Lógicas no clásicas:
Introducción a la lógica modal

Pedro Arturo Góngora Luna
pedro.gongora@gmail.com

Índice

1. Repaso

1.1. Relaciones binarias

Recordemos que el producto cartesiano de dos conjuntos A y B es el conjunto de todas las parejas cuyo primer elemento está en A y el segundo está en B, es decir,

$$A \times B \stackrel{\text{def}}{=} \{(a,b) \mid a \in A \text{ y } b \in B\}$$

Para el producto cartesiano aplicado varias veces a un mismo conjunto escribimos:

$$A^n \stackrel{\text{def}}{=} \{\underbrace{A \times \cdots \times A}_{n \text{ veces}}\}$$

Definición 1.1 (Relaciones y relaciones n-arias sobre un conjunto)**.** *Una relación* es un subconjunto de un producto cartesiano. Para cualquier conjunto A, decimos que *una relación* $R \subseteq A^n$ *es* n*-aria y sobre* A. ∎

Definición 1.2 (Tipos de relaciones binarias)**.** Dados un conjunto A y una relación binaria R sobre A, decimos que R *es:*

(a) *reflexiva* si para todo $a \in A$, $(a, a) \in R$

(b) *irreflexiva* si para todo $a \in A$, $(a, a) \notin R$

(c) *simétrica* si para todo $a, b \in A$, $(a, b) \in R$ implica que $(b, a) \in R$

(d) *asimétrica* si para todo $a, b \in A$, $(a, b) \in R$ implica que $(b, a) \notin R$

(e) *antisimétrica* si para todo $a, b \in A$, $(a, b) \in R$ y $(b, a \in R)$ implica que $a = b$

(f) *transitiva* si para todo $a, b, c \in A$, si $(a, b) \in R$ y $(b, c) \in R$, entonces $(a, c) \in R$

(g) *serial* si para todo $a \in A$, existe $b \in A$ tal que $(a, b) \in R$

(h) *euclideana* si para todo $a, b, c \in A$, si $(a, b) \in R$ y $(a, c) \in R$, entonces $(b, c) \in R$

(i) *de equivalencia* si R es reflexiva, transitiva y simétrica

(j) *total* si para todo $a, b \in A$, $(a, b) \in A$ ó $(b, a) \in A$

(k) *bien fundada* si todo subconjunto $X \subseteq A \neq \emptyset$ tiene al menos un minimal[1] ∎

1.2. Relaciones binarias como gráficas

Dados un conjunto A y una relación binaria R sobre A, muchas veces encontraremos útil darle una representación gráfica. Para esto, el proceso es muy sencillo:

- los nodos de la gráfica son los elementos de A y,
- dibujamos un arco $a \to b$ sii la pareja (a, b) está en R.

Por ejemplo, definidos A y R como sigue:

$$A \stackrel{\text{def}}{=} \{a, b, c, d\}$$
$$R \stackrel{\text{def}}{=} \{(a, b), (a, c), (b, d), (d, d)\}$$

[1]Un minimal de un conjunto A, de acuerdo con una relación $R \subseteq A^2$, es un elemento $m \in A$ tal que no existe otro $a \in A$ tal que $(a, m) \in R$.

creamos su representación gráfica:

$$\begin{tikzcd}
a \arrow[r] \arrow[d] & b \arrow[d] \\
c & d \arrow[loop right]
\end{tikzcd}$$

Si tenemos una relación *simétrica* podemos omitir la dirección de los arcos (pues sabemos que llevan implícitamente ambas direcciones). Por ejemplo, para una R' simétrica:

$$R' \stackrel{\text{def}}{=} \{(a,b),(b,a),(a,c),(c,a),(b,d),(d,b)\}$$

tenemos la siguiente gráfica:

$$\begin{tikzcd}
a \arrow[r, dash] \arrow[d, dash] & b \arrow[d, dash] \\
c & d
\end{tikzcd}$$

1.3. Composición relacional

Definición 1.3 (Composición relacional y cerraduras)**.** Sean R y S relaciones binarias sobre un conjunto A. Definimos *la composición de R con S* como sigue:

$$R \circ S \stackrel{\text{def}}{=} \{(a,b) \mid \text{existe } x \in A \text{ tal que } (a,x) \in R \text{ y } (x,b) \in S\}$$

∎

Definición 1.4 (Relación identidad y composición aplicada n veces)**.** Dado un conjunto A y una relación binaria R sobre A. Definimos la *relación identidad de A, Id_A*, de la siguiente forma:

$$Id_A \stackrel{\text{def}}{=} \{(a,a) \mid a \in A\}$$

También, definimos *la relación R^n* como sigue:

$$R^0 \stackrel{\text{def}}{=} Id_A$$
$$R^{i+1} \stackrel{\text{def}}{=} R \circ R^i$$

para cualquier $i \in \mathbb{N}$. ∎

Es fácil verificar que la relación Id_A funciona como la identidad de la composición relacional, esto es, para toda R sobre A, $R \circ Id_A = R$. También es importante aclarar que no debemos confundir la notación de la definición anterior con la del producto cartesiano.

Podemos cerrar una relación bajo composición, esto nos será útil para construir relaciones nuevas a partir de otras ya existentes.

Definición 1.5 (Cerraduras transitiva, y reflexiva-transitiva)**.** Sea una relación binaria R sobre un conjunto A. Definimos *la cerradura transitiva de* R como:

$$R^+ \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N} - \{0\}} R^n$$

Definimos *la cerradura reflexiva-transitiva de* R como:

$$R^* \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N}} R^n$$

∎

1.4. Conjuntos inductivos y sintaxis de lenguajes

Para definir *formalmente* la sintaxis de un lenguaje usaremos definiciones inductivas. Hay que recordar que un lenguaje no es más que un conjunto de cadenas de símbolos, por lo que sólo necesitamos un método para construir las cadenas de símbolos que sí pertenecen al lenguaje (o para decidir qué cadenas pertenecen y qué cadenas no).

