Modelos formales de agentes intencionales (1)

Dr. Alejandro Guerra-Hernández
Departamento de Inteligencia Artificial
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No 5, Xalapa, Ver., México 91000
aguerra@uv.mx
www.uv.mx/aguerra

27 mayo, 2006

Los métodos formales son usados en las Ciencias de la Computación para verificar que un programa sea correcto y funcione conforme a nuestras expectativas. Por ejemplo, la verificación formal de software consiste en demostrar que un algoritmo cumple una postcondición, expresada el algún lenguaje lógico como el cálculo de predicados o la lógica modal, suponiento que se ha cumplido cierta precondición. Dentro de esta tradición, encontramos dos enfoques, con propósitos ligeramente distintos[1], para especificar formalmente agentes racionales:

Lenguajes orientados a agentes. En este enfoque, el lenguaje empleado para especificar formalmente a los agentes y su comportamiento, es a su vez, un lenguaje ejecutable por el agente. Por ejemplo: Agent0 [19, 20], Golog and Congolog [12, 3], y AgentSpeak(L) [16].

Lenguajes externos a los agentes. En este enfoque, los métodos formales conducen a lenguajes no ejecutables por los agentes, metalenguajes usados para especificar el diseño de los agentes y para verificar que ciertas propiedades del comportamiento de los agentes se cumplan. Por ejemplo, las diferentes lógicas BDI [2, 17, 15, 18, 7, 21], incluyendo Lora [23]; o las diferentes especificaciones de agentes en lenguaje Z [4, 14].

Lo ideal sería contar con formalismos que atendieran ambos propósitos, pero el compromiso entre expresividad y computabilidad, hacen que esto sea en extremo difícil [22]. Debido a las propiedades que esperamos de un agente, en partícular reactividad, las arquitecuturas de agente deben usar lenguajes internos que sean computacionalmente eficientes; pero en contraparte, otras propiedades de los agentes como la iniciativa y la habilidad social, requieren de lenguajes muy expresivos para obtener

[1] Para una discusión más detallada al respecto y la consideración de algunos casos de estudio, vean el capítulo 9 de Michael Wooldrige [23] en "Reasoning about Agents".

sus especificaciones formales. Este es el origen del compromiso entre expresividad y computabilidad.

En estas sesión nos concentraremos en el uso de las lógicas BDI como metalenguajes externos para axiomatizar las propiedades esperadas de un agente, en partícular, la relación entre las diferentes actitudes proposicionales que conforman el estado mental de un agente. Las lógicas BDI incluyen un componente de primer orden [6] para tratar con los objetos en el dominio de discurso y sus propiedades; un componente temporal [5] basado generalmente en las lógicas temporales ramificadas CTL y $CTL*$; un componente de acción, inspirado en la lógica dinámica [8]; y un componente modal [10] para tratar con las creencias, deseos e intenciones. Tal aparato es lo suficientemente expresivo, como para definir formalmente los conceptos de la teoría de razonamientro práctico de Bratman [1] y, desgraciadamente, lo suficientemente complejo como para que un agente razone directamente en él. Se recomienda la revisión de las técnicas conocidas como de refinamiento, por ejemplo, la especificación en lenguaje Z [13] que Mark d'Inverno et al. [4] hacen de dMARS.

El orden de exposición es el siguiente: Comenzaremos por revisar algunos conceptos básicos sobre sistemas formales y los ejemplificaremos con el cálculo proposicional. Haremos un recordatorio rápido al cálculo de predicados y la lógica modal. Con esta base, se abordan las lógicas temporales líneales y arboriscentes. Finalmente, presentaremos diferentes sistemas formales BDI.

1. Conceptos básicos

Las aproximaciones formales en Inteligencia Artificial al concepto de agente racional, hacen uso extenso de las lógicas proposicional y de primer orden [6], modal [10], y temporal [5]. Para introducir estos sistemas, tres aspectos serán considerados:

Languaje. El lenguaje de un sistema formal está dado por un conjunto de símbolos conocido como alfabeto y una serie de reglas de construcción. Una *expresión* es cualquier secuencia de símbolos primarios. Cualquier expresión es, o no es, una fórmula bien formada (fbf). Las fórmulas bien formadas son las expresiones que pueden formarse con los símbolos a partir de las reglas de construcción.

Teoría de la prueba. La teoría de la prueba tiene como objetivo hacer de cada enunciado matemático una formula mostrable y rigurosamente deducible. Para ello, la actividad matemática debería quedar reducida a la manipulación de símbolos y suceciones de símbolos regulada por un conjunto de instrucciones dados al respecto. La construcción de tal teoría implica, además del lenguaje del sistema formal, un conjunto selecto de fbf que tendrán el papel *axiomas* en el sistema, y un conjunto de *reglas de inferencia* que regulan diversas operaciones sobre los axiomas. Las fbf obtenidas mediante la aplicación sucesiva de las reglas de inferencia a partir de los axiomas (o de teoremas obtenidos con anterioridad), se conocen como *teoremas* del sistema.

Teoría del modelo. La teoría del modelo establece la interpretación de las fbf en un sistema formal. Su función es relacionar las fbf con alguna representación sim-

plificada de la realidad que nos interesa y hace que la fbf sea falsa o verdadera. Esta versión de realidad corresponde a lo que informalmente llamamos "modelo". Sin embargo, en lógica el significado de "modelo" está intimamente relacionado con el lenguaje del sistema formal: si la interpretación M hace que la expresión α sea verdadera, se dice que que M es un *modelo* de α o que M safisface α, y se escribe $M \models \alpha$. Como veremos, los modelos pueden de gran talla y estructurados, por lo que en algunas ocasiones es necesario establecer explícitamente que componente del modelo establece el valor de verdad de una fbf, para ello se usan *índices*, por ejemplo, un momento en el tiempo, un curso posible de acciones, etc. Para un modelo M, un índice i y una fbf α, esto se escribe $M \models_i \alpha$. Una fbf es *válida* en un modelo M, si y sólo si, el modelo satisface a la fbf en todos sus índices, y esto se escribe $M \vdash \alpha$. Algunas veces, si el sistema formal en el que se trabaja necesita ser mencionado explícitamente, los símbolos $\models_{S,i}$ y $\vdash_S$ son usados, donde S es el sistema formal referido. El símbolo α se usa aquí como una variable *metalógica*, es decir una variable que tiene como referente el lenguaje del sistema formal mismo, y por lo tanto, no forma parte del lenguaje del sistema en si. Se usaran letras griegas como variables metalógicas.

