

Facsímil 12

Lógica e incertidumbre

Capítulo 06

Razonamiento no monótono y revisión de creencias

Capítulo 06: Razonamiento no monótono y revisión de creencias

Entrando en el tema

Te presento a Tweety. Te digo una sola cosa de él: es un pájaro. Sin pensarlo, ya has concluido algo que yo no he dicho: que vuela. No has consultado a Tweety, no has visto sus alas, no sabes su especie. Has aplicado una regla que llevas grabada desde niño, «los pájaros vuelan», y la has dado por buena. Es una conclusión razonable, útil y, casi siempre, correcta.

Ahora te doy un dato más: Tweety es un pingüino. En una décima de segundo retiras la conclusión anterior. Ya no crees que vuele; de hecho, ahora crees lo contrario. No has cambiado de regla, no has descubierto que los pájaros no vuelan. Simplemente has reconocido una excepción y has actualizado lo que creías. Y lo más interesante es lo que ha pasado con tu razonamiento: **una premisa nueva te ha hecho perder una conclusión que ya tenías**.

Eso, que para ti es trivial, es imposible en la lógica del capítulo 1. Allí la inferencia era **monótona**: si una conclusión se seguía de unas premisas, seguía siguiéndose por más premisas que añadieras. Añadir información solo podía hacer crecer el conjunto de cosas demostrables, nunca encogerlo. Es una propiedad maravillosa para garantizar verdades matemáticas, y un desastre para razonar sobre el mundo real, donde casi todo lo que sabemos es provisional, vale «por defecto» y está a una excepción de quedar desmentido.

Este capítulo trata de la lógica que **sí se retracta**. De cómo se formaliza «las aves vuelan, salvo las que no», cómo una base de conocimiento incorpora un hecho que contradice lo que creía sin volverse loca, y cómo razonar con argumentos que se atacan unos a otros. Es el puente entre la rigidez del capítulo 1 y el mundo lleno de excepciones de los sistemas expertos del capítulo 4, donde el pingüino que no vuela ya asomó como problema. Aquí lo formalizamos. Y al final veremos por qué esto reaparece, sin avisar, en cómo un modelo de lenguaje moderno maneja (a veces fatal) los casos por defecto y sus excepciones.

Monotonía frente a no monotonía: la lógica que no se retracta

Pongamos nombre exacto a la propiedad que rompe Tweety. En la lógica clásica, la relación de consecuencia cumple lo que se llama **monotonía**: ampliar las premisas nunca invalida una conclusión.

$$\text{Si } \Gamma \vdash \varphi \text{ entonces } \Gamma \cup \Delta \vdash \varphi$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
Γ	El conjunto de premisas de partida.	{toda ave vuela, Tweety es ave}.
$\vdash$	«deriva»: a la izquierda hay con qué demostrar lo de la derecha.	Lo que separa lo que sabes de lo que se sigue.
φ	La conclusión derivada.	Tweety vuela.
Δ	Premisas nuevas que añadimos después.	{Tweety pesa 3 kg}.
$\cup$	Unión: juntar las premisas viejas con las nuevas.	Γ más Δ .

En palabras: si algo se deduce de un conjunto de premisas, se sigue deduciendo por muchas premisas más que le echés encima; la información nueva solo puede añadir conclusiones, jamás quitarlas.

Un ejemplo numérico de que esto es deseable en matemáticas: de $\{a > 5\}$ se deduce $a > 0$; si añadido $\{a \text{ es par}\}$, sigue siendo cierto que $a > 0$. Ninguna premisa adicional sobre a va a destruir esa desigualdad. Esa estabilidad es justo lo que quieres en una demostración: un teorema demostrado no deja de ser teorema porque aparezcan nuevos axiomas compatibles.

El sentido común funciona al revés.¹ Cuando concluyes «Tweety vuela» a partir de «es un ave», no estás afirmando una consecuencia lógica necesaria: estás haciendo una **conjetura razonable** que mantienes mientras nadie te lleve la contraria. Para marcar esa diferencia se usa otra flecha, la de la consecuencia **no monótona**, que se suele dibujar ondulada:

$$\Gamma \vdash \varphi \quad \text{pero} \quad \Gamma \cup \Delta \nvdash \varphi$$

En palabras: de lo que sé concluyo φ de forma plausible, pero al añadir Δ esa conclusión puede caerse; la flecha ondulada $\vdash$ avisa de que estamos razonando «por defecto», no con garantía de hierro.

Con $\Gamma = \{\text{las aves vuelan por defecto, Tweety es ave}\}$ concluimos $\varphi = \text{Tweety vuela}$. Pero con $\Delta = \{\text{Tweety es pingüino}\}$, la conclusión se retira. La flecha gruesa $\vdash$ nunca haría esto; la ondulada $\vdash$, sí. Todo el capítulo consiste en darle a esa flecha ondulada una definición precisa, porque «retractarse cuando toca» suena fácil hasta que intentas escribir las reglas exactas de cuándo toca.