Para construir un conjunto inductivamente necesitamos los siguiente:

- Un conjunto con los símbolos básicos o *atómicos* del lenguaje.
- Una colección de funciones que combinen elementos existentes del lenguaje y para crear otros nuevos. A estas funciones le llamamos *constructores del lenguaje*.
- Definimos el lenguaje cerrando el conjunto base bajo los constructores del lenguaje.

Ejemplo 1.6 (Expresiones $+$ y $\times$)**.** Sean el conjunto base:

$$B \stackrel{\text{def}}{=} \{0, 1, 2, \ldots\}$$

y los constructores:

$$f_1(\alpha, \beta) \stackrel{\text{def}}{=} \alpha + \beta$$
$$f_2(\alpha, \beta) \stackrel{\text{def}}{=} \alpha \times \beta$$

Definimos el conjunto Exp de las expresiones de suma y multiplicación como el conjunto tal que:

(a) Si $n \in B$, entonces $n \in Exp$;

(b) Si $\alpha, \beta \in Exp$, entonces $f_1(\alpha, \beta) = \alpha + \beta \in Exp$;

(c) Si $\alpha, \beta \in Exp$, entonces $f_2(\alpha, \beta) = \alpha \times \beta \in Exp$;

(d) (Cierre) Sólo están en Exp los elementos que pueden construirse por un número finito de aplicaciones de las clausulas (a) - (c).

La clausula de cierre es muy importante, pues gracias a ella eliminamos del lenguaje cadenas no deseadas como ‘‘$++\times 1$’’, que puede construirse por $f_2(\text{“}++\text{”}, \text{“}1\text{”})$. En el caso anterior, la cadena ‘‘++’’ no puede construirse con la aplicación finita de las cláusulas (a) - (c), y por lo tanto no pertenece al lenguaje.[2] ∎

1.5. Notación BNF

Para evitar hacer definiciones tan largas, utilizaremos una notación más compacta: la notación de Backus-Naur (BNF). En esta notación, definimos todas las clausulas en una sola regla recursiva.

Por ejemplo, el conjunto Exp, lo definimos como el *menor* conjunto tal que si $\alpha \in Exp$, entonces cumple con la regla:

$$\alpha ::= n \;\Big|\; \alpha + \alpha \;\Big|\; \alpha \times \alpha$$

donde $n \in B$.

En la regla anterior, el símbolo $::=$ indica que lo que está a su izquierda se define como cualquiera de los elementos que están a su derecha. El símbolo $\Big|$ puede leerse como ‘‘ó’’.

Esto es, según la regla, una expresión α es un símbolo $n \in B$, ó la concatenación de dos expresiones con un símbolo $+$ entre ellas, o la concatenación de dos expresiones con un símbolo $\times$ entre ellas.

Es importante recalcar que en la notación BNF, un elemento como $\alpha + \alpha$ no nos dice que las dos α’s son el mismo elemento, sino que lo que está a la izquierda del $+$ es una expresión definida de acuerdo con la misma regla, y lo mismo para la expresión a la derecha.

1.6. Demostraciones por inducción estructural

Una ventaja de dar definiciones inductivas para la sintaxis de lenguajes es que podemos utilizar la técnica demostraciones por inducción estructural para verificar propiedades del lenguaje. La inducción estructural funciona casi igual que la inducción matemática, hay un caso base, una hipótesis de inducción y un paso inductivo.

Si queremos demostrar que todos los elementos del lenguaje cumplen una propiedad P procedemos de la siguiente forma:

(a) Demostramos que los elementos del conjunto base cumplen P.

(b) Definimos una hipótesis inductiva: suponemos que todos los predecesores de un elemento x cumplen con P (de acuerdo con cada uno de los constructores del lenguaje).

(c) Demostramos que x cumple P.

[2] En este ejemplo usamos los símbolos ‘‘ ’’ para delimitar (clarificar) el inicio y final de las cadenas, sin embargo, estos símbolos no pertenecen al lenguaje y no los usaremos más.

Definición 1.7 (Predecesor)**.** Decimos que a ***es un predecesor de*** x sii usamos a como el argumento de un constructor para crear x, esto es, $x = f(\ldots, a, \ldots)$.

Ejemplo 1.8.

- 1 es predecesor de $1+2$, pues $f_1(1,2) = 1+2$.
- $1+2$ y 5 son los predecesores de $1+2\times 5$, pues $f_2(1+2,5) = 1+2\times 5$. ∎

Ejemplo 1.9. Mostrar que ninguna expresión comienza con el símbolo $+$.
Por inducción sobre la estructura de una expresión $a \in Exp$.

- Caso base $a = n \in B = \{0, 1, \ldots\}$: se cumple, pues ningún elemento de B comienza con $+$.
- Hipótesis inductiva: suponemos ninguna de las expresiones α y β comienzan con $+$.
- Paso inductivo: dos casos:
  - I) Caso $a = f_1(\alpha, \beta) = \alpha + \beta$: por H.I. α no comienza con el símbolo $+$, por lo tanto la cadena $\alpha + \beta$ tampoco.
  - II) Caso $a = f_2(\alpha, \beta) = \alpha \times \beta$: por H.I. α no comienza con el símbolo $+$, por lo tanto la cadena $\alpha \times \beta$ tampoco. ∎

1.7. Funciones recursivas e interpretación

Cuando tenemos un conjunto definido inductivamente es muy fácil defini funciones recursivas sobre él. En el caso de los lenguajes, las funciones recursivas son muy útiles pues nos ayudan a dar un ‘‘significado’’ a las oraciones del lenguaje.

Ejemplo 1.10. Definir una función de evaluación para las expresiones de suma y resta (i.e., que calcule el valor representado por la expresión).