Los elementos del lenguaje y la teoría de la prueba, constituyen la sintáxis del sistema formal y su *base axiomática*. La teoría del modelo está relacionada con la semántica del sistema formal. Sintáxis y Semántica serán representados por reglas sintácticas (Sint) y semánticas (Sem) para cada una de las lógicas que presentaremos.

2. Cálculo proposicional (CalcProp)

El *Cálculo Proposicional* es una lógica utilizada para representar información efectiva. Las fórmulas de este lenguaje se contruyen a partir de *proposiciones* atómicas que expresan hechos acerca del mundo, i.e. información efectiva; y algunos *operadores* sobre ellas, por ejemplo: disyunción ($\vee$), conjunción ($\wedge$), negación ($\neg$), implicación material ($\Rightarrow$), etc. En lo que sigue, la sintáxis y la semántica del cálculo proposicional son detallados:

2.1. Sintáxis del cálculo proposicional

Los símbolos primarios del cálculo proposicional incluyen un conjunto numerable de variables proposicionales, algunos operadores sobre la proposiciones y paréntesis para evitar ambigüidad en las expresiones:

Conjunto de variables proposicionales: Var;

Operadores unarios: $\neg$ (negación);

Operadores binarios: $\vee$ (disyunción[2]);

[2]También es posible considerar la conjunción como símbolo primario y definir la disyunción en términos de ésta, lo importante es observar que cualquier fbf puede escribirse como una sucesión finita de símbolos primarios.

Parentesis: (,).

El lenguaje del cálculo proposicional $\mathcal{L}_{PC}$, se compone a partir de las siguientes reglas sintácticas:

CalcProp Sint 1 *Si* $\alpha \in Var$ *Entonces* $\alpha \in \mathcal{L}_{PC}$.

CalcProp Sint 2 *Si* $\alpha \in \mathcal{L}_{PC}$, *Entonces* $\neg\alpha \in \mathcal{L}_{PC}$.

CalcProp Sint 3 *Si* $\alpha, \beta \in \mathcal{L}_{PC}$ *Entonces* $(\alpha \vee \beta) \in \mathcal{L}_{PC}$.

Los paréntesis en la regla sintáctica 3 pueden eliminarse si no existe ambigüidad en la fbf. Como se mencionó, las letras griegas se usan como variables metalógicas. Otros operadores pueden definirse como de costumbre:

Def 1 (conjunción) $(\alpha \wedge \beta) =_{def} \neg(\neg\alpha \vee \neg\beta)$;

Def 2 (implicación material) $(\alpha \Rightarrow \beta) =_{def} (\neg\alpha \vee \beta)$;

Def 3 (equivalencia material) $(\alpha \equiv \beta) =_{def} ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$;

Def 4 (falso) $f =_{def} \neg\alpha \wedge \alpha$;

Def 5 (verdadero) $t =_{def} \neg f$

2.2. Semántica del cálculo proposicional

Los elementos de *Var* son interpretados como variables cuyo contenido son proposiciones, por lo que se conocen como *varibles proposicionales*. Las proposiciones tienen asociado un *valor de verdad* (*verdadero* o $falso$), de forma que toda proposición es verdadera o falsa, pero ninguna puede tomar ambos valores a la vez. Por lo tanto, una expresión que no es falsa, ni verdadera, o puede ser verdadera y falsa, no es una proposición. Como ejemplo, consideren nuevamente el agente inspirado en `xbiff` introducido en las sesiones anteriores: la variable proposicional *newMail* se uso para describir el hecho (información efectiva) de que el buzón electrónico tiene mensajes nuevos. En cualquier momento, esta variable puede ser falsa o verdadera, valor de verdad obtenido a través de la percepción del agente.

Los operadores pueden usarse para formar proposiciones compuestas. Aplicando el operador de negación a la proposición *newMail* produce una nueva proposición que describe el hecho de que el buzón está vacío: $\neg$ *newMail*. La negación, la disyunción y la conjunción, cuando se aplican de esta manera, son conocidos como *conectivos* y las proposiciones sobre las cuales operan se conocen como sus *argumentos*. Los operadores que sólo requieren un argumento se conocen como *unarios*, ej. la negación. Los operadores que requieren dos argumentos se conocen como *binarios*, ej. disyunción y conjunción.

El valor de verdad de la expresión $\neg$ *newMail* depende del valor de verdad de la proposición *newMail*. Esto aplica a todas las proposiciones compuestas creadas con

conectivos: sus valores de verdad dependen de sus argumentos. Por ello, se dice que los conectivos son *funcionales* con respecto al valor de verdad (truth-functionals).

Los operadores $\neg$, $\vee$ y $\wedge$ se interpretan respectivamente como "no es el caso que", "o", e "y". La implicación material ($\Rightarrow$) tiene dos argumentos: *antecedente* y *consecuente*. Se interpreta como "Si *antecedente* Entonces *consecuente*". Por ejemplo, si la proposición *userOnLine* se usa para representar el hecho de que el usuario está en línea, entonces la expresión $newMail \wedge userOnLine$ expresa que hay nuevos mensajes en el buzón y que el usuario está en línea.

