# Capítulo 3

# Lógica modal

*Clara Smith*

Supported by EU H2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 690974 for the project MIREL: MIning and REasoning with Legal texts.

## Conceptos básicos

La lógica modal fue en sus orígenes (atribuidos a Aristóteles por la mayoría de los estudiosos) la lógica de *lo necesario* y *lo posible*. Más modernamente se la usó en el estudio de construcciones lingüísticas que califican las condiciones de validez de las proposiciones. Actualmente la lógica modal se aplica en el área de la Informática para formalizar esquemas de razonamiento y sistemas donde intervienen múltiples agentes. La lógica modal en su versión proposicional es una extensión de la lógica de enunciados que también puede verse como un fragmento de la lógica de primer orden con buenas propiedades computacionales, como la decidibilidad.

Una *modalidad* es una palabra o frase que puede aplicarse a una proposición *A* para crear una nueva proposición que hace una afirmación acerca del modo de verdad de *A* o de las circunstancias bajo las cuales *A* es verdadera: cuándo, dónde o cómo *A* es verdadera. Ejemplos son: “en el futuro sucederá *A*” (*FA*), “está permitido *A*” (P*A*), “el agente sabe *A*” (K*A*), “es necesario *A*” ($\Box A$), “alguna ejecución finita del programa $\pi$ deja al sistema en un estado con información *A*” ($<\pi>A$), “es demostrable *A*”, entre muchas otras.

**Lenguaje modal.** Usamos un lenguaje proposicional clásico para trabajar. El lenguaje modal básico se funda sobre un conjunto numerable *P* de proposiciones usualmente denotadas con las letras p, q, r,… Expresiones complejas se forman sintácticamente del modo inductivo usual, usando (posiblemente) el operador $\bot$ (la constante *false*), el operador binario $\vee$ (disyunción), y el operador unario $\neg$ (negación). Como el comportamiento proposicional de esta lógica es clásico, asumimos que T (la constante *true*), $\wedge$ (conjunción), y $\rightarrow$ (condicional) se definen del modo esperado a partir de los símbolos ya provistos. A este lenguaje proposicional básico le agregamos un operador unario, que simbolizamos “$\Diamond$” y llamamos coloquialmente “diamante” o “rombo”. Con “$\Diamond$” *modalizamos* las expresiones, decimos algo de ellas colocándoles un símbolo delante: $\Diamond$p.

**Definición 3.1. Lenguaje.** Las fórmulas bien formadas (fórmulas) del lenguaje modal básico *L* se definen a partir de un conjunto de variables proposicionales *P* = {p, q, r…}, con los operadores booleanos usuales, y con un operador unario $\Diamond$, del siguiente modo:

$$p \mid q \ldots \mid \bot \mid \neg A \mid A \vee B \mid \Diamond A$$

con *A* y *B* fórmulas construidas del modo inductivo usual.

Agregamos el símbolo "□" para usarlo como una abreviatura. La relación entre ◊ y □ es *dual*: □p ≡ ¬◊¬p (el símbolo "≡" representa equivalencia lógica). Tradicionalmente, □ se lee "es necesario" y ◊ se lee "es posible". Coloquialmente también llamamos "cuadrado" al "□".

**Ejemplos de expresiones modales:**

$p \rightarrow q$ si nos haces falta entonces te llamamos

$p \rightarrow \Diamond q$ si nos haces falta entonces es posible que te llamemos

$p \rightarrow q$ si queremos aprender entonces estudiamos

$p \rightarrow \Box q$ si queremos aprender entonces es necesario que estudiemos.

Las lecturas de los símbolos □ y ◊, y también de otros símbolos modales, son muchas; diferentes lecturas de dichos símbolos ejercieron variada influencia a lo largo de los años en diferentes disciplinas, especialmente en la filosofía: se considera a la lógica modal como la herramienta por excelencia de la lógica filosófica, dando a los que la usan exquisitez y precisión para tratar con cuestiones metafísicas tales como la moral, para tratar con el tiempo, el espacio, el conocimiento, las obligaciones, etcétera. Algunos otros símbolos modales son, por citar algunos, O, F y P (por "obligatorio", "prohibido" y "está permitido") en la lógica deóntica, *F* y *P* (por "en el futuro sucederá que" y "en el pasado sucedió que") en la lógica temporal, K (por "el agente sabe que") en la lógica epistémica, $<\pi>$ y $[\pi]$ (por "alguna ejecución finita del programa $\pi$" y "toda ejecución finita del programa $\pi$") en la lógica dinámica.

**Ejemplos de expresiones de distintas lógicas modales:**

| | | |
|---|---|---|
| F(p) | prohibido pisar el césped | *lógica deóntica* |
| *P*(p) | en el pasado pisé el césped | *lógica temporal* |
| $<\pi>$q | alguna ejecución de $\pi$ arroja información q | *lógica dinámica* |

Las interpretaciones y usos actuales de la lógica modal caen dentro de dos grandes áreas: la de la *información* y la de la *acción*.

**Nota. Los Principios Generales.** Pareciera que un principio general de la lógica modal es □p → ◊p, cuya lectura intuitiva es "lo que es necesario, es posible". Sin embargo, a pesar de que dicha fórmula luce consistente desde el sentido común, no es un principio rector de la lógica modal (lo justificamos más adelante, al manejar el aparato formal). Considerar como principios generales de la lógica modal a diferentes fórmulas es difícil de decidir. También es difícil determinar qué fórmulas merecen ser consideradas principios de una lógica modal de propósito determinado como lo es la lógica modal temporal, que es una lógica modal para representar el tiempo, o la lógica deóntica, que es una lógica modal para formalizar normas.

**Discusión.** En la *lógica epistémica*, que se ocupa de precisar aspectos referidos al *conocimiento*, usamos Kp para simbolizar "el agente sabe p". Las fórmulas $Kp \rightarrow p$, $p \rightarrow Kp$, y $Kp \rightarrow KKp$, ¿podrían ser consideradas principios rectores de la lógica epistémica? ¿Cuál es la lectura intuitiva de cada una de ellas?

De aquí al final del capítulo nos concentramos en lenguajes modales con una, o a lo sumo dos modalidades, con o sin sus duales, y de aridad 1 (la aridad de un operador es la cantidad de argumentos para que el operador pueda funcionar). Pero no siempre debemos restringirnos así; existen lógicas modales con infinitos operadores (la lógica dinámica, por ejemplo), y lógicas modales con operadores de aridad mayor a 1.

Usamos también el concepto usual de sustitución uniforme, que permite reemplazar en una fórmula todas las apariciones de una subfórmula por otra. Entonces, por ejemplo, dada la fórmula $p \wedge q \wedge r$, y dada la sustitución $\sigma = \{p/(p \wedge \Box q), q/(\Diamond\Diamond q \vee r)\}$ tenemos que $[p \wedge q \wedge r]_\sigma = (p \wedge \Box q) \wedge (\Diamond\Diamond q \vee r) \wedge r$.

**Semántica.** Asociado a un lenguaje modal hay estructuras matemáticas en las que definimos las nociones de consecuencia lógica y verdad. Pasamos entonces a ver estas estructuras: frame y modelo.

Un *frame* es una dupla $F = (W,R)$ tal que W es un conjunto no vacío llamado universo (o dominio) de *F*, y R es una relación binaria sobre W. Los elementos en W se llaman *puntos*, *situaciones*, *estados*, o *mundos*, y a R se la denomina *relación de accesibilidad* entre mundos. Por ejemplo, el frame formado por los números naturales con la relación <, $F = (N,<)$, es un frame usual de la lógica temporal en el que podemos interpretar a cada mundo como un día, o una hora, o una semana. El frame formado por los números reales con la relación <, $F = (R,<)$ es también un frame usual para interpretar el tiempo y nos permite considerar al tiempo como *denso*: si cada mundo se corresponde con un número real que representa un instante de tiempo, entonces es posible identificar otro instante de tiempo entre cada par de instantes. Notar que en estos frames asumimos que tanto el pasado como el futuro son una *línea temporal*, pero tengamos en cuenta que existen concepciones del tiempo no determinísticas, donde el futuro y/o el pasado tienen una estructura de árbol.

Dado un lenguaje modal, un *modelo* es un par $M = (F,V)$ donde $F = (W,R)$ es un frame y $V: P \rightarrow \mathcal{P}(W)$ es una *función de valuación* que asigna a cada proposición p del lenguaje un subconjunto V(p) de W. Intuitivamente, V(p) es el conjunto de mundos en los que vale p. Los modelos, así presentados, son frames a los que les agregamos una función de valuación. A estos modelos se los llama *modelos de Kripke*.

Así como evaluamos fórmulas del cálculo de enunciados en el conjunto de valores de verdad booleanos representado por las constantes en el conjunto $\{true, false\}$, y así como evaluamos fórmulas del cálculo de predicados en una estructura que llamamos interpretación (que por definición consta de un conjunto no vacío de elementos llamado dominio, una colección de elementos distinguidos llamado constantes, una colección de funciones sobre elementos del dominio y una colección de relaciones sobre elementos del dominio), en la lógica

modal evaluamos fórmulas en modelos (y también en frames). Más adelante en este capítulo veremos que las estructuras de las interpretaciones de la lógica de predicados y las estructuras de los frames y modelos de la lógica modal guardan, en realidad, una muy estrecha relación entre sí.

Clásicamente tenemos la siguiente definición inductiva de cuándo una fórmula es verdadera en un modelo $M = (F,V)$ en un mundo $w$. Recordemos que $A \models B$ se lee coloquialmente "de $A$ se deduce $B$", o "$B$ es consecuencia lógica de $A$". En la definición que sigue, $M,w \models A$ se lee coloquialmente "$A$ es localmente verdadera en un mundo w en un modelo $M$".

**Definición 3.2. Verdad local.** Sea $p \in P$, sean $A, B \in L$:

$M,w \models p$ si y sólo si $w \in V(p)$ ,

Nunca sucede que $M,w \models \bot$ ,

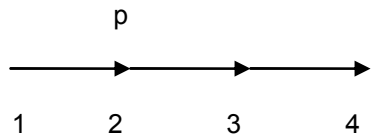
$M,w \models \neg A$ si y sólo si no sucede que $M,w \models A$,

$M,w \models A \vee B$ si y sólo si $M,w \models A$ o $M,w \models B$,

$M,w \models \Diamond A$ si y sólo si existe un mundo $v$ tal que $Rwv$ y $M,v \models A$.

**Algunos comentarios.** i) La segunda condición en la Definición 3.2 establece que en un mundo cualquiera no puede valer una contradicción (que es distinto a que una propiedad p valga o no valga en un mundo cualquiera); ello tiene sentido pues es de esperar que las contradicciones no valgan nunca en ninguna parte. ii) La idea de "posibilidad" surge naturalmente cuando notamos que para que una fórmula $\Diamond A$ sea verdadera en $w$ se requiere que $A$ sea verdadera en algún estado $v$ R-accesible desde $w$. iii) Dado que consideramos al símbolo $\Box$ una abreviatura de "$\neg\Diamond\neg$", de la última condición en la Definición 3.2 surge que $M,w \models \Box A$ si y solo si para todo mundo $v$ tal que $Rwv$ sucede que $M, v \models A$. iv) Finalmente, notemos que la noción de verdad dada en 3.2 es "local" a un mundo $w$ de un modelo $M$ para un frame $F$. Esta definición puede asimilarse al concepto de satisfactibilidad que conocemos del cálculo de enunciados y del cálculo de predicados.

**Ejemplo.** En el siguiente frame la proposición p es localmente verdadera en el mundo 2, es falsa en los mundos 3 y 4, y la proposición $\Diamond$p es localmente verdadera en 1.




**Nota. El frame bidireccional para la lógica temporal**. Definimos la estructura de un modelo básico para la lógica temporal $M = (T,R,V)$, y la semántica de sus operadores $F$ y $P$ como:

$M, t \models FA$ si y sólo existe un mundo $s$ tal que $Rts$ y $M, s \models A$, y con

$M, t \models PA$ si y sólo existe un mundo $s$ tal que R$st$ y $M, s \models A$.

Esta definición respeta la definición 3.2; a su vez, notemos cómo el operador $F$ va "hacia adelante" en R y el operador $P$ va "hacia atrás" en R, logrando el movimiento intuitivo pretendido en la línea del tiempo.

**Nota. Valuación de fórmulas.** Sea $F$ = (W,R) un frame y sea $w \in$ W un mundo en un modelo $M$ = ($F$,V). Extendemos naturalmente la función de valuación V, en el sentido inductivo usual, para que evalúe formulas: V($A$) = $\{w / M,w \models A\}$.

En el gráfico previo, la proposición $p \vee q$ es localmente verdadera en el mundo 2, y la proposición $\Box p \wedge \Diamond p$ es localmente verdadera en el mundo 1.

Además de querer saber si una fórmula es localmente verdadera o no, podemos querer saber si una fórmula es *globalmente verdadera*, esto es, si es verdadera en todos los puntos de un modelo dado. O si no lo es, claro.

**Definición 3.3. Verdad global.** Sea $A$ una fórmula, sea $M$ = ($F$,V) un modelo. $A$ es globalmente verdadera en $M$, escribimos $M \models A$, si $A$ es localmente verdadera en todos los mundos de W en $M$.

En el gráfico del ejemplo previo es fácil ver que la fórmula $\neg q$ es globalmente verdadera en el modelo.

Sabemos que si a un frame le agregamos información contingente (una valuación) tenemos un modelo. Pero podemos querer ignorar la información contingente (la que nos dice qué fórmula vale en qué mundo) y averiguar qué fórmulas son verdaderas con respecto a la estructura del frame. Esto es, podemos "olvidarnos" de la información contingente –de todos los modelos que existen para un frame- y averiguar qué información es verdadera respecto de la estructura del frame. Esta es una noción de verdad, si se quiere, "más fuerte" pues se enfoca en la estructura (más "sólida", "estática", o "fija") y no en las valuaciones (más "dinámicas", o "volátiles" que pueden suceder o no suceder):

**Definición 3.4. Validez.** Sean $A$ una fórmula, $F$ = (W,R) un frame, $w \in$ W un mundo.

a) $A$ es **válida en un mundo** $w$ en un frame $F$ ($F$,$w \models A$) si $M$,$w \models A$ para todo modelo $M$ = ($F$,V); es decir, cuando $A$ es localmente verdadera en $w$ para cualquier modelo $M$ "basado" en $F$.

b) $A$ es **válida en un frame** $F$ ($F \models A$) si $A$ es válida en todo mundo $w$ en $F$ = (W,R).

c) $A$ es **válida en una clase de frames** F (F $\models A$) si $A$ es válida en todo frame $F \in$ F.

d) $A$ es **válida** ( $\models A$) si $A$ es válida en **F**, la clase de todos los frames.