- Definimos una función $e : B \to \mathbb{N}$, esta función es necesaria, pues los elementos en B no son números, sino que son símbolos que representan números. Si asumimos que B es numerable, entonces tenemos e ‘‘gratis’’, pues por definición existe una inyección con $\mathbb{N}$.
  Haciendo un poco de abuso de notación, e se comporta de la siguiente forma:

$$\begin{aligned} e(0) &= 0 \\ e(1) &= 1 \\ &\vdots \\ e(n) &= n \end{aligned}$$

- Extendemos e a una versión recursiva $\hat{e} : Exp \to \mathbb{N}$. Esta función la definimos por casos, el caso base y un caso por cada constructor.

$$\begin{aligned}\hat{e}(n) &\stackrel{\text{def}}{=} e(n) \text{ para } n \in B \\ \hat{e}(\alpha + \beta) &\stackrel{\text{def}}{=} \hat{e}(\alpha) + \hat{e}(\beta) \\ \hat{e}(\alpha \times \beta) &\stackrel{\text{def}}{=} \hat{e}(\alpha) \times \hat{e}(\beta)\end{aligned}$$

  Sólo hay que recalcar que los $+$ y $\times$ que aparecen del lado derecho de las definiciones son las operaciones de suma y multiplicación de los naturales.

Por supuesto, la forma como hemos definido las cosas hasta ahora permiten ambigüedades en la interpretación. Por ejemplo, la evaluación de una expresión $2 + 3 \times 5$ pudiera ser 25 o 17, dependiendo de qué operación se realice primero.

Para evitar problemas de ambigüedad como el anterior, vamos a optar por una solución muy simple: encerrar en paréntesis a todas las expresiones.

Entonces la nueva regla BNF para las expresiones queda como sigue:

$$\alpha ::= n \;\Big|\; (\alpha + \alpha) \;\Big|\; (\alpha \times \alpha)$$

De esta forma, la expresión del ejemplo anterior no podría construirse, quedando como únicas opciones $((2 + 3) \times 5)$ ó $(2 + (3 \times 5))$, y ninguna es ambigua.

1.8. Cálculo de proposiciones

pendiente

2. Lógica modal

2.1. Introducción y sintaxis

La lógica modal puede pensarse como una extensión de la lógica clásica para darle un carácter puramente intencional, es decir, para hablar sobre posibilidad y necesidad.

Por ejemplo, considera las siguientes oraciones:

- Todas las personas son mortales
- Pedro es mortal

Estas oraciones denotan objetos por extensión, ya sea enumerando elemento por elemento o incluyendo a todos los elementos de algún dominio determinado (en este caso las oraciones denotan conjuntos de personas). También, observa que de estas oraciones sólo podemos afirmar si son verdad o no.

En contraste, considera las oraciones:

- *Pienso* que es correcto
- *Creo* que él sabe la respuesta
- Esto *debería* ser así
- Es *improbable* que suceda eso

Aquí se denotan objetos por intensión, esto es, describiendo una propiedad que deben cumplir (e.g., el conjunto de todas las cosas que yo creo que son verdad). En estos casos, las palabras *pienso*, *creo*, *debería* e *improbable* califican la veracidad de la oración. A este tipo de modificadores les llamamos *modalidades*.

Vamos a estudiar primero el lenguaje de la lógica monomodal proposicional, o simplemente lógica modal (después veremos otras lógicas multimodales, pero las llamaremos de acuerdo con su aplicación). La lógica modal es la lógica de proposiciones extendida con dos operadores: $\Box$ de necesidad y $\Diamond$ de posibilidad.

Definición 2.1 (Sintaxis lógica modal)**.** Sea $\mathcal{P}$ un conjunto numerable de proposiciones atómicas. *El conjunto $\mathcal{L}_{ML}$ de las fórmulas de la lógica monomodal proposicional*, se define como el menor conjunto que se construye de acuerdo con la siguiente regla BNF:

$$\phi ::= p \;\big|\; \bot \;\big|\; \neg\phi \;\big|\; (\phi \wedge \phi) \;\big|\; \Box\phi$$

donde $p \in \mathcal{P}$ y $\bot \notin \mathcal{P}$. ∎

La constante $\top$ y el resto de los conectores lógicos se definen de la forma usual. Además, definimos el operador $\Diamond$, dual de $\Box$, como sigue:

$$\Diamond\phi \stackrel{\text{def}}{=} \neg\Box\neg\phi$$

También, para evitar el exceso de paréntesis, usamos la siguiente convención de precedencia de los operadores (de mayor precedencia a menor):

- $\neg$, $\Box$ y $\Diamond$
- $\wedge$
- $\vee$
- $\Rightarrow$
- $\Leftrightarrow$

De esta forma, la fórmula:

$$\Box p \vee \neg q \Rightarrow \Diamond s \wedge t$$

debe leerse como:

$$(((\Box p) \vee (\neg q)) \Rightarrow ((\Diamond s) \wedge t))$$

2.2. Estructuras relacionales, valuaciones, modelos y satisfacción

Intuitivamente, una fórmula $\Box\phi$ se lee como ‘‘*necesariamente* ϕ es cierta’’. De la misma forma, la fórmula $\Diamond\phi$ se lee como ‘‘*es posible que* ϕ *sea cierta*’’. Pero, ¿qué quiere decir que ϕ sea posible?

La idea esencial es que además del mundo real, existen otros mundos ‘‘posibles’’, en donde las fórmulas pudieran tener otros valores de verdad. Esta idea se define formalmente con estructuras relacionales.

Una estructura relacional no es más que un conjunto y una (o varias) relación binaria sobre éste. En el contexto de la lógica modal les llamamos marcos de Kripke.

Definición 2.2 (Marcos y modelos)**.** Sea un conjunto numerable:

$$W = \{w_1, w_2, \ldots\}$$

de mundos posibles, o simplemente posibilidades. Sea una relación binaria:

$$R \subseteq W \times W$$

Definimos a *un marco de Kripke* $\mathfrak{F}$ como la pareja:

$$\mathfrak{F} = (W, R)$$

∎

La idea intuitiva es que ahora tenemos satisfacción local, esto es, una misma fórmula puede tener distintos valores de verdad en diferentes mundos posibles. De esta forma, una fórmula ϕ es *necesariamente* cierta en un mundo posible si se satisface en todos los mundos posibles accesibles a través de R.