Estas equivalencias con el lenguaje natural no deben tomarse literalmente, dado que el significado en lógica y en lenguaje natural no siempre son equivalentes, ej., la implicación material es verdadera si su consecuente es falso, independientemente del valor de verdad de su antecedente, y por lo tanto, identifica relaciones que son potencialmente irrelevantes para la expresión del lenguaje natrual "si ... entonces". El significado preciso de estos operadors está dado por la semántica del cálculo proposicional:

Abusando un poco del lenguaje, sea $M_0 =_{def} \langle L \rangle$ el modelo formal para el cálculo proposicional. $L \subseteq Var$ es una interpretación o *etiquetado* (label), que identifica el conjunto de variables proposicionales que tienen valor verdadero. Luego, el significado de las fbf del cálculo proposicional es como sigue:

CalcProp Sem 1 $M_0 \models \alpha$ *Si y sólo si* $\alpha \in L$*, donde* $\alpha \in Var$

CalcProp Sem 2 $M_0 \models \neg\alpha$ *Si y sólo si* $M_0 \not\models \alpha$

CalcProp Sem 3 $M_0 \models (\alpha \vee \beta)$ *Si y sólo si* $M_0 \models \alpha$ *o* $M_0 \models \beta$

3. Cálculo de Predicados (CalcPred)

Básicamente, el *cálculo de predicados*, también conocido como lógica de primer orden, extiende el cálculo proposicional al introducir símbolos que nos permiten expresarnos acerca de los *objetos* en un dominio de discurso dado. El conjunto de todos estos objetos se conoce como *universo de discurso*. Los miembros del universo de discurso pueden ser objetos concretos, ej., un libro, un robot, etc; abstractos, ej., números; e incluso, ficticios, ej., unicornios, etc. Un objeto es algo sobre lo cual queremos expresarnos. Para ejemplificar esto, consideren el multi citado mundo de los bloques [6] que se muestra en la figura 1. El universo de discurso para esta conceptualización de tal escenario es el conjunto que incluye los cinco bloques, la mano del robot y la mesa: $\{A, B, C, D, E, RobotHand, Table\}$.

Una *función* es un tipo de relación entre los objetos del dominio de discurso. Por ejemplo, es posible definir la función parcial *Sombrero* que mapea un bloque al bloque que se encuentra encima de él, si tal bloque existe. Las parejas correspondientes a esta función parcial son: $\{(B, A), (C, D), (D, E)\}$. El conjunto de todas las funciones consideradas en la conceptualización del mundo se conoce como *base funcional*.

Un segundo tipo de relación entre los objetos del dominio de discurso son los *predicados*. Diferentes *predicados* pueden definirse en el mundo de los bloques, ej., el predicado *Sobre* que se cumple para dos bloques, si y sólo si uno está inmediatamente encima del otro. Para la escena mostrada en la figura 1, *Sobre* se define por los pares

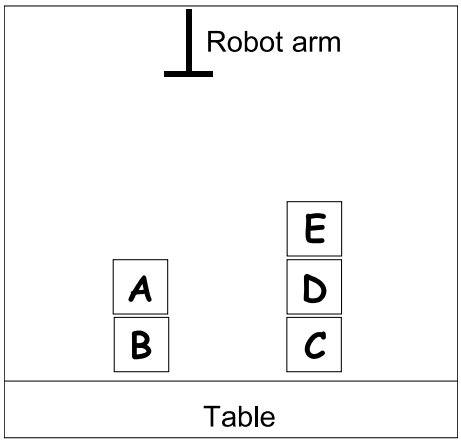

Figura 1: *El mundo de los bloques, un ambiente para ejemplificar el cálculo de predicados.*

$\{(A, B), (D, C), (E, D)\}$. Otro predicado puede ser *Libre*, que se cumple para un bloque si y sólo si éste no tiene ningún bloque encima. Este predicado tiene los siguientes elementos $\{A, E\}$. El conjunto de todos los predicados usados en la conceptualización se conoce como *base relacional*.

Para universos de discurso finitos, existe un límite superior en el número posible de predicados n-arios. Para un universo de discurso de cardinalidad b, existen b^n distintas n-tuplas. Cualquier predicado n-ario es un subconjunto de estas b^n tuplas. Por lo tanto, un predicado n-ario debe corresponder a uno de máximo $2^{(b^n)}$ conjuntos posibles.

Además de las funciones y predicados, la flexibilidad el cálculo de predicados viene del uso de cuantificadores y variables. Las variables cuyos valores son objetos del universo de discurso se suelen representar por cualquier secuencia de caracteres que inicie con una minúscula. El cuantificador "para todo" ($\forall$) nos permite expresar hechos acerca de todos los objetos en el universo del discurso, sin necesidad de enumerarlos. Por ejemplo, $\forall x \forall y\ Sobre(x, y) \Rightarrow PorArriba(x, y)$ expresa que cualquier objeto sobre otro, está por encima. El cuantificador "existe" ($\exists$) nos permite expresar la existencia de un objeto en el universo de discurso con cierta propiedad en partícular, por ejemplo, $\exists x\ Libre(x) \wedge EnLaMesa(x)$ expresa que hay al menos un objeto que no tiene bloques sobre él y aue se encuentra sobre la mesa. A continuación definiremos la sintáxis y la semántica del cálcuo de predicados.

3.1. Sintáxis del CalcPred

Los símbolos primarios del cálculo de predicados se obtienen al extender el cálculo proposiucional con un conjunto numerable de símbolos de predicado y constantes. El uso de funciones no será descrito aquí, para hacer está presentación más clara. Se asume que los miembros del conjunto *Var* toman valores en el universo de discurso. Asociado a cada predicado hay un número natural conocido como su *aridad*, que expresa en

número de argumentos del predicado. Los predicados de aridad 0 se asumen como variables proposicionales. Los operadores se definen como en el cálculo proposicional. Los símbolos primarios del cálculo de predicados son:

Conjunto de predicados: $Pred$;

Conjunto de constantes: $Const$;

Conjunto de variables: Var;

Operadores unarios: $\neg$ (negación);

Operadores binarios: $\vee$ (disyunción);

Cuantificadores: $\forall$ (cuantificador universal);

Paréntesis: (,).