Mundo cerrado y negación por fallo: lo que callas, lo niego

La forma más sencilla de razonamiento no monótono no nace de la filosofía, sino de la práctica: las bases de datos. Cuando consultas los vuelos de una compañía y el de las 14:30 a Lisboa no aparece, concluyes que **no existe**. Nadie ha escrito en ningún sitio «no hay vuelo a Lisboa a las 14:30»; lo deduces de su ausencia. A esa convención se le llama **suposición de mundo cerrado** (en inglés *closed-world assumption*, CWA): lo que no consta como verdadero se asume falso.

Es lo contrario de la lógica clásica, que vive en **mundo abierto**: si no puedo demostrar algo, no concluyo ni que es verdad ni que es mentira, simplemente no lo sé. El mundo cerrado es más atrevido y, para datos completos, mucho más útil: tu listado de vuelos es exhaustivo por diseño, así que la ausencia sí es información.

Esta idea tiene un nombre operativo en programación lógica: **negación por fallo**. Prolog no tiene una negación lógica de verdad; tiene un operador, `\+`, que significa «no consigo demostrarlo». La regla es: para evaluar `\+ Objetivo`, intenta probar `Objetivo`; si la prueba **falla**, entonces `\+ Objetivo` se da por cierto.

```
vuelo(madrid, paris, 0930).
vuelo(madrid, roma, 1145).

% "no hay vuelo a Lisboa" se concluye porque no se puede probar que lo haya
?- \+ vuelo(madrid, lisboa, _).
% Prolog responde: true
```

Aquí está la no monotonía, desnuda. Hoy `\+ vuelo(madrid, lisboa, _)` es verdadero. Si mañana alguien añade el hecho `vuelo(madrid, lisboa, 1430).`, la misma consulta pasa a ser falsa. Una premisa nueva ha retirado una conclusión: exactamente lo que la flecha gruesa del capítulo 1 tenía prohibido hacer. La negación por fallo es cómoda y rapidísima, pero arrastra un peligro que conviene tener grabado: **confunde «es falso» con «no lo sé»**. Si tu base de datos de vuelos está incompleta, concluirás que no existen vuelos que sí existen. El mundo cerrado solo es honesto cuando de verdad lo sabes todo sobre el dominio que cierras.

Lógica por defecto de Reiter: reglas con letra pequeña

La negación por fallo resuelve un caso, pero el razonamiento por defecto es más amplio: queremos poder decir «normalmente pasa esto, salvo que conste lo contrario» de forma general. En 1980, Raymond Reiter le dio a esa idea una forma matemática que sigue siendo la referencia: la **lógica por defecto**.²

Su pieza central es la **regla por defecto**, que se escribe como una fracción con tres partes:

$$\frac{\alpha : \beta}{\gamma}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
α	El prerrequisito : lo que debe constar para plantearse la regla.	$Ave(x)$.
β	La justificación : lo que debe ser consistente suponer.	$Vuela(x)$.
:	«y es consistente con todo lo demás suponer que...».	El filtro de excepciones.
γ	El consecuente : lo que se concluye si todo encaja.	$Vuela(x)$.

En palabras: si se cumple α y no hay nada en lo que ya sabes que contradiga β , entonces concluye γ ; en cuanto algo choque con β , la regla se calla y no concluye nada.

El caso de las aves es el ejemplo canónico, y se escribe así:

$$\frac{Ave(x) : Vuela(x)}{Vuela(x)}$$

Léelo despacio: «si x es un ave, y es consistente suponer que vuela, concluye que vuela». La justificación y el consecuente coinciden (esto se llama un *default normal*), y es la forma más común: la propia conclusión sirve de filtro. Mientras nada diga que x no vuela, la regla dispara. En cuanto sepamos que x **no** vuela, suponer que vuela deja de ser consistente, la regla no se aplica, y la conclusión «vuela» nunca llega a entrar.

Añadamos el conocimiento del pingüino como hechos clásicos, no como defaults, porque no son conjeturas sino verdades duras:

- $Pingüino(x) \rightarrow Ave(x)$: todo pingüino es un ave.
- $Pingüino(x) \rightarrow \neg Vuela(x)$: ningún pingüino vuela.

Ahora razonemos sobre dos individuos. Sobre **Tweety, del que solo sabemos que es ave**: el prerrequisito $Ave(tweety)$ se cumple, y suponer $Vuela(tweety)$ es consistente con todo lo que sabemos, así que la regla dispara y concluimos que vuela. Sobre **Pingu, del que sabemos que es pingüino**: de $Pingüino(pingu)$ se sigue $Ave(pingu)$, así que el prerrequisito también se cumple; pero también se sigue $\neg Vuela(pingu)$, con lo que suponer $Vuela(pingu)$ es **inconsistente**. La justificación falla, la regla se bloquea, y la conclusión sobre Pingu es que no vuela. La misma regla, dos resultados, sin contradicción.