**Ejemplo.** La fórmula $\Diamond\Diamond p \rightarrow \Diamond p$ es válida en la clase de los frames transitivos. Para comprobar esta afirmación, sea $F$ = (W,R) un frame transitivo cualquiera (esto es, $F$ verifica la propiedad

$\forall xyz$ (R$xy \wedge$ R$yz$) $\rightarrow$ (R$xz$), con $x$, $y$, $z \in$ W), y sea $w$ cualquier mundo de W. Si sucede que $\Diamond\Diamond$p es verdadera en $w$ entonces existe un $v$ tal que R$wv$ y $F,v \models \Diamond$p; y si esto ocurre entonces existe un $u$ tal que R$vu$ y $F,u \models$ p. Como $F$ es transitivo tenemos que vale R$wu$, con lo que $F,w \models \Diamond$p. Por lo tanto la fórmula $\Diamond\Diamond$p $\rightarrow \Diamond$p es válida en cualquier $w$ en cualquier $F$ transitivo y, por definición, es válida en la clase de los frames transitivos.

**Discusión.** ¿Cuáles son las fórmulas a las que hace referencia la definición 3.4d)?

Recordemos que, si bien desde un punto de vista filosófico podemos decir que existe una única "noción de verdad" que todos aspiramos conocer, cuando manipulamos un sistema formal como la lógica modal -que intenta capturar y usar la noción de verdad (y la de falsedad)- solo podemos acercarnos a ésta a través de las herramientas que el propio sistema formal nos provee. Con lo cual el camino hacia la verdad siempre se nos presenta relativo a la herramienta que usamos para llegar a ella, y por lo tanto, la noción de verdad se vuelve de algún modo *relativa al sistema formal* que usamos. Para ilustrar este punto, pensemos en que la lógica de enunciados maneja los conceptos de verdad y falsedad de un modo simple y llano armando las tablas de verdad de acuerdo a las funciones de verdad de los conectivos. Luego, cuando usamos la lógica de primer orden (que es más rica que la lógica de enunciados) aparecen nuevas estructuras sobre las que testear validez y con éstas aparecen las nociones de verdad en una interpretación y validez lógica (verdad en todas las interpretaciones). Al trabajar con la lógica modal vemos que sucede lo mismo: tenemos distintas maneras de acceder al concepto de verdad, maneras sensibles a las estructuras con las que trabaja la lógica en cuestión.

**Relaciones de consecuencia lógica.** Tenemos ya formada una intuición del concepto de consecuencia lógica, y es la que dice que la validez de las premisas garantiza la validez de la conclusión. Hemos visto esta intuición formalizada en los capítulos 1 y 2, al estudiar lógica proposicional y lógica de predicados.

En la lógica modal las consecuencias lógicas dependerán de la estructura con la que estemos trabajando. Esto quiere decir que la noción de consecuencia lógica está *parametrizada*. Definimos a continuación, semánticamente, las nociones de *consecuencia local* y *consecuencia global*. La noción de consecuencia local es la noción de consecuencia lógica que ya hemos manejado en el cálculo de enunciados y en el cálculo de predicados, trasladada a la lógica modal.

**Definición 3.5. Consecuencia lógica "local".** Sea $\Sigma$ un conjunto de fórmulas, sea $A$ una fórmula, sea S una clase de estructuras (modelos, frames,…). Decimos que $A$ es consecuencia local de $\Sigma$ sobre S, $\Sigma \models_S A$, si para todos los modelos $M$ de S (si S son modelos, para ellos mismos, si S son frames, para todos los modelos de ellos) y todos los mundos $w$ de $M$, si sucede que $M,w \models \Sigma$ entonces $M,w \models A$.

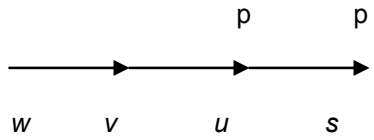
**Ejemplo.** Es fácil ver que $\{\Diamond\Diamond p\} \models_{\mathbf{Tran}} \Diamond p$, con **Tran** la clase de los frames transitivos. Pero también es directo notar que $\Diamond p$ no es una consecuencia local de $\{\Diamond\Diamond p\}$ en la clase **F** de todos los frames (para comprobarlo, proveer un *contramodelo*).

**Definición 3.6. Consecuencia lógica "global".** Sea $\Sigma$ un conjunto de fórmulas, sea *A* una fórmula, sea S una clase de estructuras (modelos, frames,…). Decimos que *A* es una consecuencia global de $\Sigma$ sobre S, $\Sigma \models^{g}_{S} A$, si para toda estructura *S* en S, si $S \models \Sigma$ entonces $S \models A$. Aquí, dependiendo del tipo de estructuras que contiene S, el símbolo "$\models$" se interpreta como validez en un frame (si S es una clase de frames), verdad global (en un modelo, si S es un conjunto de modelos), etcétera.

**Ejemplo.** Vale $\{\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p\} \models^{g}_{\mathbf{F}} \Diamond\Diamond p \rightarrow \Diamond p$; pero no vale $\{\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p\} \models_{\mathbf{F}} \Diamond\Diamond p \rightarrow \Diamond p$, siendo **F** la clase de todos los frames.

¿Por qué la primera afirmación es cierta? Porque las fórmulas $\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p$ y $\Diamond\Diamond p \rightarrow \Diamond p$ valen en los frames transitivos (probarlo); por lo tanto vale la consecuencia lógica global en la clase **F** de todos los frames (notar que en aquellos frames donde la subfórmula $\{\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p\}$ es falsa, la afirmación es verdadera).

Ahora bien, no vale la afirmación $\{\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p\} \models_{\mathbf{F}} \Diamond\Diamond p \rightarrow \Diamond p$. Esto es, la fórmula $\Diamond\Diamond p \rightarrow \Diamond p$ no es consecuencia lógica local de la fórmula $\Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p$ teniendo en cuenta la clase de todos los frames. Ello porque podemos construir el "contramodelo" *M* con la siguiente estructura finita:




y con V(p) = {*u*, *s*}. Entonces tenemos que: $M,w \models \Diamond\Diamond\Diamond p$, y $M,w \models \Diamond\Diamond p$, y por lo tanto $M,w \models \Diamond\Diamond\Diamond p \rightarrow \Diamond\Diamond p$. Y si bien $M,w \models \Diamond\Diamond p$, no es cierto que $M,w \models \Diamond p$ (pues, para que ello sucediera, deberíamos tener $M,v \models p$). Con lo que no vale $M,w \models \Diamond\Diamond p \rightarrow \Diamond p$.

Vemos a continuación algunas definiciones y herramientas sintácticas que permiten manejar las relaciones semánticas de validez y consecuencia lógica de un modo más automatizado. Esto es importante para nosotros como informáticos.

**Definición 3.7.a. Lógica modal.** Una lógica modal $\Lambda$ es un conjunto de fórmulas bien formadas que contiene todas las tautologías proposicionales, es cerrado -está *clausurado*- bajo *modus ponens* (esto es, si las fórmulas p y $p \rightarrow q$ pertenecen a $\Lambda$, entonces la fórmula q también), y es cerrado bajo sustitución uniforme (si una fórmula *A* pertenece a $\Lambda$ entonces todas sus instancias de sustituciones también).

Si una fórmula *A* pertenece a $\Lambda$ decimos que *A* es *teorema* de $\Lambda$. Si $\Lambda_1$ y $\Lambda_2$ son dos lógicas modales y $\Lambda_1 \subseteq \Lambda_2$ decimos que $\Lambda_2$ es una *extensión* de $\Lambda_1$.

**Definición 3.7.b. Lógica modal normal.** Una lógica modal $\Lambda$ es *normal* si contiene las fórmulas $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$ y $\neg\Box\neg p \leftrightarrow \Diamond p$, y es cerrada bajo Generalización (esto es, si una fórmula *A* pertenece a $\Lambda$, entonces $\Box A$ también).

Observemos que estas dos definiciones 3.7.a y 3.7.b son bien simples: identifican a una lógica como un conjunto de fórmulas que cumplen ciertas condiciones de clausura.

De la definición 3.7.a se desprende que la lógica de enunciados –tal como la estudiamos en el Capítulo 1- está contenida en una lógica modal. De ambas definiciones 3.7.a y 3.7.b podemos intuir que existen lógicas no normales, como veremos más adelante. Dicho de modo simple, la "normalidad" de una lógica modal queda determinada por la propiedad de distribución del "$\Box$" sobre el "$\rightarrow$" y por la regla de generalización.

Damos a continuación la definición sintáctica (axiomática) de una lógica modal.

**Definición 3.8. Sistema formal K de la lógica modal.**

Lenguaje

*L* (como en la definición 3.1).

Axiomas

Todas las instancias de tautologías proposicionales.

(K) $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$.

(Dual) $\neg\Box\neg p \leftrightarrow \Diamond p$.

Reglas de inferencia

· *Modus ponens*: a partir de p y de $p \rightarrow q$ obtenemos q.

· Sustitución uniforme: a partir de una fórmula *A* conseguimos una fórmula *B* sustituyendo uniformemente letras proposicionales en *A* por fórmulas arbitrarias.

· Generalización: si tenemos p obtenemos $\Box p$.

**Algunos comentarios y observaciones sobre el sistema formal K.** Traigamos a este punto la noción de "derivación" (o "deducción") que ya conocemos (lo hemos visto en el estudio del cálculo de enunciados y en el del cálculo de predicados). Sabemos que una *derivación* (o deducción) de *A* a partir de un conjunto finito $\Gamma$ de fórmulas bien formadas es una secuencia finita de fórmulas bien formadas $A_1,..., A_n$ en la que $A_n = A$ y, para todo *i*, cada $A_i$ de la secuencia es: o una instancia de uno de los esquemas de axioma provistos por el sistema formal, o es una fórmula en $\Gamma$, o se obtiene por aplicación de la regla de *modus ponens* entre dos fórmulas $A_k$ y $A_j$ que aparecen antes en la secuencia (esto es, con $k,j \leq i$); o se obtiene de la aplicación directa de la regla de Generalización sobre alguna fórmula $A_k$, con $k \leq i$. Esta última condición es la única condición "novedosa" que aparece ahora para la noción de derivación relativa a este sistema formal de la lógica modal. Cuando $\Gamma$ es el conjunto vacío, entonces

decimos que *A* es *teorema* de **K**. Para simbolizar que *A* es derivable (o deducible, o demostrable) a partir de $\Gamma$ en **K** escribimos $\Gamma \vdash_{\mathbf{K}} A$. Notemos que la noción de derivación nos permite pensar que el sistema formal **K** dado en la definición 3.8 induce una lógica modal en el sentido de la definición 3.7.a, esto es, el conjunto de las fórmulas derivables es una lógica en el sentido de la definición 3.7b.

**Otras observaciones**. El sistema modal **K** dado en la definición 3.8 es el mínimo normal, esto es, es el sistema modal normal que tiene menos restricciones. Vemos que, aplicando la regla de sustitución, podemos construir nuevas tautologías provenientes del cálculo proposicional y que ahora contengan las modalidades $\Box$ y $\Diamond$ (por ejemplo: $\neg\Diamond p \vee \Diamond p$). Es fácil ver que estas tautologías son válidas en todos los frames (queda para el lector la tarea de demostrar esta afirmación). Otras fórmulas válidas en todos los frames -como las que, por ejemplo, podemos obtener a partir del axioma K en un solo paso- no provienen por sustitución de ninguna tautología proposicional, pues el cálculo proposicional carece del símbolo "$\Box$". *Modus ponens* preserva validez en frames, verdad global y verdad local (dejamos al lector estas pruebas). Sustitución uniforme no preserva ni verdad global ni satisfactibilidad en un mundo (q se obtiene por sustitución de p, pero si p es verdad global en un modelo, no necesariamente lo es q. Probarlo construyendo un modelo apropiado).

Decimos que el axioma K permite realizar razonamiento proposicional tradicional porque el "$\Box$" se "mete" dentro del paréntesis y se "distribuye": entonces pasamos de tener una fórmula modal a tener un condicional entre dos subfórmulas modales. La regla de Generalización permite crear nuevas fórmulas modales yuxtaponiendo un "$\Box$" delante de una fórmula demostrable. Con K y Generalización tenemos entonces, de algún modo, un "interjuego" entre dos contextos: el proposicional y el modal. Generalización preserva verdad global (si p vale para todo mundo en un modelo entonces vale $\Box p$ porque en cada mundo vale p para todos sus adyacentes!) pero no preserva satisfactibilidad (si p es verdadera en un mundo no podemos afirmar que en el mundo vale $\Box p$).

**Extensiones de K.** El sistema modal **K** es un sistema formal mínimo y simple. Dado cualquier conjunto $\Gamma$ de fórmulas modales, podemos agregarlas como nuevos axiomas y formar el sistema modal **K**$\Gamma$. Esta técnica generadora de nuevos sistemas formales es sintáctica y la hemos usado ya en el estudio del cálculo de enunciados y también en el del cálculo de predicados para generar extensiones en ambos contextos.

Definir una lógica estableciendo las fórmulas que genera -esto es, agregar axiomas que para nosotros tienen algún tipo de interés- es un modo usualmente aceptado de especificar lógicas. Sin embargo, pareciera haber algo arbitrario en este proceso de definir lógicas "sintácticamente": ¿por qué agregaríamos algunas fórmulas como punto de partida de nuevos teoremas, y no otras?

Es útil también –desde el punto de vista formal- conocer cuál es la contraparte semántica de una lógica.

En muchos casos es posible describir a las extensiones del sistema **K** en términos de validez en frames. Esta es una perspectiva *semántica*. Con la siguiente definición tenemos entonces una manera diferente de especificar una lógica modal a como lo hemos hecho en 3.7.a. Describimos *semánticamente* una lógica modal mediante la identificación de la estructura de los frames en los que son válidas las fórmulas de la lógica en cuestión.

**Definición 3.9. Lógica modal (desde una perspectiva semántica).** Sea S una clase de frames. Definimos el conjunto de fórmulas $\Lambda_S = \{A \,/\, S \models A$, para todas las estructuras $S \in S\}$, con *A* fórmulas del lenguaje modal; $\Lambda_S$ es una lógica modal.

La relación entre el aspecto sintáctico y el aspecto semántico de las lógicas modales nos lleva a considerar resultados de correctitud (o adecuación, o *soundness*) y completitud (*completeness*) de dichas lógicas. Intuitivamente sabemos que la relación entre sintaxis y semántica debe ser tal que los teoremas que derivamos en la lógica son verdaderos y que todas las fórmulas verdaderas tienen una derivación sintáctica de la cual dicha fórmula es el último paso.

Existen teoremas que relacionan a las lógicas modales descriptas axiomáticamente con las estructuras de los frames en los que sus teoremas son válidos: estos teoremas a veces se conocen como *teoremas de determinación.*

**Ejemplos.** El axioma $\Diamond\Diamond p \rightarrow \Diamond p$ identifica a la lógica modal cuyas fórmulas son verdaderas en los frames transitivos, tal como hemos visto en dos ejemplos estudiados previamente en este capítulo. El axioma $p \rightarrow \Diamond p$ identifica a la lógica modal cuyas fórmulas son verdaderas en los frames reflexivos. El axioma $\Box p \rightarrow \Diamond p$ identifica a las lógicas modales cuyas fórmulas son verdaderas en los frames sin límite a derecha (*right-unbounded*). Las respectivas extensiones del sistema **K** para estas tres lógicas se llaman **K4**, **KT**, y **KD**. La clase de frames cuya relación es una relación de equivalencia (esto es, verifica reflexividad, transitividad y simetría) se identifican con la extensión de **K** conocida como **S5**.