El primer paso es definir valuaciones para las proposiciones atómicas, que junto con los marcos conforman modelos para las fórmulas.

Definición 2.3 (Valuaciones y modelos)**.** Sea $\mathfrak{F} = (W, R)$ una estructura de Kripke. Definimos *una valuación* v como una función:

$$v : \mathcal{P} \to \wp(W)$$

donde $\mathcal{P}$ es el conjunto de las proposiciones atómicas de $\mathcal{L}_{ML}$ y $\wp(W)$ es el conjunto potencia de W. A la pareja:

$$\mathfrak{M} = (\mathfrak{F}, v)$$

le llamamos *modelo*. Si tomamos un elemento w de W, entonces a la pareja:

$$(\mathfrak{M}, w)$$

le llamamos un *modelo apuntado por* w. ∎

Una valuación asigna a cada proposición un subconjunto de W en donde dicha proposición es verdadera. Con este elemento como base, podemos construir inductivamente una relación de satisfacción entre los modelos apuntados y las fórmulas del lenguaje.

Definición 2.4 (Satisfacción)**.** Sean $\mathfrak{M} = (W, R, v)$ un modelo y w un elemento distinguido de W. Definimos *la relación* $\models$ *de satisfacción entre los modelos apuntados y las fórmulas de* $\mathcal{L}_{ML}$ como la menor relación tal que:

(a) $(\mathfrak{M}, w) \not\models \bot$;

(b) $(\mathfrak{M}, w) \models p$ sii $w \in v(p)$ (para $p \in \mathcal{P}$);

(c) $(\mathfrak{M}, w) \models \neg\phi$ sii $(\mathfrak{M}, w) \not\models \phi$;

(d) $(\mathfrak{M}, w) \models (\phi \wedge \psi)$ sii $(\mathfrak{M}, w) \models \phi$ y $(\mathfrak{M}, w) \models \psi$;

(e) $(\mathfrak{M}, w) \models \Box\phi$ sii para todo $w' \in W$, si $(w, w') \in R$, entonces $(\mathfrak{M}, w') \models \phi$. ∎

2.3. Clases de modelos, validez y axiomas (correspondencia)

Así como en el cálculo de proposiciones tenemos el concepto de fórmula válida (i.e., la que es verdadera bajo cualquier valuación), en la lógica modal es posible tener un concepto análogo. Sin embargo, como la satisfacción es local, entonces podemos definir varios niveles de validez, además del concepto de consecuencia lógica.

Definición 2.5 (Validez)**.** Sea ϕ una fórmula cualquiera en $\mathcal{L}_{ML}$, entonces:

(a) Dado un marco $\mathfrak{F}$ y un elemento w en W (en $\mathfrak{F}$), se dice que ϕ *es válida en* w *en el marco* $\mathfrak{F}$ si para toda valuación v, se tiene que el modelo apuntado $(\langle \mathfrak{F}, V \rangle, w)$ satisface ϕ. En este caso se escribe $\mathfrak{F}, w \models \phi$;

(b) Dado un marco $\mathfrak{F}$, se dice que ϕ *es válida en el marco* $\mathfrak{F}$ si para toda w en W (en $\mathfrak{F}$), se tiene que ϕ es válida en w en $\mathfrak{F}$. En este caso se escribe $\mathfrak{F} \models \phi$;

(c) Dada una clase $\mathbf{C}$ de marcos, se dice que ϕ *es válida en* $\mathbf{C}$ si para todo $\mathfrak{F}$, ϕ es válida en $\mathfrak{F}$. En este caso se escribe $\models_{\mathbf{C}} \phi$;

(d) Se dice simplemente que ϕ *es válida*, si es válida en la clase $\mathbf{K}$ de todos los marcos. En este caso se escribe $\models \phi$. ∎

Definición 2.6 (Consecuencia lógica modal)**.** Sean $\mathbf{C}$ una clase de marcos y Γ un conjunto de fórmulas. Se dice que *una fórmula* ϕ *es consecuencia (local) de las fórmulas en* Γ si:

$$\mathfrak{M}, w \models \gamma \text{ implica que } \mathfrak{M}, w \models \phi$$

para todo modelo $\mathfrak{M}$ basado en los marcos en $\mathbf{C}$, para todo w en $\mathfrak{M}$, y para toda γ en Γ. Si esto es el caso, entonces también se puede escribir:

$$\Gamma \models_C \phi$$

∎

Así como se usa $\mathbf{K}$ para la clase de todos los marcos, para otras clases comunes se tiene una nomenclatura estándar. Estas clases agrupan marcos de acuerdo con las propiedades de su relación de accesibilidad. Los nombres de las clases de marcos más importantes se muestran en el cuadro 2.3.

$\mathbf{K}$	Clase de todos los marcos.
$\mathbf{K4}$	Clase de todos los marcos con relaciones transitivas.
$\mathbf{T}$	Clase de todos los marcos con relaciones reflexivas.
$\mathbf{B}$	Clase de todos los marcos con relaciones simétricas.
$\mathbf{KD}$	Clase de todos los marcos con relaciones seriales.
$\mathbf{S4}$	Clase de todos los marcos con relaciones reflexivas y transitivas.
$\mathbf{S5}$	Clase de todos los marcos con relaciones reflexivas y euclidianas (i.e., de equivalencia).
$\mathbf{K4{,}3}$	Clase de todos los marcos con relaciones transitivas y no seriales.
$\mathbf{S4{,}3}$	Clase de todos los marcos con relaciones reflexivas, transitivas y no seriales.
$\mathbf{KD45}$	Clase de todos los marcos con relaciones transitivas, euclidianas y seriales.

Cuadro 1: Las clases de marcos de Kripke más comunes

Diferentes clases de marcos permiten diferentes axiomatizaciones, dependiendo de las propiedades de la relación de accesibilidad. El sistema básico es el sistema $\mathbf{K}$, al que se le pueden agregar los axiomas asociados a las distintas clases de marcos.