Una premisa nueva retira una conclusión

Lo que la lógica monótona del capítulo 1 tiene prohibido hacer, el sentido común lo hace a diario.

ESTADO INICIAL

Sé: Tweety es un ave
regla por defecto: las aves vuelan

Concluyo: Tweety vuela
provisional, mientras nada lo niegue

LLEGA INFORMACIÓN

Tweety es un pingüino
y los pingüinos no vuelan

ESTADO REVISADO

~~Tweety vuela~~
conclusión retirada

Concluyo: Tweety no vuela
la excepción manda

La regla «las aves vuelan» no ha cambiado: ha cambiado lo que sabemos de Tweety.

Eso es razonamiento no monótono: conjeturar de más y corregir cuando hace falta.

IA para gente curiosa / Facsímil 12 / Capítulo 06 / 686f6c61

Extensiones: las visiones coherentes del mundo

Una teoría por defecto puede producir más de un mundo coherente. A cada uno de esos mundos cerrados y consistentes que se obtienen aplicando todos los defaults que se pueden aplicar, Reiter lo llama una **extensión**: un conjunto de conclusiones que está cerrado bajo la lógica clásica y que respeta las justificaciones de los defaults usados. En el caso de las aves, la extensión sobre Tweety contiene «vuela»; la extensión sobre Pingu contiene «no vuela».

El detalle que vuelve esto interesante, y a veces incómodo, es que **puede haber varias extensiones incompatibles**. El ejemplo clásico es Nixon, que era a la vez cuáquero (y los cuáqueros son pacifistas por defecto) y republicano (y los republicanos no lo son por defecto). Las dos reglas dan resultados opuestos, ninguna excepción explícita decide, y la teoría tiene dos extensiones: una en la que Nixon es pacifista y otra en la que no. La lógica por defecto no inventa una respuesta única; te muestra que el conocimiento, tal como está escrito, admite dos posturas razonables. Ese «el sistema puede tener varios modelos coherentes a la vez» es exactamente la idea que más adelante reaparece, con maquinaria moderna, en Answer Set Programming.

Circunscripción de McCarthy: minimizar lo anormal

El mismo año que Reiter, John McCarthy atacó el problema desde otro ángulo, más semántico que sintáctico: la **circunscripción**.³ Su intuición es preciosa y muy humana: **suponer que el mundo es lo más normal posible**. Las cosas son corrientes salvo que conste que son raras.

La maquinaria consiste en introducir un predicado explícito de anormalidad, *Ab* (de *abnormal*), y escribir las reglas con una cláusula de escape. «Las aves vuelan» se convierte en «las aves vuelan salvo que sean anormales en este aspecto»:

$$\forall x (Ave(x) \wedge \neg Ab(x) \rightarrow Vuela(x))$$

En palabras: sea quien sea ese x , si es un ave y no es anormal, entonces vuela; la coletilla $\neg Ab(x)$ es la puerta por la que se cuelan las excepciones.

La parte no monótona es la receta para decidir quién es anormal: **circunscribir** Ab significa quedarse con los modelos en los que el conjunto de cosas anormales es lo más pequeño posible. Si nada te obliga a declarar anormal a Tweety, no lo declaras, y vuela. Pero si añades «los pingüinos son anormales respecto a volar» ($Pingüino(x) \rightarrow Ab(x)$) y sabes que Pingu es pingüino, entonces Pingu **tiene** que ser anormal, no hay forma de minimizarlo fuera, y la regla del vuelo no se le aplica. La no monotonía está en que minimizar es una operación que el conocimiento nuevo puede deshacer: una premisa que fuerza una anormalidad nueva cambia cuál es el conjunto mínimo y, con él, las conclusiones.

Reiter y McCarthy llegan al mismo sitio por caminos distintos. Reiter escribe reglas con letra pequeña y construye extensiones; McCarthy escribe reglas con una válvula de anormalidad y minimiza esa válvula. Los dos capturan la misma idea de fondo: **lo normal por defecto, lo raro solo cuando consta.**

Revisión de creencias: cambiar de opinión con método

Hasta aquí hemos visto cómo concluir cosas provisionales. Falta la otra mitad: qué hacer cuando llega un hecho que **contradice de frente** lo que ya creías. Tu base de conocimiento decía «todos los cisnes son blancos» y aparece un cisne negro. No puedes simplemente añadir el hecho nuevo, porque entonces tu base afirmaría a la vez que ese cisne es blanco y que es negro, y de una contradicción la lógica clásica deduce cualquier cosa. Tienes que **reacomodar** lo que sabes. La teoría que estudia cómo hacerlo con orden se conoce por las iniciales de sus tres autores: **AGM**, por Alchourrón, Gärdenfors y Makinson.⁴

AGM distingue tres operaciones sobre una base de creencias K , y la diferencia entre ellas es la frase clave de toda la sección:

OPERACIÓN	SÍMBOLO	QUÉ HACE	CUÁNDO SE USA
Expansión	$K + \varphi$	Añade φ sin tocar nada más.	Cuando φ no contradice lo que ya sabes.
Revisión	$K * \varphi$	Añade φ y repara lo que haga falta para no caer en contradicción.	Cuando φ choca con lo que crees.
Contracción	$K \div \varphi$	Quita φ (y lo justo para que no se vuelva a deducir).	Cuando dejas de estar seguro de φ .

