P). Las divergencias se centran en la información negativa, ya que literales negativos no deducibles bajo la perspectiva clásica, sí pueden deducirse ahora. Estas características hacen de la programación lógica y de las semánticas ligadas a la misma un campo de investigación muy interesante, ya que se genera un concepto de negación divergente del concepto clásico y con un carácter no monótono. Przymusinski, en (prz89), señala:

> Another, perhaps more important, reason is that we really do not want classical negation in logic programs.

Otra de las semánticas declarativas que se han asociado a los programas positivos es la que nos proporciona el operador T_p[22] o el modelo mínimo de Herbrand. Cuando "leemos" un programa bajo el prisma que nos proporciona la semántica del modelo de Herbrand mínimo, aparece un problema en relación a una de las características arriba mencionadas, esto es, sea cual sea la semántica que consideremos nunca debe obtenerse *más información positiva* que la que el propio programa positivo nos proporciona a través de la semántica clásica. Si consideramos un programa con una única cláusula "Pa" y la semántica determinada por el modelo mínimo de Herbrand, entonces $\forall xPx$ es una consecuencia del programa. Para evitar esto último caben varias soluciones. La primera es considerar los modelos minimales en general, sin limitarnos a los modelos de Herbrand. Otra solucion se presenta en (ros92). Se trata de suponer la existencia en el dominio de un elemento extraño que es inombrable en el programa. En lo que sigue dejaremos de lado este problema y nos limitaremos a modelos de Herbrand.

El modelo mínimo de Herbrand nos proporciona una semántica divergente tanto con respecto a la semántica clásica como a la correspondiente a la compleción de Clark.

Si consideramos COM(P), siendo $P=\{p \leftarrow p\}$, vemos que ni p ni $\neg p$ son consecuencia del programa, mientras que $\neg p$ sí es consecuencia del programa bajo la perspectiva del modelo mínimo de Herbrand. A lo largo de este trabajo nos centraremos, especialmente, en aquellas semánticas determinadas por un único modelo de Herbrand. Este es el caso, por ejemplo, del modelo mínimo para programas positivos. También iremos señalando las divergencias y convergencias que las semánticas de modelos canónicos sostienen con respecto a las semánticas definidas sobre la noción de compleción.

3.2. Semántica del modelo perfecto para programas estratificados

Es interesante dar el paso a programas más generales que constituyan un marco más adecuado para el diseño de teorías y/o bases de conocimiento cuya implementación sobre la base de programas exclusivamente positivos supondría la necesidad de la aplicación de técnicas, en algunos casos, muy complejas. Asimismo es interesante que las semánticas que se propongan para estos programas generales colapsen en las semánticas ya analizadas, cuando el programa en cuestión sea positivo.

Un programa general tiene la misma estructura que un programa positivo, con la salvedad de que en el cuerpo de sus cláusulas se permiten literales negativos.

Es sabido que la compleción de Clark nos lleva a teorías inconsistentes, en el caso de considerar programas generales. Si el programa general es estratificado el problema desaparece, siendo posible, además, definir una semántica adecuada y definida sobre un modelo canónico para dicho programa. Nos referimos a la semántica del modelo perfecto. La semántica del modelo mínimo deja de estar bien-definida cuando damos el paso de programas positivos a programas generales, ya que un programa con literales negativos puede tener más de un modelo de Herbrand minimal.

Ejemplo 2

Sea el programa P = {Ba, Pb, Gx ← Px $\wedge$ ¬Ax}

Este programa tiene dos modelos minimales:

M_1 = {Ba, Pb, Gb} y M_2 = {Ba, Pb, Ab}[23]

Por lo tanto no tenemos un modelo canónico único que determine la semántica.

La semántica de modelos perfectos está determinada por una serie de relaciones sintácticas en el interior de los programas. Estas propiedades se fundamentan en unas prioridades relativas, establecidas entre los átomos (y relaciones) del programa, de tal forma, que los consecuentes de las reglas tienen menor prioridad que sus premisas negativas. Por otro lado, las premisas positivas tienen mayor o igual prioridad que los átomos de la cabeza de sus respectivas reglas. Estamos estableciendo un orden de prioridades entre elementos de la base de Herbrand. Sobre este "orden", posteriormente se inducirá un orden entre relaciones.

