8

Tablas recursivas para lógica epistémica temporal

Emilio F. Gómez-Caminero Parejo

Resumen. En la primera sección de este trabajo se recuerda una pequeña anécdota relacionada con el profesor Ángel Nepomuceno. En las secciones siguientes se presenta un método de tablas semánticas para sistemas sincrónicos de lógica epistémica temporal. La segunda sección presenta la sintaxis y la semántica de dichas lógicas epistémicas temporales. La sección tercera está dedicada a la axiomatización, haciendo hincapié en los sistemas con *memoria perfecta* y *no aprendizaje*. En la sección cuarta, por último, se presenta un método de tablas etiquetado con dos características especiales: el uso de dos tipos de reglas, a las que llamamos *reglas comunes* y *reglas de herencia*, por una parte, y el uso de reglas recursivas para tratar los operadores temporales.

8.1. ¿Veinte años no es nada?

Bueno, en realidad hace más de veinte años. Era exactamente el curso 1985/86. Yo era entonces un estudiante de segundo de Filosofía, y el recordado Emilio Díaz Estévez nos daba clase de Lógica I. Ni que decir tiene que muchos alumnos tenían problemas con la asignatura, a pesar de que Emilio era extremadamente generoso en la evaluación.

Nos ofreció entonces una posibilidad: asistir a clases prácticas de problemas; pero, eso sí, tendrían que ser por la tarde.

¿Por qué por la tarde, si la facultad por aquel entonces sólo tenía turno de mañana? Alguien, tal vez un alumno mayor, nos explicó la razón: la persona que iba a dar esas clases era “un señor que trabajaba en un banco” y que hacía esas cosas por la tarde poco menos que “por amor al arte”.

Ya habréis adivinado que estoy hablando de la misma persona que ahora homenajeamos: Ángel Nepomuceno. Por cierto, recuerdo bien el contenido de esas clases.

No tuve ocasión de volver a entablar contacto con él hasta muchos años más tarde. Yo había terminado la carrera y trabajaba como profesor de secundaria cuando volvió a picarme el gusanillo de la lógica y me matriculé, primero, en un curso de posgrado, luego en otro, y por fin en doctorado. El profesor que me recibió era aquel mismo señor que nos daba clases prácticas, sólo que ahora era ya profesor titular. Todo este tiempo, hasta ahora, he tenido a Ángel como tutor y maestro, y quiero pensar que también como amigo.

En estos años he ido sabiendo algo más de la historia de Ángel: no sólo era "un señor que trabaja en un banco" cuando nos daba esas clases, sino que también estudió la carrera e hizo el doctorado trabajando en ese mismo banco. Por supuesto, todo eso debería compatibilizarlo con sus responsabilidades familiares: ya estaba casado y tenía hijos cuando estudiaba.

Sé que todos conocéis esta historia, pero he querido destacarla porque me parece un mérito poco común. Siempre es agradable rendir homenaje a un maestro, y podría haber resaltado su valía como lógico, como profesor o como gestor de investigación, que sin duda es mucha; pero prefiero recordar su valor como persona, que a la postre, es lo que le hace inestimable.

8.2. Conocimiento y tiempo

Hay un amplio campo de problemas que requerirían, para su correcto análisis, la combinación de operadores epistémicos y temporales. Desgraciadamente, esta combinación da lugar a un enorme número de sistemas, y su estudio resulta de gran complejidad. Halpern, van de Meyer, y Vardi [4] cuentan hasta noventa y seis lógicas posibles resultantes de las decisiones que tomemos sobre el lenguaje empleado y sobre ciertas asunciones que afectan a los sistemas distribuidos subyacentes, cuarenta y ocho de estos sistemas consideran un tiempo lineal y cuarenta y ocho un tiempo ramificado. Nosotros no consideraremos operadores de tiempo ramificado y nos limitaremos a ciertos tipos de sistemas denominados *sistemas sincrónicos*. Empecemos presentando el lenguaje.