La expansión es el caso fácil y monótono: si te enteras de algo compatible, lo apuntas y ya está. El caso difícil es la revisión, y AGM la resuelve con una idea elegante, la **identidad de Levi**: revisar es contraer primero y expandir después.

$$K * \varphi = (K \div \neg\varphi) + \varphi$$

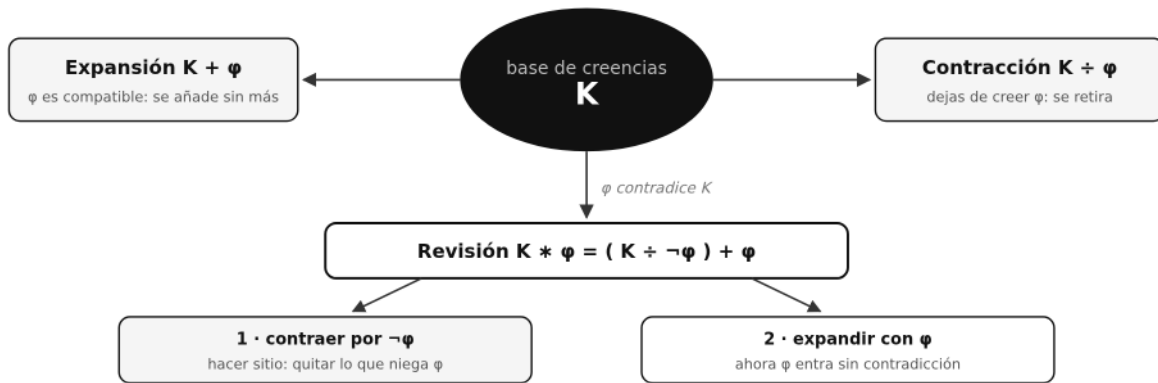
SÍMBOL O	SIGNIFICADO	EJEMPLO
K	La base de creencias actual.	«Todos los cisnes son blancos».
$*$	Revisión: incorporar φ manteniendo la coherencia.	Meter «este cisne es negro» sin romperlo todo.
$\div$	Contracción: retirar una creencia.	Soltar «todos los cisnes son blancos».
$+$	Expansión: añadir sin más.	Apuntar el cisne negro una vez hay sitio.
$\neg\varphi$	La negación de lo que queremos incorporar.	«No existe ningún cisne no blanco».

En palabras: para incorporar un hecho que te contradice, primero quita de tu base todo lo que se opone a él (contrae por su negación) y solo entonces añádelo; nunca metas la novedad encima de la creencia que la niega, porque eso te deja con las dos a la vez.

Veámoslo con el cisne. Tu base K contiene «todos los cisnes son blancos», de donde se deduce «este cisne es blanco». Llega φ = «este cisne es negro», así que $\neg\varphi$ = «este cisne es blanco». La identidad de Levi dice: primero $K \div \neg\varphi$, es decir, retira la creencia «este cisne es blanco» (y para que no se vuelva a deducir, tienes que debilitar también «todos los cisnes son blancos», quizá a «casi todos los cisnes son blancos»). Una vez tu base ya no afirma que el cisne sea blanco, expandes con «este cisne es negro». Resultado: una base coherente que reconoce la excepción sin haber explotado por el camino.

Las tres formas de cambiar de opinión

Una base de creencias se actualiza añadiendo, quitando o reparando, según el hecho que llegue.



IA para gente curiosa / Facsimil 12 / Capítulo 06 / 686f6c61

Los postulados: qué cuenta como un cambio racional

Lo que hace memorable a AGM no es la receta, sino que fijaron por escrito **qué propiedades debe cumplir cualquier forma sensata de revisar**, sin decir cómo implementarla. Son los **postulados de racionalidad**, y en lenguaje llano dicen cosas que, una vez enunciadas, parecen de cajón:

- **Éxito.** Después de revisar por φ , tu base cree φ . Si te has tomado la molestia de incorporar un hecho, que conste que lo crees.
- **Coherencia.** Si φ no es una contradicción en sí misma, la base revisada es consistente. Cambiar de opinión no debe dejarte creyendo cualquier cosa.
- **Inclusión.** Revisar no añade nada que no estuviera ya implícito al expandir; no te inventas creencias de la nada.
- **Cambio mínimo.** Y el más importante de todos: al reparar, **tira lo menos posible**. Si para aceptar el cisne negro basta con soltar «todos los cisnes son blancos», no aproveches para olvidar también dónde aparcaste el coche. La economía cognitiva es la regla de oro: conserva todo lo que puedas conservar.