En realidad al decidir cuáles son las prioridades tenemos en cuenta dos criterios estrechamente relacionados:

1) Una mayor o igual prioridad sobre lo más específico, es decir, sobre los átomos correspondientes a literales del cuerpo de las reglas. Esta mayor prioridad se traducirá, como veremos, en una "anterioridad" a la hora de ser minimizados.

2) Intuitivamente podemos decir que los literales negativos son estrictamente de mayor prioridad, ya que están asociados a aquellos símbolos de relación que señalan algún tipo de "excepcionalidad a una regla".

Si consideramos el *ejemplo 2* vemos que Aa, por ejemplo, es de mayor prioridad que Ga. Esta decisión afectará a todo el desarrollo que de la semántica se haga. Pero, a la vez, vemos que esa mayor prioridad es algo que se decide basándose en la forma elegida a la hora de "presentar" el programa. Es posible, siguiendo reglas de interdefinición de conectores de la lógica proposicional, presentar la tercera regla de P de una manera equivalente, intercambiando de "lugar" los literales Gx y Ax. En este caso, por supuesto variarían las prioridades. La semántica del modelo perfecto pretende adecuarse a unas supuestas intenciones del que diseña el programa, teniendo en cuenta que estas intenciones quedan expresadas en la "manera" de escribir dicho programa.

Sobre la relación de prioridad (<) entre elementos de la base de Herbrand, cabe establecer una relación entre símbolos de relación ($<_p$). A partir de estas nociones se desarrolla toda la semántica de modelos perfectos.

-Sean P y Q dos símbolos de relación:

P $<_p$ Q si y sólo si existen a y b tales que Pa < Qb.

-Sean N y M dos modelos distintos de P.

N es preferible a M (N << M) si y sólo si para todo átomo A $\in$ N-M, hay un átomo B $\in$ M-N, tal que A < B.

Considerando el ejemplo anterior establecemos las siguientes relaciones:

Ga < Aa, Gb < Ab, Ga $\leq$ Pa, Gb $\leq$ Pb. Vemos que las siguientes interpretaciones son modelos de P:

N = {Ba, Pb, Gb} y M = {Ba, Pb, Ab}. Se comprueba que N << M, ya que N-M = {Gb} y M-N = {Ab}, siendo Ab > Gb. Dicho de otra forma en N-M tenemos un átomo que tiene menor prioridad que el átomo de M-N. Ya hemos señalado que a la idea de "mayor prioridad" subyace la idea de un anteponerse en el proceso de minimización, por lo cual N es preferible a M ya que la relación a minimizar con anterioridad se encuentra en M y no en N. Llamamos la atención sobre el hecho de que los modelos propuestos son minimales con respecto al conjunto de todos los símbolos de relación del lenguaje de P.

Decimos que un modelo M es perfecto si y sólo si no hay modelos preferibles con respecto a M.

Recalcamos que hasta ahora no se ha impuesto ninguna restricción sobre el conjunto de los programas generales y que todas las definiciones propuestas son aplicables a dichos programas.

A partir de estas definiciones cabe mostrar que la noción de modelo perfecto coincide con la definición de modelo mínimo, cuando el programa es positivo. También cabe caracterizar la clase de programas estratificados y localmente estratificados. Un programa es estratificado si y sólo si $<_p$ es noetheriana.[24] Un programa es localmente estratificado si y sólo si < es noetheriana. Consecuencia inmediata de la definición es que todo programa estratificado es localmente estratificado.[25]