8.2.1. Sintaxis y semántica

Sea P un conjunto de variables proposicionales y A un conjunto de símbolos de agente, el lenguaje LET viene definido de la siguiente forma:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_a\varphi \mid \bigcirc\varphi \mid \varphi U \psi$$

$\widehat{K}$ se define como

$$\widehat{K}_a\varphi =_{def} \neg K_a \neg\varphi$$

Mientras que los operadores temporales $\Diamond$ y $\Box$ se definen:

$$\Diamond\varphi =_{def} \top U \varphi$$

$$\Box\varphi =_{def} \varphi U \bot$$

Intuitivamente, $K_a\varphi$ significa "el agente a sabe que φ; mientras que $\widehat{K}_a\varphi$ se podría interpretar como "Es posible, por lo que a sabe, que φ". En cuanto a los operadores temporales, $\bigcirc\varphi$, $\Box\varphi$, $\Diamond\varphi$ y $\varphi U\psi$ se leerían respectivamente como "en el siguiente momento, φ", "siempre (ahora y en el futuro) φ"; "en algún momento (ahora o en futuro) φ" y "φ hasta que ψ".

Pasemos ahora a la semántica. Para, ello, debemos antes decir algo sobre la noción de sistema.

Partiremos de la noción de *estado local* de un agente a_i, lo que denotaremos por s_{a_i}. Intuitivamente, un estado local es cada una de las situaciones posibles en las que puede estar un agente. Por ejemplo, en un juego de naipes en que cada jugador tiene en todo momento cuatro cartas, el conjunto de todos los estados posibles para cada jugador es el de todos los repartos posibles de cuatro cartas. Es conveniente, además tener en consideración aspectos relevantes del sistema que no se incluyen en la descripción del estado local de los agentes, para ello incluimos un estado local s_e que describe las características relevantes del entorno.

Sea A un conjunto de m agentes. Si L_e es el conjunto de estados posibles del entorno y L_{a_i} (para $a_i \in A$) el conjunto de estados locales posibles para el agente a_i, entonces $\mathcal{G} = L_e \times L_{a_1} \times \cdots \times L_{a_m}$ es el conjunto de *estados globales.*

Definimos ahora un *recorrido* sobre $\mathcal{G}$ como una función del conjunto de los números naturales, que elegimos como dominio temporal, en $\mathcal{G}$. Esto es, cada recorrido es una secuencia de estados globales. Hablaremos del *punto* (r, n) para referirnos al momento temporal n del recorrido r, y escribiremos $r(n)$ para referirnos al estado global del sistema en el punto (r, n). A su vez, escribiremos $r_e(n)$ para referirnos al estado local del entorno en el punto (r, n); y $r_{a_i}(n)$ para referirnos al estado local del agente a_i en ese mismo punto.

Ahora ya podemos definir un *sistema* $\mathcal{R}$ sobre $\mathcal{G}$ como un conjunto de recorridos sobre $\mathcal{G}$.

El concepto de sistema es la base sobre la que se construye la semántica de una lógica epistémica temporal. El sistema $\mathcal{R}$ tiene el mismo papel que juega el conjunto W de mundos posibles en las semánticas kripkeanas; pero para definir la verdad de una fórmula necesitamos disponer todavía de las relaciones de accesibilidad y de la función de evaluación.

Las primeras pueden definirse directamente sobre el concepto de *estado global*: dados dos estados globales de $\mathcal{R}$ $s = (s_e, s_{a_1}, \cdots, s_{a_n})$ y $s' = \left(s'_e, s'_{a_1}, \cdots, s'_{a_n}\right)$, diremos que s es indistinguible de s' para el agente a_i, lo que escribiremos $s \sim_{a_i} s'$ syss el estado local de a_i es el mismo en s y en s'; esto es, si $s_{a_i} = s'_{a_i}$. Como puede verse, esta relación es una relación de equivalencia, lo que significa que el sistema de lógica epistémica subyacente es $S5$.

Sea ahora P un conjunto no vacío de variables proposicionales[1]. Podemos definir un sistema interpretado $\mathcal{I}$ como un par $(\mathcal{R}, v)$, donde $\mathcal{R}$ es un sistema sobre $\mathcal{G}$ y v es una función que asigna valores de verdad a los elementos de P en cada estado $s \in \mathcal{G}$; esto es, $v : P \times \mathcal{G} \mapsto \{1, 0\}$[2]. Escribiremos $v(s)(p) = 1$ ó $v(s)(p) = 0$ para indicar que el valor asignado a la variable p en el estado global s es respectivamente 0 ó 1. El valor de verdad de una variable en un punto (r, n) es simplemente el valor de esa variable en el estado $r(n)$ correspondiente.