A continuación damos las definiciones de correctitud y completitud de una lógica modal.

**Definición 3.10. Correctitud.** Sea S una clase de estructuras (concentrémonos en frames). Una lógica modal normal $\Lambda$ es correcta (*sound*) con respecto a S si $\Lambda \subseteq \Lambda_S$, con $\Lambda_S = \{A \,/\, S \models A \;\; \forall \; S \in S\}$.

Notemos que esta definición es por inclusión de un conjunto de fórmulas en otro conjunto (ya hemos usado este estilo de definición en la Definición 3.7.a).

De modo equivalente puede definirse que la lógica $\Lambda$ es correcta con respecto a S si para toda fórmula *A* y todas las estructuras $S \in S$, $\vdash_\Lambda A$ implica $S \models A$. Esto es, si *A* es teorema en $\Lambda$ entonces es válida en *S*. Decimos entonces que S *es una clase de estructuras para* $\Lambda$.

**Prueba de correctitud.** Para probar la correctitud de una lógica modal normal (presentada en términos de axiomas y reglas de inferencia) respecto de una clase de frames debemos probar que los axiomas de la lógica son válidos en la clase de frames de que se trate y que las reglas de inferencia (*modus ponens*, generalización y sustitución uniforme) preservan verdad.

A continuación damos la definición de completitud "fuerte" (existe también una definición de completitud "débil"). Que un sistema formal sea completo, genéricamente hablando, significa que *lo que es cierto en el sistema entonces es demostrable en el sistema*.

**Definición 3.11. Completitud (fuerte).** Sea S una clase de frames. Una lógica modal normal $\Lambda$ es "fuertemente" completa con respecto a S si, para cualquier conjunto de fórmulas $\Gamma \cup \{A\}$, si $\Gamma \models_S A$ entonces $\Gamma \vdash_\Lambda A$.

**Teorema de Completitud de K.** El sistema formal **K** de la lógica modal es fuertemente completo respecto de la clase de todos los frames.

Los teoremas de completitud son, esencialmente, teoremas de existencia de modelos. Esto es, para probar completitud usualmente hay que probar que determinados modelos "especiales" existen. Lo importante entonces es que sepamos cómo encontrarlos o cómo construirlos.

Para demostrar el teorema de completitud de **K** necesitamos conocer algunas definiciones y hacer algunos comentarios previos. Tengamos presente que:

i) Un conjunto $\Gamma$ de fórmulas es *consistente* si, o bien $A$ o bien $\neg A$ no es teorema de $\Gamma$ (ambas no son teoremas a la vez en $\Gamma$). Pensemos que si ambas $A$ y $\neg A$ son teoremas de $\Gamma$ entonces de $\Gamma$ se deduce una contradicción ($A \wedge \neg A$) que podemos usarla como premisa para derivar todas las fórmulas del lenguaje. De un conjunto inconsistente pueden derivarse todas las fórmulas. La consistencia es una característica importante de los conjuntos de fórmulas: poco interés tiene -tanto desde el punto de vista lógico como desde el punto de vista de los sistemas informáticos- un conjunto de fórmulas a partir del cual pueden derivarse todas las demás.

ii) Un conjunto $\Gamma$ de fórmulas de una lógica $\Lambda$ es *maximal $\Lambda$-consistente* si es consistente y cualquier otro conjunto $\Delta$ de fórmulas tal que $\Gamma \subset \Delta$ es $\Lambda$-inconsistente.

iii) La propiedad de compacidad (*compactness*) establece que un conjunto $\Gamma$ de fórmulas de una lógica $\Lambda$ es $\Lambda$-consistente si y solo si todo subconjunto finito de $\Gamma$ lo es. Daremos un esquema de la demostración de esta propiedad hacia el final de esta sección.

La técnica de prueba que se usa para demostrar el teorema de completitud de **K** se conoce como de *completitud por canonicidad*: se construyen modelos, llamados *canónicos*, a partir de conjuntos maximales consistentes.

Notemos dos detalles relevantes. Por un lado, todo mundo *w* en todo modelo *M* para una lógica $\Lambda$ está asociado con el conjunto de fórmulas $\{A \mid M,w \models A\}$, esto es, el conjunto de fórmulas que son verdaderas en *w*. Es fácil verificar que este conjunto de fórmulas es maximal $\Lambda$-consistente (es decir, si *A* es verdadera en algún modelo para $\Lambda$ entonces *A* pertenece a un conjunto maximal $\Lambda$-consistente). Por otro lado, si el mundo *w* está relacionado con otro mundo *v* en un modelo *M* debe quedarnos claro que la información codificada en los conjuntos maximales $\Lambda$-consistentes de *w* y de *v* está relacionada, digamos, "de algún modo coherente". Podemos entonces formarnos la intuición de que los modelos permiten que conjuntos maximales consistentes se relacionen coherentemente entre sí.

La idea detrás de la construcción de *modelos canónicos* es poner a trabajar estos dos detalles relevantes recién señalados: partir de colecciones de conjuntos maximales consistentes "coherentemente relacionados" e intentar obtener el modelo buscado. El objetivo es probar que la afirmación "*A* pertenece a un conjunto maximal $\Lambda$-consistente" es equivalente a "*A* es verdadera en algún modelo" (esta afirmación es un *Lema de Verdad*). Se prueba construyendo un modelo especial –el modelo canónico- cuyos mundos son todos los conjuntos maximales consistentes de la lógica $\Lambda$. Veamos la definición siguiente:

**Definición 3.12. Modelo canónico.** El modelo canónico para una lógica modal normal $\Lambda$ es la terna $(W^{\Lambda},R^{\Lambda},V^{\Lambda})$ con:

· $W^{\Lambda}$, el conjunto de todos los conjuntos maximales $\Lambda$-consistentes.

· $R^{\Lambda}$, la relación canónica sobre $W^{\Lambda}$, definida como: $R^{\Lambda}wv$ si para toda fórmula *A*, $A \in v$ implica $\Diamond A \in w$.

· $V^{\Lambda}$, la función de valuación canónica, definida como $V^{\Lambda}(p) = \{w \in W^{\Lambda} / p \in w\}$.

$W^{\Lambda}$ contiene todos los conjuntos maximales $\Lambda$-consistentes. Esto es relevante porque (por el *Lema de Lindenbaum*) cualquier conjunto $\Lambda$-consistente de fórmulas es un subconjunto de algún mundo de $W^{\Lambda}$ y entonces (por el *Lema de Verdad*) cualquier conjunto $\Lambda$-consistente de fórmulas es verdadero en algún mundo del modelo.

$R^{\Lambda}$ es una relación de accesibilidad entre conjuntos maximales consistentes basada (precisamente) en el concepto de consistencia. Como los mundos en $W^{\Lambda}$ son conjuntos maximales consistentes, si en el mundo *v* adyacente a *w* la fórmula *A* no fuese cierta entonces en *v* valdría $\neg A$ pero entonces por definición de $R^{\Lambda}$ en *w* valdría $\Diamond \neg A$, lo que es absurdo por ser *w* un conjunto consistente.

Finalmente, la función de valuación canónica $V^{\Lambda}$ iguala la verdad de un símbolo proposicional en *w* con su pertenencia a *w*. Así, el modelo canónico nos permite relacionar verdad con pertenencia a un conjunto maximal consistente.

Ya estamos en condiciones de organizar un esquema de la prueba del teorema de completitud fuerte de **K**.

**Orientación para la prueba del Teorema de Completitud fuerte de K.** Para probar la completitud fuerte de **K** hay que usar la noción de compacidad. Tenemos que encontrar, para cada conjunto $\Gamma$ K-consistente de fórmulas, un modelo *M* y un mundo *w* en *M* tal que $M,w \models \Gamma$. Elegimos $M = (F^K,V^K)$, el *modelo canónico* para **K**, y elegimos que el mundo *w* sea cualquier *conjunto maximal consistente* $\Gamma^+$ que extienda a $\Gamma$. Entonces $(F^K,V^K)$, $\Gamma^+ \models \Gamma$. Ciertamente podemos elegir a $M = (F^K,V^K)$ porque un resultado auxiliar (y relevante) nos lo garantiza: el *Lema de Lindenbaum* asegura que si $\Gamma$ es un conjunto $\Lambda$-consistente de fórmulas, entonces $\Gamma^+$ existe.

Finalmente, para terminar esta presentación de la noción de completitud, mencionamos que existe aún un resultado más poderoso y general llamado *Teorema del Modelo Canónico* que afirma que toda lógica modal normal es fuertemente completa respecto de su modelo canónico. Su demostración se apoya en el Lema de Lindenbaum y en la técnica de armado de modelos canónicos vista.

**Computabilidad y complejidad de las lógicas modales.** Para los informáticos es importante conocer aspectos de computabilidad y complejidad de las lógicas modales. Esto significa conocer cuántos recursos de tiempo (pasos de computación) y de espacio (memoria) se necesitan para saber si una fórmula es satisfactible en un modelo de una lógica dada.

**Decidibilidad.** Sabemos que un conjunto $\Gamma$ de fórmulas es decidible si existe un procedimiento (un método finito y efectivo de decisión) para determinar si cualquier fórmula del lenguaje pertenece a $\Gamma$.

Decimos entonces que una lógica modal normal $\Lambda$ es decidible si el problema de $\Lambda$-*satisfactibilidad* (determinar si una fórmula *A* es satisfactible en algún modelo para $\Lambda$) es decidible.

Existe otro problema interesante referido a decidibilidad de las lógicas modales y que se basa en el problema de $\Lambda$-satisfactibilidad es el problema de $\Lambda$–*validez*, que consiste en determinar si una fórmula *A* es válida en la clase de modelos M que identifica a la lógica $\Lambda$.
A continuación presentamos informalmente el problema de cómo se establecen resultados de $\Lambda$-*satisfactibilidad* y $\Lambda$–*validez* para una lógica modal.

Hemos visto que podemos tener una lógica modal especificada de manera puramente semántica, conociendo la clase de frames que la identifican. Y que también podemos conocerla desde su aspecto puramente sintáctico, sabiendo cuáles son los axiomas y las reglas que generan la lógica. También sabemos que la computación trata de la manipulación finita de estructuras finitas.

Sin importarnos si la lógica se nos presenta desde su aspecto sintáctico o desde su aspecto semántico debemos determinar si es decidible o no. Un instrumento para demostrar la decidibilidad de una lógica es el siguiente:

**Propiedad de Modelo Finito (f.m.p., finite model property).** Sea $\Lambda$ una lógica, y M una clase de modelos para $\Lambda$. Decimos que $\Lambda$ tiene la propiedad de modelo finito con respecto a M si dada una fórmula *A* de $\Lambda$ que es satisfactible en algún modelo en M entonces *A* es satisfactible en un modelo finito en M. Un modelo es finito si su conjunto de mundos W tiene una cantidad finita de elementos, si no el modelo es infinito.

La f.m.p. es interesante para nosotros como informáticos porque es una fuente de robustez computacional de la lógica modal: no tenemos que preocuparnos por un modelo infinito porque si vale la f.m.p. para la lógica en cuestión entonces siempre podemos encontrar otro modelo finito que es, de algún modo, "equivalente" al infinito. No entraremos en detalles de las técnicas de obtención de modelos finitos, pero informalmente mencionaremos dos: *selección* y *filtrado*. La primera elige cuidadosamente un submodelo finito del modelo infinito (por ejemplo, eliminando mundos que son redundantes). La segunda encuentra una estructura finita que se corresponde con el modelo infinito de modo que la estructura infinita puede mapearse en la estructura finita.

**Discusión informal de decidibilidad para lógicas especificadas semánticamente.** Supongamos que tenemos la lógica $\Lambda$ especificada semánticamente. Supongamos también que sabemos (o probamos) que $\Lambda$ verifica una forma "fuerte" de f.m.p., esto es: no solo verifica la f.m.p. respecto de alguna clase de modelos sino que además, para cualquier fórmula *A* existe una función computable *f* tal que $f(|A|)$ es una cota superior del tamaño de los modelos necesarios para satisfacer *A* (donde $|A|$ es la "longitud" de *A*, que puede estar medida tanto en cantidad de subfórmulas como en letras proposicionales). Entonces: escribimos un programa que recibe a *A* como input, genera todos los modelos finitos (de la clase de modelos de que se trata) hasta los del tamaño $f(|A|)$ y testea satisfactibilidad de *A* en estos modelos. Como *A* es $\Lambda$-satisfactible si y solo si es satisfactible en un modelo de $\Lambda$ de a lo sumo tamaño $f(|A|)$, y como el programa que construimos examina todos estos modelos, el programa determina $\Lambda$-satisfactibilidad.

**Discusión informal de decidibilidad para las lógicas especificadas sintácticamente.** Tenemos la lógica $\Lambda$ especificada mediante sus axiomas, y ya probamos que verifica la f.m.p. para alguna clase de modelos M. Entonces escribimos dos programas: uno que usa la axiomatización de $\Lambda$ para generar las fórmulas $\Lambda$–válidas; otro que genera los modelos finitos en M. Si una fórmula *A* dada es $\Lambda$–válida entonces será generada por el primer programa; si no lo es, encontraremos con el segundo programa el modelo finito en el que es falsa.

**Ejemplo.** La lógica modal mínima K es decidible. Verifica la f.m.p. "fuerte".

La prueba de esta propiedad requiere del armado de un modelo finito. Lo hacemos aplicando la técnica de filtrado: dado un modelo $M$ = (W,R,V) que satisface una fórmula $\phi$ en algún mundo, filtramos $M$ usando el conjunto $\Sigma$ (cerrado) de todas las subfórmulas de $\phi$ y obtenemos un modelo finito $M^f$ que satisface $\phi$. Escribimos dicho modelo finito como $M^f_\Sigma$ = $(W^f,R^f,V^f)$ y lo llamamos "el modelo filtrado de $M$ a partir de $\Sigma$". Se arma así:

· $W^f$ es el conjunto de las clases de equivalencias de los mundos de W. Para definir esas clases de equivalencias usamos la siguiente relación de equivalencia:

$$w \cong_\Sigma v \text{ sii para toda } \phi \text{ en } \Sigma \text{ ocurre que } (M,w \models \phi \text{ sii } M,v \models \phi).$$

Dos mundos son $\cong_\Sigma$-equivalentes si y solo si ocurre que para toda subfórmula $\phi$, $\phi$ es verdadera en w si y solo si es verdadera en v. Escribimos $|w|_\Sigma$ para referirnos a la clase de equivalencias de un estado w en $M$ con respecto a $\cong_\Sigma$. El mapeo $w \rightarrow |w|_\Sigma$ que envía cada mundo w a su clase de equivalencia $|w|_\Sigma$ se llama *mapeo natural*. Entonces: $W^f = \{|w|_\Sigma / w \in W\}$.