Definición 2.7 (Sistema de demostración $\vdash_{\mathbf{K}}$)**.** Se define el *sistema de demostración* $\vdash_{\mathbf{K}}$ como el conjunto que contiene todas las instancias de las tautologías de la lógica proposicional, todas las instancias del axioma (K), y que está cerrado bajo las reglas (Gen) y (MP) (definidos a continuación).

$$\vdash_{\mathbf{K}} \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi) \tag{K}$$

$$\frac{\vdash_{\mathbf{K}} \phi \qquad \vdash_{\mathbf{K}} \phi \Rightarrow \psi}{\vdash_{\mathbf{K}} \psi} \tag{Modus Ponens}$$

$$\frac{\vdash_{\mathbf{K}} \phi}{\vdash_{\mathbf{K}} \Box\phi} \tag{Gen}$$

∎

De acuerdo con la forma en que definimos la sintaxis de las fórmulas, además de las tautologías de la lógica proposicional (que incluyen las definiciones de $\wedge$, $\Rightarrow$ y $\Leftrightarrow$) debemos recordar la dualidad del operador $\Diamond$. Si hubiésemos incluido el operador $\Diamond$ directamente en la sintaxis del lenguaje, y definido $\Box$ como derivado, entonces deberíamos considerar un axioma más para el sistema $\mathbf{K}$:

$$\vdash_{\mathbf{K}} \Diamond\phi \Leftrightarrow \neg\Box\neg\phi \tag{Dual}$$

Conviene resaltar que la regla (Gen) de *generalización* o *nececitación* se refiere a fórmulas válidas en cualquier marco posible, esto es, anteponer $\Box$ a una fórmula preserva su validez, mas no su satisfacción. Esto no ocurre igual en el caso de la regla (Modus Ponens), la cual sí preserva tanto validez como satisfacción.

$$\Box\phi \Rightarrow \Box\Box\phi \tag{4}$$
$$\Box\phi \Rightarrow \phi \tag{T}$$
$$\phi \Rightarrow \Box\Diamond\phi \tag{B}$$
$$\Box\phi \Rightarrow \Diamond\phi \tag{D}$$
$$\Diamond\phi \wedge \Diamond\psi \Rightarrow \Diamond(\phi \wedge \Diamond\psi) \vee \Diamond(\phi \wedge \psi) \vee \Diamond(\psi \wedge \Diamond\phi) \tag{.3}$$
$$\Box(\Box\phi \Rightarrow \phi) \Rightarrow \Box\phi \tag{L}$$

Cuadro 2: Axiomas asociados a distintas clases de marcos

Para finalizar, en el cuadro 2 tenemos varios axiomas correctos, asociados a algunas de las clases conocidas. Por ejemplo, el axioma (4) es correcto para la clase $\mathbf{K4}$ y el axioma (.3) es correcto para las clases $\mathbf{K4{,}3}$ y $\mathbf{S4{,}3}$. En conjunto con el sistema $\mathbf{K}$, los axiomas del cuadro 2 forman lógicas correctas y completas para sus clases (cf. cuadro 2.3). Las lógicas resultantes se nombran de la misma forma que las clases para las que son correctas. Por ejemplo, la lógica que contiene los axiomas (4), (T) y (D), forman la lógica $\mathbf{KD45}$, y si se le añade el axioma (B), forma la lógica $\mathbf{S5}$.

Como ejemplo, en la siguiente proposición vamos a demostrar la corrección (*soundness*) del axioma (K) y la regla (Gen). El resto de la demostración de corrección para todo el sistema $\vdash_{\mathbf{K}}$ se deja como ejercicio.

Proposición 2.8. *El axioma (K) y la regla (Gen) son correctos.*

Demostración. P.D.: (a) $\models \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi)$, y (b) $\models \phi$ implica que $\models \Box\phi$.

(a) Suponemos que $(\mathfrak{M}, w) \models \Box(\phi \Rightarrow \psi)$ para algún $(\mathfrak{M}, w)$. Entonces: para todo w', $(w, w') \in R$ implica que $(\mathfrak{M}, w') \models \phi \Rightarrow \psi$, por lo tanto, si $(\mathfrak{M}, w') \models \phi$, entonces $(\mathfrak{M}, w') \models \psi$.
Para que la implicación anterior sea cierta tenemos dos posibles casos:
(i) $(\mathfrak{M}, w') \models \phi$ en todo w' accesible desde w, entonces $(\mathfrak{M}, w) \models \Box\phi$ y por la suposición original también $(\mathfrak{M}, w) \models \Box\psi$.
(ii) Existe w' accesible desde w t.q. $(\mathfrak{M}, w') \not\models \phi$, entonces $(\mathfrak{M}, w) \not\models \Box\phi$ y por vacuidad $(\mathfrak{M}, w) \models \Box\phi \Rightarrow \Box\psi$.

(b) Suponemos que $\models \phi$, entonces $(\mathfrak{M}, w) \models \phi$ para todo $(\mathfrak{M}, w)$ y para todo w en $\mathfrak{M}$. Por lo tanto, $(\mathfrak{M}, w) \models \Box\phi$ pues ϕ se satisface en todo mundo posible, incluyendo los accesibles desde w.

∎

2.4. Completud de $\vdash_{\mathbf{K}}$

La prueba de completud para el sistema $\vdash_{\mathbf{K}}$ (y también para las extensiones a éste) es más difícil que la demostración de corrección. Para probar corrección podemos hacer inducción, pues una demostración (usando el sistema de demostración) sólo es una enumeración finita de fórmulas. Sin embargo, para probar la completud no podemos aplicar la misma técnica, pues la clase de todos los modelos no la podemos construir inductivamente. En lugar de usar una estrategia directa la prueba de completud se hace por contraposición, mostrando que toda formula (o conjunto de fórmulas) consistente es satisfactible en algún modelo.

Para esta sección vamos a introducir el concepto de *lógica modal normal*, el cual vamos a utilizar en lugar del concepto de demostración.