El lenguaje del cálculo de predicados $\mathcal{L}_{FOL}$ se especifica como sigue. La reglas sintácticas 1 y 2 expresan que las constantes y variables son fbf en el lenguaje. El conjunto $Term$ se define como $Var \cup Const$:

CalcPred Sint 1 *Si* $\alpha \in Const$ *Entonces* $\alpha \in \mathcal{L}_{FOL}$

CalcPred Sint 2 $\alpha \in Var$ *Entonces* $\alpha \in \mathcal{L}_{FOL}$

La sintáxis de la conjunción y la disyunción se define como en el cálculo proposicional:

CalcPred Sint 3 *Si* $\alpha \in \mathcal{L}_{FOL}$ *Entonces* $\neg\alpha \in \mathcal{L}_{FOL}$

CalcPred Sint 4 *Si* $\alpha \in \mathcal{L}_{FOL}$ *and* $\beta \in \mathcal{L}_{FOL}$ *Entonces* $(\alpha \vee \beta) \in \mathcal{L}_{FOL}$

La sintáxis de los predicados ímplica que estos se aplican al número correcto de argumentos, especificado por la aridad, y que los argumentos son términos:

CalcPred Sint 5 *Si* $\alpha \in Pred$ *es un predicado de aridad n, Entonces* $\alpha(t_1, \ldots, t_n)$ *está en* $\mathcal{L}_{FOL}$ *Si y sólo si para todo* $1 <= i <= n, t_i \in Term$.

La sintáxis del cuantificador universal es como sigue:

CalcPred Sint 6 *Si* $\alpha \in \mathcal{L}_{FOL}$ *y* $x \in Vars$ *es una variable que ocurre en* α*, Entonces* $\forall x\ \alpha \in \mathcal{L}_{FOL}$

Las definiciones de la conjunción, la implicación material, la equivalencia material, verdadero y falso, son como en el cálculo proposiciona (p. 4). La definición del cuantificador existencial es la siguiente:

Def 6 (cuantificador existencial) $\exists x\ \alpha =_{def} \neg(\forall x\ \neg\alpha)$

Siendo estrictos, el cuantiicador propiamente dicho, es el símbolo de cuantificador seguido de una variable, puesto que $\forall x$ y $\forall y$ tienen significados diferentes. En una fbf de la forma $\forall x\ \alpha$, se dice que la fbf α esta en el alcance del cuantificador $\forall x$. En tal caso, se dice aue la ocurrencia de x en α está *acotada*, en caso contrario se dice que la ocurrencia de la variable es *libre*. Por ejemplo, en $\exists x\ Sobre(x, y)$ la variable x está acotada, mientras y está libre.

3.2. Semántica del CalcPred

Anted de introducir las definiciones formales de la semántica del cálculo de predicados, consideremos algunas expresiones posibles en está lógica, usando como ambiente el mundo de los bloques (Figura 1). Si queremos expresar que al menos algún bloque no tiene nada encima, podemos usar los predicados *Bloque* y *Libre* en la siguiente expresión: $\exists x\ Block(x) \wedge Libre(x)$. Esta fbf expresa que existe un x tal que x es un bloque y x está libre (no tiene otro bloque encima). Observen que cuando usamos cuantificadores, siempre tenemos en mente el universo de discurso o *dominio*. El dominio puede expecificarse en término de conjuntos. Luego, si el dominio D es el conjunto de constantes $\{A, B, C, D, E, Table, RobotHand\}$, podemos decir que $B = \{A, B, C, D, E\}$ es el conjunto de bloques en D. Entonces plantear una expresión equivalente a $\exists x\ Block(x) \wedge Libre(x)$, usando la fbf $\forall x\ Libre(x)$, si específicamos que *Libre* tiene como dominio B. En tal caso, se dice que B es la *interpretación* del predicado *Libre*. Para un predicado de aridad dos, como $Sobre$ cuyos argumentos son bloques, podemos decir que su interpretación está en $B \times B$.

Para obtener un modelo para el lenguaje $\mathcal{L}_{FOL}$ formamos el par $\langle D, V\rangle$, donde D es el universo de discurso, ej. cualquier clase de objetos sobre la que queremos expresarnos, y V es una función, tal que para cualquier predicado de aridad n se obtienen las n-tuplas que corresponden a la interpretación del predicado. En el ejemplo de la figura 1, consideren el predicado $Sobre$ de aridad dos. Su interpetación es un subconjunto de $B \times B$. Para la escena mostrada, $V(Sobre) = \{(A, B), (E, D), (D, C)\}$. Para una constante, la función V regresa la misma constante, ej. $V(A) = A$. Algunas veces la expresión $V(\alpha)$, donde $\alpha \in Const \cup Pred$, se abrevia α^V y se conoce como una *interpetación*. Una posible interpretación V para la escena mostrada en al figura 1 y su base relacional es:

$$
\begin{aligned}
A^V &= A \\
B^V &= B \\
C^V &= C \\
D^V &= D \\
E^V &= E \\
Sobre^V &= \{(A, B), (E, D), (D, C)\} \\
EnLaMesa^V &= \{B, C\} \\
Libre^V &= \{A, E\} \\
PorEncima^V &= \{(A, B), (E, D), (E, C), (D, C)\}
\end{aligned}
$$

Todo esto puede especificarse formalmente con la siguiente regla semántica:

CalcPred Sem 1 *Una intepretación V,con rspecto a un dominio de discurso D es una función que safisface las siguientes propiedades: i) Si $\alpha \in Const$, Entonces $V(\alpha) = \alpha$; ii) Si $\alpha \in Pred$ es de aridad n, Entonces $V(\alpha) \subseteq D^n$.*

Observen que las variables no están incluidas en la interpretación, tal y como se definión anteriormente. Interpretar las variables de manera serapara a otros símbolos,

es una práctica aceptada. En lo que sigue, se asume que las variables en las fbf están acotadas, es decir, bajo el alcance de un cuantificador. Decimos que μ es una *asignación de variables* basada en el modelo $M_1 = \langle D, V \rangle$ si para todo $\alpha \in Var$, $\mu(\alpha) \in D$. Entonces podemos escribir $M_1 \models V_\mu(\alpha)$ para expresar que α es verdadera en el modelo $\langle D, V \rangle$ cuando las variables en α toman valores de acuerdo a la asignación μ. Luego, para las fbf atómicas tenemos que:

CalcPred Sem 2 $M_1 \models \alpha$ *Si y sólo si* $V_\mu(\alpha) \in V$.