· $R^f$ se define como:
(i) si Rwv entonces $R^f|w||v|$, y
(ii) si $R^f|w||v|$ entonces para todo $\Diamond\phi \in \Sigma$, si $M,v \models \phi$ entonces $M,w \models \Diamond\phi$.

La primera de estas condiciones relaciona dos clases de equivalencias cada vez que dos mundos *w* y *v* se relacionan en W. La segunda condición se ocupa de conectar dos clases de equivalencias si ocurre que dos mundos se vinculan en W a partir de la semántica pretendida del operador $\Diamond$.

· $V^f(p) = \{|w|_\Sigma / M,w \models p\}$, para todas las letras de proposición p en $\Sigma$. Esto es, si una proposición vale en w, en el modelo filtrado la proposición vale en la clase de equivalencias de w, $|w|_\Sigma$. Ello surge naturalmente del concepto de mapeo natural.

**Comentario.** Por qué el modelo que obtenemos con el filtrado es *finito*. Para afirmar ello necesitamos conocer dos resultados. El primero: el conjunto de subfórmulas de una fórmula es claramente finito. En el armado previo, trabajamos con un conjunto $\Sigma$ que es un conjunto cerrado (o clausurado) de subfórmulas de $\phi$. Dado que trabajamos con subfórmulas de una fórmula bien formada es fácil ver que $\Sigma$ es finito; $\Sigma$ se arma así: para todas las fórmulas $\phi$, $\phi$': i) si $\phi \vee \phi' \in \Sigma$ entonces $\phi \in \Sigma$ y $\phi' \in \Sigma$; ii) si $\neg\phi \in \Sigma$ entonces $\phi \in \Sigma$; y iii) si $\Diamond\phi \in \Sigma$ entonces $\phi \in \Sigma$. El segundo resultado que necesitamos conocer establece que si $\Sigma$ es un conjunto cerrado de subfórmulas entonces, para algún modelo $M$, si $M^f$ es un filtrado de $M$ a través de un conjunto cerrado $\Sigma$ de subfórmulas, entonces $M^f$ contiene a lo sumo $2^n$ nodos (con n tamaño de $\Sigma$). Para probar este resultado recordemos que los estados de $M^f$ son las clases de equivalencias $W_\Sigma = \{|w|_\Sigma / w \in W\}$. Sea $g$ una función con dominio $W_\Sigma$ y rango $P(\Sigma)$ definida como $g(|w|) = \{ \phi \in \Sigma /$

$M$,w $\models \phi\}$. A partir de la definición de $\cong_\Sigma$ concluimos que $g$ está bien definida y es inyectiva. Por ello, el tamaño de $W_\Sigma$ es a lo sumo $2^n$, con n tamaño de $\Sigma$.

**Expresividad. Traducción Standard.** Al comienzo de este capítulo mencionamos que la lógica modal puede verse como un fragmento de la lógica de predicados. Trabajamos a continuación esa idea.

El siguiente algoritmo de traducción de fórmulas modales a fórmulas de primer orden nos permite una conexión con un contexto lógico más amplio y bien conocido para nosotros como lo es la lógica de predicados, donde podemos estudiar aspectos de expresividad. El algoritmo ST (por las iniciales en inglés de "traducción standard") recibe una fórmula modal y retorna una fórmula de primer orden con exactamente una variable libre (digamos, $x$). Las fórmulas modales se traducen a fórmulas de primer orden (escritas en un lenguaje de primer orden) que tiene exactamente un símbolo de relación. Intuitivamente, este símbolo de relación se corresponde con la relación que subyace a un frame.

Veamos cómo trabaja el algoritmo. A medida que aparecen operadores modales mientras "parseamos" la fórmula original (esto es, la recorremos sintácticamente de izquierda a derecha), aquéllos se traducirán en variables *nuevas* (que no aparecieron hasta entonces) cuantificadas en la fórmula de salida.

A continuación, el algoritmo ST:

$ST_x(p) = p(x)$

$ST_x(\bot) = x \neq x$ (una fórmula falsa)

$ST_x(\neg A) = \neg ST_x(A)$

$ST_x(A \vee B) = ST_x(A) \vee ST_x(B)$

$ST_x(\Diamond A) = \exists y(Rxy \wedge (ST_y A))$, donde $y$ es nueva

$ST_x(\Box A) = \forall y(Rxy \rightarrow (ST_y A))$, donde $y$ es nueva.

Si, por ejemplo, no estamos en alguna ocasión convencidos del significado intuitivo de una fórmula modal, podemos usar el algoritmo ST y trabajar o analizar la fórmula equivalente en el cálculo de predicados.

**Ejemplo.** Consideremos la fórmula $\Box p \rightarrow \Diamond p$, entonces $ST_x(\Box p \rightarrow \Diamond p) = ST_x(\neg\Box p \vee \Diamond p) = ST_x(\neg\Box p) \vee ST_x(\Diamond p) = \neg ST_x(\Box p) \vee ST_x(\Diamond p) = \neg\forall y(Rxy \rightarrow ST_y(p)) \vee \exists z(Rxz \wedge ST_z(p)) = \neg\forall y(Rxy \rightarrow p(y)) \vee \exists z(Rxz \wedge p(z)) = \forall y(Rxy \rightarrow p(y)) \rightarrow \exists z(Rxz \wedge p(z))$.

El estudio de la expresividad de las fórmulas modales en relación con el cálculo de predicados cae en el marco de lo que se conoce como *Teoría de Correspondencia*. El algoritmo ST es un puente importante entre la lógica modal y el cálculo de predicados porque podemos transferir ideas, resultados e incluso algunas técnicas de demostración entre una lógica y otra. Para esto, es útil verificar que no existe distinción matemática entre modelos modales y modelos de primer orden, y que ambos son esencialmente estructuras relacionales: un modelo

modal $M$ = (W,R,V) provee una relación binaria R que puede usarse para interpretar un símbolo de relación R, y el conjunto $V(p_i)$ puede usarse para interpretar cada predicado unario $p_i$ (correspondiente cada uno de ellos a cada letra de proposición en el lenguaje modal). Dicho esto, existen dos resultados importantes que establecen:

i) **Correspondencia local entre modelos.** Para todo modelo $M$ y todos los estados $w \in M$; $M,w \models A$ sí y solo si $M \models ST_x(A)\ [w]$ (esta última expresión se lee "la expresión $ST_x(A)$, escrita en un lenguaje de primer orden, es verdadera cuando la variable x se instancia con el valor $w$").

**ii)** **Correspondencia global entre modelos.** Para todo modelo $M$; $M,w \models A$ sí y solo si $M \models \forall x\ ST_x(A)$.

La prueba de ambos resultados se hace por inducción sobre la estructura de $A$.

**Ejemplo.** Es posible usar el algoritmo ST para obtener la compacidad de la lógica modal como corolario de la prueba de compacidad para la lógica de primero orden. La propiedad de compacidad, que establece que un conjunto $\Gamma$ de fórmulas de una lógica $\Lambda$ es $\Lambda$-consistente si y solo si todo subconjunto finito de $\Gamma$ lo es. Para demostrar una de las dos implicaciones, consideremos a $\Gamma$ un conjunto de fórmulas modales en el que cada subconjunto es satisfactible. ¿El conjunto $\Gamma$ es satisfactible? Consideremos el conjunto $\{ST_x(A) / A \in \Gamma\}$: es un conjunto de fórmulas escritas en un lenguaje de primer orden. Como cada subconjunto finito de $\Gamma$ tiene un modelo, por correspondencia local entre modelos sucede que todo subconjunto finito de $\{ST_x(A) / A \in \Gamma\}$ también; y por lo tanto, por *compacidad demostrada para primer orden* (ver el Capítulo 4 de *Lógica para Matemáticos*, de A. G. Hamilton) ese conjunto de fórmulas es satisfactible en algún modelo, digamos $M$. Entonces, nuevamente por correspondencia local entre modelos, $\Gamma$ es satisfactible en $M$.

## Lógica Deóntica

La lógica deóntica es la "lógica de lo que debe ser", de lo *obligatorio* y lo *prohibido*, y como tal es fundamento de la Ética y del Derecho (*deon* viene del griego *lo que debe ser*). Últimamente se la usa también en el área de la Informática para la especificación, por ejemplo, de sistemas y protocolos de seguridad, donde hay permisos y prohibiciones de acceso. La lógica deóntica sienta las bases para el estudio de teorías de argumentación, de lógicas de la acción, de agentes, de grupos; y para el abordaje de enfoques cognitivos del Derecho. Todas estas teorías incluyen novedosas y precisas definiciones formales de conceptos tales como poder institucional, representación, obligaciones, grupos y equipos, delegación, cumplimiento y violación de normas, confianza, contratos, entre otros, con miras a ser aplicadas en sistemas computacionales inteligentes.

Una de las principales características de las reglas deónticas es que pueden ser *violadas*. Es en este aspecto en el que difieren de otras reglas, normas, o principios, por ejemplo de las

matemáticas o de la naturaleza. En esos contextos los principios no pueden quebrarse fácilmente. Por ejemplo, a ninguno de nosotros nos tomará demasiado esfuerzo violar la norma que establece que no debemos cruzar el semáforo en rojo cuando conducimos un automóvil; sin embargo, es imposible que un círculo tenga un área distinta a $\pi r^2$ o que dos moléculas de hidrógeno y una de oxígeno se unan para formar una sustancia distinta de agua.

Para los informáticos, conocer formalismos simbólicos de la lógica deóntica aumenta nuestras capacidades de razonamiento abstracto en el área de sistemas y nos prepara para enfrentar -desde un punto de vista lógico formal- muchas de las modernas teorías de sistemas donde intervienen múltiples agentes, cada uno con sus propias creencias e intenciones y que interactúan entre ellos para lograr sus objetivos en un ambiente donde hay normas de diferentes tipos y jerarquías. Las aplicaciones de la lógica deóntica a la informática normalmente se relacionan con modos de especificación computacional de normas, esto es, con formas de especificación de comportamiento ideal, de lo que "debe ser". Hay normas que regulan el funcionamiento de los sistemas de computación, el comportamiento, movimiento y seguridad de sus usuarios, y normas que gobiernan el núcleo central de procesamiento de un sistema, propiamente dicho. Los sistemas -y las organizaciones, o instituciones- junto con sus partes, sus usuarios y sus miembros integrantes (que pueden ser otras instituciones), están cruzados por normas de diferentes clases: en los sistemas hay normas de accesos y permisos, hay especificaciones de políticas de trabajo, de comunicación y de acción; hay otros tipos de reglas como las de orden y de limpieza, o guías de comportamiento, horarios de entradas, salidas, hay también restricciones de integridad y de seguridad, hay reglas que son para los usuarios y otras que son para empleados, etcétera. Cómo modelar computacionalmente normas, cómo hacerlas cumplir, cómo detectar su violación y cómo determinar y exigir un resarcimiento ante un incumplimiento son temas de los que se ocupa el área de sistemas normativos dentro del área más grande de sistemas inteligentes.

Es conveniente que conozcamos las dificultades que tiene la lógica proposicional básica para capturar formalmente el discurso normativo, donde cobra especial interés la categoría filosófica del "deber ser". La lógica proposicional clásica es insuficiente para representar dicha categoría porque las proposiciones son o verdaderas o falsas, es decir, las cosas son o no son. No podemos simbolizar que las cosas "deben ser" o "está prohibido que sean", solo podemos simbolizar la *forzosidad* de que las cosas son o no son, que los hechos ocurren o no ocurren. Mostraremos esta incapacidad de la lógica proposicional para representar el "deber ser" con un caso simple (Ejemplo de la Biblioteca del Imperial College, dado por A.I. Jones y citado por R. J. Wieringa y J-J. Ch. Meyer):

**Ejemplo. Reglas de la biblioteca.** *El lector devolverá el libro en 15 días hábiles. Si el lector devuelve el libro en 15 días hábiles, no se le aplicará el apercibimiento administrativo del artículo 20. Si el lector no devuelve el libro en 15 días hábiles, se le aplicará el apercibimiento administrativo del artículo 20.*

Si formalizamos estas reglas usando lógica proposicional, tenemos tres proposiciones: p, $p \rightarrow \neg q$ y $\neg p \rightarrow q$ para la primera, segunda y tercera regla de la biblioteca respectivamente. Supongamos que ocurre que el lector no devuelve el libro en 15 días hábiles; formalizamos ese hecho como $\neg p$. Tenemos entonces: i) una contradicción entre este hecho nuevo y la primera regla de la biblioteca; y también tenemos ii) que entre las dos primeras reglas deducimos $\neg q$ por *modus ponens*, y entre la tercera regla y el hecho nuevo $\neg p$ deducimos q, con lo cual conseguimos q y $\neg q$. Esto sorprende, porque las reglas tal como están presentadas en lenguaje natural son coherentes desde el punto de vista de lo que se debe hacer para el correcto funcionamiento de la biblioteca y porque además las hemos traducido de un modo directo al lenguaje de la lógica proposicional. No solo no hay error alguno en las reglas de la biblioteca ni en su formalización proposicional sino que tampoco hay error en el proceso de deducción llevado a cabo. Simplemente, la lógica proposicional "se queda corta" para representar que algo es forzoso.

Esta dificultad de la lógica proposicional para modelar el deber ser favoreció la búsqueda de representaciones que fueran adecuadas.

**Formalización de conceptos deónticos.** El aspecto técnico más influyente sobre las descripciones formales modernas de la lógica deóntica aparece en el trabajo seminal de von Wright, *Deontic Logic*, de 1951. Dicho trabajo define un sistema formal proposicional elemental que incluye los modos deónticos básicos P, F y O de *permiso*, *prohibición* y *obligación*, representados con tablas de verdad que incluyen los conectivos booleanos usuales. Las letras proposicionales se corresponden con acciones simples, por ejemplo, con la letra p describimos acciones tales como "pagar", "estacionar", "matar", "adeudar", "fumar", "robar", etc. Asumimos que la expresión negada $\neg p$ se interpreta como "no estar haciendo p". Conceptualmente entonces tenemos que los modos deónticos *modalizan acciones*. Algunos renglones de las tablas, por corresponderse con escenarios imposibles para un contexto normativo, no eran para von Wright combinaciones admisibles. Por ejemplo, al armar la tabla de verdad de la proposición $Pp \vee P\neg p$, cuya lectura intuitiva es "está permitido hacer p o está permitido no hacer p", la combinación falso-falso de valores de verdad para Pp y para $P\neg p$ es una combinación inadmisible porque o bien uno tiene permitido hacer p o bien uno tiene permitido hacer $\neg p$; en la realidad no se da el caso de que *poder hacer p* y *poder hacer la negación de p* sean ambas falsas pues en un momento dado o estamos haciendo p o no lo estamos haciendo. Entonces, es por ello que el renglón falso-falso en la tabla de verdad de la proposición $Pp \vee P\neg p$, para von Wright, no existe.

Luego de que von Wright explicara su sistema deóntico en 1951, el área de la lógica deóntica floreció de estudios y se descubrió que aquellos tres operadores para los cuales von Wright armaba tablas de verdad podían –con mayor o menor éxito- representarse con los operadores de necesidad y posibilidad de una lógica modal normal.