Ahora ya podemos definir el concepto de verdad de una fórmula φ en un punto (r, n) de un sistema interpretado $\mathcal{I}$. Escribiremos $(\mathcal{I}, r, n) \vDash \varphi$ para denotar que la fórmula φ es verdadera en (o satisfecha por) el punto (r, n) del sistema $\mathcal{I}$. Presentaremos las cláusulas para el lenguaje KL_m, dados un conjunto A de m agentes y un conjunto no vacío P de variables proposicionales.

[1] Asumimos que se pueden describir las propiedades relevantes del sistema usando lógica proposicional. Es posible extender el concepto de sistema interpretado a la lógica de primer orden; pero no lo haremos en este lugar, por razones de simplicidad.

[2] Esta es la forma en que lo presentan [1] y [3]; en cambio, [4] interpretan que v asigna un valor de verdad en cada punto, y no en cada estado global; con lo que el valor de verdad de las variables en puntos a los que les corresponde el mismo estado global puede ser distinto. Esta es una interpretación más amplia, pero hemos adoptado la primera por parecernos más natural.

La verdad de una fórmula atómica se define de forma completamente previsible:

$$(\mathcal{I}, r, n) \vDash p \text{ (para } p \in P) \text{ syss } \upsilon (r, n) (p) = 1$$

El caso de los operadores proposicionales se define de la forma habitual. Las cláusulas correspondientes a los operadores epistémicos son simplemente una variación formal de las habituales en lógica epistémica:

$(\mathcal{I}, r, n) \vDash K_{a_i}\varphi$ syss $(\mathcal{I}, r', n') \vDash \varphi$ para todo (r', n') tal que $(r, n) \sim_{a_i} (r', n')$.

$(\mathcal{I}, r, n) \vDash \widehat{K}_{a_i}\varphi$ syss $(\mathcal{I}, r', n') \vDash \varphi$ para algún (r', n') tal que $(r, n) \sim_{a_i} (r^{\text{‘}\prime}, n')$.

Queda, por último, definir la verdad de una fórmula en un punto de un sistema interpretado para el caso de los operadores temporales. Las cláusulas correspondientes son las siguientes:

$(\mathcal{I}, r, n) \vDash \bigcirc\varphi$ syss $(\mathcal{I}, r, n+1) \vDash \varphi$

$(\mathcal{I}, r, n) \vDash \Box\varphi$ syss $(\mathcal{I}, r, n') \vDash \varphi$ para todo $n' \geq n$

$(\mathcal{I}, r, n) \vDash \Diamond\varphi$ syss $(\mathcal{I}, r, n') \vDash \varphi$ para algún $n' \geq n$

$(\mathcal{I}, r, n) \vDash \varphi U \psi$ syss hay algún $n' \geq n$ tal que $(\mathcal{I}, r, n') \vDash \psi$ y para todo n'' tal que $n' > n'' \geq n$, $(\mathcal{I}, r, n'') \vDash \varphi$

Diremos que $\mathcal{I} \vDash \varphi$ si para todo punto $(r, n) \in \mathcal{R}$ del sistema interpretado $\mathcal{I}$, $(\mathcal{I}, r, n) \vDash \varphi$. A su vez, dado un conjunto de sistemas interpretados Θ, escribiremos $\vDash_\Theta \varphi$ para denotar que $\mathcal{I} \vDash \varphi$ para todo $\mathcal{I} \in \Theta$.

8.3. Axiomatización

Los axiomas propios de lógica epistémica son los propios del sistema $S5_m$:

A1 Todas las tautologías de la lógica proposicional.

A2 $(K_{a_i}\varphi \wedge K_{a_i} (\varphi \rightarrow \psi)) \rightarrow K_{a_i}\psi$

A3 $K_{a_i}\varphi \rightarrow \varphi$

A4 $K_{a_i}\varphi \rightarrow K_{a_i}K_{a_i}\varphi$

A5 $\neg K_{a_i}\varphi \rightarrow K_{a_i}\neg K_{a_i}\varphi$

R1 $\dfrac{\begin{array}{c}\varphi \rightarrow \psi \\ \varphi\end{array}}{\psi}$

R2 $\dfrac{\vdash \varphi}{\vdash K_{a_i}\varphi}$

Respecto a los operadores temporales, el siguiente conjunto de esquemas de axiomas y reglas constituye, junto a A1 y R1, una axiomatización correcta y completa respecto a la semántica que acabamos de presentar [2]:

1 $\bigcirc\varphi \wedge \bigcirc(\varphi \rightarrow \psi) \rightarrow \bigcirc\psi$