Desde un punto de vista, digamos, de lenguaje de programación imperativo lo que evita la estratificación local es que haya llamadas recursivas de un "procedimiento" a sí mismo a través de la negación sin ningún tipo de variación en sus argumentos, de tal manera que el proceso no tenga parada. Si tenemos el programa {p $\leftarrow \neg$p}, donde se produce una llamada recursiva a través de la negación, además sin símbolos de función que afecten a los argumentos de este procedimiento, la relación "<" deja de ser irreflexiva, ya que obtenemos p<p<p<p... y, por lo tanto, la relación < tampoco es noetheriana. Un programa lógico puede ser localmente estratificado permitiendo llamadas recursivas a través de la negación. Pero en este caso intervienen símbolos de función, de tal forma, que en la llamada el argumento del nuevo objetivo se simplifica y la cadena que forma < tiene un final. Para esta clase de programas la semántica del modelo perfecto está bien definida. También la compleción de Clark proporciona una semántica para estos programas ya que es su compleción es consistente, como se demuestra en (abw88). En cualquier caso ambas semánticas divergen y la semántica del modelo perfecto resultará preferible, pues parece que responde mejor a lo que hemos denominado "intenciones" del diseñador. Este "responder" mejor adquiere sentido siempre y cuando estemos pensando en una programación lógica en relación con el razonamiento no monótono.

3.3. Semántica del modelo bien-fundado para programas generales

Tras la obtención de semánticas canónicas para la clase de los programas (localmente) estratificados, el siguiente objetivo es el obtener dichas semánticas para clases más generales de programas lógicos. En este intento se plantean las semánticas trivalentes. Fitting, en (fit85), lo expresa de la siguiente manera:

> The use of conventional classical logic is misleading for characterizing the behavior of logic programs because a logic program, when queried, will do one of three things: succed with the query, fail with it, or no respond because it has fallen into infinite backtracking. In (kle52) Kleene proposed a three-valued logic for use in recursive function theory. The so-called third truth value was really undefined: truth valued non determined. This logic is a useful tool in logic-program especification, and in particular, for describing models.

Tanto Fitting como Kunen en (kun87) trabajan en el marco de la compleción de Clark. Con la compleción trivalente, al introducir un nuevo valor de verdad, se solucionan algunos problemas. Es el caso de la inconsistencia de la compleción de Clark bivalente para algunos programas. Fitting demuestra que la compleción de un programa tiene un modelo mínimo trivalente que "determina" su semántica. Ross, Schlipf y Van Gelder, en (rsv88), consideran que esta semántica no se ajusta de una manera adecuada a las intenciones del diseñador del programa. Esta afirmación se justifica si tenemos en cuenta que el trabajo de estos últimos autores se enmarca en la línea de investigación que tiene en cuenta la relación existente entre la programación lógica y las lógicas no monótonas, mientras que tanto Kunen como Fitting trabajan fundamentalmente con problemas internos de la programación lógica. En este apartado nos limitaremos a exponer alguna características básicas que nos permitan exponer los puntos de divergencia entre ambas semánticas.

La diferencia básica la encontramos en los órdenes parciales, sobre los que obtenemos los modelos minimales. Estas relaciones se definen sobre el universo de las interpretaciones de Herbrand trivalentes, donde caben fórmulas atómicas que no son ni verdaderas ni falsas. En la perspectiva de (rsv88) minimizamos los hechos verdaderos, maximizando los falsos. Cuando las interpretaciones de Herbrand sean bivalentes, este orden parcial colapsa con los considerados con respecto a programas estratificados y positivos. En el orden parcial "de Fitting", los modelos minimales son los que minimizan tanto la información verdadera, como la falsa, maximizando la información indefinida. El modelo de Herbrand, obtenido siguiendo la primera estrategia y caracterizado en términos de modelos preferibles, constituye el modelo bien-fundado, que determina la semántica de todo programa lógico.

Si consideramos la compleción del programa $\{p \leftarrow p\}$, el modelo mínimo en la relación de orden de Fitting es $M = \emptyset$ (esto es, p es indefinido). En cambio el modelo bien-fundado, que en este caso coincide con el modelo mínimo por tratarse de un programa positivo, es $\{\neg p\}$.[26] Se aprecia que el modelo bien-fundado es un modelo de COM(P), pero no coincide con el modelo que se obtiene en la semántica de Fitting. Lo mismo ocurre en la relación entre la semántica del modelo perfecto y la semántica inducida por la compleción bivalente. El modelo perfecto es un modelo de COM(P), pero no tiene por qué ser un modelo minimal de la compleción.