Ese principio de cambio mínimo es el que conecta AGM con todo lo anterior. Tanto los defaults de Reiter como la circunscripción de McCarthy son, en el fondo, formas de «cambiar lo menos posible» ante información nueva: mantén tus conjeturas normales mientras puedas, y retoca solo lo que la realidad te obligue a retocar.

Argumentación: cuando las razones se pelean

Hay una última manera de domar la no monotonía, y es quizá la más cercana a cómo discutimos los humanos: no calcular extensiones ni contraer bases, sino **poner los argumentos a competir** y ver cuáles sobreviven. En 1995, Phan Minh Dung dio a esta idea

un marco abstracto de una sencillez sorprendente.⁵

Un **marco de argumentación** es asombrosamente austero: un conjunto de argumentos y una relación de ataque entre ellos. Nada más. No importa qué dice cada argumento por dentro; solo importa quién ataca a quién.

$$AF = \langle \mathcal{A}, \mathcal{R} \rangle, \quad \mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$$

SÍMBOL O	SIGNIFICADO	EJEMPLO
$\mathcal{A}$	El conjunto de argumentos en juego.	«Tweety vuela porque es ave».
$\mathcal{R}$	La relación de ataque: qué argumento contradice a cuál.	«Tweety es pingüino» ataca al anterior.
$\langle \rangle$	El marco completo: argumentos más sus ataques.	El debate entero, dibujado como un grafo.

En palabras: un marco de argumentación es un mapa de un debate donde cada flecha significa «este argumento tumba a aquel»; razonar es decidir qué grupo de argumentos puede sostenerse a la vez sin atacarse y defendiéndose de todo lo que les ataca.

La clave es la noción de **defensa**: un argumento está defendido por un conjunto si, para cada ataque que recibe, alguien del conjunto contraataca al atacante. A partir de ahí, Dung define conjuntos de argumentos que se aceptan juntos, las **extensiones** (el nombre vuelve, no por casualidad). Una extensión debe ser **libre de conflicto** (sus miembros no se atacan entre sí) y **admisibile** (defiende a todos sus miembros). Sobre Tweety, el argumento «es pingüino, no vuela» ataca a «es ave, vuela» y no recibe contraataque, así que sobrevive solo: la extensión aceptada concluye que no vuela. La excepción gana el debate.

Lo verdaderamente potente del resultado de Dung es que este marco minimalista **engloba a casi todos los anteriores**: la lógica por defecto, la negación por fallo de la programación lógica y otras formas de razonamiento no monótono se pueden traducir a marcos de argumentación. Detrás de «calcula extensiones» y «contrae tu base» late, una y otra vez, la misma pregunta: ¿qué conjunto de afirmaciones puede sostenerse coherentemente frente a las objeciones?

Las herramientas que lo ejecutan

Nada de esto es solo pizarra. Hay motores reales que razonan así todos los días:

- **Prolog**, con su negación por fallo ($\neg$), es la puerta de entrada práctica al mundo cerrado y la base de incontables sistemas de reglas.

- **Answer Set Programming** (ASP) es la heredera moderna y más expresiva. Se apoya en la **semántica de modelos estables** de Gelfond y Lifschitz, que da sentido preciso a la negación no monótona y permite que un programa tenga varios modelos coherentes, igual que una teoría por defecto tiene varias extensiones.⁶ Solvers como **clingo** resuelven con ella problemas de planificación, configuración y diagnóstico donde las excepciones son la norma.
- Para la revisión de creencias hay razonadores específicos que implementan operadores AGM, y la idea vive además dentro de cualquier sistema que tenga que mantener una base de conocimiento coherente mientras llegan hechos que se contradicen.

Para un mapa de conjunto de toda esta familia, el texto de referencia sigue siendo el panorama de Brewka, Dix y Konolige, que ordena defaults, circunscripción, argumentación y semántica de modelos estables bajo un mismo techo.⁷

En el día a día

Aunque no escribas nunca una regla por defecto ni la identidad de Levi, esta maquinaria está actuando bajo cosas que tocas a diario.

Cada vez que tu móvil asume que sigues en tu ciudad porque no le consta que hayas viajado, está haciendo mundo cerrado. Cada vez que un sistema de permisos concede acceso «por defecto al equipo, salvo excepción explícita para este usuario», está aplicando una regla con su letra pequeña, exactamente la estructura del default de Reiter. Cada vez que un GPS recalcula la ruta porque te has saltado una salida, está revisando sus creencias: tenía un plan, llegó un hecho que lo contradecía, y reparó lo justo sin tirar el destino. Y cada vez que un corrector o un asistente «cambia de idea» a mitad de frase porque una palabra posterior aclara el sentido, está haciendo algo que se parece, de lejos, a retractar una conclusión provisional.

En los sistemas expertos del capítulo 4, la no monotonía es la sangre de las excepciones: la regla de especificidad que hacía ganar a «si es ave y pingüino, no vuela» sobre «si es ave, vuela» es razonamiento por defecto disfrazado de prioridad. Y en las bases de conocimiento que evolucionan (un grafo de empresa, una ontología médica que incorpora un fármaco nuevo que contraindica a otro), la revisión de creencias es lo que evita que la base se vuelva incoherente cada vez que el mundo cambia.