CalcPred Sem 3 $M_1 \models \neg\alpha$ *Si y sólo si* $M_1 \not\models \alpha$.

CalcPred Sem 4 $M_1 \models (\alpha \vee \beta)$ *Si y sólo si,* $M_1 \models \alpha$ *o* $M_1 \models \beta$.

Para formular las reglas semánticas de las expresiones que contienen cuantificadores, necesitamos el concepto de asignación de variables alternativa. La idea es que queremos que $V_\mu(\forall x\alpha)$ sea verdadera, no solo cuando $V_\mu(\alpha)$ es verdadera, sino cuando $V_\rho(\alpha)$ es verdadera para cualquier valor en D regresado por cualquier función de asignación ρ. Observen que las variables libres en α deben conservar el mismo valor en tal asignación. Se dice que la asignación de variables ρ es *x-alternativa* de μ, si y sólo si para toda variable α, excepto posiblemente x, $\rho(\alpha) = \mu(\alpha)$. Entonces:

CalcPred Sem 5 $M_1 \models \forall x\ \alpha$ *Si y sólo si* $V_\rho(\alpha) \in V$ *para cada asignación de variable x-alternativa* ρ *de* μ.

CalcPred Sem 6 $M_1 \models \exists x\ \alpha$ *Si y sólo si existe una asignación de variables x-alternativa* ρ *de* μ*, tal que* $V_\rho(\alpha)$ *es verdadera.*

Una fbf α es *válida* en el model M_1 si y sólo si $M_1 \models \alpha$ para toda asignación de variable μ basada en $\langle D, V \rangle$. Una fbf aue es válida en todo modelo se conoce como *universalmente válida*.

Un resultado estándar importante es el *principio de remplazamiento*, expresado de la siguiente forma: Sea α cualquier fbf y x e y dos variables cualesquiera. Sea M_1 cualquier modelo de $\mathcal{L}_{FOL}$. Sea μ cualquier asignación de variables. Entonces cuando ρ es casi igual a μ excepto que $\rho(x) = \mu(y)$, $V_\rho(\alpha) = V_\mu(\alpha[y/x])$. Otro resultado es el principio de *concordancia* que expresa que cuando μ y ρ concuerdan en todas las variables libres de una fbf α, enconces $V_\rho(\alpha) = V_\mu(\alpha)$. Como consecuencia, cuando α no contiene variables libres, entonces $V_\rho(\alpha) = V_\mu(\alpha)$ para cualquier ρ y μ.

4. Lógica Modal Proposicional (Modal)

La *lógica modal proposicional* trabaja con conceptos cuyo valor de verdad no es funcional. Estos conceptos generalmente expresan un "modo" o "manera" en la cual una proposición puede ser verdadera. Uno de estos modos es la expresión necesariamente. Podemos agregar un operador unario al cálculo proposicional, llamado L con la siguiente regla sintáctica: Si α es una fbf, también lo es $L\alpha$. Este opedor se conoce como *operador de necesidad* y es usado para expresar conceptos como necesariamente, debe ser, está sujeto a, etc. Las interpretaciones pueden variar de entre "debe ser" a

“moralmente debería ser el caso que” o “siempre es el caso que” o “es sabido que”. Lo común entre estas expresiones es que todas ellas son no funcionales con respecto a su valor de verdad, es decir, el valor funcional de la proposición p no es suficiente para determinar el valor de verdad de Lp.

Un operador modal para expresar “posibilidad”, “puede ser” o “quizás” puede definirse como $M =_{def} \neg L \neg \alpha$. Para toda interpretación de L habrá una de M. En algunos textos L se representa por $\Box$, y M por $\Diamond$. La imposibilidad se clasifica tambien como un concepto modal que puede ser expresado por $\neg M$ o $L \neg \alpha$. Las proposiciones aue no son necesarias o imposibles se conocen como *contingentes*.

4.1. Sintáxis de la lógica modal proposicional

El lenguaje de la lógica modal proposicional incluye los siguientes símbolos primarios:

Variables proposicionales: Var;

Operadores unarios: $\neg$, L;

Operadores binarios: $\vee$;

Paréntesis: (,).

Las reglas de formación son las siguientes:

Modal Sint 1 *Si $\alpha \in Var$ Entonces α es una fbf;*

Modal Sint 2 *Si α es una fbf, entonces $\neg\alpha$ and $L\alpha$ son fbf;*

Modal Sint 3 *Si α y β son fbf, también lo es* $(\alpha \vee \beta)$.

La conjunción ($\wedge$), la implicación material ($\Rightarrow$), y la equivalencia ($\equiv$) se definen como en el cálculo proposicional (p. 4). El operador de *posibilidad* se define como sigue:

Def 7 (posiblemente) $M =_{def} \neg L \neg \alpha$.

4.2. Semántica de la lógica modal proposicional

La teoría de la pruebla de lógica modal está basada en las siguientes tres ideas: i) Determinar el valor de verdad de una expresión modal implica tomar en consideración estados alternativos a los que realmente se han dado (Como podrían haber sido las cosas); ii) Para cada estado posible existe un conjunto de estados posibles relativos al primero; iii) En un estado dado $M\alpha$ es verdadero si y sólo si α mismo es verdadero en al menos uno de los estados posibles relativos al estado dado, y $L\alpha$ es verdadero si y sólo si α mismo es verdadero en todos los estados posibles relativos al estado dado.

Los estados posibles se conocen generalmente como *mundos posibles*. Se asume un conjunto no vacío W de ellos. Una relación R de *accesibilidad* entre estos mundos,

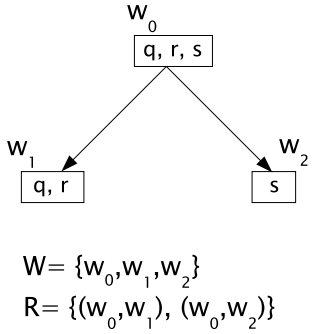

Figura 2: *Un marco* $\langle W, R\rangle$.

también es asumida. Se dicde que w' es accesible desde w si y sólo si $(w, w') \in R$. La estructura $\langle W, R\rangle$ se llama *marco*. La figura 2 muestra gráficamente un marco donde $W = \{w_0, w_1, w_2\}$ y $R = \{(w_0, w_1), (w_0, w_2\}$. Las letras en cada mundo posible w_i son las proposiciones que son verdaderas en ese mundo posible.