Definición 2.9 (Lógica modal normal)**.** Una *lógica modal* es un conjunto de fórmulas que contiene a todas las instancias de las tautologías de la lógica proposicional y está cerrado bajo Modus Ponens y sustitución uniforme. Una lógica modal es *normal* si contiene el axioma (K) y está cerrada bajo la regla (Gen). Escribimos $\vdash_{\mathbf{C}} \phi$ para indicar que la fórmula ϕ pertenece a la lógica modal $\mathbf{C}$. ∎

Podemos listar varios casos que cumplen con la definición de lógicas normales. Por ejemplo, la lógica modal trivial (i.e., la que tiene a todas las fórmulas modales) es normal. Sin embargo, las lógicas más usadas son las que están asociadas a alguna clase de estructuras relacionales en particular (e.g., $\mathbf{K4}$, $\mathbf{S5}$, etc.). Nosotros nos enfocaremos en el sistema $\mathbf{K}$, pero la misma técnica puede generalizarse para otros casos.

Definición 2.10 (Deducibilidad)**.** Dada una lógica $\mathbf{C}$ y un conjunto de fórmulas $\Gamma \cup \{\phi\}$, decimos que ϕ *es deducible en* $\mathbf{C}$ *desde* Γ, si $\vdash_{\mathbf{C}} \phi$ ó existe un conjunto de fórmulas $\{\gamma_1, \ldots, \gamma_n\} \subseteq \Gamma$ tales que $\vdash_{\mathbf{C}} (\gamma_1 \wedge \cdots \wedge \gamma_n) \Rightarrow \phi$. ∎

Definición 2.11 (Consistencia)**.** Sean una lógica $\mathbf{C}$ y un conjunto de fórmulas Γ. Decimos que Γ *es consistente en* $\mathbf{C}$ si $\Gamma \not\vdash_{\mathbf{C}} \perp$, de lo contrario decimos que es inconsistente. *Una fórmula* ϕ *es consistente* si el conjunto $\{\phi\}$ es consistente. ∎

Definición 2.12 (Conjuntos Consistentes Maximales)**.** Sea Γ un conjunto consistente de fórmulas. Decimos que Γ *es un Conjunto Consistente Maximal (CCM),* si para todo $\phi \notin \Gamma$, $\Gamma \cup \{\phi\}$ es inconsistente. ∎

Lema 2.13 (Lindenbaum)**.** *Todo conjunto consistente de fórmulas está contenido en un CCM.*

Demostración. Sea Γ un conjunto consistente. Sea

$$\phi_0, \phi_1, \ldots$$

una enumeración de todas las fórmulas de un lógica modal. Construimos el conjunto Γ^+ de la siguiente forma:

$$\Gamma^0 \stackrel{\text{def}}{=} \Gamma$$

$$\Gamma^{i+1} \stackrel{\text{def}}{=} \begin{cases} \Gamma^i \cup \{\phi_i\} & \text{si } \Gamma^i \cup \{\phi_i\} \text{ es consistente} \\ \Gamma^i & \text{en el otro caso} \end{cases}$$

$$\Gamma^+ \stackrel{\text{def}}{=} \bigcup_{n\in\mathbb{N}} \Gamma^n$$

Claramente Γ^+ es un CCM, pues las únicas fórmulas que no se le agregaron a son las que lo vuelven inconsistente. Además, como $\Gamma \subseteq \Gamma^+$ se demuestra lo que se quería. ∎

Teorema 2.14 (Propiedades de un CCM)**.** *Sean Γ un CCM y $\phi, \psi \in \mathcal{L}_{ML}$, entonces:*

(a) $\perp \notin \Gamma$

(b) $\neg\phi \in \Gamma$ *sii* $\phi \notin \Gamma$

(c) $(\phi \wedge \psi) \in \Gamma$ *sii* $\phi \in \Gamma$ *y* $\psi \in \Gamma$

(d) $(\phi \vee \psi) \in \Gamma$ *sii* $\phi \in \Gamma$ *ó* $\psi \in \Gamma$

(e) $(\phi \Rightarrow \psi) \in \Gamma$ *sii* $\phi \notin \Gamma$ *ó* $\psi \in \Gamma$

(f) Γ *está cerrado bajo modus ponens*

(g) Γ *está cerrado bajo deducibilidad*

Definición 2.15 (Modelo canónico)**.** *El modelo canónico $\mathfrak{M}^C = (W^C, R^C, v^C)$* se compone de los siguientes elementos:

$$W^C \stackrel{\text{def}}{=} \{\Gamma \mid \Gamma \text{ es un CCM}\}$$

$$R^C \stackrel{\text{def}}{=} \{(\Gamma, \Gamma') \mid \{\phi \mid \Box\phi \in \Gamma\} \subseteq \Gamma'\}$$

$$v^C(p) \stackrel{\text{def}}{=} \{\Gamma \mid p \in \mathcal{P} \text{ implica que } p \in \Gamma\}$$

∎

Lema 2.16 (Verdad)**.** *Sea una fórmula $\phi \in \mathcal{L}_{ML}$ y Γ un CCM, entonces:*

$$\phi \in \Gamma \quad \textit{sii} \quad (\mathfrak{M}^C, \Gamma) \models \phi$$

Demostración. Por inducción sobre la estructura de ϕ.
Base $\phi = p \in \mathcal{P}$:

$p \in \Gamma$	sii	$\Gamma \in v^C(p)$	(def. de v^C)
	sii	$(\mathfrak{M}^C, \Gamma) \models p$	(def. de $\models$)

Hipótesis inductiva: para dos fórmulas cualesquiera α y β, y un CCM Γ:

$$\alpha \in \Gamma \text{ sii } (\mathfrak{M}^C, \Gamma) \models \alpha$$

$$\beta \in \Gamma \text{ sii } (\mathfrak{M}^C, \Gamma) \models \beta$$

Paso inductivo:

- Caso $\phi = \bot$: directo por propidades de Γ.