Al hablar de modelos canónicos, hemos considerado única y exclusivamente modelos de Herbrand, salvo para el caso de la compleción de Clark. Se han señalado la existencia de divergencias de las semánticas definidas sobre el modelo mínimo, perfecto y bien-fundado, con respecto a las complecciones tanto bivalente como trivalente. También hemos comprobado que el paso de lógicas bivalentes a trivalentes ha hecho posible el establecimiento de semánticas bien-definidas para los programas lógicos generales.[27]

4. Circunscripción y programación lógica

Anteriormente, ya hemos señalado que la circunscripción está muy relacionada con los procesos de minimización. Efectivamente, la circunscripción, en sus distintas estrategias, tiene una semántica siempre asociada a modelos minimales.[28] La circunscripción predicativa es correcta y completa con respecto a la semántica de los modelos minimales débiles.[29] Si permitimos que algunas relaciones de la teoría vayan variando mientras se circunscriben otras (circunscripción paralela), entonces la semántica viene determinada por un subconjunto del conjunto de los modelos minimales anteriores. Esta semántica permite inferir "más cosas", al ser menor el número de modelos que la determinan. En esta progresión, si ahora además establecemos prioridades en los procesos de minimización (circunscripción con prioridad), también es un subconjunto del conjunto de los modelos minimales para la circunscripción paralela el que determina la semántica.

La circunscripción paralela permite inferir más hechos que la circunscripción predicativa, ya que al definir las semánticas de ambas así como la relación sobre la que establecemos los elementos minimales, dicha relación es en el primer caso un orden parcial y en el segundo caso un preorden. Esto hace que elementos no relacionables en el marco de la circunscripción predicativa, sí lo sean para la circunscripción paralela, disminuyendo el número de elementos minimales. Un proceso análogo cabe identificar al relacionar la circunscripción paralela con la circunscripción con prioridad.

Teniendo en cuenta la relación existente entre la semántica asociada a la programación lógica y a la circunscripción, al descansar ambas sobre modelos minimales, y las restricciones sintácticas asociadas a los programas lógicos, cabe pensar en la vía de relación entre las mismas. Para ello la primera exigencia es la adecuación sintáctica. El siguiente nivel de la adecuación afecta a la semántica. Los modelos que se consideran son los modelos de la teoría más los axiomas de la unicidad del nombre y además deben ser minimales con respecto a una política que considere la circunscripción de todos los predicados. En la programación lógica no hay manera de distinguir entre relaciones minimizables y no-minimizables. Por lo tanto, las únicas estrategias circunscriptivas que pueden admitir tratamiento mediante la programación lógica son las que consideran minimización para toda relación, ya sea minimización priorizada o no. Esto hace que la traducción de un programa lógico a una teoría con circunscripción siempre sea posible, mientras que la relación conversa no se cumple.

Consideremos una teoría T y una política de circunscripción predicativa en la que se minimizan todos las relaciones de T CIRCUN(T; Q; ∅)[30], siendo Q el conjunto de todos los símbolos de relación que aparecen en T). Si esta teoría es traducible a un programa positivo P, entonces se cumple:

P $\models_{min} \phi$ si y sólo si U, CIRCUN(T; Q; ∅) $\models \phi$.[31]

Pensemos ahora en aquellos programas para los cuales cabe definir una estratificación del conjunto de símbolos de relación que aparecen en él.[32] Nos estamos refiriendo a los programas localmente estratificados. Hemos visto que la semántica del modelo perfecto es una semántica adecuada para estos programas. El modelo perfecto es seleccionado de entre los modelos minimales con arreglo a algunas prioridades establecidas en los procesos de minimización de los símbolos de relación que aparecen en el programa. Esta idea concuerda con el objetivo que McCarthy en (mcc86) se marca cuando plantea la circunscripción con prioridades. Si queremos que circunscripción y programación lógica participen del mismo concepto de consecuencia semántica deberemos presuponer, como antes hemos señalado, que las interpretaciones con las que trabajamos verifican los axiomas de la unicidad del nombre.