Una función de asignación de verdad V se define como sigue: $V(\alpha, w) = true$ si y sólo si V asigna verdadero a la proposición α en el mundo w. $V(\alpha, w) = falso$ Si y sólo si V asigna falso a la proposición α en el mundo w. Observen que V corresponde al etiquetado o interpretación L en el cálculo proposicional, pero aquí el valor de verdad de una proposición puede cambiar entre los mundos posibles. En el marco que se muestra en la figura 2, $V(q, w_0) = verdadero$, pero $V(q, w_1) = falso$. La función de asignación de verdad puede definirse también como: $V : W \to 2^{Var}$ denotando el conjunto de variables proposicionales que son verdad en un mundo. En el caso del ejemplo, $Var = \{q, r, s\}$, y $V(w_0) = \{q, r, s\}$, pero $V(w_1) = \{q, r\}$, y $V(w_2) = \{s\}$. U *modelo* es una tripleta $\langle W, R, V\rangle$ donde $\langle W, R\rangle$ es un marco, y V es una función de verdad como se definió anteriormente. Tal modelo se conoce como estructura de Kripke , por Saul A. Kripke [11][3].

Sea $M_2 = \langle W, R, V\rangle$ el modelo para la lógica modal proposicional. La semántica de los operadores puede definirse de la siguiente manera:

Modal Sem 1 $M_2 \models_w \alpha$ *Si y sólo si* $\alpha \in V(w)$ *y* $\alpha \in Var$*;*

Modal Sem 2 $M_2 \models_w \neg\alpha$ *Si y sólo si* $M_2 \not\models_w \alpha$*;*

Modal Sem 3 $M_2 \models_w (\alpha \vee \beta)$ *Si y sólo si* $M_2 \models_w \alpha$ *o* $M_2 \models_w \beta$*;*

Modal Sem 4 $M_2 \models_w L\alpha$ *Si y sólo si, para todo* w'*, tal que* $(w, w') \in R$*,* $M_2 \models_{w'} \alpha$*;*

Modal Sem 5 $M_2 \models_w M\alpha$ *Si y sólo si, para algún* w' *s.t.* $(w, w') \in R$*,* $M_2 \models_{w'} \alpha$*;*

Una consecuencia de estas definiciones es que habrá tantos tipos diferentes de validación, como existen arreglos diferentes de mundos posibles, representados por la

[3]La semantica de los mundos posibles fuer originalmetne formulada por Jacob Hintikka [9] quien llamó a los mundos *alternativas epistḿicas*, pero el desarrollo de los sistemas modales normales se debe a Kripke.

relación de accesibilidad. Estos diferentes tipos de validez, corresponden a diferentes significados esperados para los operadores modales.

Una fbf es válida en el marco $\langle W, R\rangle$ si y sólo si, para cada modelo $\langle W, R, V\rangle$ basado en este marco, y para cada $w \in W$, $\alpha \in V(w)$. Existen algunas fbf que son válidas en cualquier arreglo de mundos posibles, y se conocen como *K-válidas*. Todas las fbf válidas de la lógica proposicional, son K-válidas. En especial, una fbf modal que es K-válida es la siguiente: $L(p \Rightarrow q) \Rightarrow (Lp \Rightarrow Lq)$.

Para mostrar la relación correcta entre el sistema lógico y la definición de validez es este sistema, debemos probar:

1. Que todo teorema en el sisteme es válido, siguiendo la definición presentada de validez.

2. Que toda fbf válida en el sistema es un teorema.

Si (1) es cierta, se dice aue el sistema es *consistente*. Si (2) es cierta, se dice que el sistema es *completo*. La completitud es más difícil de establecer.

A continuación, diferentes sistemas modales les serán presentados, pero antes es necesario introducir cierta terminología. Cuando una fórmula α es un teorema en un sistema dado S, decimos que α pertenece a S, o que α está contenido en S, o simplemente que α está en S. Si dos sistemas axiomáticos S y S' tienen diferentes bases axiomáticas, pero contienen exactamente los mismos teoremas, decimos que son *deductivamente equivalentes*, o simplemente que son equivalentes. Si cada teorema de S es también un teorema de S', sin tomar en cuenta si S' tiene o no otros teoremas, decimos que S' contiene a S. Dos sistemas son deductivamente equivalentes si cada uno contiene al otro. Si S' contiene a S y a otros teoremas, decimos que S' contiene propiamente a S, o que es una *extensión propia* de S, que S' es más fuerte que S y que S es el más débil de los dos sistemas.

4.3. El sistema K

Los teoremas de este sistema son aquellas formulas que son fbf válidas en la lógica proposicional y una sola fbf modal distintiva conocida como K[4]:

K 1 (PC-válida) *Si α es una fbf válida en el cálculo proposicional, entonces α es un axioma en K.*

K 2 (K) $L(p \Rightarrow q) \Rightarrow (Lp \Rightarrow Lq)$

Y las siguientes reglas de inferencia:

Inf R 1 (Substitución Uniforme, SU) $\vdash \alpha \rightarrow \vdash \alpha[\beta_1/p_1, \ldots \beta_n/p_n]$*;*

Inf R 2 (Modus ponens, MP) $\vdash \alpha,\ \alpha \Rightarrow \beta\ \rightarrow \vdash \beta$*;*

Inf R 3 *[Necesidad, N]* $\vdash \alpha \rightarrow \vdash L\alpha$

[4]Nuevamente, por Samuel Kripke.

La expresión $\alpha[\beta_1/p_1, \ldots \beta_n/p_n]$ denota las fbf que resulta de substituir las variables $p_1, \ldots, p_n$ en α, uniformemente por cualquier expresión $\beta_1, \ldots, \beta_n$; y $\rightarrow$ denota que la fbf a la derecha es derivable a partir de la de la izquierda.