T2 $\bigcirc\neg\varphi \leftrightarrow \neg\bigcirc\varphi$

T3 $\varphi U\psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi U\psi))$

RT1 $\dfrac{\vdash \varphi}{\vdash \bigcirc\varphi}$

RT2 $\dfrac{\vdash \varphi' \rightarrow \neg\psi \wedge \bigcirc\varphi'}{\vdash \varphi' \rightarrow \neg(\varphi U\psi)}$

La combinación de estos axiomas y reglas es suficiente para axiomatizar completamente un sistema de lógica epistémica temporal, incluso si deseamos exigir que tenga las propiedades de *sincronía* y *estado inicial único*[3]. Si además añadimos el esquema de axioma KT2:

$$K_{a_i}\bigcirc\varphi \rightarrow \bigcirc K_{a_i}\varphi \text{ (para todo } a_i \in A)$$

tendremos una axiomatización completa respecto al conjunto de todos los sistemas con las propiedades de *memoria perfecta*[4] y *sincronía*. Por su parte, el axioma KT5:

$$\bigcirc K_{a_i}\varphi \rightarrow K_{a_i}\bigcirc\varphi \text{ (para todo } a_i \in A)$$

expresa la combinación de *no aprendizaje*[5] y *sincronía*.

Es posible añadir en lugar de los anteriores axiomas que expresen exclusivamente las propiedades de *memoria perfecta* o *no aprendizaje*, pero no la de *sincronía*. Nuestro propósito, sin embargo, es presentar un método de tablas para sistemas sincrónicos, por lo que tales axiomas no resultan pertinentes para este trabajo.

8.4. Tablas semánticas

Llegamos ahora al punto central de este trabajo, que consiste precisamente en presentar un método de tablas etiquetadas para sistemas sincrónicos de lógica epistémica temporal: método que tiene además la peculiaridad de utilizar reglas recursivas. Empecemos presentando el sistema de etiquetas que utilizaremos.

Una etiqueta es una secuencia de la forma $\sigma(t)$, donde $\sigma \in \Sigma$ y $t \in \mathbb{N}$. Diremos que σ es el segmento epistémico y t el segmento temporal de la etiqueta.

El conjunto Σ de todos los segmentos epistémicos se define:

1. $1 \in \Sigma$

2. Si $\sigma \in \Sigma$ y $a \in A$, $\sigma.an \in \Sigma$

[3]Un sistema $\mathcal{R}$ es sincrónico syss para todo agente $a_i \in A$ y cualesquiera puntos $(r,n),(r',n') \in \mathcal{R}$, si $(r,n) \sim_{a_i} (r',n')$ entonces $n = n'$. Un sistema $\mathcal{R}$ tiene un estado inicial único si para cualesquiera recorridos $r, r' \in \mathcal{R}$, $r(0) = r'(0)$; lo cual significa que para todo agente $a_i \in A$, $(r,0) \sim_{a_i} (r',0)$.

[4]El agente a_i tiene memoria perfecta en el sistema $\mathcal{R}$ si para cualesquiera puntos $(r,n),(r',n') \in \mathcal{R}$, si $(r,n) \sim_{a_i} (r',n')$ entonces la secuencia de estados locales que adopta a_i a lo largo de r es la misma en (r,n) y en (r',n').

[5]Diremos que el agente a_i no aprende en el sistema $\mathcal{R}$ si para cualesquiera puntos $(r,n),(r',n') \in \mathcal{R}$, si $(r,n) \sim_{a_i} (r',n')$ entonces la secuencia de estados locales futuros del agente a_i a lo largo de r es la misma en (r,n) y en (r',n').

La primera etiqueta de toda tabla es $1\,(0)$.

Con vistas a introducir una restricción que evita que las tablas se vuelvan infinitas en ciertos casos en que el conjunto de fórmulas que funciona como asunción de la tabla tiene un modelo finito, necesitamos definir previamente el concepto de *ascendiente epistémico* de una etiqueta:

1. Una etiqueta es un ascendiente epistémico de sí misma respecto a cualquier agente.
2. $\sigma.a_i n(t)$ es un ascendiente epistémico (respecto a a_i) de $\sigma.a_i n.a_j m(t)$ syss $a_i = a_j$.
3. Si $\sigma(t)$ es un ascendiente epistémico de $\tau(t)$ respecto a a_i y $\tau(t)$ lo es de $\kappa(t)$ entonces $\sigma(t)$ es un ascendiente epistémico respecto a a_i de $\kappa(t)$.