Sea P un programa estratificado para una teoría T y $\{S_1, S_2, \ldots, S_n\}$ una estratificación del mismo:

P $\models_{perf} \phi$ si y sólo si U, CIRCUN(T; $S_1 > S_2 > \ldots > S_n$; ∅)$\models \phi$.[33]

El problema de la "traducción" de una teoría en un programa lógico no es una cuestión trivial, ya que las equivalencias semánticas de la lógica clásica no resultan válidas en dicho proceso. Téngase en cuenta que la estratificación se define basándose en el modo en que está escrito el programa, y cambios en éste afectan a la semántica. Por ejemplo, la aplicación de una equivalencia lógica como "$\neg p \rightarrow q \equiv \neg q \rightarrow p$", puede afectar al sistema de prioridades de manera esencial y asimismo al concepto de consecuencia semántica que induce el nuevo modelo perfecto.[34]

Hasta ahora hemos mostrado cómo la circunscripción, tal como se ha planteado en los trabajos sobre lógicas no monótonas, puede relacionarse, de forma natural y con algunas limitaciones, con la programación lógica. También hemos visto que la obtención de una semántica para todo programa lógico nos lleva a dar el salto a marcos trivalentes. Teniendo en cuenta estos dos hechos Przymusinski, en (prz91), cierra la relación entre la programación lógica y la circunscripción, proponiendo una circunscripción trivalente en relación con la semántica bien-fundada. El proceso hacia la circunscripción trivalente es relativamente sencillo. Se trata de definir el concepto de orden en el universo de las interpretaciones trivalentes. Este concepto de orden hace que en los elementos minimales haya una minimización de información verdadera y una maximización de información falsa.

En (prz91) y sobre la base de los resultados de (prz88) se presenta el concepto de estratificación dinámica, aplicable a todo programa, con un último estrato de átomos indefinidos. También se nos propone un concepto de circunscripción trivalente con prioridad, para finalmente establecer el paralelismo existente entre la circunscripción trivalente y la programación lógica bajo la perspectiva de la semántica bien-fundada.

Es decir, sea P un programa y $\{S_1, S_2,..., S_n\}$ una estratificación (dinámica) del mismo. Tenemos que:

$P \models_{bf} \phi$ si y sólo si $CIRCUN3(T; S_1>S_2>...>S_n) \models \phi$.

Conclusiones

Hemos visto que el recurso a la trivalencia constituye la vía para el cierre coherente en el establecimiento de semánticas para lenguajes lógicos cada vez más expresivos. Decimos "cierre coherente" ya que todas las semánticas propuestas para las distintas clases de programas colapsan en las semánticas para las clases más específicas.

Hemos señalado también cómo pueden relacionarse de una forma natural la programación lógica y las lógicas no monótonas. Para ello no ha resultado necesario "forzar" ninguno de los dos campos, que se han desarrollado de una manera relativamente independiente.

Finalmente el estudio de esta relación, ha llevado a plantear la lógica no monótona trivalente, que hasta donde nosotros conocemos, no había tenido tratamiento en los trabajos sobre la no-monotonía. Consideramos que la misma se han planteado motivada fundamentalmente por un intento de cierre, también coherente, de las investigaciones sobre las relaciones existentes entre los dos citados campos. Las motivaciones son teóricas. No ha sido el manejo de bases de conocimiento o ciertas aplicaciones concretas de los formalismos no monótonos las que han provocado el planteamiento de la circunscripción trivalente. En la literatura, hasta donde nosotros conocemos, no encontramos aplicaciones, ni tan siquiera consideraciones relevantes de estos formalismos no monótonos trivalentes.[35] La estrecha relación existente entre la semántica del modelo perfecto y la circunscripción bivalente ha generado la necesidad de pensar en la circunscripción trivalente en aras a un ajuste entre esta última y la semántica del modelo bien-fundado de carácter trivalente. Las motivaciones subyacentes a las lógicas no monótonas trivalentes no provienen desde los estudios de la lógica no monótona.

Concluimos, pues, que estos estudios, en lo que a marcos trivalentes se refieren, tienen el interés de marcar los límites y posibles pautas para el diseño de entornos basados en la programación lógica donde quepa la implementación de los formalismos no monótonos. En cualquier caso, pensamos que, para el tratamiento de problemas habituales de razonamiento ordinario, el caso bivalente cubre "generosamente" los casos situados en la franja intermedia de sencillez-dificultad.