Es plausible que la substitución uniforme (SU) es requerida en cualquier sistema lógico. El Modus Ponens (MP) refleja la verdad funcional de la implicación material. Ambos son válidos y preservan la validez. La regla de necesidad (N) es una regla modal que preserva la validez con respecto a la definición de K-válido.

Algunos teoremas en K son como sigue:

K 3 $L(p \wedge q) \Rightarrow (Lp \wedge Lq)$

K 4 $(Lp \wedge Lq) \Rightarrow L(p \wedge q)$

K 5 (ley de distribución) $L(p \wedge q) \equiv (Lp \wedge Lq)$

K 6 $\vdash \alpha \Rightarrow \beta \rightarrow \vdash L\alpha \Rightarrow L\beta$

K 7 $(Lp \vee Lq) \Rightarrow L(p \vee q)$

K 8 $\vdash \alpha \equiv \beta \rightarrow \vdash L\alpha \equiv L\beta$

K 9 $Lp \equiv \neg M \neg p$

K 10 $M(p \vee q) \equiv (Mp \vee Mq)$

Si α es un teorema y β difiere de α solo por que tiene la fbf δ en uno o más lugares, donde α tiene γ, entonces si $\gamma \equiv \delta$, γ puede ser remplazado por δ en cualquier teorema, no necesariamete de manera uniforme, el resultado será un teorema. Esto se conoce como *substitución de equivalentes*(Eq).

K es el más débil de los sistemas modales. Todos los demás sistemas serán extensiones propias de K. Todos los sistemas que incluyen propiamente a K se conocen como sistemas modales *normales* .

4.4. El sistema T

Este sistema resulta al agregar el axioma $Lp \Rightarrow p$ al sistema K. Aún cuando esta fbf no es K-válida, es aceptable para representar algunas nociones de necesidad:

T 1 (T) $Lp \Rightarrow p$

Otros teoremas de este sistema son:

T 2 $p \Rightarrow Mp$

T 3 $M(p \Rightarrow Lp)$

T es válida unicamente en marcos donde R es reflexiva. El sistema T es consistente con respecto a todos los marcos reflexivos.

4.5. El sistema D

Si queremos expresar obligación, o necesidad moral, no desearíamos contar con el axioma T, puesto que queremos expresar cualquier cosa que es obligatorio, es al menos permisible. Está iterpretación de L se conoce como *deóntica* y se formula agregando el axioma D al sistema K:

D 1 (D) $Lp \Rightarrow Mp$

Un teorema fácil de derivar en D es:

D 2 $M(p \Rightarrow p)$

Observen aue si α es un teorema en el sistema D, también lo es $M\alpha$. Si cualquier sistema que sea una extensión de K, incluye algún teorema de la forma $M\alpha$, entonces el sistema contiene D. T es una extensión propia de D y D lo es de K.

Un mundo que no puede accesar a ningún otro mundo se conoce como *punto muerto*. En puntos muertos D no es válida porque $L\alpha$ y $M\alpha$ son verdadera y falsa, respectivamente, sin tomar en cuenta α. Los marcos sin puntos muertos son aquellos donde R es serial. D es válida en todo marco serial y el sistema es consistente con respecto a la clase de todos los marcos seriales.

4.6. Los sistemas S4 y S5

Formulas como $Lp \Rightarrow LLp$ deberían al menos dejarnos perplejos. Contienen secuencias como LL, conocidas como *modalidades iteradas*, que nos llevan a preguntas como –¿Qué significa necesariamente necesario? $LLp \Rightarrow Lp$ es una substitución de T, así que si agregamos $Lp \Rightarrow LLp$ como un teorema, el nuevo sistema tendrá q $Lp \equiv LLp$ como teorema. Este tipo de teoremas se conoce como *leyes de reducción*. T no contiene ninguna regla de reducción.

El sistema S4 se forma agregando el axioma 4 a K y T:

S4 1 (4) $Lp \Rightarrow LLp$

Una *modalidad* es definida como cualquier secuencia ininterrumpida de cero o más operadores unarios. El caso de cero operadores se expresa como '–'. Si una modalidad se expresa sin ninún signo de negación, o con sólo uno, se dice que está expresada en forma *estándar*. Se dice que una modalidad está iterada si contiene dos o más operadores modales. Una modalidad es *afirmativa* si no contiene signos de negación, y *negativa* si contiene uno. Dos modalidades A y B son equivalentes si y sólo si el resultado de remplazar una por otra en cualquier formula es siempre equivalente en el sistema a la formula original.

En S4 toda modalidad es equivalente a una u otra de las siguientes modalidades o sus negaciones: i)–; ii) L; iii) M; iv) LM; v) ML; vi) LML; vii) MLM. El axioma 4 es válido con respecto a marcos transitivos. El sistema S4 es válido para todo marco reflexivo y transitivo.

El sistema S5 resulta al condiserar el siguiente axioma junto a K y T:

S5 1 (5) $Mp \Rightarrow LMp$

S5 contiene a S4 puesto que el axioma 5 no es un teorema en S4. El sistema contiene seis modalidades distintivas: i)–; ii) L; iii) M; y sus negaciones. El axima 5 es válido en todo marco que es reflexifo, transitivo y simétrico. El sistema S5 es consistente con respecto a marcos reflexivos, transitivos y simétricos, es decir cuando R es una relación de equivalencia.

Observen aue los axiomas de los sistemas modales proposicionales, corresponden a ciertas propiedades de la relación de accesibilidad R. Esto se conoce como *Teoría de correspondencia* [10]. Para resumir, el cuadro 1 muestra la teoría de correspondencia para las axiomas T,D,4, and 5.