Por qué debería importarte

Porque la diferencia entre un sistema que parece inteligente y uno que de verdad lo aguanta está, muchas veces, en cómo maneja las excepciones y la información que falta. Un sistema que solo sabe lógica monótona es frágil: en cuanto el mundo le presenta un pingüino, o se

queda mudo o se contradice. Un sistema que razona por defecto sigue siendo útil con información incompleta y se corrige con elegancia cuando llega el dato que faltaba.

Esto conecta de lleno con los modelos de lenguaje. Un LLM está empapado de defaults estadísticos: «un ave vuela», «un médico es un hombre», «un cumpleaños se celebra». Esos defaults lo hacen sonar razonable, y son justo los que arrastra mal cuando aparece la excepción: tiende a no retractarse con limpieza, a mantener la conjetura por defecto aunque el contexto la haya desmentido tres líneas antes, o a no distinguir «es falso» de «no me consta». Entender la diferencia entre una conclusión monótona (garantizada) y una por defecto (revisable) te da el vocabulario para saber cuándo exigirle a un sistema una garantía dura y cuándo aceptar una conjetura blanda, y para diseñar la capa que se encarga de retractarse cuando el modelo no sabe hacerlo solo.

Dónde solía tropezar yo

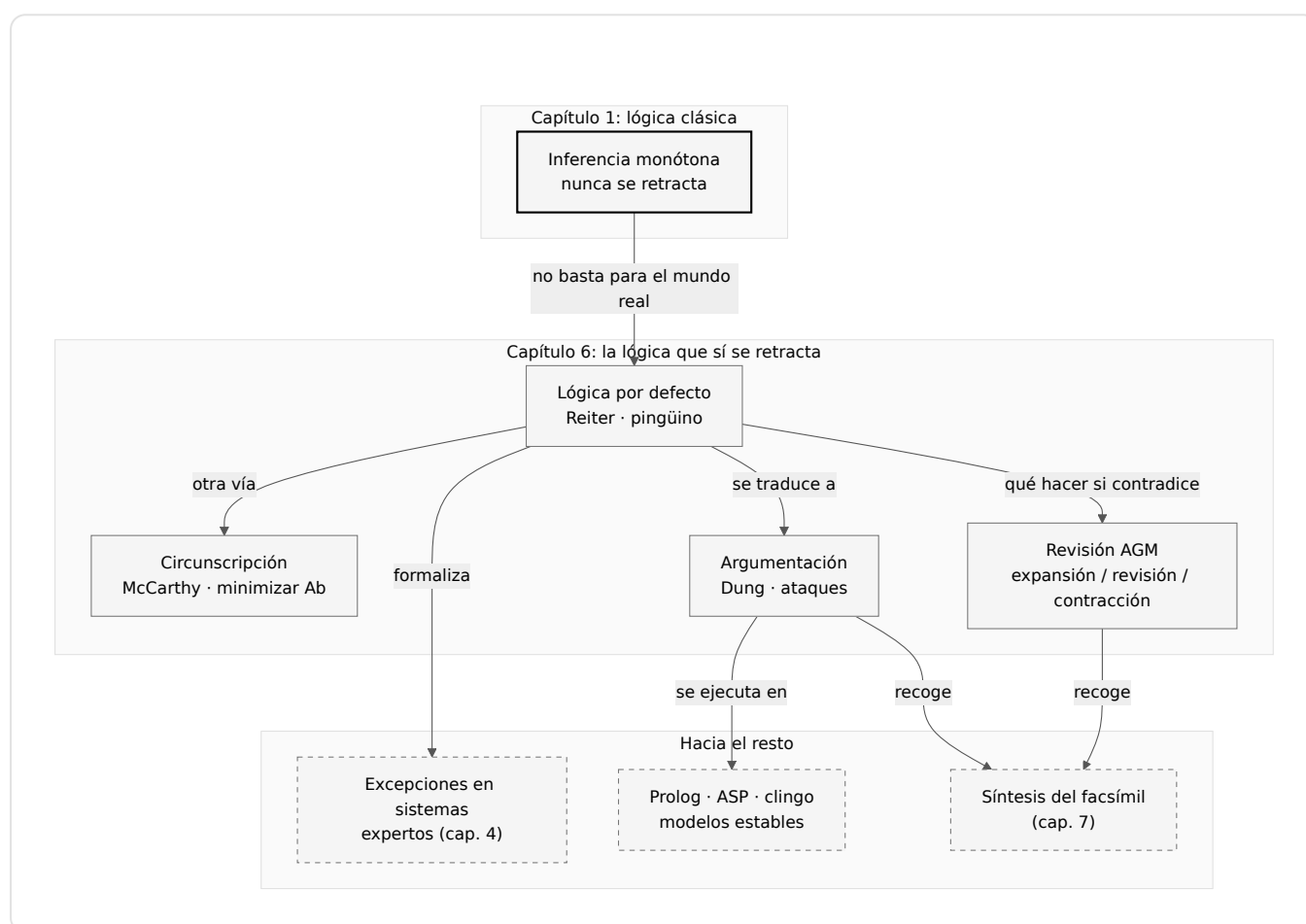
ERROR	POR QUÉ ES UN ERROR	ANTÍDOTO
Confundir «es falso» con «no lo puedo probar»	La negación por fallo concluye falso desde la ausencia. Si la base está incompleta, niegas cosas que sí son verdad.	Antes de cerrar el mundo, pregúntate si de verdad lo sabes todo sobre ese dominio. Si no, usa mundo abierto.
Esperar una única extensión	Una teoría por defecto puede tener varias visiones coherentes e incompatibles (el caso Nixon).	Acepta que a veces el conocimiento, tal como está escrito, admite varias posturas; decide qué política sigues (crédula o escéptica).
Revisar añadiendo encima de la contradicción	Si metes φ sin quitar $\neg\varphi$, tu base afirma las dos y se vuelve inconsistente.	Aplica la identidad de Levi: contrae por $\neg\varphi$ primero, expande con φ después.
Tirar demasiado al revisar	Para aceptar un hecho nuevo, olvidas medio sistema «por si acaso».	Respetar el cambio mínimo: retira solo lo que de verdad choca con la novedad.
Creer que un default es una verdad	«Las aves vuelan» es una conjetura revisable, no un teorema. Tratarla como ley te impide ver la excepción.	Marca mentalmente qué conclusiones son por defecto (flecha ondulada) y cuáles son lógicas duras (flecha gruesa).