Agradecimientos

Nuestro agradecimiento al Gobierno Vasco, ya que este trabajo ha sido realizado en el Center for the Study of Language and Information (CSLI), gracias a una beca posdoctoral y parcialmente al proyecto de investigación PGV9225. Las facilidades ofrecidas por el staff del CSLI han permitido que la finalización de este artículo haya sido sencilla y agradable.

Nuestro agradecimiento, también, a Jesús María Larrazábal, L.A. Pérez Miranda y a Eduardo Alonso con quienes en numerosas ocasiones hemos analizado los problemas que en este trabajo se tratan. Finalmente agradecer a los informantes anónimos sus interesantes observaciones, tanto sobre aspectos estilísticos, como de contenido.

Manuscrito recibido, 21 de julio, 1994.
Versión final recibida, 11 de abril, 1995.

* Departamento de Lógica y Filosofía de la Ciencia
Universidad del País Vasco-Euskal Herriko Unibertsitatea
Apartado 1.249
20080 Donostia-San Sebastián
E-mail: ylparura@sf.ehu.es

Notas

1 Decimos únicamente "podremos" ya que dependerá de la *independencia* de la nueva premisa con respecto a las primitivas.

2 Pensamos en nuevo conocimiento que se añade al *conocimiento explícito* anterior, sin necesariamente revisarlo. Es decir, en este momento estamos pensando más en términos de actualización que en términos de revisión.

3 Ver (mcc88) y (min75).

4 Destacamos como manuales de lógica no-monótona: (eth88), (bes89), (bre91) y (luk90)

5 Ver (kon87).

6 Ver, especialmente, (gpp89), (gpp90) y (prz91).

7 Para todo enunciado básico se cumple que dicho enunciado o su negación es consecuencia de la teoría, bajo la asunción del mundo cerrado.

8 Señalar que "el cierre del dominio" supone considerar que los únicos individuos a tener en cuenta en el desarrollo de la teoría son los que son denotados por medio de los símbolos de constante individual que aparecen en ella. Con la asunción de la unicidad del nombre se indica una propiedad que cumplen, por definición, las interpretaciones de Herbrand: dos símbolos de término distintos denotan individuos distintos del dominio de la interpretación. Estas asunciones son habituales también en los lenguajes para bases de datos, donde preguntas, por ejemplo, en torno a si todos los individuos cumplen una determinada propiedad son contestadas afirmativamente en el caso de que, efectivamente, los individuos que aparecen explícitamente en la base de datos cumplan esa propiedad, siempre según la información almacenada en la misma, esto es, suponiendo que no hay más individuos que los que aparecen explícitamente en la base y suponiendo también que nombres distintos denotan individuos distintos.

9 En (gpp89) se demuestra la equivalencia entre la asunción del mundo cerrado extendida y la circunscripción paralela.

10 A excepción de la circunscripción punto a punto (pointwise circumscription).

11 Ver (moo83).

12 Utilizo el término propuesto por Pérez Miranda en (per93).

13 Existe una diferencia entre la lógica no-monótona de McDermott-Doyle y la lógica de Moore. Mientras en la primera el agente no es consciente más que de lo que no cree, es decir, de lo que es consistente con la teoría ($S = \text{Teoría}(T \cup \{\neg B\alpha: \alpha \notin S\})$, siendo T la teoría de partida y S la definición de la teoría generada a través de la lógica de McDermott-Doyle), en la segunda esta introspección afecta tanto a lo que se cree como a lo que no se cree: $S = \text{Teoría}(T \cup \{B\alpha: \alpha \in S\} \cup \{\neg B\alpha: \alpha \notin S\})$.