- Caso $\phi = \neg\alpha$:

$$\begin{array}{lll} \neg\alpha \in \Gamma & \text{sii} & \alpha \notin \Gamma \qquad \text{(propiedades de } \Gamma) \\ & \text{sii} & (\mathfrak{M}^C, \Gamma) \not\models \alpha \qquad \text{(H.I.)} \\ & \text{sii} & (\mathfrak{M}^C, \Gamma) \models \neg\alpha \qquad \text{(def. de } \models) \end{array}$$

- Caso $\phi = (\alpha \wedge \beta)$:

$$\begin{array}{lll} (\alpha \wedge \beta) \in \Gamma & \text{sii} & \alpha \in \Gamma \text{ y } \beta \in \Gamma \qquad \text{(propiedades de } \Gamma) \\ & \text{sii} & (\mathfrak{M}^C, \Gamma) \models \alpha \text{ y } (\mathfrak{M}^C, \Gamma) \models \beta \qquad \text{(H.I.)} \\ & \text{sii} & (\mathfrak{M}^C, \Gamma) \models (\alpha \wedge \beta) \qquad \text{(def. de } \models) \end{array}$$

- Caso $\phi = \Box\alpha$:

($\Rightarrow$) Si $\Box\alpha \in \Gamma$, entonces para toda Γ', $(\Gamma, \Gamma') \in R^C$ implica que $\alpha \in \Gamma$ (por def. de R^C). Luego $(\mathfrak{M}^C, \Gamma') \models \alpha$ (por H.I.) y por lo tanto $(\mathfrak{M}^C, \Gamma) \models \Box\alpha$.

($\Leftarrow$) Suponemos que $(\mathfrak{M}^C, \Gamma) \models \Box\alpha$. Entonces para toda Γ', $(\Gamma, \Gamma') \in R^C$ implica que $(\mathfrak{M}^C, \Gamma') \models \alpha$, y por H.I. $\alpha \in \Gamma'$. Por def. de R^C podemos encontrar un conjunto $\{\sigma_1, \ldots, \sigma_n \mid \Box\sigma, \ldots, \Box\sigma_n \in \Gamma\} \subseteq \Gamma'$ y por las propiedades de Γ', $(\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \alpha \in \Gamma'$ y por consistencia $\vdash_{\mathbf{K}} (\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \alpha$. Ahora procedemos utilizando las herramientas del sistema $\vdash_{\mathbf{K}}$:

$$\begin{array}{ll} \vdash_{\mathbf{K}} (\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \alpha & \\ \vdash_{\mathbf{K}} \Box((\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \alpha) & \text{(Gen)} \\ \vdash_{\mathbf{K}} \Box((\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \alpha) \Rightarrow (\Box(\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \Box\alpha) & \text{(K)} \\ \vdash_{\mathbf{K}} \Box(\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \Box\alpha & \text{(MP)} \\ \vdash_{\mathbf{K}} (\Box\sigma_1 \wedge \cdots \wedge \Box\sigma_n) \Rightarrow \Box\alpha & (^3) \end{array}$$

Como por hipótesis $\Box\sigma_1, \ldots, \Box\sigma_n \in \Gamma$, entonces por las props. de Γ, también $\alpha\Box \in \Gamma$.

∎

Ahora tenemos todas las herramientas necesarias para demostrar la completud del sistema $\mathbf{K}$. También, ya mencionamos que la demostración no es directa, sino que se demuestra que un conjunto consistente es satisfactible[4]en un modelo de la clase $\mathbf{K}$. Esta caracterización de completud queda establecida en el siguiente teorema.

Teorema 2.17. *(a) Una lógica $\mathbf{C}$ tiene la propiedad de completud fuerte si todo conjunto consistente de fórmulas es satisfactible en un modelo de la clase $\mathbf{C}$.*

(b) Una lógica $\mathbf{C}$ tiene la propiedad de completud débil si toda fórmula consistente de es satisfactible en un modelo de la clase $\mathbf{C}$.

[3]Ejercicio: probar que $(\mathfrak{M}, w) \models \Box(\sigma_1 \wedge \cdots \wedge \sigma_n) \Rightarrow \Box\alpha$ implica que $(\mathfrak{M}, w) \models (\Box\sigma_1 \wedge \cdots \wedge \Box\sigma_n) \Rightarrow \Box\alpha$.

[4]Decimos que un conjunto de fórmulas es satisfactible si existe un modelo que satisface a todas las fórmulas del conjunto.

Demostración. (a) Por contraposición, suponemos que la lógica no es fuertemente completa, es decir, existe un conjunto de fórmulas $\Gamma \cup \{\phi\}$ tal que $\Gamma \models_{\mathbf{C}} \phi$ pero $\Gamma \nvdash_{\mathbf{C}} \phi$. Por lo tanto $\Gamma \cup \{\neg\phi\}$ es consistente pero no satisfactible en ningún modelo.

∎

Teorema 2.18 (Completud fuerte de $\vdash_{\mathbf{K}}$)**.** *Para todo conjunto de fórmulas* $\Gamma \cup \{\phi\}$*, si* $\Gamma \models_{\mathbf{K}} \phi$*, entonces* $\Gamma \vdash_{\mathbf{K}} \phi$*.*

Demostración. Por el teorema anterior basta demostrar que todo conjunto consistente de fórmulas tiene un modelo en la clase $\mathbf{K}$. Sea Γ un conjunto consistente de fórmulas. Por el Lema de Lindenbaum tenemos que Γ^+ es un CCM que contiene a Γ. Por el Lema de la Verdad $(\mathfrak{M}^C, \Gamma^+) \models \gamma$ para todo $\gamma \in \Gamma^+$. Por lo tanto, Γ es satisfactible. ∎