Cómo encaja todo

Este capítulo es la bisagra del facsímil. El capítulo 1 te dio una lógica perfecta y rígida: correcta, completa y monótona, incapaz de retractarse de nada. Era exactamente lo que necesitabas para demostrar teoremas, y exactamente lo que te falla en cuanto sales a la calle,

donde casi todo lo que sabes vale «por defecto». Lo que aquí hemos hecho es añadirle a esa lógica la capacidad de cambiar de opinión: reglas con letra pequeña (Reiter), una válvula de anomalía que se minimiza (McCarthy), un método para reparar la base cuando un hecho la contradice (AGM) y un modo de poner los argumentos a competir (Dung).

El hilo más corto va al **capítulo 4**, el de los sistemas expertos. Allí, la regla de especificidad hacía que «si es ave y pingüino, no vuela» ganara a «si es ave, vuela», y se quedó como un truco del motor. Aquí ese truco tiene nombre y teoría: es razonamiento por defecto. Las excepciones que volvían frágiles a XCON y a MYCIN son, vistas desde este capítulo, defaults que se bloquean y bases de conocimiento que hay que revisar sin romper. Y el hilo hacia delante lleva al **capítulo 7**, donde recogeremos todo el facsímil: la lógica dura del capítulo 1, la probabilidad de los capítulos intermedios y esta lógica blanda que se retracta, tres formas complementarias de que una máquina razone con honestidad sobre un mundo incompleto.



Vocabulario aprendido

TÉRMINO	DEFINICIÓN
Razonamiento monótono	Razonamiento en el que añadir información nunca retira una conclusión previa.
Razonamiento no monótono	Razonamiento que puede retractar conclusiones al llegar información nueva.
Suposición de mundo cerrado	Asumir falso todo lo que no se puede demostrar verdadero.
Negación por fallo	Concluir la negación de algo porque no se logra demostrarlo, como en Prolog.
Lógica por defecto	Lógica de Reiter con reglas que valen salvo excepción, mientras sea consistente suponerlas.
Extensión	Conjunto de conclusiones coherente que se obtiene aplicando los defaults posibles.
Circunscripción	Método de McCarthy que minimiza lo anormal para razonar por defecto.
Revisión de creencias	Incorporar a una base de conocimiento un hecho nuevo manteniéndola consistente.
AGM	Marco de Alchourrón, Gärdenfors y Makinson para expansión, revisión y contracción de creencias.
Argumentación	Razonar con argumentos que se atacan entre sí y decidir cuáles son aceptables.
Answer Set Programming	Programación lógica no monótona basada en modelos estables para resolver problemas con reglas y excepciones.

Antes de pasar página

- ☐ ¿Sé enunciar la monotonía y explicar con Tweety por qué el sentido común la rompe? (Si no, vuelve a «Monotonía frente a no monotonía».)
- ☐ ¿Distingo «es falso» de «no lo puedo probar» y veo el riesgo de la negación por fallo? (Si no, vuelve a «Mundo cerrado y negación por fallo».)
- ☐ ¿Puedo leer una regla por defecto $\frac{\alpha:\beta}{\gamma}$ y decir cuándo se bloquea ante una excepción? (Si no, vuelve a «Lógica por defecto de Reiter».)
- ☐ ¿Entiendo qué es una extensión y por qué puede haber varias incompatibles? (Si no, vuelve a «Extensiones».)