En el resto de esta sección estudiamos a la lógica deóntica descripta como una lógica modal normal.

**Operadores modales en su interpretación deóntica.** El operador "□" es comúnmente usado para modalidades de carácter universal; esta intuición ya la manejamos porque conocemos la semántica del operador □. Notemos que esta universalidad del □ coincide con nuestra idea básica de obligatoriedad: algo es obligatorio si necesariamente debe ser cumplido sea cual sea la situación o el estado de cosas.

Entonces, para trabajar en un contexto deóntico, simplemente reescribimos □ como O (por "obligatorio"). Así, tenemos que dado un mundo *w* la fórmula O*A* es verdadera en *w* si *A* es verdadera en todos los mundos o situaciones que son adyacentes a *w*: la semántica para O es *verdad en todos los mundos R-accesibles*.

El dual del operador O es el operador P cuya lectura intuitiva es "permitido": Notemos que la dualidad $Op \leftrightarrow \neg P\neg p$ se ajusta a nuestra intuición de permiso pues algo es obligatorio si no ocurre que está permitido que su negación suceda. Por ejemplo, "es obligatorio hacer silencio" es equivalente a "no se permite no hacer silencio".

Finalmente, el tercer operador comúnmente usado en la lógica deóntica es una abreviatura: definimos el operador F, cuya lectura intuitiva es "prohibido", como $Fp \leftrightarrow \neg Pp$, esto es, si algo está prohibido es porque no está permitido que lo llevemos a cabo. Por ejemplo "prohibido fumar" es equivalente a "no se permite fumar". Es fácil ver que Fp también puede escribirse como $O\neg p$: reemplazando adecuadamente vemos que $O\neg p \equiv \neg P\neg(\neg p) \equiv \neg Pp \equiv Fp$.

A continuación describimos formalmente desde el punto de vista sintáctico un sistema modal básico de lógica deóntica.

**Definición 3.13. Sistema modal KD de la lógica deóntica.** Definimos el sistema de la lógica deóntica como sigue:

Lenguaje

El lenguaje proposicional estándar, al que sumamos los símbolos O, P, F.

Axiomas

Todas las tautologías del cálculo proposicional

(K) $O(p \rightarrow q) \rightarrow (Op \rightarrow Oq)$

(P) $Pp \leftrightarrow \neg O\neg p$

(D) $Op \rightarrow Pp$

(F) $Fp \leftrightarrow O\neg p$

Reglas de inferencia

*modus ponens*, generalización y sustitución uniforme.

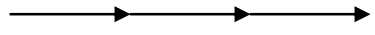
Algunos comentarios sobre esta definición. La lógica deóntica así definida -como una extensión de la lógica modal normal mínima- captura la intuición que tenemos sobre el universo de las normas. El axioma K es el axioma de distribución de toda lógica normal. Los axiomas (P) y (F) se corresponden con el dual de O y con una abreviatura, respectivamente. El axioma (D) establece el principio deóntico de que si algo es obligatorio entonces está permitido, lo que

tiene sentido, pues si pretendemos que algo sea impuesto por una norma entonces ese algo tiene que estar permitido o habilitado.

**Discusión.** Notemos que podemos aplicar la regla de generalización a las tautologías proposicionales por ser éstas teoremas del sistema de la lógica deóntica. Pero, ¿las tautologías son obligatorias?

**Discusión conexa.** La fórmula $\neg O\bot$ es un teorema del sistema formal de la lógica deóntica. Para probarlo, veamos que la fórmula OT es demostrable en el sistema por aplicación de la regla de Generalización (con T constante *true*, definida como abreviatura, tal como hicimos en la Sección 3.1). Seguidamente, invocamos el axioma (D) y conseguimos $OT \rightarrow PT$. Aplicando *modus ponens* y a continuación (P) obtenemos $PT \equiv \neg O\neg T \equiv \neg O\bot$. Este teorema le da cierta coherencia fundamental a cualquier sistema de normas, impidiéndole tener normas que sean contradicciones.

**Semántica.** A la semántica formal de esta lógica deóntica se la llama **KD** (por abuso de lenguaje se suele mencionar el nombre del cálculo sintáctico, normalmente en la jerga se dice: "este sistema tiene semántica KD estándar"). Los modelos para esos frames tiene estructura $(W, R_O, V)$, con W mundos, V función de valuación y $R_O$ relación de accesibilidad tal que cumple que para todo $w \in W$ existe un $v \in W$ tal que $R_O wv$ ($\forall w \exists v\ R_O wv$). Esto es, la lógica deóntica presentada es fuertemente completa respecto de los frames *seriales* o "sin límite a derecha"; son frames en los que la relación de accesibilidad entre mundos se denomina *serial*: siempre para cada mundo hay otro mundo accesible.




**ejemplo de frame serial**

Para probar que la lógica **KD** es fuertemente completa respecto de los frames sin límite a derecha es suficiente con mostrar que el modelo canónico para **KD** es sin límite a derecha. Esto requiere de una prueba de existencia. Sea *w* cualquier punto en el modelo canónico para **KD**, debemos probar que existe un *v* tal que $R_{\mathbf{KD}}wv$. Como *w* es un conjunto **KD**-maximal consistente, entonces contiene la fórmula $\Box p \rightarrow \Diamond p$. Por lo tanto, por clausura de los conjuntos maximales consistentes y por sustitución uniforme, *w* contiene a $\Box T \rightarrow \Diamond T$ (con T constante *true*). Como las tautologías pertenecen a toda lógica modal normal, por aplicación de la regla de generalización $\Box T$ también, y entonces, por *modus ponens*, $\Diamond T \in w$ y por lo tanto existe *v* sucesor $R_{\mathbf{KD}}$-accesible de *w* por aplicación del *Lema de Existencia*. Este lema establece que para toda lógica normal $\Lambda$ y cualquier estado $w \in W^{\Lambda}$, si $\Diamond A \in w$ entonces existe un estado $v \in W^{\Lambda}$ tal que $R^{\Lambda}wv$ y $A \in v$, con $W^{\Lambda}$ y $R^{\Lambda}$ como en la Definición 3.12.

**Algunos teoremas de KD.** Algunos de los más relevantes son:

$(O\wedge)$ $O(p \wedge q) \leftrightarrow (Op \wedge Oq)$

$(P\wedge)$ $P(p \wedge q) \rightarrow (Pp \wedge Pq)$

$(F\wedge)$ $(Fp \vee Fq) \rightarrow F(p \wedge q)$

$(O\vee)$ $(Op \vee Oq) \rightarrow O(p \vee q)$

$(P\vee)$ $P(p \vee q) \leftrightarrow (Pp \vee Pq)$

$(F\vee)$ $F(p \vee q) \leftrightarrow (Fp \wedge Fq)$

Es importante notar que en estos teoremas hay implícita una suposición de *cotemporalidad*, es decir, los actos o hechos representados por las letras proposicionales en cada teorema se consideran como ocurriendo *a la vez*, *en simultáneo*. Vemos que la obligación de una conjunción de, digamos, dos actos, es equivalente a la conjunción de las obligaciones de cada acto por separado ($O\wedge$), y que el permiso de una disyunción es equivalente a la disyunción de los permisos ($P\vee$). Ambas equivalencias seguramente nos resultan intuitivas, para ($O\wedge$) por ejemplo una posible lectura en lenguaje natural es: “es obligatorio permanecer de pie” y “es obligatorio guardar silencio”, y “es obligatorio permanecer de pie y guardar silencio”. Siguiendo, notemos que el permiso de una conjunción implica la conjunción de los permisos ($P\wedge$); pero en el otro sentido la implicación no vale: por ejemplo, que esté permitido conducir un automóvil y también esté permitido hablar por teléfono móvil no implica que estén permitidas a la vez ambas acciones. Dejamos al lector la lectura intuitiva (y también la demostración) de los teoremas restantes teniendo en cuenta la suposición de cotemporalidad.

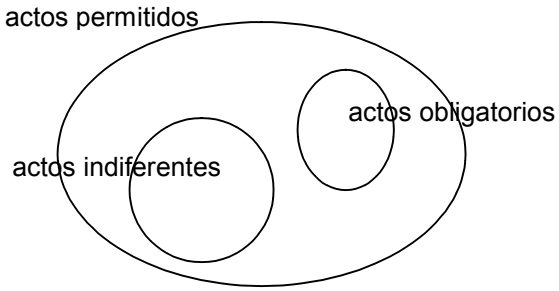
**Ventajas del enfoque modal de la lógica deóntica.** La descripción lógica de normas usando operadores deónticos permite descubrir patrones normativos. A partir de allí, es posible explorar nuestras capacidades de diseño y análisis de normas a distintos niveles de jerarquías de normas y también en cuanto a normas de distintos tipos: jurídicas, morales, de tránsito, de etiqueta, etc. Adquirimos así mayor capacidad “ingenieril” en el sentido de poder definir y determinar las formas lógicas de normas, relaciones entre ellas, y nuevas categorías normativas.

**Ejemplo. Definición de nuevas categorías normativas.** von Wright dio la definición para el concepto de *acto indiferente* usando el operador deóntico P del siguiente modo: un acto (simbolizado con una letra proposicional) es indiferente si el acto está permitido y su negación también. Por ejemplo, en una plaza está permitido fumar, y también está permitido no fumar (en símbolos: $Pf \wedge P\neg f$). Además, von Wright explicó que, si bien todos los actos indiferentes están permitidos (($Pf \wedge P\neg f) \rightarrow Pf$), aquello que está permitido no es indiferente (por ejemplo, que esté permitido honrar a la patria no implica que esté permitido honrar a la patria y que también esté permitido deshonrarla). Los actos indiferentes pueden lucir triviales en su estructura lógica; sin embargo, pueden resultar relevantes a la tarea de diseño de sistemas normativos, esto es, en un sentido “ingenieril” de un cuerpo normativo: cuando ciertos actos son identificados como indiferentes, seguramente no integrarán ninguna obligación, no serán parte de ninguna norma. Al identificar actos indiferentes podemos “purgar” o “limpiar” un cuerpo normativo (una base de

datos normativa) de ellos. von Wright también sostuvo que lo que es obligatorio está permitido (axioma D) pero no es indiferente; ello es fácil de ver porque por el axioma D tenemos $Op \rightarrow Pp$ pero no es posible derivar $Op \rightarrow (Pp \wedge P\neg p)$ en el sistema.

Notemos que estas estructuras de fórmulas, que representan diferentes categorías normativas, son aplicables a actos considerados aisladamente, "de a uno". von Wright presentó también conceptos deónticos que se aplican a pares de actos, como la idea de *actos incompatibles*: dos actos son incompatibles si su conjunción está prohibida: $F(p \wedge q)$, como por ejemplo conducir un automóvil y hablar por teléfono celular. También presentó la idea de *compromiso*: un acto nos compromete a (hacer) otro acto si la implicación entre ambos es obligatoria. Por ejemplo, hacer una promesa nos compromete a cumplirla: vemos que $O(p \rightarrow q) \equiv \neg P\neg(p \rightarrow q) \equiv \neg P\neg(\neg p \vee q) \equiv \neg P(p \wedge \neg q)$, que puede leerse intuitivamente como "si uno se obliga prometiendo que si p es el caso, entonces cumplirá con q". Vemos que no está permitido prometer p y no cumplir con q ($\neg q$).

Gráficamente tenemos:




**Paradojas deónticas.** Las paradojas deónticas son expresiones que son verdaderas en el sistema **KD** pero que carecen de significado o son, directamente, contradictorias cuando las analizamos desde el sentido común. Algunos ejemplos son:

| | |
|---|---|
| Paradoja de Ross | $Op \rightarrow O(p \vee q)$ |
| Paradoja del Penitente | $Fp \rightarrow F(p \wedge q)$ |
| Obligación derivada | $Op \rightarrow O(q \rightarrow p)$ |
| Sistema normativo vacío | OT, con T *true* (cualquier tautología) |

Una posible lectura en lenguaje natural de la paradoja de Ross es "es obligatorio que lleves esta carta al correo, entonces es obligatorio que o lleves esta carta al correo o la quemes". La paradoja del Penitente puede ejemplificarse en lenguaje natural con "está prohibido matar, por lo tanto están ambos prohibidos matar y arrepentirse". Finalmente, notar que la paradoja de la obligación derivada proviene de la definición de la función de verdad del condicional tal como lo conocimos al estudiar la lógica proposicional (que en la jerga algunos denominan "implicación material").

von Wright consideró que las tautologías no necesariamente debían ser obligatorias, y que tampoco las contradicciones deben estar prohibidas. Estos dos escenarios deónticos le

resultaban a von Wright innecesarios, ajenos a cualquier sistema normativo que se preciara de ser coherente con la realidad, fundando esto en el hecho de que no estamos obligados a hacer cosas verdaderas y en que muchas veces hacemos contradicciones. A estos dos escenarios los consideró integrantes de lo que llamó el *Principio de Contingencia Deóntica*, que podemos formalizar $\neg OT \wedge \neg F\bot$. Ahora bien, notemos que tal como hemos lo hemos planteado en la discusión conexa a la Definición 3.13 la fórmula OT es teorema del sistema formal **KD**, y también notemos que $OT \equiv (\neg P\neg)T \equiv F(\neg T) \equiv F\bot$. Justamente las fórmulas OT y $F\bot$ hacen caer el Principio de Contingencia Deóntica considerado válido por von Wright para quien OT y $F\bot$ no deben ser teoremas de ningún sistema de normas. Así, vemos que **KD** entra en colisión con el sistema original propuesto por von Wright.

**Decidibilidad del sistema KD.** Dejamos al lector la prueba de decidibilidad de **KD** teniendo en cuenta que ya sabemos que: **KD** es axiomatizable mediante un número finito de esquemas de axioma, y que el axioma D determina la clase de los frames seriales (lo hemos probado más arriba). Queda por demostrar, para la prueba de decidibilidad, que **KD** posee la propiedad de modelo finito. Ello puede hacerse mediante un filtrado, siguiendo los pasos descriptos hacia el final de la Sección 3.1.

**Otros enfoques para la representación de normas.** El enfoque que usa solo la lógica proposicional clásica -despojada de operadores deónticos, como en el ejemplo de las reglas de la biblioteca- se ubica en lo que en el área se denomina “enfoque factual”, es decir, relativo a los hechos, a “lo que es”. A este enfoque también pertenecen intentos de representar normas usando lógica de predicados -tal como la hemos estudiado en el Capítulo 2-. En este contexto, las normas se ven como *definiciones* en lugar de obligaciones, permisos y prohibiciones. De este modo, las normas se formalizan como predicados de un lenguaje de primer orden, o como cláusulas en un programa Prolog. Esta versión del diseño de normas presenta las conocidas ventajas y características que posee la lógica de predicados por sobre las de la lógica de enunciados; pero tengamos en cuenta que las cláusulas Prolog o los predicados de primer orden no permiten capturar la categoría deóntica, es decir, no permiten diferenciar entre lo que “es” y lo que “debe ser” sino que *formalizan conceptos*. Autores como A. I. Jones han remarcado que, claramente, los enfoques factuales de las normas son limitados en su capacidad de modelización pero *no tienen nada de malo* dado que permiten estudiar, por ejemplo, cómo diferentes definiciones o conceptos legales se aplican a un caso en estudio, o cómo analizar textos legales, etc.