14 Shoham en (sho88), Etherington en (eth88), Nilsson en (nil91) y Levesque en (lev88) abordan el problema.

15 P es la clase de complejidad de los problemas tratables.

16 Ver (bdg88), y las definiciones de las funciones "tiempo" y "espacio".

17 Sobre la complejidad de las lógicas no monótonas y de la lógica autoepistémica, ver (got92).

18 Ver (kopa88), pp. 468-469.

19 Ver (lif85), (rab89).

20 Przymusinski en (prz90), p. 50.

21 Utilizamos el término canónico en el sentido de (rsv88) y (geli88b).

22 Ver (llo87), (she88).

23 Los modelos de Herbrand los representamos como conjuntos de fórmulas atómicas cerradas o básicas.

24 Una relación < definida en un conjunto C es noetheriana, si no existe una cadena infinita creciente.

25 Para más detalles ver (prz88).

26 En el marco trivalente representamos las interpretaciones como conjuntos consistentes de literales básicos.

27 En (ros89), Ross propone ir más allá de lo que hemos denominado programas generales, proponiéndose la semántica bien-fundada para las bases disyuntivas. Estas últimas permiten la disyunción de átomos en la cabeza de las cláusulas. Encontramos otra generalización en (lave92). En este trabajo se plantean semánticas para programas que permiten la negación en la cabeza de las reglas.

28 Las ideas básicas sobre la semántica de la circunscripción las presenta McCarthy en (mcc80).

29 Ver (luk90).

30 CIRCUN(T; A; B), representa la circunscripción de la teoría T, siendo A el conjunto de relaciones minimizadas y B el conjunto de relaciones que pueden variar en el proceso de minimización.

31 U es el axioma de unicidad del nombre. El concepto de consecuencia lógica mencionado, es el inducido por la semántica del modelo mínimo. Posteriormente haremos referencia al concepto de consecuencia semántica inducido por el modelo perfecto(perf) y por el modelo bien fundado (bf).

32 Para cualquier programa estratificado cabe encontrar una estratificación (partición) de los símbolos de relación que aparecen en él. Ver (prz88).

33 Sobre este resultado y el anterior, ver, por ejemplo, (prz90).

34 Motivaciones de este orden llevan a Gelfond y Lifschitz, en (geli88a), a introducir una fase de compilación.

35 Utilizamos el plural ya que consideraciones similares son posibles, por lo menos, con respecto a la lógica autoepistémica.

BIBLIOGRAFIA

(abw88) Apt, K.R., Blair, H. and A. Walker: 1988, 'Towards a theory of declarative knowledge', in Minker, J. (ed.): *Foundations of deductive databases and logic programming*, Morgan Kaufmann, 89-148.

(bdg88) Balcázar, J.L., Díaz, J. and J. Gabarró: 1988, *Structural complexity I*, Springer-Verlag.

(bes89) Besnard, P.: 1989, *Introduction to default logic*, Springer-Verlag.

(bre91) Brewka, G.: 1991, *Nonmonotonic reasoning: logical foundations of common sense*, Cambridge University Press.

(eth88) Etherington, D.W.: 1988, *Reasoning with incomplete information*, Pitman.

(fit85) Fitting, M.: 1985, 'A Kripke-Kleene semantics for logic programs', *Journal of Logic Programming* 2, 295-312.

(geli88a) Gelfond, M. and V. Lifschitz: 1988, 'Compiling circumscriptive theories into logic programs', *Proceedings second workshop on nonmonotonic reasoning*, Springer Verlag,.

(geli88b) Gelfond, M. and V. Lifschitz: 1988, 'The stable model semantics for Logic Programming', *Proceedings 5th international Conference and Symposium on Logic Programming*, 1070-1080,.

(got92) Gottlob, G.: 1992, 'Complexity results for Nonmonotonic Logics', *Journal of Logic and Computation* 2(3), 397-427.

(gpp89) Gelfond, M., Przymusinska, H. and T. Przymusinski: 1989, 'On the relationship between circunmscription and negation as failure', *Artificial Intelligence* 38, 75-94.

(gpp90) Gelfond, M., Przymusinska, H. and T. Przymusinski: 1990, 'On the relationship between CWA, Minimal model and minimal Herbrand model semantics', *International journal of intelligent systems* 5, 549-564.

(kon87) Konolige, K.: 1987, *On the relation between default and autoepistemic logic. Technical note 407*, SRI International.