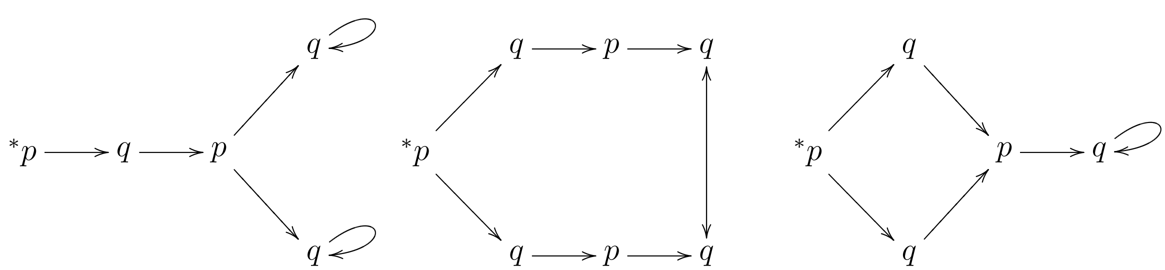
Observa que para demostrar la completud del sistema $\mathbf{K}$ fueron necesarias dos cosas: que el modelo canónico estuviera en la clase $\mathbf{K}$ y que para demostrar el Lema de la Verdad se utilizaron las herramientas del sistema. Para demostrar completud de otro sistema necesitaríamos hacer los siguientes ajustes. Primero, debemos cambiar la definición del modelo canónico para que pertenezca a la clase deseada. Finalmente, por el cambio en el modelo canónico, para demostrar el Lema de la Verdad va a ser necesario utilizar los axiomas asociados con la clase deseada.

2.5. Bisimulaciones

Un modelo puede estudiarse desde el punto de vista de un sistema de transiciones. Desde esta perspectiva, un mundo posible representa un estado de un sistema donde se satisfacen ciertas condiciones, y la relación de accesibilidad entre mundos posibles representa las posibles transiciones entre los estados del sistema.

De esta forma, dos estados o mundos posibles son equivalentes si, y sólo si, satisfacen las mismas condiciones. En este caso, dichas condiciones son las proposiciones atómicas. Extendiendo este razonamiento, puede decirse que dos modelos son equivalentes, o *bisimilares*, si para cada transición posible en el primero, existe otra en el segundo que lleva a un estado equivalente al que se llegó en el primero, y esta condición se vuelve a cumplir sucesivamente en los siguientes estados.

Por ejemplo, considera los siguientes modelos:



En los modelos anteriores no es importante el nombre de cada mundo posible, así que etiquetamos directamente el nodo con las proposiciones atómicas que se satisfacen ahí.

Usamos el símbolo $*$ para indicar que los modelos están apuntados por ese nodo. Claramente vemos que en cualquiera de los tres, si recorremos la gráfica nodo a nodo, tendríamos exactamente las mismas opciones en los tres casos.

Definición 2.19 (Bisimilaridad de modelos $\leftrightarrows$)**.** Sean dos modelos:

$$\mathfrak{M} = (W, R, v)$$
$$\mathfrak{M}' = (W', R', v')$$

y una relación simétrica:

$$S \subseteq W \times W'$$

Se dice que S *es una bisimulación entre los modelos* $\mathfrak{M}$ *y* $\mathfrak{M}'$ si se cumple que:

(a) si $(w, w') \in S$, entonces w y w' satisfacen exactamente las mismas proposiciones atómicas y,

(b) para toda u tal que $(w, u) \in R$, existe u' tal que $(w', u') \in R'$ y $(u, u') \in S$.

Si S es una bisimulación entre $\mathfrak{M}$ y $\mathfrak{M}'$ y $(w, w') \in S$, entonces se dice que *los modelos apuntados,* $(\mathfrak{M}, w)$ *y* $(\mathfrak{M}', w')$ *son bisimilares, y se escribe* $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$. Se dice simplemente que $\mathfrak{M}$ *y* $\mathfrak{M}'$ *son bisimilares, y se escribe* $\mathfrak{M} \leftrightarrows \mathfrak{M}'$, si existe una bisimulación entre ellos. ∎

Teorema 2.20 ($\leftrightarrows$ preserva $\models$)**.** *Sean* $(\mathfrak{M}, w)$ *y* $(\mathfrak{M}', w')$ *dos modelos apuntados tales que* $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$*. Entonces,* $(\mathfrak{M}, w) \models \phi$ *sii* $(\mathfrak{M}', w') \models \phi$*.*

Demostración. Sea S la bisimulación entre $\mathfrak{M}$ y $\mathfrak{M}'$. Por inducción estructural en el orden sintáctico de las fórmulas.
Caso base: Se tiene que por definición si $(w, w') \in S$, entonces w y w' satisfacen exactamente las mismas proposiciones atómicas.
Hipótesis inductiva: Sean dos fórmulas α y β, entonces:

$$(\mathfrak{M}, w) \models \alpha \text{ sii } (\mathfrak{M}', w') \models \alpha$$
$$(\mathfrak{M}, w) \models \beta \text{ sii } (\mathfrak{M}', w') \models \beta$$

Paso inductivo:

- Caso $\phi = \neg\alpha$: entonces por H.I. en ambos modelos no se satisface α, y por definición de satisfacción, en ambos modelos se satisface $\neg\alpha$.
- Caso $\phi = \alpha \wedge \beta$: por definición de satisfacción en $(\mathfrak{M}, w)$ se satisfacen α y β, y por H.I. se satisfacen en ambos modelos y por satisfacción también se satisface $\alpha \wedge \beta$ en ambos modelos.
- Caso $\phi = \Box\alpha$: se tiene que α es el caso en todo mundo accesible desde w en R, entonces cada uno de esos elementos de R tiene su símil en R'; en el sentido contrario, si en el segundo modelo existiera un mundo accesible desde w' por R', donde α no fuese el caso, entonces existiría otro equivalente en R, pero no es así, pues $\mathfrak{M}, w \models \Box\alpha$.

∎

Referencias

[1] Blackburn, de Rijke, Venema: *Modal Logic*, Cambridge University Press, 2001

[2] Popkorn: *First Steps in Modal Logic*, Cambridge University Press, 1995

[3] Cresswell, Hughes: *A New Introduction to Modal Logic*, Routledge, 1996