**Operadores deónticos relativizados.** Hasta aquí hemos visto a la lógica deóntica, que tiene un operador modal O de obligación. Este operador O es *genérico* en el sentido de que es *impersonal* pues asume una *referencia tácita a todos los obligados*, que somos *todos los individuos* o agentes integrantes del grupo o de la comunidad de que se trate. Así, entendemos al operador O como de *obligación general*. Pero podemos hacer una distinción y referirnos a quiénes son los individuos obligados introduciendo operadores de obligación relativizados a

dichos individuos, operadores que autores como H. Herrestad y C. Krogh llaman *obligaciones especiales*. Estos autores definen un nuevo operador deóntico $O_xA$ cuya lectura intuitiva es "es obligatorio *A* para el individuo o agente *x*" (y su semántica es KD). Supongamos que consideramos la posibilidad de crear una extensión del sistema **KD** agregándole esta modalidad, entonces aparecen relaciones interesantes que debemos considerar, por ejemplo, seguramente pretenderemos que valga en la extensión el axioma $OA \rightarrow O_xA$ que establece que si algo es obligatorio en términos generales entonces es obligatorio para el individuo *x*. Del mismo modo que relativizamos obligaciones para denotar normas individuales podemos relativizar el operador deóntico para que indique a favor de qué individuo debe *x* cumplir la obligación de *A*. Por ejemplo, podemos definir $O^y{}_xA$ cuya lectura intuitiva es "*x* está obligado a *A* en el interés de *y*". Este tipo de obligación relativizada es muy específica pues denota las dos partes: el deudor y el beneficiario de una obligación individual.

Estudiamos en la sección siguiente más aspectos referidos a operadores individuales para cada agente.

**Síntesis.** Hemos mencionado a la lógica proposicional y su limitación para representar satisfactoriamente la categoría del "deber ser" de las normas. Hemos estudiado cómo el enfoque modal deóntico sí hace una distinción precisa entre las categorías filosóficas del "deber ser" (lo ideal) y del "ser". Muchas veces las distinciones y debates filosóficos no tienen aplicaciones concretas en la realidad; la formalización de la lógica deóntica como una lógica modal computable es un ejemplo de un concepto filosófico que puede materializarse y ser puesto en uso en un sistema computacional.

## Sistemas Multiagente

El área de sistemas multiagente (MAS, por las siglas en inglés de multi-agent systems) se ocupa principalmente de modelar agentes cognitivos (actores humanos o entidades computacionales que *saben* y *conocen*) o reactivos (que *actúan* y *reaccionan*), que dependen unos de otros para lograr sus objetivos individuales o grupales, e interactúan en varios y diferentes ambientes.

Hay al menos cuatro usos actuales de sistemas multiagente que describen la segmentación del campo de estudio: i) el diseño de sistemas distribuidos o híbridos, ii) la formulación, simulación y resolución de problemas haciendo foco en unidades sociales, grupos y organizaciones, iii) el desarrollo de teorías socio-filosóficas, y iv) la comprensión de temas sociales y hechos sociales.

MAS hace énfasis en el comportamiento visible de los agentes, en el conocimiento que manejan, en los diferentes tipos de normas que regulan el accionar de los agentes, y en las agrupaciones de agentes que se comportan como unidades sociales de distintas envergaduras. Los agentes artificiales imitan (o intentan imitar) atributos humanos y capacidades humanas que, en el área, se describen con términos provenientes de las ciencias cognitivas: "pensar",

“adaptarse”, “aprender”, “argumentar”; ser “racional”, ser “emotivo”, o “rutinario”. Las estructuras de grupos de agentes y las relaciones entre agentes se describen usando terminología sociológica: “organización”, “comunidad”, “coalición”, “grupo”, “poder”, “solidaridad”, “normas”, “contratos”, “institución”, etc.

A partir de aquí usamos la sigla MAS no sólo para referirnos al área de estudio sino también como abreviatura de la expresión “sistema(s) multiagente”, cuando no hay confusiones.

**Descripción formal de sistemas multiagente (MAS).** La lógica modal es -por su flexibilidad y naturalidad en la escritura- una herramienta ampliamente aceptada para el diseño y el desarrollo de MAS. Con el fin de dar una definición de los estados mentales y cognitivos de los agentes, se formaliza con distintas lógicas modales especiales la postura del agente hacia su entorno: lo que el agente sabe, cuáles son sus creencias, cuáles son sus objetivos, cómo actúa, etc.

Los sistemas más conocidos e influyentes de este tipo son los llamados de *creencia-deseo-intención* BDI (por el inglés *belief-desire-intention systems*). Los agentes BDI se describen a través de: i) un estado “mental” dado en términos de creencias (beliefs) correspondientes a la información que el agente tiene sobre el entorno (que “cree” que sucede alrededor); ii) los deseos (desires), que son opciones que tiene el agente, y iii) las intenciones (intentions) que representan deseos elegidos por el agente (para ser cumplidos, o intentar ser cumplidos). Las creencias son vistas como “información” del agente. Los deseos e intenciones son vistos como actitudes *motivacionales*, como una inspiración para la actividad del agente.

Para representar cada uno de estos aspectos existen lógicas específicas de poder expresivo limitado, por ejemplo: una lógica de creencias, una lógica de intenciones, una lógica de objetivos, una lógica del actuar, etc. De manera análoga a la discutida en la presentación de la lógica deóntica, una descripción formal de estas lógicas específicas puede hacerse a través de una lógica proposicional extendida con una colección de operadores modales $\Box_x$ indexada por una colección A de agentes ($\Box_x$ / $x \in A$). Típicamente, para cada $x \in A$, $\Box_x$ funciona como un operador modal normal que satisface axiomas extra que capturan algún aspecto relevante del agente.

**Ejemplo. Lógica de creencias.** En la lógica de creencias escribimos $Bel_x\ A$ por $\Box_x A$. $Bel_x\ A$ es una modalidad epistémica (o “del conocimiento”). Es usada para representar “el agente *x* cree que *A*”, con *A* proposición.

El conjunto de creencias de un agente representa su “estado mental”. Para la lógica de creencias, requerimos: $Bel_x A \wedge Bel_x(A \rightarrow B) \rightarrow Bel_x B$ (distribución de creencias), $\neg Bel_x \bot$ (consistencia de creencias), $Bel_x A \rightarrow Bel_x(Bel_x\ A)$ (introspección positiva), $\neg Bel_x A \rightarrow Bel_x(\neg Bel_x\ A)$ (introspección negativa), y *de A se obtiene* $Bel_x A$ (regla de generalización para creencias). El axioma de consistencia de creencias nos asegura que el agente no cree en contradicciones, esto es, no cree en algo y en lo opuesto. El axioma de introspección positiva afirma que si el agente cree algo entonces cree en lo que cree, y el axioma de introspección negativa establece

que si un agente no cree en algo entonces cree en que no cree ese algo. Finalmente, la regla de generalización establece que el agente cree en algo si ese algo pudo probarse como cierto.

**Ejemplo. Lógica de objetivos.** En la lógica de creencias escribimos $\text{Goal}_x\ A$ por $\Box_x A$. La expresión $\text{Goal}_x\ A$ representa "el agente *x* tiene el objetivo *A*", con *A* proposición, que refleja algún estado particular de cosas (por ejemplo: "viajar") que el agente elige, que el agente quiere llevar a cabo. Para la lógica de objetivos requerimos: $\text{Goal}_x A \wedge \text{Goal}_x(A \rightarrow B) \rightarrow \text{Goal}_x B$ (distribución de objetivos) y *de A se obtiene* $\text{Goal}_x A$ (regla de generalización para objetivos).

**Ejemplo. Lógica de intenciones.** En la lógica de creencias escribimos $\text{Int}_x\ A$ por $\Box_x A$. La expresión $\text{Int}_x\ A$ significa "el agente *x* tiene la intención de que *A* sea verdadero", con *A* proposición. Las intenciones son vistas en MAS como inspiración para actividades. Para la lógica de intenciones requerimos: $\text{Int}_x\ A \wedge \text{Int}_x(A \rightarrow B) \rightarrow \text{Int}_x\ B$ (distribución de intenciones), $\neg\text{Int}_x\bot$ (consistencia de intenciones) y *de A se obtiene* $\text{Int}_x\ A$ (regla de generalización para intenciones). Las intenciones son objetivos seleccionados por el agente para intentar convertirse en verdaderos.

**Comentarios.** $\text{Bel}_x$, $\text{Int}_x$ y $\text{Goal}_x$ capturan la configuración *interna* de un agente. Notemos que estas tres lógicas -de creencias, de objetivos y de intenciones- tienen en su descripción, cada una, una instancia del esquema general de distribución $\Box A \wedge \Box(A \rightarrow B) \rightarrow \Box B$ que define cierto grado básico de "racionalidad" (notemos que el axioma de distribución guarda una estructura parecida a la de la regla *modus ponens*). La lógica de creencias y la de intenciones tienen ambas el axioma de consistencia, pero la de objetivos no. Esto se asemeja bastante a lo que nos sucede usualmente a los seres humanos, que podemos (y solemos) tener objetivos contradictorios; pero cuando elegimos objetivos para que sean nuestras intenciones e intentar concretarlas hacemos esa elección de intenciones de modo tal que no se contradigan entre sí.

Notemos que, tal como está presentada, la lógica de objetivos no es más que una colección de modalidades con semántica K básica.

En el área de MAS normalmente se asimila la noción de "lo que el agente cree" con aquella de "lo que el agente sabe", como un modo de establecer que el agente efectivamente cree en lo que sabe, esto es, que lo que cree y lo que sabe son lo mismo. Sin embargo, en otras ocasiones, cuando hay que distinguir con precisión entre lo que el agente sabe y lo que el agente cree, se usa la lógica llamada epistémica (vimos un ejemplo al principio de este capítulo) que usa la modalidad $K_x A$, para representar "el agente sabe *A*" (la K proviene del inglés *knows*).

A continuación presentamos el operador $\text{Does}_x$. A diferencia de las tres modalidades anteriores este operador indica el actuar visible, *externo*, de un agente:

**Ejemplo. Lógica de la acción.** $\text{Does}_x$ representa actividad exitosa del agente *x*. Su lectura intuitiva es: "el agente *x* lleva a cabo la acción *A*", con *A* proposición. La lógica del Does en su

definición axiomática tiene los siguientes esquemas: $\text{Does}_x A \rightarrow A$, $(\text{Does}_x A \wedge \text{Does}_x B) \rightarrow \text{Does}_x (A \wedge B)$, y $\neg\text{Does}_x \text{T}$ (con T abreviatura de *true*). El primer axioma establece efectividad en el actuar: si el agente x lleva a cabo la acción *A*, entonces *A* sucede. El segundo axioma, conocido como Axioma de Aglomeración, se refiere a la cotemporalidad implícita en la lógica modal proposicional: si el agente lleva a cabo la acción *A* y lleva a cabo la acción *B* entonces el agente lleva a acabo las dos acciones en el mismo tiempo (por ejemplo, hizo el backup del dispositivo y lo apagó). El tercer axioma de algún modo formaliza la idea de que un agente lleva a cabo acciones que son plausibles y, principalmente, evitables. La noción de acción es de algún modo un concepto de control: ningún agente lleva a cabo acciones inevitables, y las tautologías son inevitables (intuitivamente podemos asumir que las tautologías "se hacen solas" sin la intervención de ningún agente).

**Observación.** Si bien D. Elgesem acepta el Axioma de Aglomeración, no acepta el esquema converso, llamado M: $\text{Does}_x (A \wedge B) \rightarrow (\text{Does}_x A \wedge \text{Does}_x B)$. Esto porque, en presencia de sustitución uniforme por equivalentes lógicos, a partir de $\text{Does}_x A$ podemos conseguirnos la equivalencia $(\text{Does}_x A) \leftrightarrow (\text{Does}_x A \wedge (B \vee \neg B))$, y si aplicamos el axioma M obtenemos $\text{Does}_x (A \wedge (B \vee \neg B)) \rightarrow \text{Does}_x A \wedge \text{Does}_x (B \vee \neg B)$ y entonces conseguimos $\text{Does}_x \text{T}$ que es para Elgesem contraintuitivo en una lógica de la acción, contradice el axioma $\neg\text{Does}_x \text{T}$.

Además, M junto a Generalización y sustitución de equivalentes nos da una instancia de la paradoja de Ross tal como la vimos al estudiar lógica deóntica, esto es $\text{Does}_x A \rightarrow \text{Does}_x (A \vee B)$ y que es inaceptable en la intuición de una lógica de la acción. Dejamos al lector esta comprobación.

**Lógicas no normales.** En su definición de la lógica de la acción D. Elgesem explica que Does no puede ser un operador normal porque si lo fuera entonces su comportamiento no sería el que esperamos para representar acciones. Veamos: si la lógica del Does fuese normal entonces fácilmente la podríamos definir como una extensión de K agregándole a K el axioma de éxito $\text{Does}_x A \rightarrow A$ (es decir, agregándole a K el esquema modal llamado **T**) pues es en virtud de este único axioma que la lógica refleja lo que esperamos del actuar de un agente: éxito y control en el actuar. Entonces adoptaríamos el sistema normal **KT** para la lógica de la acción. Ahora bien, por ser normal entonces la lógica del Does verificaría la regla de generalización *si* $\vdash A$ *entonces* $\vdash \text{Does}_x A$ que nos lleva a derivar, dentro del sistema, la fórmula $\text{Does}_x \text{T}$ (con T abreviatura de *true*) que, hemos dicho, no queremos que sea verdadera pues las cosas que son factibles de ser hechas tienen que ser evitables y no tautologías. Con lo cual la regla de generalización no es deseable para definir una lógica de la acción. Además, si la lógica del Does fuese normal, verificaría también el axioma K de distribución: $\text{Does}_x (A \rightarrow B) \rightarrow (\text{Does}_x A \rightarrow \text{Does}_x B)$ lo que nos permitiría derivar, por ejemplo, proposiciones indeseadas para una lógica de la acción, tal como $\text{Does}_x A \rightarrow \text{Does}_x (A \vee B)$ (esto a partir del teorema $A \rightarrow (A$

$\vee B$), generalización, axioma K de distribución y *modus ponens*). Por todo esto es que Elgesem decide que la lógica del Does no puede tener una semántica modal normal.

**Definición 3.14. Lógica modal no-normal.** Una lógica modal es *no normal* cuando no satisface el axioma K de distribución (ver Sección 3.1, Definiciones 3.7b y 3.8).

Así las cosas, los modelos de Kripke no son suficientes para dar una semántica de lógicas no-normales. Tenemos entonces una semántica diferente, llamada *de tipo Scott-Montague*. La intuición detrás de esta semántica es la siguiente: en lugar de tener una relación entre mundos, tenemos un conjunto de colecciones de mundos conectados a *w*. Esas colecciones se llaman *neighbourhoods* o vecindarios de *w*.