(kopa88) Kolaitis, P.G. and C.H. Papadimitriou: 1988, 'Some computational aspects of circumscription', *Proceedings of AAAI88*, Minessota, 465-470,.

(kun87) Kunen, K.: 1987, 'Negation in logic programming', *Journal of Logic Programming* 4, 289-308.

(lave92) Laenens, E. and D. Vermeir: 1992, 'Assumption-free semantics for ordered logic programs: on the relationship between well-founded and stable partial models', *Journal of logic and computation* 2(2), 133-172.

(lev88) Levesque, H.J.: 1988, 'Logic and the complexity of reasoning', *Journal of philosophical logic* 17, 355-389.

(lif85) Lifschitz, V.: 1985, 'Computing circumscription', *Proceedings IJCAI-85*, Los Angeles, 121-127.

(luk90) Lukaszewicz, W.: 1990, *Non-monotonic reasoning. Formalization of commonsense reasoning*, Ellis Horwood,.

(llo87) Lloyd, J.W.: 1987, *Foundations of Logic Programming*, Springer Verlag (second edition).

(mcc80) McCarthy, J.: 1980, 'Circumscription-a form of non-monotonic reasoning', *Artificial Intelligence* 13, 27-39.

(mcc86) McCarthy, J.: 1986, 'Applications of circumscription to formalizing commonsense knowledge', *Artificial Intelligence* 28, 89-116.

(mcc88) McCarthy, J.: 1988, *Mathematical logic in AI*, DAEDALUS, Winter.

(mcdo80) McDermott, D. and J. Doyle: 1980, 'Non-monotonic logic I', *Artificial Intelligence* 13 (1-2), 41-72.

(min75) Minski, M.: 1975, 'A framework for representing knowledge', in Winston, P. (ed.): *The Psychology of Computer Vision*, McGraw-Hill, 211-277.

(moo83) Moore, R.C.: 1983, *Semantic considerations on nonmonotonic logic. Technical note 284*, SRI International.

(nil91) Nilsson, N.J.: 1991, 'Logic and Artificial Intelligence', *Artificial Intelligence* 47, 31-56.

(per93) Pérez Miranda, L.A.: 1993, *Fundamentos Lógico-Conceptuales para una Teoría general de la Racionalidad: Razonamiento Teórico y Práctico*, Tesis Doctoral, Universidad del País Vasco, 1993.

(prz88) Przymusinski, T.: 1988, 'On the declarative semantics of stratified deductive databases and logic programs', in Minker, J. (ed.): *Foundations of deductive databases and logic programming*, Morgan Kaufmann, 193-216.

(prz89) Przymusinski, T.: 1989, 'On the declarative and procedural semantics of logic programs', *Journal of Automated Reasoning* 5, 167-205.

(prz90) Przymusinski, T.: 1990, 'Non-monotonic reasoning versus logic programming: a new perspective', in Partridge, D., Y. Wilks (eds.): *The foundations of artificial intelligence. A sourcebook*, Cambridge University Press, 49-71.

(prz91) Przymusinski, T.: 1991, 'Three-valued nonmonotonic formalisms and semantics of logic programs', *Artificial Intelligence* 49, 309-343.

(rab89) Rabinov, A.: 1989, 'A generalization of collapsible cases of circumscription', *Artificial Intelligence* 38, 111-117.

(ros89) Ross, K.A.: 1989, 'The well founded semantics for disjunctive logic programs', *Proceedings DOOD' 89.*

(ros92) Ross, K.A.: 1992, 'A procedural semantics for well-founded negation in logic programs', *Journal of Logic Programming* 13, 1-22.

(rsv88) Ross, K.A., Schlipf, J.S. and A. Van Gelder: 1988, 'Unfounded sets and well-founded semantics for general logic programs', *ACM symposium on principles of database systems*, 221-230.

(she88) Shepherdson, J.C.: 1988, 'Negation in logic programming', in Minker, J. (ed.): *Foundations of deductive databases and logic programming*, Morgan Kaufmann, 19-88.

(sho88) Shoham, Y.: 1988, *Reasoning about change*, MIT Press.