Una vez explicado estos conceptos previos, es el momento de presentar nuestro método de tablas etiquetadas. Una fórmula etiquetada es una secuencia de la forma $\sigma :: \varphi$, donde σ es una etiqueta y $\varphi \in LET$. Una tabla semántica es un conjunto de secuencias de fórmulas etiquetadas, a las que llamaremos ramas. Una rama es cerrada si ella aparecen una fórmula atómica y su negación, precedidas de la misma etiqueta. Una tabla es cerrada si todas sus ramas lo son.

Las reglas para los operadores proposicionales son las habituales salvo, naturalmente, por lo que hace a la etiqueta, que permanece constante. Por esta razón, no consideramos necesario exponerlas aquí. Las reglas de los operadores epistémicos y temporales se dividen en dos categorías: reglas comunes y reglas de herencia. Las propias de la lógica epistémica son las siguientes:

Si la fórmula etiquetada a marcar es $\sigma\,(t) :: K_{a_i}\alpha$ se escribe $\sigma\,(t) :: \alpha$ al término de la rama y se marca con K.

si la fórmula etiquetada a marcar es $\sigma(t) :: \neg K_{a_i}\beta$, se escribe $\sigma(t) :: \widehat{K}_{a_i}\neg\beta$ al término de la rama y se marca con $\neg$.

Si la fórmula etiquetada a marcar es $\sigma\,(t) :: \widehat{K}_{a_i}\alpha$ se escribe $\sigma.a_i n\,(t) :: \alpha$ al término de la rama (donde n es el primer entero positivo tal que $\sigma.a_i n\,(t)$ es nueva en la rama) y se marca con $\widehat{K}(\sigma.a_i n\,(t))$. Restricción: salvo que $\tau\,(t) :: \alpha$ aparezca en la rama y $\tau\,(t)$ sea un ascendiente epistémico de $\sigma\,(t)$ respecto a a_i, en cuyo caso se da la regla por aplicada y se marca la fórmula con $\widehat{K}(\tau\,(t))$.

si la fórmula etiquetada a marcar es $\sigma(t) :: \neg\widehat{K}_{a_i}\beta$, se escribe $\sigma :: K_{a_i}\neg\beta$ al término de la rama y se marca con $\neg$.

A las que hay que añadir la siguiente regla de herencia:

HKS5:

a. Si en una rama del árbol aparece una fórmula etiquetada de la forma $\sigma\,(t) :: K_{a_i}\alpha$, se escribe al término de la rama $\sigma.a_i n\,(t) :: K_{a_i}\alpha$, para toda etiqueta $\sigma.a_i n\,(t)$ que aparezca en la rama.

b. Si en una rama aparece una fórmula etiquetada de la forma $\sigma.a_i n\,(t) :: K_{a_i}\alpha$, se escribe $\sigma\,(t) :: K_{a_i}\alpha$ al término de la rama.

Se habrá podido observar que el segmento epistémico de las etiquetas permanece constante en todas las reglas. Este hecho es precisamente el que garantiza que la tabla nos permite encontrar un modelo con la propiedad de sincronía.

En cuanto a los operadores temporales, es aquí donde entra en juego la recursividad: esto ocurre con las fórmulas de la forma $\Diamond\alpha$, $\alpha U\beta$ y $\neg(\alpha U\beta)$. En los casos de estos tres tipos de fórmulas la regla consta de dos partes; una primera, a la que llamaremos la condición de parada, que en caso de dar lugar a una rama abierta da por concluida la aplicación de la regla; y una segunda que se aplica sólo si la primera da lugar a una rama cerrada y que característicamente genera una rama que contiene las fórmulas $\bigcirc\Diamond\alpha$, $\bigcirc(\alpha U\beta)$ o $\bigcirc\neg(\alpha U\beta)$ (en los casos de $\Diamond\alpha$, $\alpha U\beta$ o $\neg(\alpha U\beta)$, respectivamente). Llamaremos a esta segunda parte la cláusula recursiva de la regla. Por supuesto, puede ocurrir que la condición de parada nunca genere una rama abierta, bien porque la fórmula no sea satisfactible, bien porque sólo lo sea en sistemas interpretados que constan de infinitos puntos. En estos casos, la regla da lugar a una rama infinita.