Axioma	**Esquema**	**Condición en R**	**CalcPred**
T	$L\alpha \Rightarrow \alpha$	reflexiva	$\forall w \in W (w, w) \in R$
D	$L\alpha \Rightarrow M\alpha$	serial	$\forall w \in W\ \exists w' \in W (w, w') \in R$
4	$L\alpha \Rightarrow LL\alpha$	transitiva	$\forall w, w', w'' \in W\ (w, w') \in R \wedge (w', w'') \in R \supset (w, w'') \in R$
5	$M\alpha \Rightarrow LM\alpha$	euclideana	$\forall w, w', w'' \in W\ (w, w') \in R \wedge (w, w'') \in R \supset (w', w'') \in R$

Cuadro 1: *Teoría de correspondencia para los axiomas T,D,4, y 5.*

Es posible construir $2^4 = 16$ diferentes sistemas modales proposicionales con estos cuatro axiomas[5], pero algunos de ellos son equivalentes. La figura 3 muestra la relación de inclusión entre los diferentes sistemas modales proposicionales normales, construidos con estos cuatro axiomas. Una flecha del sistema A al sistema B en el diagrama puede interpretarse como B es un superconjunto estricto de A, es decir, todo teorema de A es un teorema de B, pero no a la inversa. $A = B$ significa que A y B contienen exactamente los mismos teoremas. Recuerden que, como estos sistemas son ampliamente utiliwados, se conocen como sigue: *KT* se conoce como *T*; *KT4* se conoce como *S4*; *KD45* se conoce como *weak-S5*; y *KT5* se conoce como *S5*.

5. Ejercicios.

1. Demuestre que la extensión de un predicado n-ario (p. 6) debe corresponder a uno de máximo $2^{(b^n)}$ conjuntos posibles.

2. Demuestre el axioma de necesidad (p. 12).

3. ¿Qué implicaciones tiene para la modelización de agentes que K esté incluído en todo sistema modal proposicional normal?

[5] K es común a todos los sistemas.

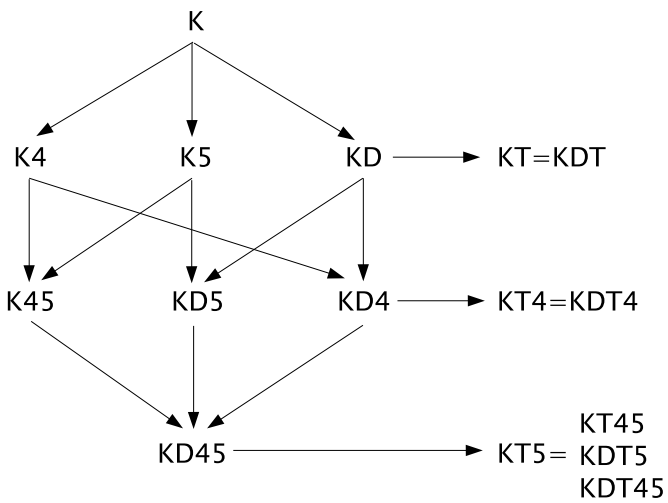

Figura 3: *Sistemas modales proposicionales normales, usando los axiomas T,D,4, and 5.*

References

[1] M.E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA., USA, and London, England, 1987.

[2] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.

[3] G. De Giacomo et al. Reasoning about concurrent execution, prioritized interrupts, and exogenous actions in situated calculus. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1221–1226, Nagoya, Japan, August 1997.

[4] M. d'Inverno et al. A formal specification of dmars. Technical Note 72, Australian Artificial Intelligence Institute, Carlton, Victoria, April 1997. Appeared in Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages, ATAL'97.

[5] A.E. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, Amsterdam, Holland, 1990.

[6] M.R. Genesereth and N.J. Nilsson. *Logical Foundations for Artificial Intelligence*. Morgan Kauffman Publishers, Inc., Palo Alto, CA., USA, 1987.

[7] A. Haddadi. *Communication and Cooperation in Agent Systems, A Pragmatic Theory*. Number 1056 in Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin-Heidelberg, Germany, 1995.

[8] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II — Extensions of Classical Logic*, pages 497–604. D. Reidel Publishing Company: Dordrecht, The Netherlands, 1984.

[9] J. Hintikka. *Knowledge and Belief.* Cornell University Press, Ithaca, N.Y., USA, 1962.

[10] G.E. Hughes and M.J. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, England, 1998 edition, 1996.

[11] S.A. Kripke. A completeness theorem in modal logic. *The Journal of Symbolic Logic*, 24:1–14, 1959.

[12] H. Levesque et al. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84, 1997.

[13] D. Lightfoot. *Formal Specification Using Z.* Macmillan Computer Science Series. The Macmillan Press LTD, London, UK, 1991.

[14] M. Luck and M. d'Inverno. A conceptual framework for agent definition and development. *The Computer Journal*, 44(1):1–20, 2001.

[15] A.S. Rao. Decision procedures for propositional belief-desire-intention logics. Technical Note 44, Australian Artificial Intelligence Institute, Carlton, Victoria, September 1993.

[16] A.S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In Rudy van Hoe, editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.

[17] A.S. Rao and M.P. Georgeff. Modelling rational agents within a bdi-architecture. Technical Note 14, Australian Artificial Intelligence Institute, Carlton, Victoria, February 1991.

[18] A.S. Rao and M.P. Georgeff. Formal models and decision procedures for multi-agent systems. Technical Note 61, Australian Artificial Intelligence Institute, Carlton, Victoria, June 1995.

[19] Y. Shoham. Agent-oriented programming. Technical Report STAN–CS–1335–90, Computer Science Department, Stanford University, Stanford, CA., USA, 1990.

[20] Y. Shoham and S.B. Cousins. Logics of mental attitudes in AI. In *ECAI Workshop on Knowledge Representation and Reasoning*, pages 296–309, 1992.

[21] M.P. Singh. *Multiagent Systems: A theoretical framework for intentions, know-how, and communication*. Number 799 in Lecture Notes in Computer Sciences. Springer Verlag, Berlin-Heidelberg, Germany, 1995.

[22] M.P. Singh, A.S. Rao, and M.P. Georgeff. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter Formal Methods in DAI: Logic-Based Representation and Reasoning, pages 331–376. MIT Press, Cambridge, MA., USA, 1999.

[23] M. Wooldridge. *Reasoning about Rational Agents*. MIT Press, Cambridge, MA., USA, 2000.