Formalmente: un frame de Scott-Montague es un par ordenado <W,N> donde W es un conjunto (de mundos, puntos, situaciones, etc) y N es una función que asigna a cada elemento $w \in W$ un conjunto de subconjuntos de W (los *neighbourhoods* de w). Un modelo de Scott-Montague es una terna <W,N,V> donde <W,N> es un frame de Scott-Montague y V es una función de valuación como en lo modelos de Kripke, y la definición de verdad en *w* es: $\Box A$ es verdadera en *w* si y sólo si los elementos de W donde *A* es verdadera es uno de los conjuntos en N(*w*), esto es, un neighbourhood de *w*.

Esta es una generalización de la semántica tradicional de Kripke. Es fácil ver que un frame de Kripke es equivalente a un frame de Scott-Montague con el mismo W y donde N(*w*) se define como $\{v / wRv\}$. Debe quedar claro que, por el contrario, hay frames de Scott-Montague que no se corresponden con frames de Kripke. Además, lamentablemente, hemos perdido el paralelismo entre $\Box$ y $\Diamond$ y los cuantificadores universal y existencial que sí hay en los frames de Kripke.

**Lógicas de propósitos especiales. Comentarios.** En los sistemas BDI la actividad de un agente comienza a partir de objetivos. Un agente tiene en general muchos objetivos, muchos de los cuales no serán perseguidos, no estarán relacionados con acciones. Esto permite que un agente pueda comportarse consistentemente aún cuando tenga objetivos inconsistentes. Un agente elige un número finito de sus objetivos para que sean sus intenciones, es decir, sus motivaciones para actuar. No nos resulta relevante cómo una intención se forma a partir de un conjunto de objetivos, solo nos concentramos en el hecho de que los objetivos son elegidos por el agente de modo tal que se preserve consistencia (ver el axioma de consistencia de intenciones). Un problema con las lógicas modales estándar para creencias (y conocimiento) es que los agentes son formalizados como *omniscientes*: creen en todos los teoremas así como en las consecuencias lógicas de sus creencias. Cualquier lógica modal estándar con semántica de Kripke en la que se modela creencia como un operador de necesidad tendrá esta propiedad. El problema aquí es que la omnisciencia no se aplica a los humanos, que tenemos normalmente poco tiempo disponible y racionalidad limitada, es irreal asumir que creemos en cada teorema (los hay muy complicados!). Finalmente, la lógica del Does no satisface el converso del axioma de aglomeración. Si lo hiciese, en presencia del Axioma de Aglomeración

y de la regla de sustitución uniforme la lógica de la acción se volvería inconsistente. Dejamos al lector la comprobación de ello.

**Discusión.** Dejamos al lector la lectura intuitiva de cada uno de los axiomas de las lógicas descriptas para creencias, objetivos, intenciones y acción.

**Creación de sistemas multimodales multiagente BDI.** El mecanismo es, desde el punto de vista ingenieril, el siguiente: se seleccionan diferentes lógicas modales específicas, también llamadas de *propósitos especiales* o *de propósito determinado*, que son -casi siempre- monomodales, es decir, con un único operador modal, con o sin su dual. Se las *combina* de algún modo y se obtiene lo que se llama una *lógica multimodal resultante de la combinación*.

Normalmente se unen lógicas específicas pues tendría poco sentido poner a trabajar juntas lógicas que tengan poder expresivo general.

**Combinación de lógicas.** Combinar lógicas es una técnica que está actualmente en estudio y expansión, inspirada principalmente en el interés por la modularidad. Permite definir sistemas formales altamente especializados. ¿Ensamblar lógicas nos ofrece algo nuevo? La respuesta es sí: no hay un estudio sistematizado de cómo combinar lógicas, tampoco hay un cuerpo establecido de resultados. Lo que sí existe es un núcleo de nociones y combinaciones exitosas que han surgido para una clase importante de lógicas.

Por un lado, como intuimos, existe un aspecto "ingenieril" o de diseño que nos lleva –como informáticos- a considerar a las lógicas pequeñas como bloques o unidades de manejo de conocimiento con los que podemos construir sistemas más grandes: podemos reutilizar los bloques, montar bloques unos con otros, sustituir un bloque por otro con iguales o mejores prestaciones. Por otro lado, debemos prestar atención al aspecto "lógico" de la combinación o montaje de bloques lógicos: como lo que estamos combinando son lógicas que poseen determinadas propiedades que seguramente consideramos ventajosas por algún motivo (como la decidibilidad, por ejemplo) pretendemos que las lógicas resultantes conserven las buenas propiedades de sus bloques componentes.

Los lógicos y los lógicos computacionales que se dedican a estos temas llaman a dicha cuestión, dice C. Areces, *el problema de transferencia*: sean $\Lambda_1$ y $\Lambda_2$ dos lógicas y sea P una propiedad que las lógicas puedan tener (como decidibilidad, f.m.p., alguna cota de complejidad, etc). Si $\oplus$ es un modo de combinar $\Lambda_1$ y $\Lambda_2$, ¿posee $\Lambda_1 \oplus \Lambda_2$ la propiedad P? Es importante resolver algunos problemas de transferencia al proponer una lógica combinada. Un primer principio relativo a transferencia parece indicar que si no hay interacción entre las lógicas (es decir, las lógicas no comparten símbolos excepto conectivos booleanos), la propiedad en cuestión se preserva. Pero aún en formas leves de interacción la transferencia de propiedades puede fallar.

A continuación presentamos sintaxis y semántica para un MAS diseñado como una combinación de lógicas de propósito determinado: ponemos a trabajar juntas las lógicas específicas que hemos presentado para creencias, objetivos, intención y acción. Luego de

describir la combinación comentamos sobre la transferencia de propiedades en la lógica resultante.

**Lenguaje de un sistema multiagente BDI.** Al lenguaje modal básico de la Sección 3.1 le agregamos un conjunto finito de agentes A = {*x*, *y*, *z*...}. Las expresiones complejas son construidas del modo inductivo usual con los operadores lógicos clásicos y con los operadores unarios modales $Bel_x$, $Int_x$, $Goal_x$ y $Does_x$.

**Axiomas "puente".** Existen relaciones entre creencias, objetivos e intenciones de agentes, que describimos axiomáticamente como: $Goal_x\ A \rightarrow Bel_x(Goal_x\ A)$ (introspección positiva de objetivos), $Int_x A \rightarrow Bel_x(Int_x\ A)$ (introspección positiva de intenciones), $\neg Goal_x A \rightarrow Bel_x(\neg Goal_x\ A)$ (introspección negativa de objetivos), $\neg Int_x A \rightarrow Bel_x(\neg Int_x\ A)$ (introspección negativa de objetivos), y $Int_x\ A \rightarrow Goal_x\ A$ (intención implica objetivo).

Los axiomas que expresan interdependencias entre creencias y las actitudes motivacionales (objetivos e intenciones) permiten ver que los agentes son conscientes de los objetivos e intenciones que tienen, así como de los que no tienen. Así como los hemos definido, los axiomas para actitudes motivacionales y sus aspectos combinados son mínimos en el sentido de que pretendemos manejarnos con condiciones necesarias y suficientes, tal como los definen B. Dunnin-Keplickz y R. Verbrugge. Notemos que no hay axiomas de "realismo fuerte" como puede considerarse a los axiomas $Goal_x\ A \rightarrow Bel_x\ A$ y $Int_x\ A \rightarrow Bel_x\ A$ que corresponderían, por ejemplo, a las ideas de que un agente cree que puede alcanzar sus objetivos e intenciones mediante la elección cuidadosa de sus acciones.

Tengamos en cuenta, de todos modos, que aspectos adicionales de creencias, deseos e intenciones siempre pueden modelarse agregando nuevos axiomas y creando extensiones más refinadas de estas lógicas mínimas.

**Expresividad del sistema. Restricción.** Solamente a los efectos de facilitar la presentación técnica del MAS como una combinación de lógicas establecemos la siguiente restricción: una fórmula de la forma $Does_x\ A$ siempre es aplicada a átomos que representan acciones simples (por ejemplo "comprar", "vender", "contestar"). Ejemplo: $Bel_x$ ($Does_y$ Pagar) intuitivamente se lee "el agente x cree que el agente y paga". Con esta restricción los operadores modales normales interactúan con el operador Does de una manera limitada: no es posible escribir fórmulas como $Does_x$ ($Does_y$ Pagar) o como $Does_x$ ($Goal_y\ A$) (que puede ser vista como una forma de persuasión: "el agente x hace que el agente y tenga a *A* como objetivo").
Esta restricción no impide que, en otras configuraciones diferentes para otros MAS, algunos operadores puedan aparecer dentro del alcance de un Does.

**Semántica del sistema multiagente.** La estructura del sistema multiagente es una extensión de la definición de frame dada en la Sección 3.1, como sigue:

$$F = \langle A, W, \{B_i\}_{i \in A}, \{G_i\}_{i \in A}, \{I_i\}_{i \in A}, \{D_i\}_{i \in A} \rangle ,$$

donde:

· A es un conjunto finito de agentes.

· W es un conjunto de situaciones, o mundos.

· $\{B_i\}_{i \in A}$ es un conjunto de relaciones de accesibilidad para los operadores de creencias $Bel_x$ (hay un operador para cada agente x, por lo tanto tenemos una relación de accesibilidad para cada uno de esos operadores). Las relaciones de accesibilidad $B_i$ son transitivas (cumplen $\forall xyz\ (Rxy \wedge Ryz) \rightarrow Rxz$), euclideanas (cumplen $\forall xyz\ (Rxy \wedge Rxz) \rightarrow Ryz$) y seriales (sin límite a derecha, como las deónticas). Los frames con estas características son precisamente los que quedan deteminados por los axiomas que definen a la lógica de las creencias (por ello a la lógica de creencias se la llama de tipo **KD45**).

· $\{G_i\}_{i \in A}$ es el conjunto de relaciones de accesibilidad para cada uno de los operadores de objetivos $Goal_x$, cuya semántica es de necesidad estándar, **K** (Definición 3.8, Sección 3.1).

· $\{I_i\}_{i \in A}$ es el conjunto de relaciones de accesibilidad respecto de los operadores de intenciones $Int_x$, relaciones que son seriales (la lógica de intenciones es **KD**, como la deóntica).

· $\{D_i\}_{i \in A}$ es una familia de conjuntos de relaciones de accesibilidad para los operadores Does, relaciones que son reflexivas, seriales, y satisfacen ciertas condiciones especiales de clausura (ver *On the Axiomatisation of Elgesem´s Logic for Agency and Ability*, de G. Governatori y A. Rotolo).

Finalmente, definimos un *modelo* para nuestros sistemas multiagente como una estructura de la forma $M = \langle F, V \rangle$ en el que:

· *F* es un frame como definimos más arriba, y

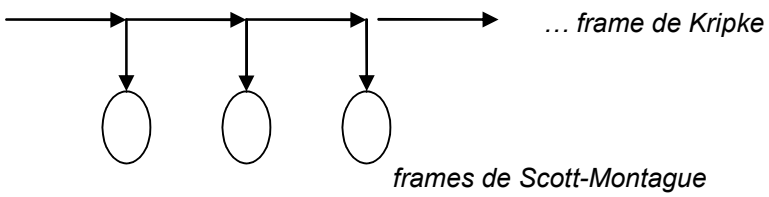
· V es una función de valuación definida como sigue:

1. condiciones booleanas estándar,
2. $V(w, Bel_i\ A) = true$ sí y sólo sí $\forall v$ (si $B_i wv$ entonces $V(v, A) = true$),
3. $V(w, Goal_i\ A) = true$ sii $\forall v$ (si $G_i wv$ entonces $V(v, A) = true$),
4. $V(w, Int_i\ A) = true$ sii $\forall v$ (si $I_i wv$ entonces $V(v,A) = true$),
5. $V(w, Does_i\ A) = true$ sii $\exists\ D_i \in D_i$ tal que $\forall v\ (D_i wv$ sii $V(v, A) = true)$;

V está definida como para los modelos de Kripke excepto para los operadores Does: la fórmula $Does_i\ A$ es verdadera en *w* si y sólo si existe un vecindario $D_i$ de *w,* $D_i \in D_i$ (con $D_i$ conjunto de todos los vecindarios de *w*) en el que la fórmula *A* es verdadera.

**Evaluación de fórmulas. Navegación dentro del frame.** Notemos que es posible identificar dos “redes” de relaciones sobre W. La primera “red”, tal como está definido *F*, corresponde al “cableado” de los operadores normales. La segunda red corresponde a las relaciones de

accesibilidad para las modalidades Does. Podemos representar gráficamente a *F* como si hubiese un frame de Kripke "exterior", y frames de Scott-Montague "interiores":




Viéndolo de este modo, la intuición detrás de la valuación de las fórmulas en el sistema multiagente es la siguiente: cuando *parseamos* una fórmula –esto es, cuando la recorremos para evaluarla- navegamos por el modelo de Kripke "exterior" hasta que una subfórmula que comienza con el operador Does aparece para ser evaluada. Ahí nos "metemos" en un modelo de Scott-Montague. Evaluamos la subfórmula Does en el modelo de Scott-Montague y sustituimos en la fórmula original la subfórmula del Does con el resultado de esta evaluación (la sustitución la haremos con algún objeto que pertenezca al dominio de los modelos de Kripke tal como una variable proposicional o un valor de verdad) y continuamos con la evaluación que, de algún modo, en este punto, ha sido "homogeneizada".

**Fibrado.** El modo en el que hemos reorganizado la vista gráfica de nuestro MAS como un frame de Kripke "exterior" y frames de Scott-Montague "interiores" se corresponde con la técnica de combinación de lógicas llamada *fibrado*. La estructura fibrada resultante se define como una función fibrada $f_{\Lambda 1,\Lambda 2}$, una función total que asigna a cada elemento $w \in M_{\Lambda 1}$ un modelo $M_{\Lambda 2}$, con $M_{\Lambda 1}$ y $M_{\Lambda 2}$ modelos de las lógicas $\Lambda_1$ y $\Lambda_2$ respectivamente (la de Kripke y la de Scott-Montague). Así, una expresión del lenguaje $L_2$ de la lógica $\Lambda_2$ es evaluada en $M_{\Lambda 1}$ –donde es indefinida- a través de $f_{\Lambda 1,\Lambda 2}(w)$.

La restricción de expresividad que impusimos sobre el uso del Does (C. Smith y A. Rotolo) en este MAS favorece el fibrado presentado ya que las lógicas fueron puestas a trabajar de una manera simple y de modo tal que, luego, el algoritmo de evaluación de fórmulas no resulta complejo: se trabaja en un modelo para lógicas normales, cuando se encuentra un Does se evalúa la subfórmula en un modelo no normal. Hemos dicho que es posible definir otros MAS donde los operadores modales sí pueden aparecer dentro del Does, por ejemplo donde sea posible escribir y evaluar fórmulas del tipo $Does_x(Goal_y A)$. En ellos se aplican otras técnicas de combinación de lógicas y otros algoritmos de evaluación de fórmulas.