Las reglas en cuestión son las siguientes:

Si la fórmula etiquetada a marcar es $\sigma(t) :: \bigcirc\alpha$ se escribe $\sigma(t+1) :: \alpha$ al término de la rama y se marca con $\bigcirc$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \neg\bigcirc\alpha$ se escribe $\sigma(t+1) :: \neg\alpha$ al término de la rama y se marca con $\neg$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \Box\alpha$ se escribe $\sigma(t) :: \alpha$ al término de la rama y se marca con $\Box$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \neg\Box\alpha$ se escribe $\sigma(t) :: \Diamond\neg\alpha$ al término de la rama y se marca con $\neg$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \Diamond\alpha$ se abre una subrama, se escribe a su término $\sigma(t) :: \alpha$ y se marca $\Diamond 1$. Si esta subrama se cierra, se abre otra subrama, se escribe $\sigma(t) :: \bigcirc\Diamond\alpha$ y se marca con $\Diamond 2$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \neg\Diamond\alpha$ se escribe $\sigma(t) :: \Box\neg\alpha$ al término de la rama y se marca con $\neg$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \alpha U\beta$ se abre una subrama, se escribe a su término $\sigma(t) :: \beta$ y se marca con $U1$. Si esta subrama se cierra, se abre otra subrama, se escribe consecutivamente $\sigma(t) :: \alpha$ y $\sigma(t) :: \bigcirc(\alpha U\beta)$, y se marca con $U2$.

Si la fórmula etiquetada a marcar es $\sigma(t) :: \neg(\alpha U\beta)$ se abre una subrama, se escribe consecutivamente $\sigma(t) :: \neg\alpha$ y $\sigma(t) :: \neg\beta$; y se marca con $\neg U1$. Si esta subrama se cierra, se abre otra subrama, se escribe consecutivamente $\sigma(t) :: \alpha$, $\sigma(t) :: \neg\beta$ y $\sigma(t) :: \bigcirc\neg(\alpha U\beta)$; y se marca con $\neg U2$.

A éstas hay que añadir la siguiente regla de herencia:

HT: Si en una rama del árbol aparece una fórmula etiquetada de la forma $\sigma(t) :: \Box\alpha$, se escribe al término de la rama $\sigma(t') :: \alpha$, para toda etiqueta $\sigma(t')$ que aparezca en la rama tal que $t' > t$.

Con estas reglas, disponemos de un procedimiento para el caso de los sistemas sincrónicos. Si además queremos añadir la propiedad de *memoria perfecta*, deberemos incorporar la siguiente regla de herencia:

HKT(MP): Si en una rama del árbol aparece una fórmula etiquetada de la forma $\sigma\,(t) :: K_{a_i} \bigcirc \alpha$, se escribe al término de la rama $\sigma\,(t+1) :: K_{a_i}\alpha$.

Si queremos un método de tablas para sistemas con la propiedad de no aprendizaje, la regla que debemos añadir es la siguiente:

HKT(NA): Si en una rama del árbol aparece una fórmula etiquetada de la forma $\sigma\,(t) :: K_{a_i}\alpha$, se escribe al término de la rama $\sigma\,(t-1) :: K_{a_i} \bigcirc \alpha$ (salvo que $t = 0$).

Por supuesto, si queremos tener ambas condiciones no tenemos más que añadir las dos reglas.

Este método de tablas constituye un procedimiento correcto y completo para sistemas sincrónicos de lógica epistémica temporal. Pero esto es algo que no podemos demostrar en este lugar.

Bibliografía

[1] Fagin, R., Halpern, J.Y., Moses, Y. y Vardy, M.Y. (1995): *Reasoning About Knowledge*. Cambridge. The MIT Press.

[2] Gabbay, D.; Pnueli, A.; Shelah, S. y Stavi, J. (1980): "On the Temporal Analysis of Fairness". En *Poc. 7th ACM Symp. on Principles of Programming Languages*, pp. 163-173.

[3] Halpern, J.Y. y Vardi, M.Y. (1989): "The complexity of Reasoning about Knowledge and Time". I: Lower Boundns. En *Journal of Computer and Systems Science*, 38, pp. 195-237.

[4] Halpern, J.Y.; van der Meiden, R. y Vardi, M.Y. (1999): "Complete axiomatizations for reasoning about Knowledge and Time.". En *SIAM Journal on Computing* 33:2, 2004, pp. 674-703.