- ☐ ¿Sé qué minimiza la circunscripción y cómo se cuela una excepción por $\neg Ab$? (Si no, vuelve a «Circunscripción de McCarthy».)
- ☐ ¿Distingo expansión, revisión y contracción, y sé aplicar la identidad de Levi? (Si no, vuelve a «Revisión de creencias».)
- ☐ ¿Entiendo el principio de cambio mínimo y por qué es el postulado clave? (Si no, vuelve a «Los postulados».)
- ☐ ¿Puedo dibujar un marco de argumentación y decir qué argumento sobrevive? (Si no, vuelve a «Argumentación».)

En resumen

IDEA FUERZA	DETALLE
La lógica clásica es monótona; el sentido común no.	Añadir premisas nunca retira conclusiones en el capítulo 1; en el mundo real, un pingüino te hace retractarte.
Lo que callas, el mundo cerrado lo niega.	La negación por fallo es rápida y útil, pero confunde «falso» con «no me consta»; solo es honesta si lo sabes todo.
Los defaults son reglas con letra pequeña.	$\frac{\alpha:\beta}{\gamma}$ concluye γ salvo que la excepción haga inconsistente la justificación; pueden surgir varias extensiones.
Revisar es contraer y luego expandir.	La identidad de Levi incorpora un hecho que contradice la base quitando primero su negación y cambiando lo mínimo.
Esta lógica blanda sostiene la IA robusta.	Excepciones en sistemas expertos, bases que evolucionan y los defaults que los LLM manejan mal viven todos aquí.

Para saber más

Alchourrón, C. E., Gärdenfors, P. y Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2), 510-530. <https://doi.org/10.2307/2274239>

Brewka, G., Dix, J. y Konolige, K. (1997). *Nonmonotonic Reasoning: An Overview*. CSLI Publications.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2), 321-357. [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X)

Gelfond, M. y Lifschitz, V. (1988). The stable model semantics for logic programming. En *Proceedings of the Fifth International Conference and Symposium on Logic Programming* (pp. 1070-1080). MIT Press.

McCarthy, J. (1980). Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2), 27-39. [https://doi.org/10.1016/0004-3702\(80\)90011-9](https://doi.org/10.1016/0004-3702(80)90011-9)

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1-2), 81-132. [https://doi.org/10.1016/0004-3702\(80\)90014-4](https://doi.org/10.1016/0004-3702(80)90014-4)

Russell, S. y Norvig, P. (2021). *Artificial intelligence: a modern approach* (4.^a ed.). Pearson.

Notas

1. Russell, S. y Norvig, P. (2021). *Artificial intelligence: a modern approach* (4.^a ed.). Pearson. Los capítulos de representación del conocimiento presentan el razonamiento no monótono, la lógica por defecto y la circunscripción como respuestas al carácter incompleto y revisable del conocimiento de sentido común. ↵
2. Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1-2), 81-132. [https://doi.org/10.1016/0004-3702\(80\)90014-4](https://doi.org/10.1016/0004-3702(80)90014-4) El artículo introduce las reglas por defecto y la noción de extensión, y es el texto fundacional del razonamiento no monótono formal. ↵
3. McCarthy, J. (1980). Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2), 27-39. [https://doi.org/10.1016/0004-3702\(80\)90011-9](https://doi.org/10.1016/0004-3702(80)90011-9) McCarthy propone razonar por defecto minimizando la extensión de un predicado de anormalidad, formalizando la idea de que las cosas son normales salvo prueba en contra. ↵
4. Alchourrón, C. E., Gärdenfors, P. y Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2), 510-530. <https://doi.org/10.2307/2274239> El artículo fundacional de la revisión de creencias define las tres operaciones y los postulados de racionalidad que deben cumplir. ↵
5. Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2), 321-357. [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X) El artículo introduce los marcos de argumentación abstracta y demuestra que unifican buena parte del razonamiento no monótono previo. ↵
6. Gelfond, M. y Lifschitz, V. (1988). The stable model semantics for logic programming. En *Proceedings of the Fifth International Conference and Symposium on Logic Programming* (pp. 1070-1080). MIT Press. Define la semántica de modelos estables, fundamento de Answer Set Programming. ↵

7. Brewka, G., Dix, J. y Konolige, K. (1997). *Nonmonotonic Reasoning: An Overview*. CSLI Publications. Una panorámica unificada de los principales formalismos no monótonos y sus relaciones. ↵
-

Cuadernos para practicar

Escanea el código con el móvil para abrir el cuaderno en Google Colab y ejecutarlo sin instalar nada.



Razonamiento no monótono: retractarse cuando llega un dato nuevo

<https://colab.research.google.com/github/686f6c61/iaparagentecuriosa-notebooks/blob/main/Facs%C3%ADmil%2012/06-razonamiento-no-monotono.ipynb>