**Completitud.** La lógica resultante de la combinación de lógicas específicas tal como fue presentada es completa. Para ello es suficiente con construir un modelo canónico para dicha combinación y establecer que la lógica es fuertemente completa con respecto al modelo canónico. La prueba de completitud para la porción no normal de la lógica es, como imaginamos, intrincada, una muy clara exposición de la prueba de completitud de una lógica no normal Does aparece en la tesis de grado de F. Carbonari, publicada por la Universidad de La Plata.

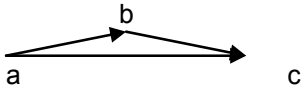
**Decidibilidad.** Si las lógicas componentes son decidibles, es posible que la combinación resultante también lo sea. La lógica del Does posee la f.m.p., y las lógicas monomodales que usamos también. La lógica resultante de la combinación de las lógicas también es decidible. Dejamos al lector el armado de la prueba, que es larga pero no demasiado compleja.

**Complejidad.** La lógica combinada presentada, a pesar de ser decidible, es EXPTIME completa (su problema de decisión tiene tiempo de ejecución exponencial, y otros problemas en la misma clase de problemas pueden reducirse a aquél). Existen algunas técnicas para reducir esta complejidad, como por ejemplo limitar la profundidad de fórmulas modales, poniendo un límite a la cantidad de operadores modales que pueden aparecer en las subfórmulas de una fórmula. M. Dziubinski, R. Verbrugge y B. Dunnin-Keplicz han hecho algunas propuestas para el tratamiento de complejidad en MAS.

**Extensiones para la lógica combinada.** Podemos extender sin mayores conflictos el MAS que hemos presentado con el operador deóntico de obligaciones O. También podríamos hacerlo con una lógica temporal. Por ejemplo, temporalizar la lógica presentada se reduce simplemente a montar sobre el MAS la maquinaria temporal con el mismo espíritu con el que visualizamos la maquinaria normal organizada sobre la no normal. Este fibrado ha sido descripto en un trabajo conjunto con A. Rotolo, A. Ambrosio y L. Mendoza, lo puntualizamos brevemente a continuación. Consideremos el modelo $(T,<,g,t_0)$, entonces tenemos un frame "externo" $(T,<)$ que se corresponde con la línea de evolución temporal, y $t_0$ es el instante inicial de tiempo. El sistema evoluciona en el sentido de que nuevos grupos, creencias, relaciones y obligaciones se van creando y también desarmando a lo largo del tiempo. Así, *g* es la función total que para cada punto $t_i$ de la línea temporal "trae" un modelo *M* (como el definido más arriba) para evaluar.

## Ejercicios para el Capítulo 3

1. Sea el siguiente frame:




Se sabe que la fórmula p es verdadera en el mundo b y que la fórmula q es verdadera en los mundos a y c. Demostrar si:

i- $\Box p$ es verdadera en a.

ii- $\Diamond p$ es verdadera en a.

iii- $p \vee \Box q$ es verdadera en b.

iv- $\Box q \rightarrow \Diamond p$ es verdadera en b.

2- Probar que la fórmula $\Box p \rightarrow \Diamond p$ de la lógica modal no es válida en la clase de todos los frames.

3- Probar que la fórmula $(\Box p \wedge \Box q) \rightarrow \Box(p \wedge q)$ es válida en la clase de todos los frames.

4- Simbolizar utilizando los operadores deónticos O, P, F de obligación, permiso y prohibición según convenga. Algunas de las sentencias no tienen carácter deóntico, indicar cuáles.

i- El lector devolverá el libro en 15 días hábiles. Si el lector devuelve el libro en 15 días hábiles, no se le aplicará el apercibimiento administrativo del artículo 20. Si el lector no devuelve el libro en 15 días hábiles, se le aplicará el apercibimiento administrativo del artículo 20.

ii- Un círculo no puede ser cuadrado.

iii- Juan promete pagarle a Pedro $50.

iv- Juan firma: “Prometo pagarle a Pedro $50”.

v- Juan se pone así mismo bajo la obligación de pagarle a Pedro $50.

vi- Juan está obligado.

vii- Juan debe pagar a Pedro $50.

viii- Fumar es perjudicial para la salud.

ix- Prohibido fumar.

x- Puede besar a la novia.

xi- Es obligatorio que lleves esta carta al correo. Por lo tanto, es obligatorio que o lleves esta carta al correo o la quemes.

xii- Tienes permitido o bien llevar la carta al correo o bien quemarla.

xiii- Debe haber paz en el mundo.

xiv- Está prohibido matar. Por lo tanto, están prohibidos ambos matar y arrepentirse.

5- a- Suponer que la expresión $\Diamond p$ significa “p es tolerable”.

i- ¿Cuál es la lectura intuitiva del dual $\Box$ si definimos $\Box p \equiv \sim\Diamond\sim p$?

ii- Se quiere formalizar un sistema moral (social/religioso/mafioso/de etiqueta y ceremonial, etc.) que captura esta interpretación de $\Diamond$ y $\Box$. Listar fórmulas que puedan considerarse principios lógicos para el sistema. Por ejemplo: $\Diamond p \vee \Diamond q \rightarrow \Diamond(p \vee q)$.

iii- Simbolizar la proposición “si algo sucede, entonces es tolerable”. La incluiríamos como principio lógico en la lista previa? Fundamentar.

iv- ¿Incluiríamos la fórmula $\Box(\Box p \rightarrow p) \rightarrow \Box p$ en la lista de principios? Fundamentar. ¿Cuál es su lectura intuitiva?

v- ¿Y $p \rightarrow \Box\Diamond p$ ? Fundamentar. ¿Cuál es su lectura intuitiva?

vi- ¿Cómo se modela en este sistema el comportamiento altruista?

b- Se quiere formalizar lógicamente la coexistencia de un sistema de la tolerancia caracterizado como en el inciso previo con los operadores $\Box$ y $\Diamond$ y un sistema de normas jurídicas

caracterizado por los operadores O, P, y F, donde O tiene una semántica de necesidad, P de posibilidad y Fp ≡ ~Pp. Simbolizar las proposiciones a continuación, y determinar (fundadamente) para cada una de ellas si constituyen o no principios lógicos de esta coexistencia de sistemas normativos.

i- Lo tolerable está permitido. (Pensar en el robo de la señal de cable)
ii- Si algo está permitido, es tolerable. (Pensar en sistemas morales que, por ej., legitiman el aborto, ¿qué pasa con los sectores que están bajo esas normas pero en contra de esa práctica?)
iii- Si algo está permitido, entonces es obligatorio tolerarlo. (Pensar en el derecho a huelga, manifestaciones, etc.)
iv- Si algo está prohibido, es obligatorio no tolerarlo. (ídem anterior, pensar)

c- ¿Cómo se enuncian en lenguaje natural las siguientes proposiciones de la lógica combinada de normas/tolerancia del ítem previo? ¿Constituyen principios de la coexistencia de ambas lógicas? Fundamentar.

i- $Op \rightarrow \Diamond p$.

ii- $\Box p \rightarrow \Box Op$.

iii- $Op \rightarrow O(\Diamond Op)$.

iv- $\sim(Pp \rightarrow \Box p)$.

6- Formalizar las siguientes reglas de comportamiento con los operadores de obligación, permiso y prohibición O, P, y F.

<u>De las conductas indecorosas en la mesa de mi señor</u>:

*(Texto anónimo, aunque atribuido a Leonardo Da Vinci, quien trabajó para los Médici, ca.1600)*

Estos son (algunos de) los hábitos indecorosos que invitados a la mesa de mi señor no deben cultivar (y baso esto en mi observación de aquellos que frecuentaron la mesa de mi señor durante el año pasado).

Ningún invitado ha de sentarse sobre la mesa, ni de espaldas a la mesa, ni sobre el regazo de cualquier otro invitado.

Tampoco ha de poner la pierna sobre la mesa.

Tampoco ha de sentarse bajo la mesa en ningún momento.

No ha de limpiar su armadura sobre la mesa.

No ha de tomar comida de la mesa y ponerla en su bolso o faltriquera para después comerla.

No ha de hacer figuras modeladas ni prender fuegos ni adiestrarse en hacer ruidos en la mesa (a menos que mi señor se lo pida).

No ha de tocar el laúd o cualquier otro instrumento que pueda ir en perjuicio de su vecino de mesa (a menos que mi señor se lo pida).

No ha de cantar, ni hacer discursos, ni vociferar improperios ni tampoco proponer acertijos obscenos si está sentado frente a una dama.

No ha de conspirar en la mesa (a menos que lo haga con mi señor).

Tampoco ha de prender fuego a su compañero mientras permanezca en la mesa.

No ha de golpear a los sirvientes (a menos que sea en defensa propia).

7- Algunas de las reglas de decoro en la mesa previas tienen contenido temporal. Identificarlas y modelarlas en el contexto de una lógica deóntica que se ha combinado con una temporal que tiene los operadores *P* y *F* para simbolizar “en el pasado” y “en el futuro”.

8- Manejamos un lenguaje modal proposicional fundado sobre un conjunto finito A de agentes y un conjunto numerable de proposiciones, denotadas con p, q, r,… Expresiones complejas se forman sintácticamente a partir de ellas, en el modo inductivo usual, usando un operador $\perp$, el operador binario $\vee$, y modalidades unarias O y $\text{Does}_x$ (donde el subíndice corre sobre el conjunto de agentes). Como el comportamiento proposicional de esta lógica es clásico, asumimos que T, $\vee$, $\rightarrow$ se definen del modo usual. El operador Does debe entenderse para representar éxito en el actuar.
En esta lógica combinada, formalizar algunas reglas de decoro en la mesa del Ejercicio 6.

7- Se tiene una lógica multiagente que provee el operador Does. Para las siguientes fórmulas, dados dos agentes x e y cualesquiera, dar su lectura intuitiva:
i- $\text{Does}_x\, A \rightarrow (\text{Does}_x(\text{Does}_x\, A))$.
ii- $(\text{Does}_y(\text{Does}_x\, A)) \rightarrow \text{Does}_y\, A$.

Indicar si esta última fórmula puede ser considerada un principio axiomático de una lógica de la acción. Fundamentar.

8- La noción de abstención dice que un agente se abstiene de hacer algo sí y solo si puede hacerlo pero no lo hace. Definir la noción de abstención usando una lógica multiagente con los operadores deónticos usuales combinados con el operador de la acción $\text{Does}_x\, A$.

9- Estudiar el impacto de los teoremas O**T** y $\text{Does}_x$**T** en las semánticas pretendidas para las lógicas deóntica y de la acción, respectivamente. Comparar los esquemas de axioma $\neg$OT de la lógica deóntica (que von Wright acepta en su sistema original) y $\neg\text{Does}_x$ T de la lógica de la acción.

10- Probar que la fórmula de la lógica epistémica $(\neg \text{K}A \rightarrow \neg \text{K}B) \rightarrow (\text{K}B \rightarrow \text{K}A)$ es una verdad lógica más allá de su contenido epistémico. Ofrecer su lectura intuitiva en lenguaje natural.

# Referencias

Areces, C., Monz, C., de Nivelle, H., y de Rijke, M. *The Guarded Fragment: Ins and Outs*. En J. Gerbrandy, M. Marx, M. de Rijke, and Y. Venema, editors, Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday, Vossiuspers, AUP, Amsterdam, 1999. Recuperado de http://www.loria.fr/~areces/content/papers/files/j50b.pdf

Blackburn, P., De Rijke, M., Venema Y. (2001). Modal Logic. Cambridge University Press.

Carbonari, F. Pruebas interesantes para una lógica multi-modal multi-agente: Un caso de estudio sobre Completitud. Tesis de grado, Facultad de Informática UNLP, noviembre de 2015.

Chellas, B. F. *Modal Logic. An Introduction.* Cambridge University Press, 1980.

Dunnin-Keplickz, B., y Verbrugge, R. *Collective Intentions*. Fundamenta Informaticae. Recuperado de http://rinekeverbrugge.nl/PDF/Articles%20in%20refereed%20journal/Collectiveintentions-FI02.pdf

Dziubinski, M., Verbrugge, R., Dunnin-Keplicz, B. *Complexity issues in multiagent logics*. Fundamenta Informaticae, 75 (1-4):239-262, 2007.

Elgesem, D. *The Modal Logic of Agency*. Nordic Journal of Philosophical Logic. Vol 2 (2), 1-46. 1997. Scandinavian University Press. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.2796&rep=rep1&type=pdf

Governatori, G. y Rotolo, A. *On the Axiomatisation of Elgesem´s Logic for Agency and Ability*. En Journal of Philosophical Logic. August 2005, Volume 34, Issue 4, pp 403-431. Recuprado de http://web.stanford.edu/class/cs222/Governatori.pdf

Hamilton, A. G. Lógica para Matemáticos. (Traducción de M. R. Artalejo). Paraninfo, Madrid, 1981.

Hansson, B. y Gardenfors, P. *A Guide to Intensional Semantics*. En Modality, Morality and Other Problems of Sense and Nonsense. Essays dedicated to Sören Hallden. CWK Gleerup, Lund, pp. 151-167, 1973.

Herrestad, H. y Krogh, C. *Deontic Logic Relativised to Bearers and Counterparties*. En Bing, J. y Torvund, O. (editores) Anniversary Anthology in Computers and Law; COMPLEX – TANO, 1995, pp 453-522.

Jones, A. J. I. *Deontic logic and legal knowledge representation*. En Ratio Juris, 3:237-244, 1990.

Meyer, J.-J. Ch., Wieringa, R.J., Dignum, F.P.M. *The Role of Deontic Logic in the Specification of Information Systems*. En Logic for Databases and Iinformation Systems. pp 71-115, Kluwer Academic Publishers Norwell, MA, USA, 1998. Recuperado de http://doc.utwente.nl/18384/1/role-deontic_meyer.pdf

Prolog. Lenguaje y ambiente de programación en versión para educación e investigación. Recuperado de http://www.swi-prolog.org/

Smith, C., Ambrossio, A., Mendoza, L., Rotolo, A. *Combinations of Normal and Non-normal Modal Logics for Modeling Collective Trust in Normative MAS*. En AI Approaches to the Complexity of Legal Systems. M. Palmirani (et al.) editores. LNAI 7639, pp. 189-203, Springer, 2012.

Smith, C. y Rotolo, A. *Collective Trust and Normative Agents*. Logic Journal of the IGPL (2010) 18 (1): 195-213. Springer.

von Wright, G. H. (1951). *Deontic Logic*. Mind, LX (237), 1-15.

Wieringa, R. J., J-J. Ch. Meyer. *Applications of Deontic Logic in Computer Science: A Concise Overview*. En Deontic Logic in Computer Science, J-J. Ch. Meyer, R. J. Wieringa (editores). pp 17-40, John Wiley & Sons, Inc. New York, NY, USA, 1994. Recuperado de http://eprints.eemcs.utwente.nl/10663/01/applications-of-deontic-logic.pdf